COLLEGE MATHS, TELECOMS
DOCTORAL INFORMATIQUE, SIGNAL
BRETAGNE SYSTEMES, ELECTRONIQUE

Université de Rennes

THÈSE DE DOCTORAT DE

L'UNIVERSITÉ DE RENNES

ÉCOLE DOCTORALE N° 601
Mathématiques, Télécommunications, Informatique, Signal, Systèmes, Électronique
Spécialité : *INFO*

Par
Gautier EVENNOU

Detection and Explanation of Semantic Changes in Multimedia Content

From Passive Detection to Active Watermarking

Soutenance prévue à Rennes, le 30 septembre 2026
Unité de recherche : IRISA

Rapporteur·trice·s avant soutenance :

Patrick BAS — Directeur de Recherche, CNRS, UMR CRIStAL, Lille
William PUECH — Professeur, Université de Montpellier, LIRMM, Montpellier

Composition du Jury :

Examinateur·trice·s :	Damien LOLIVE	Professeur, Université de Bretagne Sud, IRISA, Vannes
	Patrick BAS	Directeur de Recherche, CNRS, UMR CRIStAL, Lille
	William PUECH	Professeur, Université de Montpellier, LIRMM, Montpellier
	Pierre FERNANDEZ	Chercheur, Meta, Paris
Dir. de thèse :	Ewa KIJAK	Professeure, Université de Rennes, IRISA, Rennes
Co-dir. de thèse :	Vivien CHAPPELIER	Chercheur senior, LABEL4.AI, Rennes

Acknowledgments

Il est de coutume de remercier les personnes m'ayant accompagné durant ce moment particulier qu'est la thèse. Pour moi, il s'agit d'un parcours qui a commencé bien avant la thèse et qui forme un tout cohérent.

La première personne à qui j'ai envie de rendre hommage est Ewa, qui a été ma directrice de thèse et qui m'a introduit à l'intelligence artificielle et au monde de la recherche grâce à un stage dans l'équipe LinkMedia. C'est mon effet papillon, celui duquel tout le reste découle. Ewa, merci pour ton encadrement, je me rends compte de la chance que j'ai eue d'avoir bénéficié de tant de compétences et de bienveillance à un rythme aussi soutenu.

Comment ne pas remercier Teddy, qui m'aura accompagné durant ce stage de ses précieux conseils. Je dois beaucoup de mes références et de mon émerveillement pour la recherche à cette période, durant laquelle j'ai rencontré Antoine, Thibault, Benoît et Karim. Je me souviendrai toujours de ces parties de jeux de société/babyfoot au cœur de l'été ! Grâce à Teddy j'ai eu une recommandation pour effectuer un stage chez IMATAG.

La transition au monde de l'industrie à IMATAG a été très inspirante, notamment grâce à Antoine et à Vivien qui m'auront montré à quoi ressemblait un ingénieur-chercheur, définissant ainsi un standard de qualité que je m'efforce d'atteindre un jour. En plus ce sont de formidables personnes, ce qui ne gâche rien ! Et suffisamment fou pour ça, j'ai poursuivi avec une thèse sur un sujet qui me passionne. Je remercie Mathieu de m'avoir fait confiance et toute l'équipe d'IMATAG pour ces bons moments passés en votre compagnie : Mathieu, Vivien, Antoine, Vedran, Corinne, Jon, Morgan, David K., David M., Christine, Sarah, Mylène, Guillaume, Igor, Thierry, Gaëtan, Titouan, Agathe, Nouhaila et Clément.

Côté laboratoire, j'ai eu la joie de connaître deux équipes de recherche différentes,

LinkMedia et Artishau. Merci à tous je me suis toujours senti chez moi parmi vous : Eva, Morgane, Laurent, Teddy, Erwan, Paul, Thomas, Jean-Rémi, Lilas, Pierre, Hugo, Enoal, Ryan, Victor, Chloé, Kyrian, Rossana, Timothée, Océane, Giulio, Virgile, Gurvan, Paul et Adèle.

Pour l'équipe de babyfoot, je tiens à vous remercier pour tous ces bons moments, véritablement cathartiques et salvateurs pour moi. Je ne doute pas que l'héritage baby-footballistique de cette ère vivra en chacun d'entre vous.

Eva et Enoal particulièrement, merci pour toutes nos discussions, parfois « philosophiques », parfois sur l'actualité de notre domaine qui n'a pas toujours de sens, elles m'ont fait grandir. Paul, Rossana, Océane et Kyrian, j'ai passé de super moments dans votre bureau ; j'espère que mes quelques conseils vous seront utiles pour la suite de votre parcours. Merci Adèle d'apporter ton groin de sel à cette équipe, tu auras la référence j'en suis sûr ! Chloé merci pour ta clairvoyance et de m'avoir fait plonger dans Magic, mon porte-monnaie m'en veut encore ! Virgile, Timothée et Gurvan, ce fut un plaisir de partager officiellement ce bureau avec vous, on a sans doute trop parlé parfois mais je ne regrette rien.

Pour basculer du côté plus intime de ces remerciements, je tiens à remercier mes parents ; vous m'avez donné de formidables outils et inculqué des valeurs qui sont autant de repères balisant ce chemin bizarre et incertain que je n'aurais jamais cru arpenter.

Enfin, la meilleure pour la fin, merci Lucie ; autant que moi, tu étais impliquée dans ce chemin et dans ses contraintes. En tant que bon finistérien, je me permettrai une métaphore datée mais maritime : sans le savoir, tu m'as toujours indiqué la direction à prendre, agissant à la fois comme un phare dans la nuit et comme une ancre à mes pensées. Ce chemin aurait moins de saveur sans toi.

Résumé substantiel en français

Introduction générale

L'avènement des modèles génératifs a transformé le rapport que nous entretenons aux contenus multimédias. Produire ou éditer une image photoréaliste ne demande plus d'expertise particulière : ces outils sont désormais intégrés aux smartphones, aux réseaux sociaux et à des services accessibles au plus grand nombre. Cette facilité a un revers. Toute image peut être altérée, et les altérations qui en modifient le *sens*, comme déplacer un objet, substituer un visage, ou falsifier une scène, sont précisément celles qui nourrissent la désinformation. La confiance que l'on accordait par défaut à une photographie n'existe plus.

Face à ce constat, deux besoins complémentaires se dégagent. Il faut pouvoir *détecter* qu'une altération a eu lieu, mais aussi l'*expliquer*, c'est-à-dire caractériser ce qui a changé. La détection seule, binaire, dit peu de chose : savoir qu'une image a été modifiée sans savoir comment ne suffit ni à un journaliste, ni à un modérateur de plateforme, ni à un régulateur. Cette thèse développe des outils sémantiques pour répondre à ces deux exigences, en se concentrant sur l'image. Le fil directeur en est la sémantique : qu'elle soit décrite explicitement, insérée dans le contenu ou attaquée, elle constitue le signal sur lequel s'appuient à la fois la détection des altérations et la résistance à celles-ci.

Cet enjeu n'est pas purement technique. La loi de Brandolini rappelle qu'il est bien plus coûteux de réfuter une fausse information que de la propager : un montage trompeur atteint des millions de personnes là où son démenti passe inaperçu. Les outils proposés ici visent donc à agir le plus en amont possible dans la chaîne de production et de diffusion des contenus.

Contributions

Les travaux de cette thèse s'organisent autour de trois axes.

Explication passive. Nous abordons d'abord la tâche de description des différences entre deux images (*Image Difference Captioning*, IDC), selon le principe « expliquer, c'est détecter ». Plutôt que de signaler qu'une modification a eu lieu, nous en produisons une description textuelle fine. Cet axe est qualifié de *passif* car il n'intervient pas en amont sur le contenu : il observe l'image disponible et la compare à son originale. Notre contribution, BLIP2IDC, atteint l'état de l'art à faible coût en misant sur le préentrainement des grands modèles multimodaux et montre que l'augmentation synthétique par modèles génératifs permet d'apprendre avec peu de données réelles.

Tatouage sémantique. La principale limite de l'axe précédent est le recours à l'image originale et à des modèles lourds. Nous proposons de lever cette dépendance en insérant directement dans l'image une représentation sémantique d'elle-même, que l'on pourra ensuite comparer au contenu observé. Deux contributions concrétisent cette idée : SWIFT, qui tatoue une légende compressée, puis LatentSeal, qui tatoue un vecteur latent, ici issu d'un auto-encodeur de texte, plus rapide et généralisable à d'autres modalités.

Robustesse du tatouage à l'édition d'image basée sur la GenAI. Un système de tatouage n'a de valeur que s'il résiste aux tentatives de suppression et aux éditions génératives devenues banales. Notre dernière contribution, la détection forensique de suppression de tatouage (WRD), étudie le coût forensique d'une telle suppression et montre que ces attaques, y compris les méthodes récentes d'édition d'images, laissent des traces détectables.

Introduction technique

Cette section rassemble certaines notions réutilisées par les contributions. Elle est développée dans le chapitre de contexte technique du manuscrit et résumée ici.

Le **tatouage numérique** consiste à dissimuler un signal dans un média support (*cover*), ici une image. On distingue le tatouage *multi-bits*, qui transmet une séquence de bits, du tatouage *zéro-bit*, qui n'insère qu'un vecteur dont seule la présence importe et offre

de fortes garanties statistiques de détection, utile pour prouver la provenance d'un contenu. Quatre propriétés structurent ces systèmes : la *robustesse* (survie aux transformations : compression JPEG, altérations colorimétriques ou géométriques), l'*imperceptibilité* (le signal reste invisible à l'œil comme pour un réseau de neurones), la *capacité* (taille du message ou dimension du vecteur, et donc garantie de détection), et la *sécurité* (loi de Kerckhoffs : la sûreté ne doit reposer que sur une clé secrète). Nous privilégions le tatouage *a posteriori*, versatile et simple à mettre en œuvre, plutôt que le tatouage *à la génération* qui suppose de ré-entraîner les modèles génératifs et de se limiter aux images générées par ces modèles.

Les méthodes profondes *post-hoc* reposent sur une architecture **encodeur–décodeur** apprenant deux opérations : l'*encodage*, qui fabrique un signal robuste et le dissimule de façon imperceptible dans le support, et le *décodage*, qui le récupère. L'apprentissage combine plusieurs objectifs : la fidélité du message (entropie croisée binaire entre le message m et sa version décodée $\sigma(\hat{m})$), la contrainte d'imperceptibilité (MSE, LPIPS (R. Zhang et al. 2018), un discriminateur de type GAN (Goodfellow et al. 2014), éventuellement un modèle psychovisuel (Jinjian Wu et al. 2017)), et une robustesse obtenue en intégrant les transformations dès l'entraînement — ce qui crée un compromis direct entre robustesse et imperceptibilité.

La **décision de détection** repose sur des tests statistiques. Dans le cas zéro-bit, une similarité cosinus entre le vecteur décodé et le vecteur recherché est comparée à un seuil, avec une p-valeur explicite exprimée via la fonction bêta incomplète régularisée. Dans le cas multi-bits, le nombre b de bits correctement décodés sur n_{bits} suit une loi binomiale sous l'hypothèse nulle d'obtenir ce résultat à partir d'une image non tatouée, fournissant là encore une p-valeur exacte. Ces grandeurs fondent les métriques de confiance utilisées dans nos travaux de tatouage.

Contributions détaillées

Explication passive avec BLIP2IDC

Nous nous plaçons sur l'axe de l'**explication**, plus riche en sémantique que la simple détection. La tâche d'IDC consiste à décrire textuellement, de façon fine, les différences entre deux images. Un obstacle majeur tient à la taille des modèles vision-langage, parfaitement adaptés à ce cas d'usage : BLIP2 (J. Li, D. Li, Savarese, and S. C. H. Hoi 2023)

compte 2,7 milliards de paramètres, rendant tout entraînement complet hors de portée. Nous suivons donc une approche d'ajustement par matrices de rang faible (LoRA) (Edward J Hu et al. 2022), n'entraînant qu'un sous-ensemble minimal de paramètres. Couplée à une astuce simple de concaténation des deux images, cette stratégie atteint l'état de l'art à faible coût.

Pour aller plus loin, nous mobilisons les modèles génératifs comme outils d'**augmentation synthétique** dans un régime de rareté des données. Nous observons que remplacer intégralement les données réelles par des données synthétiques donne des résultats comparables, au prix d'un facteur dix sur le volume nécessaire. Compte tenu des progrès rapides des modèles génératifs depuis, ce résultat est d'autant plus pertinent. La principale limite de cet axe, l'accès à l'image originale, motive directement le deuxième axe.

Tatouage sémantique avec SWIFT et LatentSeal

SWIFT. Pour s'affranchir de l'accès à l'image originale, l'idéal serait d'insérer l'image dans elle-même par tatouage. C'est impossible : une image $256 \times 256 \times 3$ représente environ 1,5 Mo, qu'une compression JPEG agressive ramène au mieux à 20 Ko, tandis que les meilleures méthodes multi-bits (Fernandez, Sablayrolles, et al. 2022; Zhu et al. 2018a) plafonnent à une trentaine de bits, et les plus récentes comme ChunkySeal (Petrov et al. 2025) à 1 Ko, au détriment de la vitesse. Nous insérons donc une représentation compacte de l'image : sa légende, choix cohérent avec l'axe précédent. Le message est compressé par un codeur arithmétique (Pasco 1977) alimenté par les probabilités de symboles d'un OPT-125M via LLMZip (Valmeekam et al. 2023a; S. Zhang, Roller, Goyal, Artetxe, Moya Chen, S. Chen, Dewan, M. T. Diab, et al. 2022), gagnant 30 % de capacité, puis modulé dans $\mathbb{R}^{256}$ par une modulation TCCSK (Dillard et al. 2003). Un ajustement du modèle de tatouage basé apprentissage profond HiDDeN (Zhu et al. 2018a) permet d'apprendre à insérer ce vecteur, complétant le pipeline. Le protocole imaginé permet à Alice de cacher la légende de son image avec une clé secrète, puis de confier cette clé à un régulateur Bob, qui pourra comparer la légende extraite au contenu observé pour en vérifier l'intégrité sémantique. Nous atteignons l'état de l'art et apportons, grâce à la géométrie de l'espace des messages, une métrique de confiance attestant à l'inférence de la fiabilité du message décodé.

LatentSeal. SWIFT souffre de deux limites : faible vitesse de décodage et faible capacité, causées respectivement par le coût de l'étape LLMZip et les interférences en-

tre porteuses de la modulation TCCSK. Profitant de la montée en capacité du domaine, nous construisons sur le modèle de tatouage VideoSeal (Fernandez, Elsahar, et al. 2024) et l'ajustons pour insérer des vecteurs dans $\mathbb{S}^{255}$. Notre apport tient à l'**auto-encodeur de texte** qui produit un vecteur décodable : le jeton `[CLS]` d'encodeur de texte ModernBERT-base est projeté sur $\mathbb{S}^{255}$ par un réseau de neurones à couches, avec un ajout de bruit visant à simuler celui de l'extraction imparfaite du tatouage. Une analyse extensive des architectures et hyperparamètres de décodage fixe la meilleure configuration. Par rapport à VideoSeal, LatentSeal décode bien plus vite et possède une meilleure capacité. Nous reprenons l'analogue de la métrique de confiance de SWIFT pour assurer la fiabilité du système. Puis nous exploitons la normalisation de l'espace latent pour appliquer une matrice de rotation aléatoire (avec échantillonnage conditionné par une clé) inversible, masquant le contenu à qui ne possède pas la clé. LatentSeal permet ainsi de comparer de façon fiable, rapide et sûre un message extrait avec l'image reçue.

Robustesse du tatouage à l'édition GenAI avec WRD

Coût forensique de la suppression. Les attaques de suppression de tatouage les plus récentes, telles WMForger (Souček et al. 2025), laissent des traces perceptibles. Nous cherchons à déterminer si elles sont trivialement détectables. Après comparaison d'extracteurs de caractéristiques modernes, nous retenons ConvNextV2 (Woo et al. 2023) comme modèle pour calculer les caractéristiques des images tatouées puis attaquées. Ce modèle constitue la colonne vertébrale de notre détecteur forensique. À partir des modèles de tatouage VideoSeal et TrustMark réglés à 40 dB pour mieux garantir la robustesse, nous tatouons 5000 images, appliquons quatre attaques (Y. Liu et al. 2024; Nie et al. 2022; Souček et al. 2025; Xie et al. 2025), entraînons notre détecteur puis calibrons sur 100 000 images pour fixer des taux de fausses alarmes précis. Nous obtenons de bonnes performances, avec un fort taux de vrais positifs à faible taux de faux positifs : les attaques intentionnelles sont bien détectables.

De l'attaque dédiée à l'édition d'image. Nous étendons ensuite l'étude à l'édition d'image, désormais omniprésente et accessible (réseaux sociaux, Gemini intégré aux smartphones, modèles d'édition d'OpenAI parfois ouverts à l'échelle d'un pays). En réutilisant le détecteur forensique précédent, nous montrons que l'édition d'image constitue une menace *incomplète* : soit elle modifie disproportionnellement l'image support et devient trivialement détectable, soit elle échoue à retirer le tatouage et manque son but. Une

étude temporelle des tendances de l'édition d'image confirme que la littérature, en cherchant à préserver au maximum la qualité de l'original favorise le tatouage. Ce résultat rend nos travaux antérieurs d'autant plus pertinents qu'ils ne sont pas contournés par un sous-produit de l'édition, à savoir l'effaçage du tatouage.

Conclusion

Synthèse des contributions

Cette thèse développe des outils pour détecter et expliquer les altérations sémantiques des images. Le chapitre 2 et le chapitre 3 dressent l'état de l'art en matière de modèles vision-langage et de tatouage d'images ; les trois chapitres de contribution traitent ensuite, tour à tour, de la description des altérations, de la transmission de l'information nécessaire pour les repérer, et de la mesure de la robustesse du tatouage numérique face aux attaques modernes.

Le chapitre 4 aborde l'axe *explication*. BLIP2IDC adapte un grand modèle de légendage à l'*Image Difference Captioning* par ajustement LoRA et une simple astuce de concaténation d'images, atteignant l'état de l'art à faible coût. Pour pallier la rareté des données, il mobilise des modèles génératifs pour l'augmentation synthétique, et montre que les données synthétiques peuvent intégralement remplacer les données réelles, au prix d'un facteur dix.

Le chapitre 5 lève le besoin de l'image originale en y insérant une représentation compacte d'elle-même, sa légende, directement dans l'image. SWIFT (Sec. 5.2) compresse la légende, la module, puis l'insère à l'aide d'un modèle de tatouage ajusté nommé HiDDeN, améliorant la robustesse et la capacité par rapport aux références multi-bits tout en fournissant une métrique de confiance sur le message décodé. LatentSeal (Sec. 5.3) s'appuie sur le cadre de SWIFT en exploitant un modèle de tatouage plus récent, VideoSeal, et un auto-encodeur de texte pour insère un latent décodable sur $\mathbb{S}^{255}$, atteignant un décodage nettement plus rapide à plus forte capacité, assorti d'un schéma de sécurité fondé sur une rotation.

Le chapitre 6 pose la question de savoir si retirer un tatouage laisse une trace. À partir des *features* de ConvNextV2, nous entraînons un détecteur forensique contre quatre attaques de suppression et le calibrons pour des taux de faux positifs contrôlés, atteignant un fort taux de vrais positifs à faible taux de faux positifs. Nous étudions le comportement

de la tâche de détection de suppression de tatouage et constatons qu'elle se distingue nettement de la tâche de détection de contenus GenAI. Une extension journal étudie ensuite les attaques par édition d'image, préoccupantes en raison de leur large disponibilité sur smartphones et sur les plateformes en ligne. Nous montrons que les attaques d'édition d'image à l'état de l'art constituent une menace *incomplète* : soit elles altèrent suffisamment le support pour être aisément détectées, soit elles échouent à retirer le tatouage.

Limitations

BLIP2IDC repose sur un modèle de légendage de très grande taille, qu'un modèle plus récent et plus compact pourrait égaler voire surpasser, par exemple FastVLM (Vasu et al. 2025) ; l'augmentation synthétique impose par ailleurs une asymétrie d'un facteur dix entre données réelles et synthétiques. Sur le tatouage sémantique, l'usage d'un modèle de représentation textuelle pour comparer légendes originale et éditée ne répond pas à la question « cette altération est-elle problématique ? » ni « cette similarité est-elle significative ? » : ce choix découle de notre philosophie d'outils réutilisables en aval plutôt que sur-spécialisés, qui ne seraient adaptés qu'à un unique cas d'usage et inutilisable pour tous les autres. Enfin, comme tout système d'apprentissage profond, nous restons vulnérables aux attaques adverses qui représentent une limitation classique, hors du périmètre de cette thèse.

Problèmes ouverts

Plusieurs questions restent ouvertes. La plus fondamentale est définitionnelle : qu'est-ce qui rend une altération sémantiquement problématique, et contre quelles transformations un tatouage doit-il être robuste ? Une *fragilité sélective*, c'est-à-dire un tatouage volontairement vulnérable à certaines éditions, pourrait même être souhaitable dans certains cas (tatouage comme tampon d'authentification), mais définir ces éditions dépend du cas d'usage. Sur le plan pratique, réduire les besoins en augmentation synthétique, améliorer la défense contre les attaques en boîte noire et déployer le tatouage sémantique à l'échelle d'une plateforme demeurent des problèmes ouverts.

Perspectives

Extension de nos outils

Image Difference Captioning Nos travaux sur l'IDC laissent des défis ouverts. Le premier est la *robustesse* du modèle de légendage. Si une simple compression JPEG modifie sensiblement la légende, cela devient problématique — une propriété jamais évoquée dans la littérature IDC. Par ailleurs, un déploiement à l'échelle suppose de réduire le coût d'une telle solution. Reste à déterminer si cette amélioration passera par la quantification du modèle, une architecture plus efficace ou une définition plus fine des différences : nous le laissons à de futurs travaux.

LatentSeal : un caméléon Une perspective intéressante ouverte par ce travail est l'insertion, dans les images, de *features* de haute dimension issus d'extracteurs très efficaces. Tout vecteur de la bonne dimension (dans $\mathbb{S}^{255}$) peut être inséré par LatentSeal : des *features* de parole, de visage ou encore des transcriptions textuelles peuvent ainsi être embarqués dans la vidéo correspondante afin de contrer l'édition. Ce dernier travail constitue l'aboutissement de notre axe de tatouage sémantique et ouvre plusieurs pistes de recherche : exploiter cette charge utile dans davantage de cas d'usage, et projeter des *features* sur $\mathbb{S}^{255}$ sans trop dégrader les performances, restent à étudier. Dans cette optique, LatentSeal suggère des implémentations possibles pour l'authentification multimodale, comme l'insertion de *features* audio dans une séquence vidéo. Plusieurs défis se posent pour la vidéo, autour de la vérification des propriétés du tatouage dans ce cadre : est-il robuste aux algorithmes de compression, comment tatouer imperceptiblement un support en mouvement, tatouer un grand nombre d'images fournit-il à un adversaire assez d'information pour briser le tatouage, et la capacité est-elle meilleure ou moindre que pour les images ?

Tatouage et régulation

Conformité aux régulations L'IA fait l'objet d'une régulation croissante. En Europe, elle prend la forme de l'*EU AI Act*, dont l'article 50 (European Parliament and Council of the European Union 2024) dispose que tout contenu généré ou édité au moyen de l'IA doit être *marqué* dans un format lisible par machine. Le tatouage est un moyen de marquer un contenu, mais imparfait : son détecteur n'est généralement pas public, alors que l'EU AI Act l'exigera. Cela soulève la question de la sécurité dès lors que quiconque

peut interroger le détecteur et mener une attaque en boîte noire pour retirer le tatouage. L'interopérabilité est par ailleurs évoquée : plusieurs acteurs devraient être chargés de la détection du marquage et pouvoir vérifier mutuellement leurs sorties. La manière de protéger la propriété intellectuelle dans ce cadre reste incertaine.

Faire appliquer les régulations La Californie introduit une interdiction de la suppression des métadonnées du côté des fournisseurs de GenAI (Wicks, Becker, and Zbur 2025), qui entrera en vigueur en janvier 2027. Une métadonnée désigne ici un registre C2PA, un tatouage ou tout signal portant des données de provenance attaché au média. Les plateformes ne devront donc pas fournir d'outils permettant de retirer de telles métadonnées. Pour faire appliquer cette régulation, disposer d'outils de détection des attaques de suppression est essentiel. Pour le tatouage, nous avons amorcé cette direction avec Evennou and Kijak 2026, mais un meilleur contrôle du taux de faux positifs est nécessaire pour un déploiement fiable à grande échelle, à des fins d'audit des fournisseurs de GenAI.

Le tatouage au service de la confiance dans les médias

Le tatouage pour restaurer la confiance dans les médias Tout au long de cette thèse, le tatouage s'est imposé comme une solution naturelle pour insérer une information ensuite exploitée afin de signaler un changement. Reste à savoir si ce signalement peut être déployé sans heurts pour communiquer et prévenir les retombées d'une information trompeuse. La philosophie que l'on peut envisager pour mettre en œuvre un tel système, et que nous soutenons, consiste à utiliser le tatouage comme outil de modération en amont de la publication de l'image. Cette idée découle de la loi de Brandolini, selon laquelle il est bien plus difficile de défaire les effets de la désinformation que de la propager : signaler préventivement les contenus suspects peut donc être bénéfique. Contrôler en amont ce que partagent les entités influentes constitue une étape significative. Un exemple récent où cela aurait pu être utile s'est produit lors de la campagne présidentielle française de 2026 (Gabet 2026), lorsqu'un responsable politique a réalisé un montage de séquences de réunion d'un adversaire afin de le discréditer. Comme toujours, la publication trompeuse initiale a atteint des millions de personnes tandis que le démenti est passé quasi inaperçu.

Une autre perspective consiste à insérer le tatouage directement dans les systèmes de prise de vue, comme c'est déjà le cas dans certains appareils photo Nikon [1]. Il s'agit d'une collaboration avec l'Agence France-Presse permettant aux reporters d'amorcer la

1. Nikon Corporation 2024.

chaîne d'authenticité dès le début du pipeline de données, l'acquisition de l'image, en y ajoutant des informations de source (qui) et de provenance (où et quand). Appliqué massivement, jusqu'aux téléphones des utilisateurs ordinaires, ce principe pourrait aider à regagner une part de la confiance dans les images, tragiquement perdue ces dernières années. Cela ouvre la porte à des défis supplémentaires de sécurité, de passage à l'échelle et de frugalité du système de tatouage. Grâce à sa modularité, nos travaux de thèse se transposeront aisément si un tel paradigme est adopté à l'avenir.

L'amélioration de la GenAI L'essentiel de cette thèse s'est déroulé au début de l'« ère GenAI », ce qui laisse attendre des gains considérables de qualité et/ou de performance après la publication de ce manuscrit. Notre dernier travail Evennou and Kijak 2026 le valide : les modèles d'édition ont déjà beaucoup gagné en qualité. À l'heure d'écrire ces lignes, mi-2026, une nouvelle tendance émerge avec l'édition de vidéo, du côté propriétaire avec Gemini-Omni [2] comme du côté des initiatives à poids ouverts telles que Bernini (Chenchen Liu et al. 2026). Cela permet de passer à l'échelle l'édition de vidéo de haute qualité, autorisant des séquences toujours plus réalistes, ou toujours plus trompeuses. Plus largement, il en va de même pour la parole [3], ce qui nous rappelle que l'objectif de détecter et d'expliquer l'altération sémantique ne se limite pas au médium image : explorer d'autres modalités est une évolution naturelle de cette thèse.

2. Kavukcuoglu 2026.
3. ElevenLabs 2026; O. Team 2026.

Table of Contents

Acknowledgments **i**

Résumé substantiel en français **iii**

1 Introduction **1**

1.1 Misinformation: the menace 2

1.1.1 Progress in Generative AI 2

1.1.2 The Need for Detection and Explanation Tools 3

1.2 Motivation and Challenges 5

1.3 Outline and contributions 7

2 Related Work **9**

2.1 Multimodality in the Deep Learning Era 10

2.1.1 Taming text 10

2.1.2 Laying the Foundation for Image 11

2.1.3 Visual Language Models 13

2.1.4 Describing Image Differences 14

2.2 Detection of edited contents 15

2.2.1 Image Generative Models 15

2.2.2 Image Editing 17

2.2.3 Forensics 18

2.3 Image Watermarking 22

2.3.1 Old-fashioned image watermarking 22

2.3.2 Deep-learning based watermarking 22

2.3.3 Watermark Removal Attacks 23

2.4 Encoding Text into a Compact Payload . . . 24
2.4.1 Text Compression . . . 24
2.4.2 Text Autoencoders . . . 26

3 Technical Background on Deep-Learning Watermarking for Images 28
3.1 Properties and definitions . . . 29
3.2 Evaluation . . . 29
3.3 The encoder–decoder architecture . . . 31
3.4 Statistical detection tests . . . 33

4 Reframing Image Difference Captioning with BLIP2IDC and Synthetic Augmentation 37
4.1 Introduction . . . 38
4.2 BLIP2IDC . . . 39
4.2.1 Adaptation . . . 40
4.2.2 Efficient fine-tuning . . . 41
4.2.3 Advantages over existing IDC models . . . 41
4.3 Datasets . . . 42
4.3.1 Existing datasets . . . 43
4.3.2 IDC evaluation . . . 47
4.3.3 IDC task issues . . . 50
4.3.4 Synthetic augmentation . . . 51
4.4 Experiments . . . 53
4.4.1 Evaluation protocol . . . 53
4.4.2 BLIP2IDC evaluation . . . 54
4.4.3 Automatically generated dataset . . . 57
4.4.4 Qualitative analysis . . . 58
4.4.5 Limitations . . . 60
4.5 Conclusion . . . 60

5 Semantic Watermarking for Authentication 63
5.1 Introduction . . . 64
5.2 SWIFT: Semantic Watermarking for Image Forgery Thwarting . . . 65
5.2.1 Method . . . 66
5.2.2 Experiments & Results . . . 72

5.3 Fast, Secure, and High-Capacity Image Watermarking with Autoencoded Text Vectors . . . 77
5.3.1 Method . . . 78
5.3.2 Experimental setting . . . 83
5.3.3 Watermarking model robustness and perceptibility . . . 84
5.3.4 Text Autoencoder . . . 86
5.3.5 End-to-end comparison . . . 91
5.4 Conclusion . . . 99

6 Detecting Watermark Removal **100**
6.1 Introduction . . . 101
6.2 Method . . . 102
6.3 Experiments & Results . . . 106
6.3.1 Implementation Details . . . 106
6.3.2 Removal attacks effectiveness and perceptual quality . . . 107
6.3.3 Detection Performance . . . 111
6.3.4 Extension to Image Editing . . . 114
6.4 Conclusion . . . 120
6.4.1 Limitations . . . 120
6.4.2 Discussion . . . 121

7 Conclusion **123**
7.1 Summary of Contributions . . . 124
7.2 Perspectives . . . 124
7.2.1 Extending our tools . . . 125
7.2.2 Watermarking and regulation . . . 126
7.2.3 Watermarking for media trust . . . 126

Appendices **128**

A Reframing Image Difference Captioning with BLIP2IDC and Synthetic Augmentation **129**
A.1 BLIP2 . . . 129
A.1.1 Implementation details . . . 129
A.1.2 Memory requirements . . . 130

A.1.3 LLM limitations . . . 130
A.2 Syned . . . 133
A.2.1 Limitations . . . 133
A.2.2 Samples . . . 133

B Fast, Secure, and High-Capacity Image Water- marking with Autoencoded Text Vectors 141
B.1 Autoencoder details . . . 142
B.1.1 Architecture . . . 142
B.1.2 Training . . . 142
B.2 Watermark embedder-extractor details . . . 144
B.2.1 Architecture . . . 144
B.2.2 Training . . . 145
B.3 Rotation with secret key . . . 146
B.3.1 Sampling Uniformly Random Rotations on $\mathbb{S}^{255}$ . . . 146
B.3.2 Impact on performances . . . 148
B.4 Qualitative examples . . . 149
B.4.1 Text data samples . . . 149

Bibliography 155

List of Figures

1.1 Evolution of generative AI image synthesis. 3

2.1 Comparison of MLM and autoregressive pretraining objectives. 11
2.2 Vision Transformer (ViT). Borrowed from Dosovitskiy et al. 2021. 13
2.3 DDPM (left) and DDIM (right) generation steps. The DDPM is notably markovian and requires each denoising step to be performed. The right figure depicts a non-markovian process, allowing to skip states and thus generation steps, this is DDIM's core idea. Borrowed from J. Song, Meng, and Ermon 2021. 16
2.4 Latent Diffusion Model. Borrowed from Rombach et al. 2021. 18
2.5 Diverse image editing results from InstructPix2Pix, from Brooks, Holynski, and Alexei A Efros 2022. 19
2.6 Progress in instruction-based image editing, shown on a single EMU-Edit (Sheynin et al. 2023) test case (id 1088) for the instruction *"change the background to a college campus."* The reference input (red frame) is edited by seven models released between 2022 and 2026, arranged chronologically. Early models leave the target largely intact while degrading foreground elements (IP2P) or replace the background with a coarse, low-fidelity structure (MagicBrush), often disturbing the foreground. Recent models (FLUX.2 Klein, FireRed 1.1) synthesize coherent, photorealistic campus backdrops while preserving the subjects, their pose, and scene lighting, illustrating the steady gain in edit fidelity and instruction adherence over time. 20
2.7 Fourier Spectrum of generative models. Borrowed from Corvi et al. 2023. . 21

2.8 Process of Arithmetic Encoding and Decoding (Encoder). 25
2.8 Process of Arithmetic Encoding and Decoding (Decoder). 26
2.9 Illustration of the learned latent geometry by autoencoders before and after introducing x perturbations. With high-capacity encoder/decoder networks, a standard autoencoder has no preference over x–z couplings and can thus learn a random mapping between them (*Left*). Trained with local perturbations $\mathcal{C}(x)$ (word deletion), they learn to map similar x to close z to best achieve the denoising objective (*Right*). Borrowed from Shen et al. 2020. 27

3.1 Comparison of PSNR. Watermarked = 40 dB, WMForger= 29 dB and CTRLRegen=24 dB. WMForger is the worst version of the image but have the best PSNR among attacks. 30
3.2 Encoder-decoder paradigm for multi-bit deep learning watermarking. . . . 32
3.3 Zero-bit detection geometry on the unit hypersphere. 34

4.1 Standard IDC pipeline. 40
4.2 BLIP2IDC end-to-end pipeline. We first resize and concatenate the image inputs before feeding them to the BLIP2 architecture finetuned using LoRA. 41
4.3 CLEVR and CLEVR-DC 3D images. Changes comprises color, texture and position modifications. Figure from H. Kim et al. 2021. 44
4.4 Examples from the Spot-the-Diff dataset. Surveillance-style image pairs where the difference is a change in the scene (appearing, disappearing or moving objects). 45
4.5 Examples from the Image Editing Request (IER) dataset. Each pair shows an original image and its user-requested edit (face swap, background removal, object insertion). 46
4.6 Synthetic dataset creation pipeline leveraging a prompt-based image editing and large language models. 52
4.7 Qualitative samples of image editing 52
4.8 Ablation Study of BLIP2IDC's CIDEr according to finetuned modules on different datasets. 55

4.9 Attention maps from BLIP2IDC ViT for different attention heads. The ViT tries to attend to corresponding parts of the images and is able to find new entities in the scenes. The overlay image assigns weights to pixels based on their attention scores: brighter pixels indicate higher attention. . 58
4.10 Qualitative samples of state-of-the-art Image Difference Captioning models. 59
4.11 Ethnic bias 60
4.12 Geographic bias 61

5.1 Overview of our Semantic Watermarking pipeline. 66
5.2 Length distribution of BLIP2 captions using LLMZip. 67
5.3 Hide-$\mathbb{R}$ architecture. 68
5.4 Performance of different encoding schemes under additive white Gaussian noise. 70
5.5 Representation of the message space after modulation. 74
5.6 Examples of recovered messages under different transforms. 76
5.7 LatentSeal overview illustrated through an image tampering detection application. 77
5.8 Text autoencoder architecture. 80
5.9 Qualitative samples of LatentSeal. 88
5.10 Autoencoder decoding speed vs LLMZip. 89
5.11 ROC of $\ell = -\log_{10} \rho$ on non-catastrophic transforms on Wikitext/PixMoCap. 89
5.12 Semantic fidelity under 21 image transforms on COCOVal2017 with PixmoCap sentences. 89
5.13 BLEU4 and SBERT similarity between original and retrieved watermarks after image editing on cover. 92
5.14 Samples of images edited with HiDream 93
5.15 Novelty study of the tokens in text datasets. 95
5.16 Estimation of bit length using LLMZip. 96
5.17 Proportion of over-length samples. 97

6.1 Pipeline of Watermark Removal Detection 103
6.2 Watermark removal attacks samples, with residuals and Fourier spectrum. 105
6.3 Detector robustness under post-processing. 110
6.4 Watermark removal rate per image editing models. 115

6.5 Mean log-ratio of the 2D Fourier spectrum of each editor's perturbation relative to the average across all editors (local edits only). 116
6.6 LPIPS and compound success rate performance of Image Editing Models. . 116

A.1 Samples. The top row from left to right is the original and the Emu Edit version. 135
A.2 Local modification sample. While the wreath is not in the middle window, our variations still provide diversity so that the model can generalize better. 136
A.3 Color modification sample. Each variation respects the original prompt. . . 137
A.4 Text modification samples. The diffusion model was not able to write text. 138
A.5 Style modification samples. In most cases we struggle to perform the edit and when we suceed it change the whole scenery. 139
A.6 Global modification samples. While the prompt has been mostly respected, we observe highly contrasted images with very different scenery than the original image. 140

List of Tables

2.1 Landscape of robust image watermarking methods. Payload capacity and baselines 23

4.1 IDC and IE datasets: settings, editing methods, and ground-truth counts. . 43
4.2 CIDEr scores on CLEVR-Change for the 5 categories of changes. 54
4.3 Image Captioning results on CLEVR-DC, STD and IER 55
4.4 BLIP2IDC scores on IER with training only on other datasets. Other models are listed to provide current state-of-the art results obtained with in-domain training. We exhibit state-of-the-art performance on the main metric, CIDEr. 56
4.5 BLIP2 zero-shot performance on real-world IDC datasets 57
4.6 CIDEr scores on Emu Edit test set. 57
4.7 CIDEr Performance Metrics by Change Category and Dataset Version. . . 59

5.1 Message Recovery Rate by Transform Type. 73
5.2 Message Recovery Rate and Cumulative Distribution Function on Emu Edit for different ρ thresholds with SWIFT. 75
5.3 Robustness comparison on PixMo-Cap. 85
5.4 Scaling factor α sweep and corresponding perceptibility metrics with cosine similarity between embedded watermarks and extracted ones. 86
5.5 Watermark model evaluation 87
5.6 VideoSeal vs. LatentSeal (Ours) comparison. 90
5.7 BLEU4 scores at different sentences size. 90
5.8 Ablation study of the autoencoder. 91
5.9 PSNR statistics for HiDream and ip2p image-editing models. 92

5.10 SBERT similarity scores demonstrating graceful degradation against 21 transforms on COCOVal2017 images with PixmoCap sentences. 94
5.11 Comparison of p-values between standard methods in watermarking. 98
5.12 Operating thresholds for our confidence metric 98

6.1 Image quality metrics between original and variant images. 108
6.2 Watermark survival and forensic detectability. 109
6.3 Watermark Removal Detection rates. 109
6.4 Watermark-agnosticism of the detector. 112
6.5 Leave-one-attack-out generalization results for the forensic detector on COCO test split. 113
6.6 AEROBLADE baseline vs. our forensic detector in the leave-one-out setting (average AUC on COCO test split, per attack). 113
6.7 Comparison of defense strategies against watermark removal. 114
6.8 Perceptual quality of edited images. 117
6.9 Zero-shot compound success rate (%) per editor at FPR = 10^{-3} 118
6.10 Watermark survival and Zero-shot Forensic TPR rates (%) 119
6.11 Watermark survival and Zero-shot Forensic TPR rates (%) per editor, broken down by edit type (local vs. global). 119

A.1 BLIP2IDC hyper-parameters according to the different IDC datasets. . . . 130
A.2 Memory requirements for different versions of BLIP2 with batch size = 1. LoRA Rank are specified with a float16 version of BLIP2. 130
A.3 Performance Metrics by change category and dataset version 134

B.1 Autoencoder architecture with implementation details. 143
B.2 Dataset statistics. We use custom splits for **PixMo-Cap** and **COCO-2017** validation/test. 143
B.3 Hyperparameter search for the Text Autoencoder training at 30 tokens on Wikitext. 145
B.4 Watermarker training hyperparameters 151
B.5 Data augmentations and their parameter ranges used during training. . . . 152

B.6 Median per-matrix generation time after warm-up. `torch_batched` benefits from GPU and batching; `torch_keyed` adds per-item generator setup for deterministic, secret-key–conditioned streams. All methods are Haar-correct on $SO(d)$. 153

1

Introduction

> *The amount of energy needed to refute bullshit is an order of magnitude bigger than that needed to produce it.*
>
> — Alberto Brandolini, 2013

Contents

1.1 Misinformation: the menace 2

1.1.1 Progress in Generative AI 2

1.1.2 The Need for Detection and Explanation Tools 3

1.2 Motivation and Challenges 5

1.3 Outline and contributions 7

1.1 Misinformation: the menace

The past decade has seen generative models advance from producing recognizable artifacts to synthesizing images indistinguishable from photographs (Cai et al. 2025; Goodfellow et al. 2014; Labs 2024, 2025; Mordvintsev, Olah, and Tyka 2015; Stability AI 2022). Driven by large-scale training data, more capable architectures, and the wide availability of image-editing tools embedded in everyday platforms, the ability to alter visual content has moved from specialists to anyone with a smartphone. The same progress that fuels creativity also lowers the cost of producing misinformation: an image can now be edited to change what it means, faster and more convincingly than ever, while the effort required to detect and refute such alterations grows in the opposite direction, following Brandolini's law (Brandolini 2013). This tension motivates the present work. In this chapter, we first situate our subject within the broader problem of misinformation, then we motivate the need for detection and explanation, and their respective challenges, before outlining the contributions and structure of this manuscript.

Misinformation is rampant. It has already been used during national elections to sway political opinions or attack political opponents popularity (Haeck and Hartog 2025) through fabricated media. With the advent of generative tools, propelled by Deep Learning breakthroughs, misinformation is scalable and easily accessible to anyone with a computer. Even worse, high quality models are available on high-end smartphones, making it easy for anyone to edit an image using Gemini, Google's own generative model. While the practice of strategic deception is ancient, Sun Tzu's *Art of War*(Tzu 1910) already framed warfare as fundamentally based on deception, the term *disinformation* itself is far more recent. Following Fallis (2015), it denotes misleading information whose function is to mislead, which distinguishes it from *misinformation*, where the misleading content may be spread without the intent to deceive. This manuscript both tackles *misinformation* and *disinformation* as it is thorny to determine the intent behind misleading content.

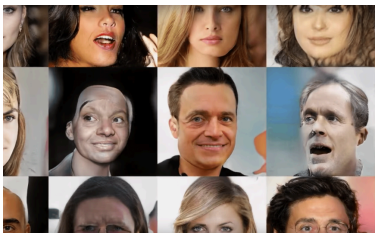
1.1.1 Progress in Generative AI

Due to the rise of Deep Learning (Dosovitskiy et al. 2021; Goodfellow et al. 2014; LeCun, Boser, et al. 1989; Rombach et al. 2021; Sohl-Dickstein et al. 2015; Vaswani et al. 2017), Generative AI (GenAI) has thrived. While in the beginning only low-quality and artifact-heavy images were able to be generated, see Fig. 1.1, recent models reach photo-

Foundational



(a) DeepDream feature visualisation (Mordvintsev, Olah, and Tyka 2015).



(b) GAN-generated faces (Goodfellow et al. 2014).



(c) Animals generated by StableDiffusion (Rombach et al. 2021).

Recent



(d) Dog generated by Flux 2 (Labs 2025).



(e) Dog generated by GPT Image 2.



(f) Man generated by Gemini.

Figure 1.1 – Evolution of generative AI image synthesis.

realistic quality. Especially, we are interested in models able to edit images. Deepfakes or faceswap are ancestors of these models, while modern ones are able to take a prompt and an image as inputs to output an edited image. This prompts further scrutiny, as it enables to use any existing image and edit it to bypass image retrieval filters on social networks, or to spread misinformation. While educating people on GenAI is important, it is not enough to guarantee trust into existing content, as we discuss next.

1.1.2 The Need for Detection and Explanation Tools

A telling illustration of why human judgment is no longer sufficient occurred in May 2026, when an artist posted one of Monet's *Water Lilies* on X (formerly Twitter), stated it was "Made with an AI" specialized in Monet's work, and asked users to explain what made the "generated" image inferior to a real Monet (SHL0MS 2026). Thousands obliged: painters and critics produced detailed takedowns of the composition, brushstrokes, and color, dissecting what was in fact an authentic masterwork from Monet. The post reached

millions before the deception was revealed. A single unverified label was enough to reorganize aesthetic judgment, and once viewers had classified the image as synthetic the instinct to verify disappeared. This is a toy failure, but one can imagine what happen when news images are tagged as AI and vice versa. This is the failure mode that authenticity infrastructure must address: a provenance claim that can be freely asserted, spoofed, or removed does not inform perception so much as steer it.

Moreover, generative and editing models are openly accessible on platforms such as HuggingFace and GitHub ((Labs 2025; Stability AI 2022; S. I. Team et al. 2026; C. Wu et al. 2025)), making misuse cheap and widespread. In response, the EU AI Act mandates the labeling of synthetic content: Article 50 requires that AI-generated or manipulated images, audio, and video be marked in a machine-readable format and disclosed as artificial (European Parliament and Council of the European Union 2024). At the same time, platform-side safeguards are eroding, as Meta has scaled down its fact-checking and Trust & Safety operations (France 24 2025).

The Monet episode shows why declarative labels alone are not trustworthy and motivates content-bound robust signals. Watermarking appears most suited to address this issue, as it enables to take an image as input and hide an imperceptible and highly robust signal in it. That is why regulatory enforcement strongly considers watermarking generated and natural images to make provenance verifiable.

While watermarking alone is a necessary tool to build back trust, it usually focuses on authenticity and ownership, not on explanation. We mean explanation in the semantic way, no link with the XAI field which aims to understands inner mechanisms of deep-learning models. Being able to explain at scale what happened to an image or communicate enough information to do so is important to counter fake news. On one side detection is useful to flag dubious content and on the other side explanation unlocks the ability to understand what was changed in an image and by doing so to tell whether this change is acceptable or not, given a certain system of values. Detection and explanation are both representative of different philosophy; passive — which do not adds any signal to content and aims to analyze it after processing to obtain specific information — and active — which purposely introduces into the content a signal conveying information for transmission. These two axis are at the core of the tools we investigate in this thesis: Image Difference Captioning to automatically explain in a passive fashion the change between two versions of an image through Vision Language Models (VLM) and Semantic Watermarking to communicate actively meaningful information for a future semantic

verification of the watermarked image.

1.2 Motivation and Challenges

In this section we detail the use cases we consider, corresponding challenges and research questions explored throughout this thesis.

Explaining and detecting semantic alterations is a rather underexplored avenue in modern computer vision literature. Adjacent literature is mostly focused on Generative AI detection through image watermarking and digital forensics. Narrow fields for explainability do exist, namely Image Difference Captioning, but are rather unknown. Image change detection in itself is non-trivial, as it prompts to define and detect an adequate signal.

The first task investigated is Image Difference Captioning, which aims to learn to describe the difference between an original image and its edited version. Modern approaches automatically generate such description. It has mainly been used for satellite imagery to monitor landscapes evolution (Chouaf et al. 2021) or for medical imagery (F. Liu et al. 2021). We consider a setting where an original image is archived in a database and then published on social networks. Later, crawling the web with a Content-Based Image Retrieval (CBIR) system finds redistributed copies, possibly edited, of that image. Detecting whether, and how, a copy was modified then serves two ends: protecting the owner against copyright infringement and supporting fact-checking when the content is political or news-adjacent.

This introduces our first challenge: describing differences requires deciding how to compare different versions of an image, a question with no clear consensus. Previous work in Image Difference Captioning rather uses a dual-stream approach, extracting features from each version of the image before proceeding to a fusion step where most of the literature is focused on improvement. Another challenge is the lack of ad-hoc data, with two images and the textual description of the change. At the beginning of this thesis, the lack of these datasets providing extensive information about edits was a significant hurdle. The literature bypassed this data scarcity regime by leveraging 3D rendering engines such as Blender or Unity to generate variations of an image, at the cost of staying in the toy setting. Those observations lead to the subsequent research questions:

RESEARCH QUESTION 1

Which space is best suited to represent the difference between two images?

RESEARCH QUESTION 2

How can we cope with the lack of suitable training data in Image Difference Captioning?

Assuming such a dataset exists, the dependence on the source image still hurts adoption. In the best case it must be retrieved through a CBIR system, with all the attendant difficulty and no guarantee of success; in the worst case it is simply unavailable. For instance, a reporter named Alice publishes an image on a social network A. Eve steals the image, modifies and publishes it at for her own benefit on a social network B, which is moderated by Bob. Bob would like to assess if the image is trustworthy or not, and has access to nothing but the image version in the social network B. In this realistic setting, Bob cannot tell anything about the semantic integrity of the image.

To sidestep this source-image dependence, one turns to Semantic Watermarking, the second tool studied in this thesis. Where standard watermarking embeds an arbitrary payload — typically an identifier, Semantic Watermarking embeds a payload whose content describes the image itself, so that a single received image can be checked against its own embedded reference. Note that 'semantic' here qualifies the payload, not the embedding space, in contrast with latent-space schemes such as Tree-Rings (Wen et al. 2023) which sometimes use the same name. How to define the content of the message embedded by the watermark remains a challenge, especially on the capacity-side. Some use cases, require high capacity, like describing the semantic content of an image, while for others only few bits are enough, like identifiers. Moreover, ensuring the integrity of the message at the reception end is crucial for trust. Given those challenges, one uncover the following research questions:

RESEARCH QUESTION 3

How can we robustly and securely embed a textual description of an image into itself?

RESEARCH QUESTION 4

How can we guarantee the trustworthiness of the received message?

Using image watermarking as a tool to enforce regulations seems adequate, but the

literature prompts caution, as GenAI models, based on Latent Diffusion Models, are reportedly able to remove watermarks, either by regenerating the image or by editing some part of it. It remains unclear under which settings those observations transfer to more modern attack/defense scenarios and, if so, if they are a death sentence to future regulations. Hence, the associated research question we focus on is:

RESEARCH QUESTION 5

To what extent do GenAI-based removal attacks and editing models succeed against digital watermarking?

Overall, these challenges and gaps expose limitations in the literature that raise the five research questions outlined.

1.3 Outline and contributions

This thesis introduces methods to answer the different challenges outlined before, through both passive and active approaches. Chapter 2 first presents related work. Chapter 3 then covers the technical background on watermarking. The contributions are organized as follows.

- **Chapter 4: Reframing Image Difference Captioning with BLIP2IDC and Synthetic Augmentation.** This chapter tackles the *passive* case, where the pristine image is available. It adapts an Image Captioning model to Image Difference Captioning by encoding image pairs at the pixel level in a single stream rather than through the usual two-stream scheme, letting the model attend to differences early while keeping the architecture untouched. Remarkably, we leverage GenAI to provide training samples countering data scarcity.
  > Based on: *Evennou et al., Reframing Image Difference Captioning with BLIP2IDC and Synthetic Augmentation.* published at WACV2025.
  > Code at: github.com/gautierevn/BLIP2IDC

- **Chapter 5: Semantic Watermarking for Authentication.** This chapter moves to the *active* case, where the original image is unavailable, and embeds a semantic description of the image into itself. Our first work of this chapter, SWIFT,

focuses on the introduction of the Semantic Watermarking task, aiming at optimizing message capacity and trustworthiness. The second work of this chapter, LatentSeal, builds upon recent advancements in the Image Watermarking literature and change the message compression algorithm from SWIFT by a more scalable one, with adapted security and trustworthiness procedures.
> Based on: *Evennou et al., SWIFT: Semantic Watermarking for Image Forgery Thwarting*, published at WIFS2024, and *Evennou et al., Fast, Secure, and High-Capacity Image Watermarking with Autoencoded Text Vectors*, 2025., under review.
> Code at: github.com/gautierevn/swift_watermarking

- **Chapter 6: Detecting Watermark Removal.** This chapter studies the robustness of the watermarking ecosystem itself. It uncovers that watermark removal attacks, while degrading the watermark without visible distortion, leave distinct statistical traces. It formalizes Watermark Removal Detection (WRD) as a third evaluation axis alongside attack success and perceptual quality, and shows that a lightweight classifier flags removals and image editing models spanning 2022–2026.
> Based on: *Evennou and Kijak, The Forensic Cost of Watermark Removal*, 2026 published at IH&MMSec 2026 and its extension *Evennou and Kijak, The Forensic Cost of Watermark Removal: From Dedicated Attacks to Image Editing*, under review.

2

Related Work

Contents

2.1 Multimodality in the Deep Learning Era 10
2.1.1 Taming text 10
2.1.2 Laying the Foundation for Image 11
2.1.3 Visual Language Models 13
2.1.4 Describing Image Differences 14
2.2 Detection of edited contents 15
2.2.1 Image Generative Models 15
2.2.2 Image Editing 17
2.2.3 Forensics 18
2.3 Image Watermarking 22
2.3.1 Old-fashioned image watermarking 22
2.3.2 Deep-learning based watermarking 22
2.3.3 Watermark Removal Attacks 23
2.4 Encoding Text into a Compact Payload 24
2.4.1 Text Compression 24
2.4.2 Text Autoencoders 26

Below, we cover the background on Vision Language Models, Diffusion Models, and Image Watermarking that the rest of the manuscript builds on.

2.1 Multimodality in the Deep Learning Era

Vision Language Models are models able to take both image and text as inputs, and answer to specific questions regarding the prompt and the associated image. They are the result of an evolution coming from text tokenizer to current models able to answer elaborated questions. This section explores their history and provides insights on the different breakthroughs achieved to reach this point. We leverage those advances both in Chapter 4 which focuses on the description of differences between images, in Chapter 5 which focuses on designing a heavily semantic payload for image watermarking and lastly in Chapter 6 which aims at detecting traces of watermark removal in watermarked images.

2.1.1 Taming text

Learning how to generate human-like sentences is the dream of the Natural Language Processing (NLP) field. The bedrock of this endeavor is the design of the tokenizer (Gage 1994; Marcus, Marcinkiewicz, and Santorini 1993), which splits a sentence in words or sub-words called "tokens". A modern tokenizer would be the Byte-Pair Encoding one (Gage 1994), which divides a text corpus into letters, or tokens, and computes what is the most occurring consecutive pair of token. It is merged, and the algorithm iterates until reaching a suitable vocabulary size. This vocabulary provides the atomic units fed to subsequent architectures. Then the vectorization step projects the tokenized sentence into a vector for future use cases. Classical vectorization relies on one-hot encoding, bag-of-words (Harris 1954), or TF-IDF (Spärck Jones 1972). These representations are sparse, high-dimensional, and treat every token as independent, no two tokens share any notion of similarity. This limitation motivated the shift toward learned dense embeddings. Recurrent neural networks (RNNs) (Graves 2013; Hochreiter and Schmidhuber 1997) were long the standard for sequence modeling tasks such as translation, but their sequential nature limited parallelization and made long-range dependencies hard to capture. Seminal work on the attention mechanism through the Transformer architecture (Vaswani et al. 2017) enabled parallelization and a direct path between tokens. The main training paradigm became two-phased (Devlin et al. 2019): the pretraining step, which involved massive use of data and a specific objective, and the finetuning step to adapt to the downstream task. Improved transformer pretraining objectives such as Masked Language Modeling (MLM) (Devlin et al. 2019; Taylor 1953), which leverages bidirectional context, and Autoregressive Generation (Radford and Narasimhan 2018), which models the distribution of the

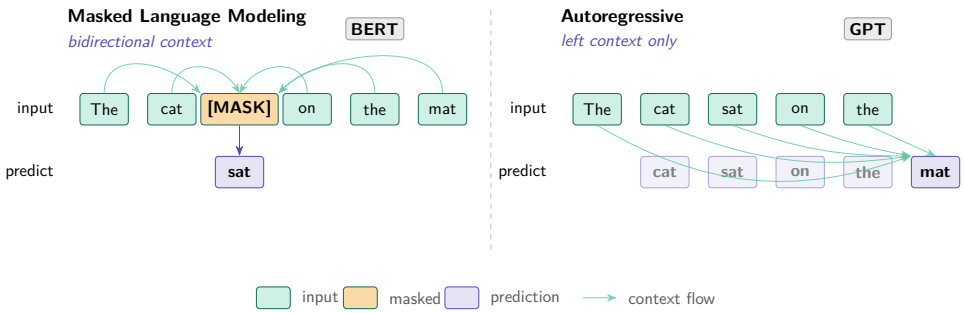


Figure 2.1 – Comparison of MLM and autoregressive pretraining objectives.

next token given all previous ones, fueled significant strides toward better representations. Fig. 2.1 better depicts the two objectives.

Following those two objectives, encoder-only and decoder-only architectures thrived, with their main representatives being Bidirectional Encoder Representations from Transformers (BERT)(Devlin et al. 2019) and Generative Pretrained Transformer (GPT)(Radford and Narasimhan 2018). On one side, the encoder-only architecture is most suited for representational tasks, like information retrieval or classification (sentiment analysis, spam filtering, Named Entity Recognition). On the other side, the decoder-only path is especially efficient to answer questions, following a specific *prompt*, as demonstrated by GPT2-3 (Brown et al. 2020; Radford, Jeff Wu, et al. 2019). Modern NLP is now mostly focused on autoregressive generation aimed at answering questions, for which Reinforcement Learning from Human Feedback (RLHF) enables specializing the model to better behave to answer questions (Ouyang et al. 2022). The first industrial success came from the ChatGPT[1]. This sparked the Large Language Model era, which is still on today. Improvements are mainly focused on data quality, better compute/data scaling and reinforcement learning for chain-of-tools, reasoning, test-time reasoning.

2.1.2 Laying the Foundation for Image

After the rise of the Transformers architecture (Vaswani et al. 2017) for the text modality, the community aimed at leveraging this breakthrough to learn visiolinguistic representations. One of the first manifestations of this endeavor was VilBERT, which used a standard CNN alongside with a BERT-like encoder (J. Lu et al. 2019; Ren et al. 2015) as features extractors and a parallel transformer layer switching *keys* and *values* inputs

1. OpenAI 2022.

from each modality to enable multimodal conditioning of their respective outputs. This results in a two-stream set of features that respectively was the product of the interaction with the other domain, h_v for vision features and h_w for word features. Consequently, the finetuning phase was the mere training of an MLP on top of (h_v, h_w) for classification tasks. But CNN lacked the ability to ingest debilitatingly large amount of data without plateauing performances.

That is why a Transformer-native image feature extractor was designed with Vision Transformers (ViT) (Dosovitskiy et al. 2021) and validated they benefit more from data scaling (Dosovitskiy et al. 2021), while classic Convolutional Networks such as ResNet (He et al. 2015; LeCun, Boser, et al. 1989) plateau during pretraining. Alike text, the image is divided into patches, which are then projected into feature space. They become the image version of tokens with this patch tokenization. Unfortunately, it still produces myopic representation of the image, as all patches are 16×16 pixels wide, while some objects can be smaller than those patches. This is suited for classification (identifying the main subject of the image) but when looking for detection (locating *where* objects are in the image using bounding boxes) or segmentation (labeling exactly which pixels belong to which class) one needs higher granularity of the image representation. SwinTransformer (Ze Liu et al. 2021) proposes to add hierarchical representation with 4×4 patches that progressively get merged back to the 16×16 dimension. By going through this process, the network is able to represent the input image with respect to more details at the earlier stage, without discarding the global meaning of the image at the 16×16 scale. Concurrently, DINO (Caron et al. 2021) found 8×8 patches with self-distillation pretraining of the Transformer architecture achieved very dense representation of an image, allowing for better performance on segmentation or detection.

In an effort to bring back CNNs to the spotlight, ConvNext (Zhuang Liu et al. 2022) investigates whether the success of Vision Transformer, especially the SwinTransformer version, came from the superiority of the attention mechanism or rather from the continual improvement of the training recipe. The conclusion was that, when given improvement from the CNN literature (Howard et al. 2017) alongside with the closest training recipe from the ViT, ConvNext was on-par with SwinTransformer. Like a distant parent of the MLM objective in NLP, ConvNextV2 (Woo et al. 2023) was trained using the Fully Convolutional Masked AutoEncoder (FCMAE). The idea is to use an autoencoder, meaning an encoder and a decoder, and perform a sparse convolution on the encoder side — meaning a fraction of the image is not involved in the convolution computation — while

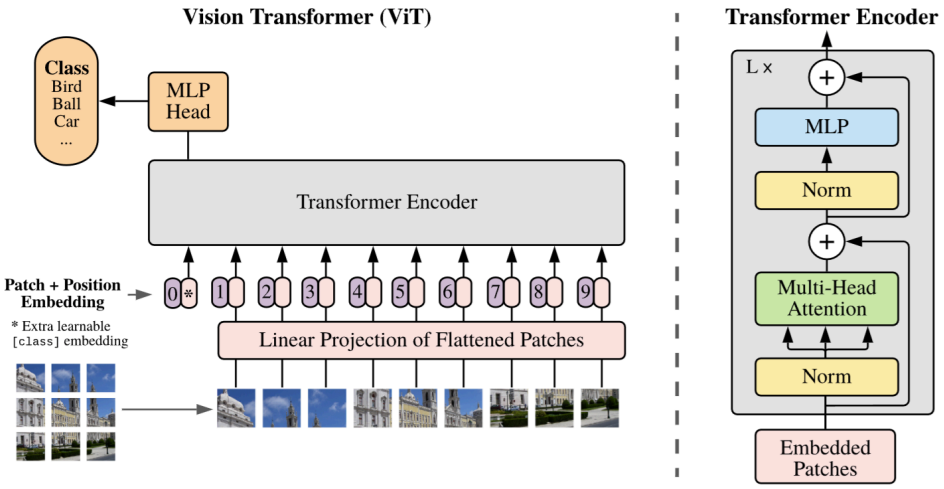


Figure 2.2 – Vision Transformer (ViT). Borrowed from Dosovitskiy et al. 2021.

requiring from the decoder the ability to reconstruct the full image, even the pixels not considered by the sparse convolution.

2.1.3 Visual Language Models

On the encoder side, multimodality was the next step, with CLIP (Radford, J. W. Kim, et al. 2021) which enabled to find a common embedding space to represent text and images via a contrastive learning objective. First, consider an image and its caption as respective positives, and in a batch of n elements compute cosine similarities between each pair (features(Image), features(caption)). Then each embedding is repelled from all the other negatives in the batch following a contrastive loss. This has the effect to construct a space in which you can query a caption such as *a cat sat on the mat* and obtain an image close from the meaning of the query. It has been used for image retrieval, captioning and explainability (Mokady, Hertz, and Bermano 2021; Odom 2022; Radford, J. W. Kim, et al. 2021).

On the generation side, Image Captioning models such as BLIP, BLIP2 and OFA (J. Li, D. Li, Savarese, and S. C. H. Hoi 2023; J. Li, D. Li, Xiong, et al. 2022; P. Wang et al. 2022) aim to describe the content of an image with fine-grained captions. They are trained on web-scale datasets (Hodosh, Young, and Hockenmaier 2013; T.-Y. Lin, Maire, Belongie, Bourdev, et al. 2015) to harness as much visual knowledge as possible, and connect text and images with various solutions: syntactic trees from image features

(Mitchell et al. 2012), RNN decoding of CNN features with (K. Xu et al. 2015) or without attention (Bahdanau, Cho, and Bengio 2015; Vinyals et al. 2015), pre-training tasks using object-based anchors (X. Li et al. 2020), Seq2Seq learning with unified vocabulary for all the linguistic and visual tokens (P. Wang et al. 2022) or leveraging additional embeddings to make a bridge between vision tokens and text tokens (J. Li, D. Li, Savarese, and S. Hoi 2023). Multimodal models specialized in VQA use cross-modal attention to tune the visual features extraction to the text prompt, enabling interaction and focusing the model on information relevant to the question. While highly efficient, they are prone to inconsistency due to their prompt dependency (Touvron et al. 2023; Weihan Wang et al. 2024). A model of particular interest for the rest of this manuscript is BLIP-2 (J. Li, D. Li, Savarese, and S. Hoi 2023), an Image Captioning model, which introduces the Querying Transformer (QFormer) as the main block to connect text and image representations. Chapter 4 adapts BLIP-2 to the task of describing image differences and introduces a better adaptation of its visual world knowledge by comparing images in the pixels space instead of the features space.

2.1.4 Describing Image Differences

Image Difference Captioning (IDC) (Park, Darrell, and Rohrbach 2019) is a recent approach that goes beyond traditional image analysis by generating detailed textual descriptions of the differences between two images. It finds applications across various fields, from detecting subtle changes in satellite imagery (Chouaf et al. 2021; Jhamtani and Berg-Kirkpatrick 2018; Chenyang Liu et al. 2023; Qiu et al. 2021) to identifying anomalies in medical imagery (F. Liu et al. 2021) or misinformation explanation. IDC focuses on telling apart the discrepancies between two near-identical images. The differences changing the meaning of the image (semantic changes) are the ones the model should focus upon. On the contrary, editorial changes (non-semantic) such as compression or rescale should be ignored. Prior work struggles with two main hardships: how to represent the difference features and how to gather this specific kind of data. Deep learning based architectures, such as CNN, CLIP-based encoding and RNN are widely used in this topic to learn the features (Chouaf et al. 2021; Hosseinzadeh and Y. Wang 2021; Jhamtani and Berg-Kirkpatrick 2018; Chenyang Liu et al. 2023; Qiu et al. 2021; Shi et al. 2020a; L. Yao, Weiying Wang, and Jin 2022). The standard procedure is to extract features for each image independently, thus discarding the images' correlation in the pixel space. After this extraction step, they fuse embeddings using concatenation before feeding them to either a

difference encoder for change representation or directly to a decoder to generate the text descriptions (Park, Darrell, and Rohrbach 2019; Qiu et al. 2021; Tu, L. Li, Su, Zha, et al. 2023; L. Yao, Weiying Wang, and Jin 2022). Another axis of work is to define pretraining tasks to find a suitable representation space for the differences (Tu, L. Li, Su, Zha, et al. 2023). Recently, the use of a multimodal model such as CLIP (Radford, J. W. Kim, et al. 2021) enables for a better representation. Yet, current adaptation to IDC (Guo, T.-J. Wang, and Laaksonen 2022) struggles with exploiting all its pretraining. Moreover, because high-quality data is hard to find, most methods are trained and evaluated on 3D rendered datasets, that do not accurately represent real-world performance. Chapter 4 addresses both points.

2.2 Detection of edited contents

A convenient way to frame the detection of semantic alterations is through the detection of edited content. To ground this, we trace the progression of image generative models and how they work. The rest of the manuscript treats them as a black box for generality, but here we open it up. We also cover recent work on detecting their outputs, which could be leveraged for our own goal: the image *forensics* field offers useful insights into these models' properties.

2.2.1 Image Generative Models

Prior to the surge in editing models, generative AI first made significant strides with Generative Adversarial Networks (Goodfellow et al. 2014).Such networks work in pairs: a generator G and a discriminator D are trained jointly with opposing objectives. D learns to distinguish real images from generated ones, while G maps a random noise vector z to an image $G(z)$ and is trained to fool D. This adversarial game drives G to progressively approximate the training data distribution, until its samples become indistinguishable from real images. At inference, a noise vector is sampled and fed to G to produce a new image. A key weakness of vanilla GANs is the lack of conditioning: they offer no control over what is generated — one samples noise and obtains an arbitrary sample from the data distribution. For instance, subsequent work tried to improve conditioning by input concatenation in cGAN (Mirza and Osindero 2014) or by adding an auxiliary classification loss (Odena, Olah, and Shlens 2017), so that at inference the operator could encode "1"

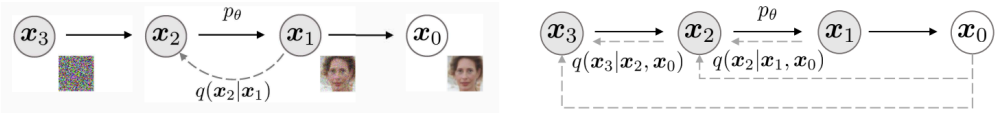


Figure 2.3 – DDPM (left) and DDIM (right) generation steps. The DDPM is notably markovian and requires each denoising step to be performed. The right figure depicts a non-markovian process, allowing to skip states and thus generation steps, this is DDIM's core idea. Borrowed from J. Song, Meng, and Ermon 2021.

in a one-hot vector, concatenate it with the sampled noise and proceed to the generation of the corresponding hand-written digit from the MNIST dataset (LeCun, Cortes, and Burges 2010). Then D predicts if the image is real and its class label. But the lack of training stability of GANs, due to the generator/discriminator adversarial balancing, limited their adoption — while they remain useful especially for super-resolution methods (Ledig et al. 2017; X. Wang et al. 2018).

The next evolution came with diffusion models, beginning with Denoising Diffusion Probabilistic Models (DDPM) (Ho, Jain, and Abbeel 2020). The term *diffusion* is a reference to thermodynamics (Sohl-Dickstein et al. 2015) and models a Markov process adding Gaussian noise to an image x_0 for T consecutive steps, with x_T being fully noised. This forward pass is followed by the reverse pass, aimed at denoising x_T into x_0. This procedure constitutes the training phase of the diffusion model that learns to model $p_\theta(x_{t-1}|x_t)$. At inference, a noise is sampled and denoised progressively into a plausible image. Denoising Diffusion Implicit Models (DDIM) (J. Song, Meng, and Ermon 2021) introduced the idea of training on the full number of steps, but only generating using a subset τ of those steps, for instance 50 steps. This works due to the closed-form approximation of the noise between x_T and x_0. Because it is only an approximation, several steps are still needed to achieve satisfactory results. Fig. 2.3 illustrates DDPM and DDIM differences in generation. Now we detail Latent Diffusion Model (LDM), as it is the backbone of several editing models studied throughout this manuscript. Rombach et al. 2021 found that denoising in pixel space, as done in DDPM and DDIM papers (Ho, Jain, and Abbeel 2020; J. Song, Meng, and Ermon 2021), is a waste of compute as the same process can be done in latent space. They first use an image compression model $\mathcal{E}$, either a Variational Autoencoder (Kingma and Welling 2014) or a Vector Quantized GAN (VQGAN) (Esser, Rombach, and Ommer 2021), to project the high-dimensional image x_0 into a low-dimension latent vector z. Then z goes through the diffusion process, either DDPM at training time or DDIM at inference, into z_T. The core idea is to leverage the spatial structure of z, which

is shaped as $64 \times 64 \times 4$, with a most-suited U-net (Ronneberger, P. Fischer, and Brox 2015) to train. The U-net in itself is an elaborated autoencoder with downsampling/upsampling transforms and skip connection between encoding and decoding steps allowing for efficient learning with few samples. Moreover, it enables better representation as it takes advantage from the inductive bias in CNNs to ensure better local spatiality conservation from the original input. z goes through T steps of denoising into $\hat{z}$ and finally the decoder part $\mathcal{D}$ of the image compression model decodes it back to an approximation of x, $\tilde{x}$. Importantly, to be able to use prompts, image edges or embeddings as conditioning at generation time, a condition encoder $\mathcal{T}_\theta$ project its input into a compatible latent space to enable to enable either concatenation with the latent representation z or integration via cross-attention layers, similarly to the procedure in cGAN (Mirza and Osindero 2014). The difference lies in the unconditional training (Ho and Salimans 2022), which randomly drops the conditioning signal at training time, see the switch in Fig. 2.4. This unconditional training enables the application of Classifier-Free Guidance (CFG) during inference: a hyperparameter that controls the extent to which the model adheres to the conditioning (e.g., a text prompt or reference image). This process is ubiquitous even in modern models and is widely used by image editing models that are evoked in Chapters 4 to 6. LDMs led to the release of the famously known StableDiffusion [2], which enabled in generation watermarking. The last step came with the rectified flow transformer (Esser, Kulal, et al. 2024; Lipman et al. 2023) which enables modeling the trajectory as a straight line between the noise distribution and the data distribution, achieving faster generation thanks to fewer generation steps.

2.2.2 Image Editing

The recent progress in generative models (Goodfellow et al. 2014; Rombach et al. 2021) has enabled realistic generation of images. This further propelled work (Brooks, Holynski, and Alexei A. Efros 2023; Meng et al. 2021) on image editing, where semantic modifications are performed to an image using a textual prompt, while keeping the main subject of an image. Recent works (Sheynin et al. 2023; K. Zhang et al. 2023a) focused on using already existing real-world text-image datasets such as MSCOCO (T.-Y. Lin, Maire, Belongie, Bourdev, et al. 2015) to perform editions on those images, by building on modifications of the associated caption.

2. Stability AI 2022.

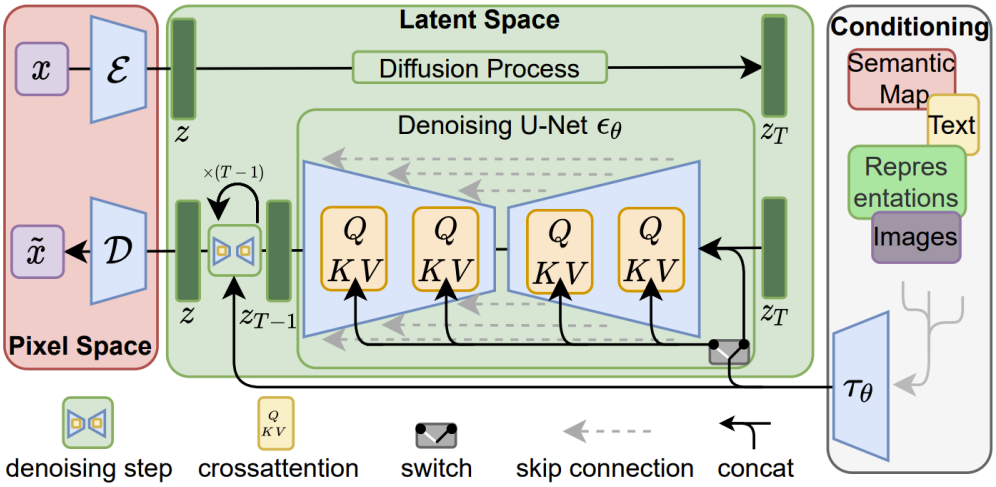


Figure 2.4 – Latent Diffusion Model. Borrowed from Rombach et al. 2021.

Demand for editing models able to perform very light modifications skyrocketed. InstructPix2Pix (Brooks, Holynski, and Alexei A. Efros 2023) introduces classifier-free guidance for both input image and text conditioning. This is fundamentally different from LDMs as the authors assume the network will have a reference image to edit as input. Then the model has to learn to respect the structure of the original image *and* follow the textual prompt. Additionally, they heavily used synthetic data to gather a dataset of triplets $(image, edit_instruction, caption)$, hence achieving a diverse set of editing, some samples in Fig. 2.5. MagicBrush (K. Zhang et al. 2023a) improved on synthetic data, and subsequent models incrementally improved either image quality or diversity in the space of edits. QwenEdit (C. Wu et al. 2025) scaled the editing model up to 20B parameters. The use of reinforcement learning during training enabled better prompt adherence and image quality, with FireRed (S. I. Team et al. 2026) achieving state-of-the-art performance at low cost, and recently the FLUX.2 klein (Labs 2025) model was released. Figure 2.6 show the evolution of model editing in a cow setting.

Image Editing models are also known to remove watermarks as a byproduct (S. Lu et al. 2025), posing a threat to regulatory initiatives involving watermarking. Chapter 6 evaluates them as attack vectors.

2.2.3 Forensics

Passive forensic methods detect alterations of a piece of content, possibly malicious ones. They utilize noise residuals or high-frequency features as input to highlight ma-

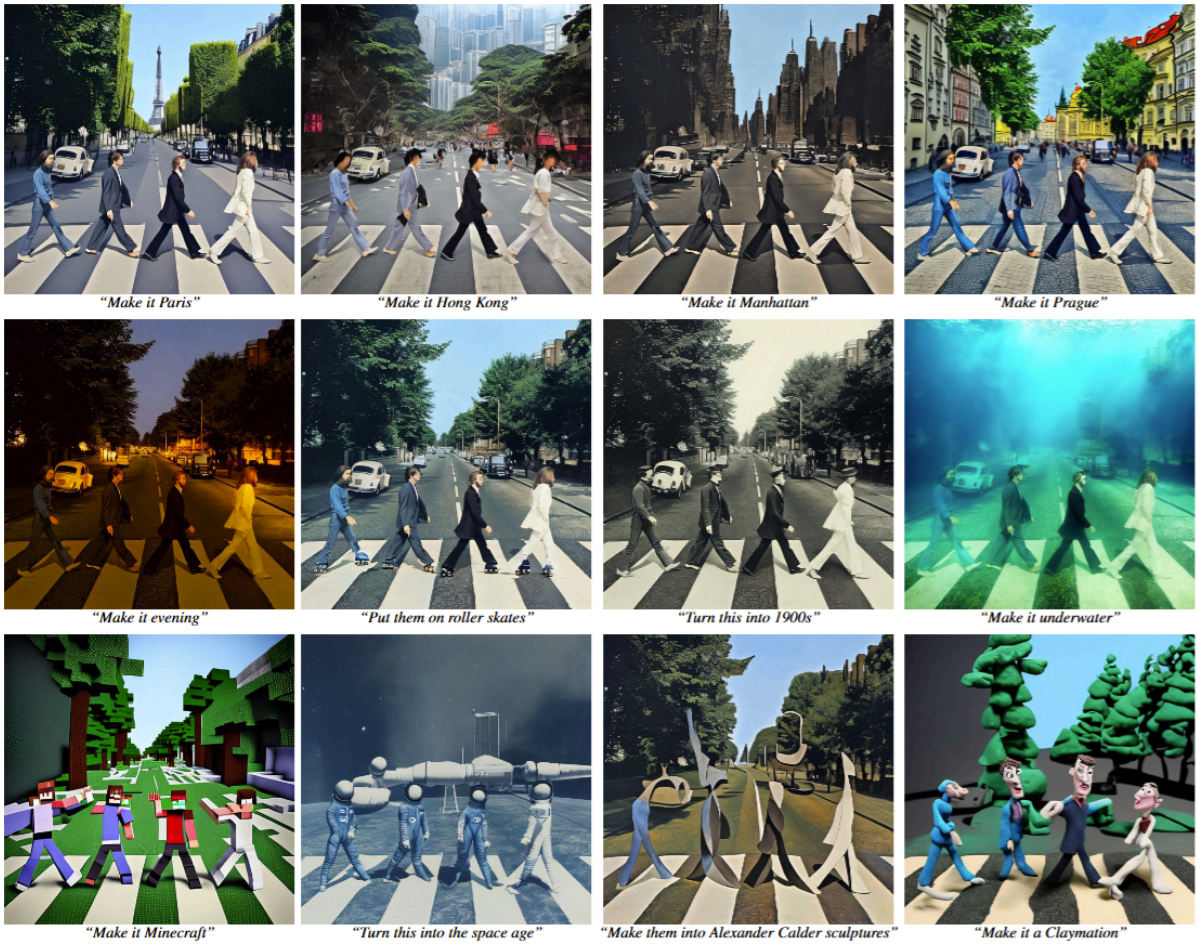


Figure 2.5 – Diverse image editing results from InstructPix2Pix, from Brooks, Holynski, and Alexei A Efros 2022.

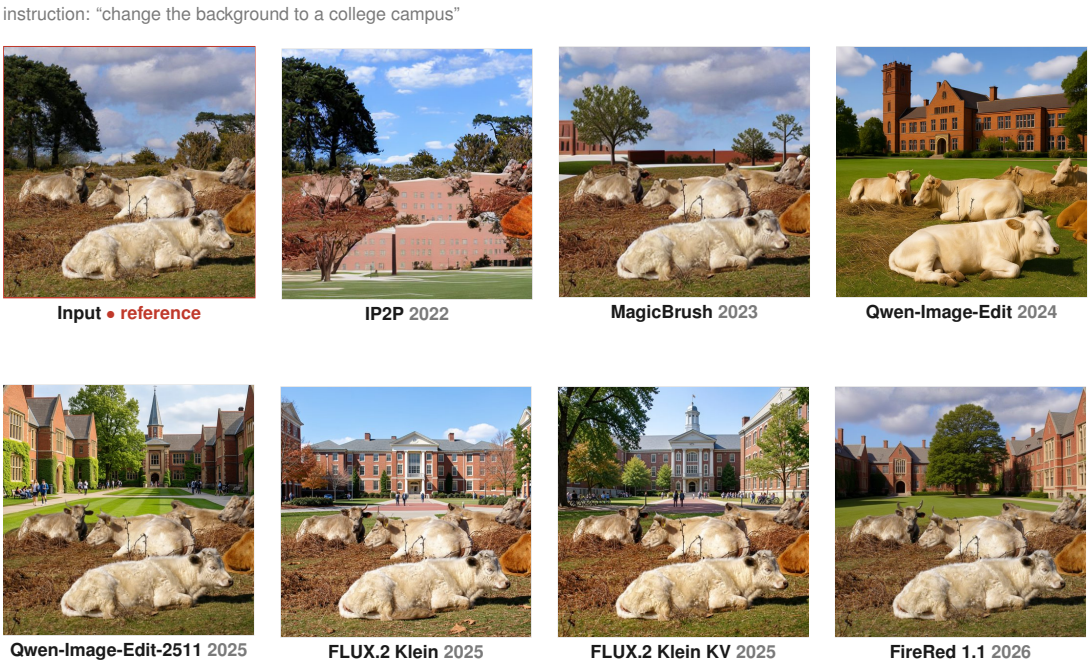


Figure 2.6 – Progress in instruction-based image editing, shown on a single EMU-Edit (Sheynin et al. 2023) test case (id 1088) for the instruction *“change the background to a college campus.”* The reference input (red frame) is edited by seven models released between 2022 and 2026, arranged chronologically. Early models leave the target largely intact while degrading foreground elements (IP2P) or replace the background with a coarse, low-fidelity structure (MagicBrush), often disturbing the foreground. Recent models (FLUX.2 Klein, FireRed 1.1) synthesize coherent, photorealistic campus backdrops while preserving the subjects, their pose, and scene lighting, illustrating the steady gain in edit fidelity and instruction adherence over time.

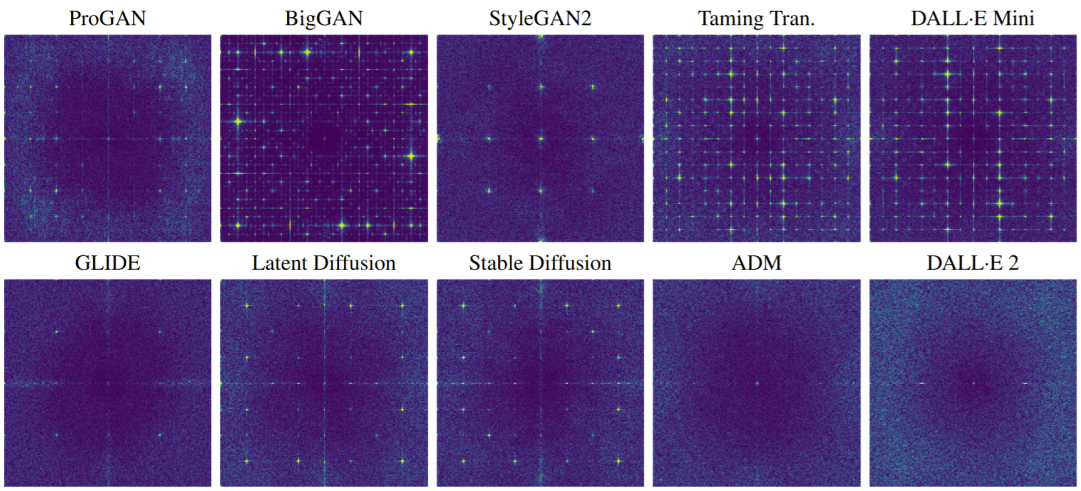


Figure 2.7 – Fourier Spectrum of generative models. Borrowed from Corvi et al. 2023.

nipulation traces. These methods are limited to providing localized insights into specific alterations. Copy-move forgery detection uses Siamese networks (Y. Wu, Abd-Almageed, and Prem Natarajan 2018) while splicing detection leverages two-stream architectures (Y. Wu, AbdAlmageed, and Premkumar Natarajan 2019; Zhou et al. 2018). Inpainting detection methods focus on traces left by deep inpainting models (H. Li and J. Huang 2019). Forensics methods lack the capacity to offer a global semantic perspective due to the absence of contextual information from the original image.

Detecting AI-generated images is a more recent task. Classifiers trained on spectral features, GAN fingerprints, or diffusion-specific artifacts (Corvi et al. 2023; Ricker, Lukovnikov, and A. Fischer 2024; Z. Wang et al. 2023) achieve strong separation between real and synthetic images, though performance degrades under compression and resizing. Fig. 2.7 illustrates specific patterns into generated images that are picked up by detectors. Notably, (Ricker, Lukovnikov, and A. Fischer 2024) scores images by minimum reconstruction error across a set of VAEs, in a training-free fashion. An issue with the use of detectors to assess the nature of an image is the adversarial paradox; being able to detect an image was generated is a strong signal, one that can be used for additional training of the generative model in order to fool the detector, reminding the generator/discriminator training of GANs. Chapter 6 differs in a key respect: we do not distinguish real from AI-generated, but rather genuine clean images from images that have undergone a watermark-removal process — a subtler signal that off-the-shelf GenAI detectors miss.

2.3 Image Watermarking

2.3.1 Old-fashioned image watermarking

Traditional watermarking schemes embed invisible marks within multimedia content to assert copyright ownership (*robust* watermarking) or authenticate content (*fragile* watermarking). A third class, *semi-fragile* watermarking, sits between the two: it should be robust to benign processing such as compression or resizing but fragile to deeper transformations that modify the semantics. Classic techniques involve manipulating spatial or frequency domain representations of the media (I. Cox et al. 2007; Katzenbeisser and Petitcolas 1999), typically hiding the payload in mid- or high-frequency coefficients where the human visual system is less sensitive. Spread-spectrum approaches such as Broken Arrows (Furon and Bas 2008) provide strong theoretical security guarantees but limited capacity. Fragile schemes for content authentication (Fridrich, Goljan, and Baldoza 2000; Utku Celik et al. 2002) are explicitly designed to be destroyed by specific or/and unexpected edits, which is a philosophy we follow in Chapter 5.

2.3.2 Deep-learning based watermarking

Two paradigms coexist in the current deep-learning based watermarking literature: *in-generation* and *post-hoc*. The idea of the *in-generation* is to nudge the output distribution so that it can be detected as watermarked. One of the first iterations of this idea came with StableSignature from Fernandez, Couairon, et al. 2023a, in which they finetuned the VAE part of StableDiffusion to directly generate watermarked images. In this thesis we focus on *post-hoc*, as we find it more convenient to evaluate and cheaper to use as it is not intertwined with the generation process.

End-to-end deep watermarking treats the task as learning an embedder/extractor network pair that hides an L-bit string in a cover image and recovers it even after attacks. Pioneering CNN systems like HiDDeN (Zhu et al. 2018a) introduced differentiable distortion layers to gain robustness to JPEG, cropping and noise; later self-supervised variants (e.g. SSL-Watermark (Fernandez, Sablayrolles, et al. 2022)) leveraged the feature extractor pretraining. Capacity and imperceptibility were boosted by TrustMark (Bui, Shruti Agarwal, and Collomosse 2025), which couples a UNet encoder with learned error-correcting codes to reach up to 100 bits at 98% bit accuracy, while the recent ConvNeXt-v2/UNet VideoSeal hybrid (Fernandez, Elsahar, et al. 2024) extends the idea

to video, with the most recent version able to hide up to 256 bits per frame.

Table 2.1 summarizes standards deep-learning based watermarking methods, their capacity and their perceptibility.

Table 2.1 – **Landscape of robust image watermarking methods.** Payload capacity is the number of bits embedded per image (RAW is a zero-bit detector that only tests for watermark presence). Only *VideoSeal* openly ships a 256-bit robust and highly imperceptible model. MBRS reaches high payloads but relies on heavy distortion (36 dB) and is not robust beyond JPEG compression, which is why we do not use it as a baseline.

Watermarking method	**Payload capacity (bits)**	**PSNR (dB)**
DwtTDcT Al-Haj 2007	**32**	40
RivaGAN K. A. Zhang et al. 2019	**32**	40
StegaStamp Tancik, Mildenhall, and Ng 2020	**32–64**	42
MBRS Jia, Fang, and W. Zhang 2021	**64–625**	36
CIN Ma et al. 2022	**30**	36–39
StableSignature Fernandez, Couairon, et al. 2023b	**48**	30
TrustMark Bui, Shruti Agarwal, and Collomosse 2023	**100**	40–44
RAW Xian et al. 2024	**zero-bit**	27
VideoSeal Fernandez, Elsahar, et al. 2024	**256**	40–44
WAM Sander et al. 2025	**32**/region	40

Chapter 5 reframes the watermark payload from a bit string to a continuous unit-norm vector encoding the semantics of the cover image itself.

2.3.3 Watermark Removal Attacks

A growing body of work targets the removal of watermarks. Watermarks-in-the-Sand (H. Zhang et al. 2024) uses an image-quality oracle, a proxy model telling whether the image is watermarked, to iteratively refine perturbations that degrade the watermark while preserving visual fidelity. WMForger (Souček et al. 2025) achieves state-of-the-art black-box removal by training a preference model via reinforcement learning to distinguish watermarked from clean images, using Fourier-domain transforms as a proxy for watermark behavior. Both methods optimize for attack success rate and distortion amplitude.

A second family relies on generative purification. DiffPure (Nie et al. 2022) introduced diffusion-based purification by projecting the watermarked image onto the generative manifold through a forward–reverse diffusion pass. CtrlRegen (Y. Liu et al. 2024) improves structural preservation via semantic conditioning with ControlNet (L. Zhang, Rao, and Agrawala 2023). VAE-based approaches offer faster but typically weaker al-

ternatives. These methods inherit artifacts of their generative backbone: diffusion-based attacks introduce characteristic frequency signatures, while VAE reconstructions suffer from smoothing (Corvi et al. 2023). Overall, the removal process trades one detectable signal (the watermark) for another (the generative artifact). Chapter 6 formalizes this trade-off and turns it into a defense.

The idea of detecting adversarial modifications has precedent in the adversarial examples literature (Carlini and Wagner 2017), where methods such as feature squeezing (W. Xu, Evans, and Qi 2018) and input-transformation detectors (Nesti, Biondi, and Buttazzo 2023) identify adversarially perturbed inputs. In watermarking, however, the analogous question — *can we detect that a removal attack occurred?* — remains largely unexplored. Existing watermark robustness benchmarks (An et al. 2024) evaluate only whether the watermark survives, not whether the attack itself is detectable. The closest alternative defense is adversarial training of the watermarker against known removal methods (S. Lu et al. 2025), which trades robustness against capacity or imperceptibility. Chapter 6 introduces Watermark Removal Detection as a complementary, post-hoc defense that can be deployed on top of any existing watermarking scheme.

2.4 Encoding Text into a Compact Payload

A central design choice of Chapter 5 is to use the watermark channel to carry not a random bit string but a textual description of the cover image. Doing so requires reducing a sentence to a compact representation that fits the bit-budget of the watermarking scheme while remaining decodable after channel noise. Two complementary lines of work are relevant: lossless text compression, which targets the smallest possible bitstream, and text autoencoders, which target a fixed-size latent vector.

2.4.1 Text Compression

On the lossless side, LLMZip (Valmeekam et al. 2023a) leverages an LLM to feed symbol probabilities to an arithmetic encoder (Pasco 1977), crafting a losslessly compressed variable-length bitstream. The compression ratio is bounded by the cross-entropy of the language model on the input distribution, making it the strongest available baseline for bit-efficient text embedding. The main idea is to use an LLM as an oracle to predict the next token t_i likelihood $P(t_i|t_0, ..., t_{i-1})$. For reference, without an LLM to predict likeli-

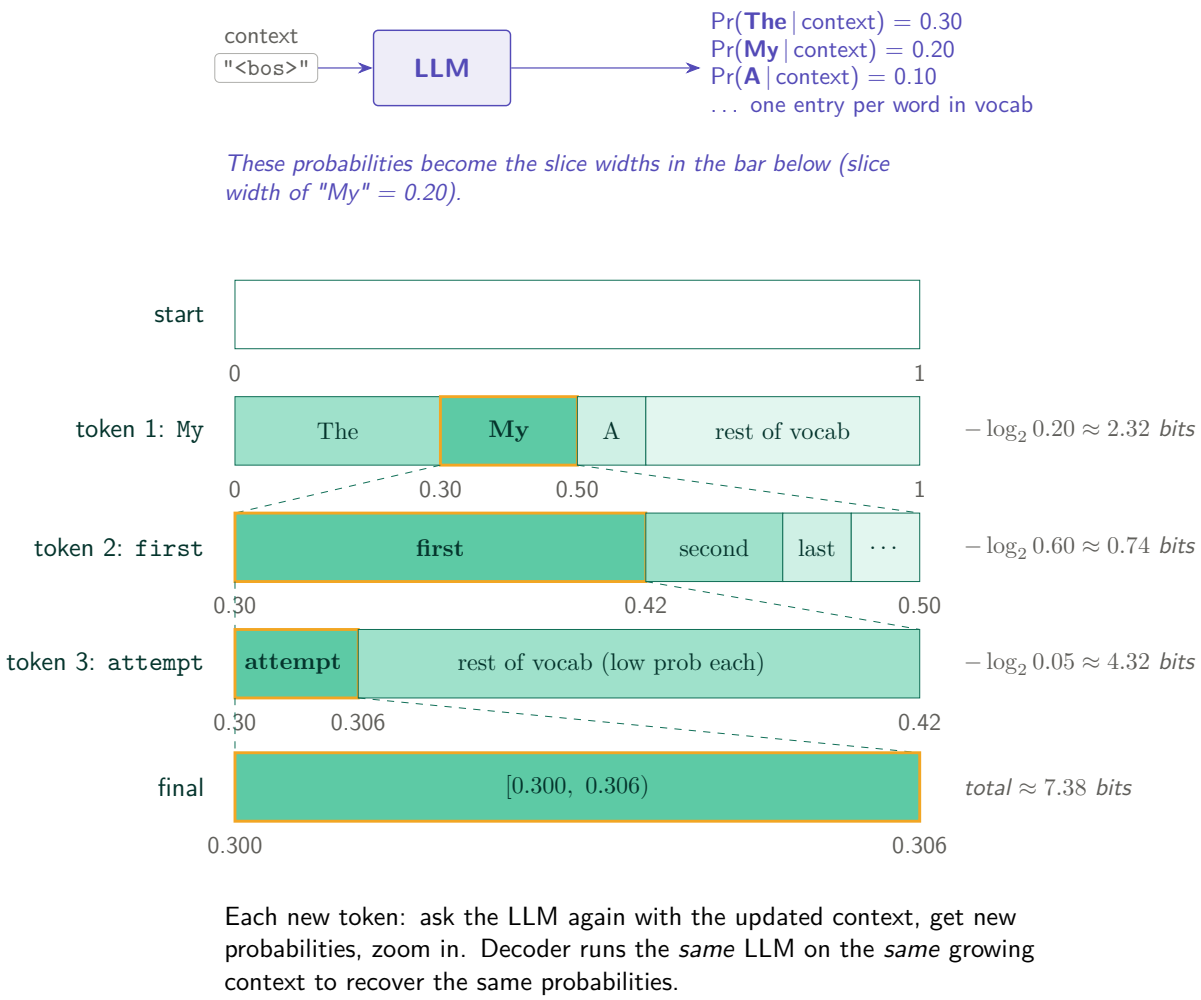


Figure 2.8 – Process of Arithmetic Encoding and Decoding (Encoder).

hood and assuming uniform distribution of tokens for the next token probability, the cost of encoding each token would be $\log_2 |\mathcal{V}|$, with $\mathcal{V}$ being the vocabulary. An interesting technical detail is that LLMs tend to have huge vocabulary size, going up to 128k different tokens, while most of them are not used quite often. This means there is headroom to learn a particular distribution to better weigh each token probability when performing compression. Fig. 2.8 details the mechanism used by the LLMZip to perform the encoding and decoding step. In Chapter 5, we leverage opt-125m (S. Zhang, Roller, Goyal, Artetxe, Moya Chen, S. Chen, Dewan, M. Diab, et al. 2022) as LLM with a vocabulary size of 32,000. In the worst case, the model's performance will cap at $\log_2(32000) \simeq 15$ bits/token, representing uniform uncertainty over the 32,000-token vocabulary. Chapter 5

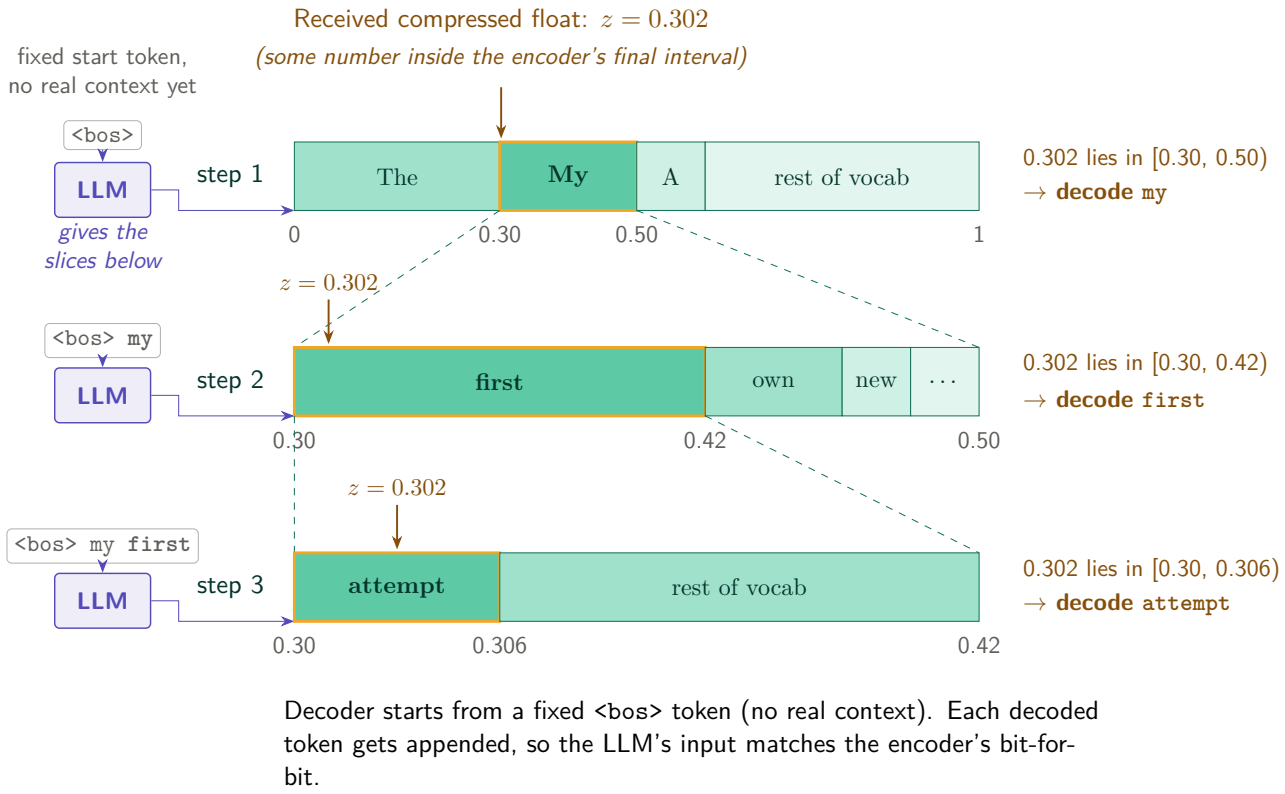


Figure 2.8 – Process of Arithmetic Encoding and Decoding (Decoder).

uses LLMZip both in the message layer of our first scheme (SWIFT) and as the baseline against which our learned vector autoencoder (LatentSeal) is compared.

2.4.2 Text Autoencoders

Modern text autoencoders are typically framed around denoising objectives, where the input is intentionally corrupted, and the model is trained to reconstruct it. Approaches like MASS (K. Song et al. 2019) and BART (Lewis et al. 2019) have demonstrated that masking or permuting text spans yields latent representations that are beneficial for downstream generation tasks. T5 (Raffel et al. 2020) further unifies this concept into a text-to-text framework. Earlier variational approaches, such as the Neural VAE proposed by Bowman et al. (2016) and its refinement by Shen et al. (2020), explicitly aim to encourage smooth semantic latent spaces. One of the main focus of this field is to achieve a coherent latent space: inputs close in meaning must be closer than sentences with very different meaning. Shen et al. 2020 add noise to the input space to enforce such property, resulting

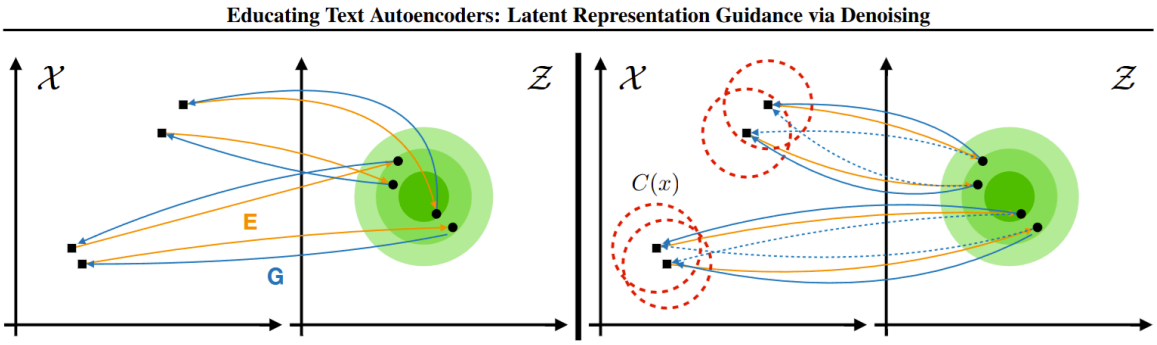


Figure 2.9 – Illustration of the learned latent geometry by autoencoders before and after introducing x perturbations. With high-capacity encoder/decoder networks, a standard autoencoder has no preference over x–z couplings and can thus learn a random mapping between them (*Left*). Trained with local perturbations $\mathcal{C}(x)$ (word deletion), they learn to map similar x to close z to best achieve the denoising objective (*Right*). Borrowed from Shen et al. 2020.

in a more coherent latent space as depicted in Fig. 2.9. Autoencoder approaches never add Gaussian noise to the latent space during training, as they optimize for reconstruction accuracy and this procedure is more aimed at improving robustness, a gap Chapter 5 closes by training a noise-robust autoencoder that maps sentences onto the unit hypersphere.

3

Technical Background on Deep-Learning Watermarking for Images

Contents

3.1 Properties and definitions 29
3.2 Evaluation 29
3.3 The encoder–decoder architecture 31
3.4 Statistical detection tests 33

In this chapter we detail the properties of deep-learning digital watermarking, alongside with standard training procedures and statistical tests for evaluation. These explanations aim to provide a better understanding to the reader of how modern watermarking architectures operate.

3.1 Properties and definitions

The idea of watermarking is to hide a signal within a media, often called *cover*, in this thesis an image. The nature of the signal and its goal define the kind of watermarking, whether *multi-bit* when conveying a sequence of bits or *zero-bit* when embedding a vector holding no information besides its presence. Multi-bit watermarking is about sending messages through the watermark payload while zero-bit enables to have high-statistical guarantee about detection of the watermark. This comes handy when trying to prove provenance of a content. Beyond this dichotomy between zero-bit and multi-bit, they both share the following properties:

- *Robustness* requires watermarks to survive transforms such as JPEG compression, valuemetric transforms (contrast, brightness) or geometric ones (rotation, flips, resize). It ensures no adversary can trivially evade the watermark.
- *Perceptibility* limits the amplitude of the watermark signal so that it remains invisible to the human eye. So finding the best place to hide the signal is required to enable this property.
- *Capacity* aims to maximize the size of the message sent and/or the statistical guarantee of the detection test. This translates directly into the number of bits of the message or the embedded vector dimension.
- *Security* is rather a design rule advising to follow Kerckhoffs's law: the security of a system relies only on a secret key, sampled from a huge space to prevent brute-force attacks, and without it adversaries cannot attack the watermarking system. But limiting security to this law is blind to another caveat watermarking may encounter; content-agnostic watermarking repeats the same pattern on every watermarked image, thus being able to have enough images to compute the common component between them enables to remove or add the watermark.

As defined in Chapter 2, we especially adopt *post-hoc* watermarking method due to their versatility and ease of implementation, as opposed with *in-generation* watermarking that requires training GenAI models to generate watermarked images.

3.2 Evaluation

We detail standard practice to evaluate digital watermarking and elaborate on common pitfalls.



Figure 3.1 – Comparison of PSNR. Watermarked = 40 dB, WMForger= 29 dB and CTRL-Regen=24 dB. WMForger is the worst version of the image but have the best PSNR among attacks.

Robustness Evaluating robustness means performing valuemetric or geometric transforms and then computing a p-value through statistical tests using the extracted watermark and a reference. A standard threshold for watermark detection is a p-value lower than $1e-6$, meaning once in a million we tell an image is watermarked when it is not. We detail in Sec. 3.4 how to perform those tests.

Perceptibility A standard way of reporting watermark perceptibility is through the Peak Signal to Noise Ratio metric:

$$\text{PSNR} = 10 \cdot \log_{10} \left(\frac{\text{MAX}^2}{\text{MSE}} \right), \tag{3.1}$$

with MAX being the maximum pixel value, and the Mean Squared Error(MSE) which measures the amplitude of the difference between the original image and its watermarked version.

This metric is reporting the amplitude of the difference between two images without taking into account the Human Visual System and how it perceives images. This metric is not linked with quality, Fig. 3.1 shows that lower PSNR does not equate to lower quality with an example of different attacks degrading the cover image and their respective PSNR. LPIPS addresses this by measuring the distance in the feature space of a pretrained network rather than in pixel space, which correlates better with human judgement:

$$\text{LPIPS}(x, x_0) = \sum_l \frac{1}{H_l W_l} \sum_{h,w} \left\| w_l \odot \left(\hat{\phi}_l(x)_{hw} - \hat{\phi}_l(x_0)_{hw} \right) \right\|_2^2, \tag{3.2}$$

where ϕ_l is the activation of layer l of a pretrained backbone (e.g. AlexNet or VGG), $\hat{\phi}_l$ denotes its unit-normalization along the channel dimension, $H_l \times W_l$ is the spatial resolution of that layer, and w_l are learned per-channel weights calibrated on human perceptual judgements. Like PSNR it is computed per image pair, but operates on perceptual features instead of raw pixels.

While PSNR and LPIPS are local, per-image metrics, Frechet Inception Distance (FID) is a global metric comparing the *distributions* of two image sets. Both sets are passed through an Inception network, and the resulting features are modeled as multivariate Gaussians (μ_r, Σ_r) and (μ_a, Σ_a) for the reference (original) and attacked sets respectively. FID is the Fréchet distance between these Gaussians:

$$\mathrm{FID} = \|\mu_r - \mu_a\|_2^2 + \mathrm{Tr}\Big(\Sigma_r + \Sigma_a - 2\left(\Sigma_r \Sigma_a\right)^{1/2}\Big). \tag{3.3}$$

A low FID indicates the attacked images remain close to the original distribution, capturing degradations that per-pixel metrics may miss.

Capacity There is no special way of "evaluating" capacity besides stating it. Usually, the more the better. It is part of a trade-off with robustness and perceptibility. Standard values for capacity ranges from 32 bits to 1024 bits.

Security As stated before, security must rely on a secret key. And one should avoid constant pattern for watermarking, as they open the way to trivial removal attacks. Besides those guidelines, an empirical way to evaluate security is to somewhat perform attacks against watermark detector/decoder and report the attack success rate and the distortion introduced by such attacks. This comprises attacks ranging from the adversarial field, which optimized a perturbation to fool a detector, to the GenAI field, which enables to regenerate the image and by doing so discard high frequency details which carry watermark information.

3.3 The encoder–decoder architecture

Post-hoc deep learning watermarking systems leverage the encoder-decoder framework to learn how to perform two operations; *Embedding/Encoding* - the task of crafting a robust signal and hiding it imperceptibly within the cover - and *Extraction/Decoding* which focuses on retrieving the signal.

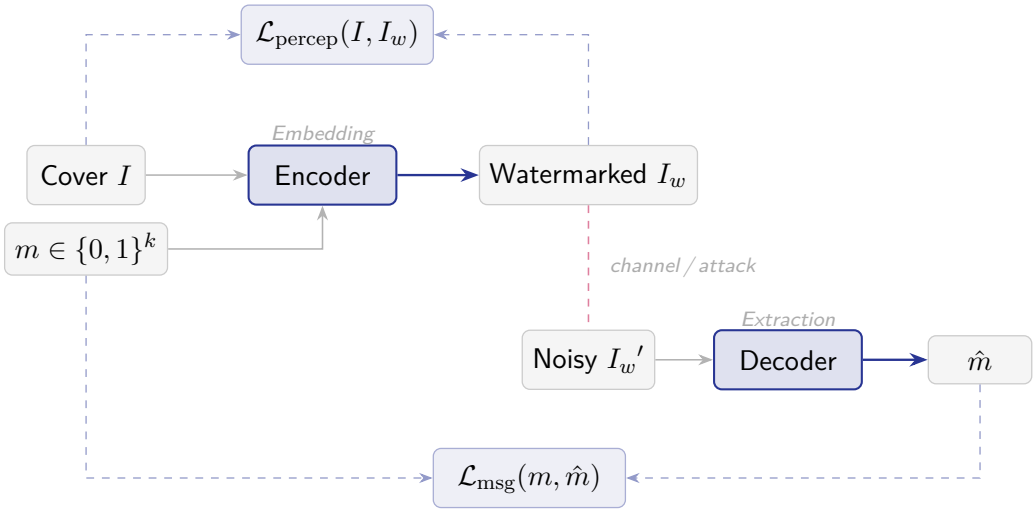


Figure 3.2 – Encoder-decoder paradigm for multi-bit deep learning watermarking. The encoder hides a binary message $m \in \{0,1\}^k$ into a cover image I, yielding watermarked image I_w. A perceptibility loss $\mathcal{L}_{\text{percep}}$ constrains the visual distortion between I and I_w. After passing through a degradation channel, the decoder recovers $\hat{m}$; training also minimizes the reconstruction error $\mathcal{L}_{\text{msg}}(m, \hat{m})$.

In Fig. 3.2, we detail the classic pipeline of post-hoc deep learning watermarking, notably with the use of two terms in the objective function $\mathcal{L}_{percep}$ and $\mathcal{L}_{msg}$.

Message fidelity The goal of multi-bit watermarking is to transmit a sequence of bits. To that end, computing reconstruction loss of the *Decoder* is mandatory and is usually done through Mean Squared Error (MSE) or Binary Cross Entropy (BCE) between m, the ground truth, and the sigmoid $\sigma(\hat{m})$ of the decoded version:

$$\mathcal{L}_{BCE} = -\frac{1}{k}\sum_{i=1}^{k}\left[m_i \log \sigma(\hat{m}_i) + (1 - m_i)\log(1 - \sigma(\hat{m}_i))\right] \tag{3.4}$$

Perceptibility There are several ways to constraint the watermark signal amplitude. A simple MSE to minimize difference amplitude, LPIPS features maps to compare more human-like perceptual shifts (R. Zhang et al. 2018) or GAN-like discriminator (Goodfellow et al. 2014). Besides those standard losses, one can additionally leverage Just Noticeable Difference (JND) models to add prior knowledge on the best places to hide the signal. This is an attenuation phase operating during the embedding step:

$$I_w = I + \alpha \cdot JND(I) \cdot E(I, m), \tag{3.5}$$

where α is a scalar factor to globally control the watermark power, JND is a psychovisual mask computed by the JND model enforced during training to better allocate perturbation budget to specific part of the image (Jinjian Wu et al. 2017) and E is the encoder.

Robustness To achieve robustness, there is no free lunch. The only way to provide a robust model with stable outputs under transformations is to use those transformations during training. Besides data augmentation, the amplitude of the watermark signal is important to as it becomes less prone to be overwritten by subsequent transformations. This introduces a trade-off between robustness and perceptibility. Moreover, robustness training often degrades a bit peak performance of the model, trading it with better stability across transforms.

Capacity Also called *payload*, the capacity is optimized by increasing the dimension of the embedded signal. It is less clear what is a good value of capacity. Like always, it depends on the use case, sometimes 32 bits are enough but to transmit contextual information we want the most capacity available. Again, this is a trade-off with robustness and perceptibility, as increasing capacity makes the task for the watermarking model harder to hide the signal robustly in the image.

3.4 Statistical detection tests

In Chapters 5 and 6, we leverage statistical tests to verify the statistical guarantees provided by watermarking algorithms and re-purpose them into confidence metrics when suited. While state-of-the-art watermarking methods operate across different domains and are trained for diverse tasks, we use the p-value interface as a common quantitative ground. This frames verification as a formal statistical detection test, establishing the exact probability of observing a detection score at least as extreme as the empirical result strictly by chance under the null hypothesis $\mathcal{H}_0$, when examining an unwatermarked cover.

Zero-bit setting In a zero-bit detection framework, a watermark is deemed present if the cosine similarity $C_0 = Y^\top X_0$ computed between the extracted vector $Y \in \mathbb{R}^D$ and the secret carrier sequence $X_0 \in \mathbb{R}^D$ exceeds a predefined detection threshold c. Under the null hypothesis $\mathcal{H}_0$, the extracted vector is assumed to be completely independent of the

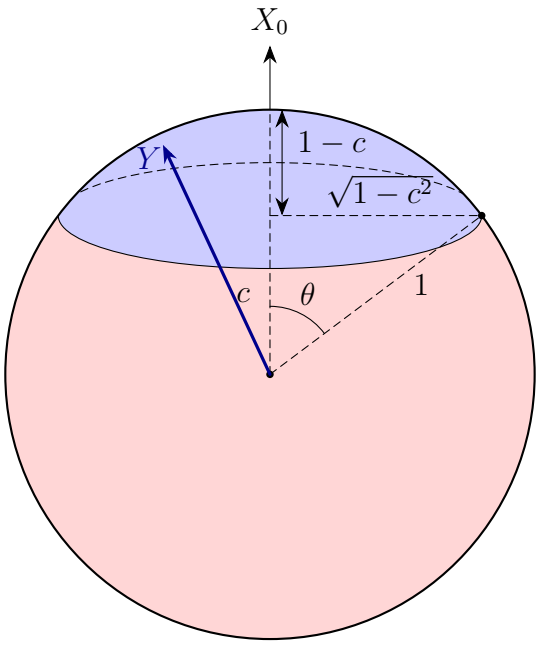


Figure 3.3 – Zero-bit detection geometry on the unit hypersphere. The carrier X_0 sets the pole and the detection region $\{Y : Y^\top X_0 \geq c\}$ is the spherical cap of half-angle $\theta = \arccos c$ (blue): an extracted vector Y is flagged when its cosine $C_0 = Y^\top X_0$ exceeds the threshold c. Under $\mathcal{H}_0$, Y is uniform on the surface, so the false-positive rate $\rho_0(c)$ equals the cap's surface area divided by the sphere's total surface area.

secret key and uniformly distributed on the surface of the D-dimensional unit hypersphere. For any meaningful detection threshold $c > 0$, the closed-form p-value corresponds to the ratio of the surface area of the hypersphere spherical cap above c to the total surface area of the hypersphere:

$$\rho_0(c) = P(C_0 \geq c \mid \mathcal{H}_0) = \frac{1}{2} I_{1-c^2}\left(\frac{D-1}{2}, \frac{1}{2}\right), \tag{3.6}$$

where $I_x(a, b)$ denotes the regularized incomplete beta function. The hypersphere spherical cap is depicted in Fig. 3.3.

Multi-bit setting In a multi-bit payload regime, the extraction/decoding part of the watermarking architecture yields an estimated binary sequence $\hat{m} \in \{0,1\}^{n_{\text{bits}}}$ which must be authenticated against the reference message m. Under the null hypothesis $\mathcal{H}_0$, the extractor is assumed to extract maximum-entropy noise, meaning the decoding of each individual bit behaves as an independent Bernoulli trial with a success probability

of 0.5.

Given an observed count of correctly decoded bits $b = n_{\text{bits}} \cdot \text{bit}_{\text{accuracy}}(m, \hat{m})$, the discrete test statistic B follows a binomial distribution $B \sim \text{Binomial}(n_{\text{bits}}, 1/2)$ under $\mathcal{H}_0$. The resulting p-value is the probability of an unwatermarked image yielding an accuracy at least as extreme as the observed match purely by chance. This is given by the complementary cumulative distribution function of the binomial distribution:

$$\rho(m, \hat{m}) = P(B \geq b \mid \mathcal{H}_0) = \sum_{k=b}^{n_{\text{bits}}} \binom{n_{\text{bits}}}{k} \left(\frac{1}{2}\right)^{n_{\text{bits}}} = I_{1/2}(b,\, n_{\text{bits}} - b + 1)\,. \tag{3.7}$$

Expressing the upper tail of the binomial distribution through the regularized incomplete beta function ensures numerical stability even when evaluating extreme bit lengths and vanishingly small probabilities. Furthermore, because Eq. (3.7) is parameterized by the message capacity n_{bits}, it provides a standardized mathematical interface to fairly compare multi-bit watermarking methods of varying payload sizes. Both expressions feed the confidence reported in Chapter 5 and the survival criteria in Chapter 6.

Assumptions The zero-bit test assumes *isotropy*: under $\mathcal{H}_0$, the normalized extracted vector $Y/\|Y\|$ is uniformly distributed on $\mathcal{S}^{D-1}$, equivalently its law is rotationally invariant and exhibits no component aligned with X_0. This holds exactly when the coordinates of Y are i.i.d. zero-mean Gaussian, and in practice requires the extractor's features to be whitened so that no preferred direction leaks into the null. The multi-bit test instead assumes that, under $\mathcal{H}_0$, the extractor returns maximum-entropy noise: each bit is decoded correctly with probability exactly 1/2 (an unbiased decoder), and the n_{bits} trials are mutually independent, so their sum is genuinely $\text{Binomial}(n_{\text{bits}}, 1/2)$ rather than over-dispersed by inter-bit correlation.

4

Reframing Image Difference Captioning with BLIP2IDC and Synthetic Augmentation

Contents

4.1 Introduction 38
4.2 BLIP2IDC 39
4.2.1 Adaptation 40
4.2.2 Efficient fine-tuning 41
4.2.3 Advantages over existing IDC models 41
4.3 Datasets 42
4.3.1 Existing datasets 43
4.3.2 IDC evaluation 47
4.3.3 IDC task issues 50
4.3.4 Synthetic augmentation 51
4.4 Experiments 53
4.4.1 Evaluation protocol 53
4.4.2 BLIP2IDC evaluation 54
4.4.3 Automatically generated dataset 57

4.4.4 Qualitative analysis 58
4.4.5 Limitations 60
4.5 Conclusion 60

This chapter is based on the paper Gautier Evennou, Antoine Chaffin, Vivien Chappelier and Ewa Kijak, Reframing Image Difference Captioning. In *Winter Conference on Applications of Computer Vision (WACV)*, 2025

4.1 Introduction

Misinformation, frequently propagated through manipulated or out-of-context images, poses a significant challenge. Image Difference Captioning (IDC) offers a solution by generating textual descriptions that enable humans to readily ascertain whether an image has undergone alterations. This work delves into IDC (Park, Darrell, and Rohrbach 2019), a recent approach that goes beyond traditional image analysis by generating detailed textual descriptions of the differences between two images. IDC finds its application across various fields, from detecting subtle changes in satellite imagery (Chouaf et al. 2021; Jhamtani and Berg-Kirkpatrick 2018; Chenyang Liu et al. 2023; Qiu et al. 2021) to identifying anomalies in medical imagery (F. Liu et al. 2021).

Although single-image captioning presents considerable challenges on its own, IDC introduces further complexities, as it involves describing the subtle differences present in a pair of similar images. Ideally, captions should disregard common objects between the images and instead emphasize the nuanced changes between them. The advancement of IDC in recent years relies significantly on the emergence of vision-language models and cross-domain learning techniques.

The second challenge of IDC is the availability of sufficiently large and diverse dataset for the task. Creating a high-quality IDC dataset is particularly difficult and resource-intensive, as it requires image pairs with detailed descriptions of their differences, encompassing various types of changes. This process can rely on expensive and time-consuming crowd-sourced labor (K. Zhang et al. 2023b), or alternatively, on the use of 3D rendered scenes (Park, Darrell, and Rohrbach 2019) or on capturing temporal variations (Tan et al. 2019). Existing IDC training/evaluation pipelines suffer from shortcomings such as suboptimal metrics, inconsistencies in ground truths, small scale of real-world datasets, or the missing categorization of modifications.

In this work, we propose an innovative use of synthetic data and advanced multimodal architectures to address IDC limitations in terms of both data and model. We leverage generative models (Brooks, Holynski, and Alexei A. Efros 2023; Rombach et al. 2021) to produce synthetic data, providing IDC models with diverse and complex image pairs. Concerning the advancements in multimodal models, we discuss how the pre-training of multimodal models like CLIP (Guo, T.-J. Wang, and Laaksonen 2022) and BLIP2 (J. Li, D. Li, Savarese, and S. C. H. Hoi 2023) is relevant for IDC. We explore the potential of BLIP2 in IDC, offering a more 'out-of-the-box' solution contrasting sharply with the previous models' reliance on complex, multi-step training and separate processes for image encoding. Despite its larger size and the associated costs for fine-tuning, we demonstrate the feasibility and effectiveness of LoRA (Edward J Hu et al. 2022) to adapt BLIP2 to IDC, capitalizing on its extensive model capacity while maintaining a low adaptation cost. The main contributions are :

1. We provide a framework for synthetic augmentation to design well-suited IDC datasets based on diffusion models. We release **Syned** [1], our synthetic augmented version of Emu Edit (Sheynin et al. 2023), an image editing dataset, to provide a challenging benchmark for the IDC task. We demonstrate the significant improvement brought by this targeted synthetic augmentation.
2. We propose BLIP2IDC, a new state-of-the-art IDC model based on BLIP2. Extensive experiments conducted on synthetic and real benchmark datasets demonstrate the strong performances of BLIP2IDC in real-world scenarios, and good generalization to unseen content (Fig. 4.10).
3. We present a comprehensive evaluation of several leading models on a new real IDC dataset based on Emu Edit outputs, contributing to a clearer understanding of the current model capabilities in IDC.

4.2 BLIP2IDC

BLIP2 (J. Li, D. Li, Savarese, and S. C. H. Hoi 2023) is a multimodal model that introduces the Querying Transformer (QFormer) as the main block to connect text and image representations. It relies on a two-stage pretraining: a vision-language representation learning stage with a frozen image encoder, and a vision-to-language generative learning

1. The code, weights and dataset are available at `https://github.com/gautierevn/BLIP2IDC`

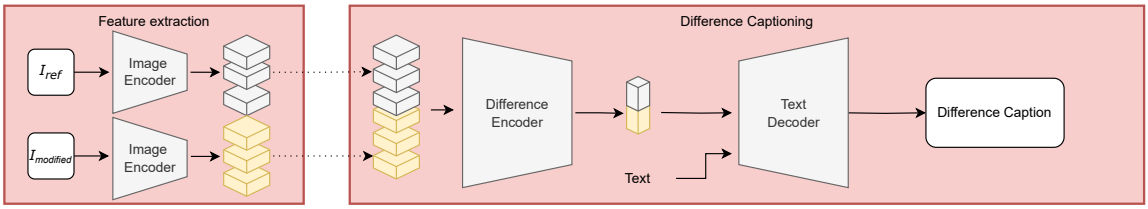


Figure 4.1 – Standard IDC pipeline. The feature extraction step is performed before training. The training step leverages the features from a frozen image encoder by concatenating them and then learns the difference representation from these concatenated features. The output is then fed to the text decoder to perform the generation of the difference caption in an autoregressive manner.

stage with a frozen LLM. The QFormer is composed of two transformers modules. The first one attends to the image through cross attention between learned embeddings, called queries, and vision encoder outputs. The second module interacts with the ground truth text and with the queries embeddings to ensure vision-language alignment. BLIP2 is pretrained to jointly optimize three pre-training objectives that are Image-Text Contrastive Learning, Image-Text Matching, and Image-grounded Text Generation. As a contribution, we take advantage of its pretraining for Image Captioning and adapt BLIP2 to IDC at a low computational cost without changing its architecture.

4.2.1 Adaptation

We argue that the classic IDC two-stream encoding scheme featured in Fig. 4.1 results in a suboptimal comparison of images at the features level. For standard image captioning, BLIP2 takes one image as input and uses a ViT model (Dosovitskiy et al. 2021; Vaswani et al. 2017) as a frozen image encoder. In the context of IDC, feeding BLIP2 jointly with the two images to be compared enables the attention layers of the visual encoder and QFormer to focus early on the differences between the two images, and to encode the two images in relation to each other rather than separately. Our BLIP2 adaptation pipeline is shown in Fig. 4.2. We give a single image as input, resulting from the vertical concatenation of two images, allowing the model to attend to the differences early, while avoiding any modification of the architecture of the model. Unlike BLIP2, which only trains the QFormer, the ViT and the LLM should be finetuned to compensate for this modification of the input domain.

Although images are stretched when concatenated, this process does not undermine

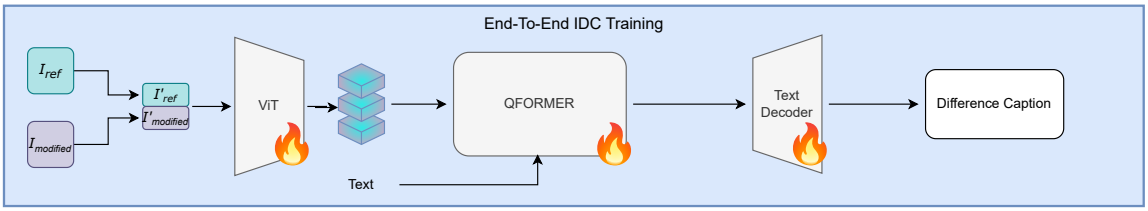


Figure 4.2 – BLIP2IDC end-to-end pipeline. We first resize and concatenate the image inputs before feeding them to the BLIP2 architecture finetuned using LoRA.

the performances. We conjecture this behavior is due to the way BLIP2 was pretrained, where images are randomly cropped then stretched to a fixed square size, without padding. Thus, the BLIP2 model learned to be robust to various stretching operations.

4.2.2 Efficient fine-tuning

Modules fine-tuning. Although BLIP2IDC does not require to change the architecture of the model, the model should be finetuned for the IDC task to adapt to the new task and type of data. Unlike the original training of BLIP2 which only fine-tunes the QFormer, all the components, including the ViT, QFormer and image-grounded text decoder, are finetuned to get the best IDC performance (see Fig. 4.8 in the result section).

Low Rank Adaptation (LoRA). We use Low Rank Adaptation (LoRA) to reduce training resources while maintaining performance. By fine-tuning just 0.1% of all parameters—specifically, the attention modules' Q, K, V layers—we achieve top performance with minimal resource use. This approach allows BLIP2IDC to adapt to IDC tasks, balancing high performance with resource efficiency.

4.2.3 Advantages over existing IDC models

The latest leading models for IDC are CLIP4IDC (Guo, T.-J. Wang, and Laaksonen 2022) and SCORER (Tu, L. Li, Su, Zha, et al. 2023). CLIP4IDC utilizes a two-step training approach, first to adapt image-pairs visual representations for captioning and then to generate captions based on visual differences encoding, through contrastive and cross-entropy losses respectively. SCORER follows the traditional IDC framework as depicted in Fig. 4.1, enhancing performance with innovative modules for view-shift invariance and caption informativeness, and is presented as showing particular strength on the CLEVR-DC dataset.

These models are characterized by complex training procedures with several steps and separate image encoding processes. In contrast, our model BLIP2IDC simply concatenates images for joint encoding, employing early attention mechanisms while learning in a single training phase. This approach proves crucial for IDC, as evidenced by our experimental results in Sec. 5.3.2. BLIP2IDC crucially benefits from knowledge gained through the large-scale unsupervised pre-training of BLIP2 for image captioning, making it compelling given the data-scarcity challenge raised by IDC.

4.3 Datasets

The structure of IDC datasets is a triplet $(I_{ref}, I_{modified}, GT)$ which respectively represent the original image I_{ref}, the modified image $I_{modified}$, and the ground-truth set of reference descriptions of the differences GT. We compare our approach to recent IDC models on standard IDC datasets: CLEVR-Change (Park, Darrell, and Rohrbach 2019), CLEVR-DC (H. Kim et al. 2021), Spot-The-Diff (Qiu et al. 2021) and Image Editing Request (Tan et al. 2019). These datasets are distinguished by various properties, summarized in Tab. 4.1: the number of image pairs $(I_{ref}, I_{modified})$, the origin of the I_{ref} images, which may be real images or generated by 3D models, the number of reference captions (ground-truths GT) per triplet, the way in which $I_{modified}$ and GT are obtained, the number and type of transformations performed, and whether some manual intervention was required to create the dataset. This may happen at the level of triplet filtering or the creation of ground-truth captions.

These properties can limit the creation of these datasets, which can be costly, as well as their usefulness, for example when the diversity of transformation types is reduced. We discuss the limitations of these datasets hereafter, as well as the issues posed by the evaluation of the IDC task. To handle those limitations, we propose a method based on text-guided image editing methods for creating IDC datasets without manual intervention, which can then be adapted to different use cases. Furthermore, instruction-based image editing capabilities are evaluated on some benchmarks (IE datasets), like MagicBrush (K. Zhang et al. 2023b) or Emu Edit test set generations (EE) (Sheynin et al. 2023). Those datasets are also composed of triplets $(I_{ref}, T_{instruction}, I_{edited})$, with $T_{instruction}$ the modification that maps I_{ref} to I_{edited} and I_{edited} the edited image. We thus propose to adapt those datasets for the IDC task.

Table 4.1 – IDC and IE datasets. Setting indicates the origin of reference images as 3D scenes or from the real world. Editing method refers to the way modified images are created or collected. GT is the number of ground-truths in the validation and test set.

Dataset	# image pairs	Setting	Editing method	Curated	# GT
CLEVR-Change	79,606	3D Scenes	3D engine	✓	5.0
CLEVR-DC	48,000	3D Scenes	3D engine	✓	5.0
STD	13,192	Real-world	Temporal change		1.86
IER	3,939	Real-world	Human editing	✓	3.0
MagicBrush	10,308	Real-world	DALL-E 2 platform	✓	1.0
EE	5,612	Real-world	Automatic generation	✓	1.0
Syned(ours)	28,720	Real-world	Automatic generation		5.0

4.3.1 Existing datasets

CLEVR-Change. Introduced by Robust Change Captioning (Park, Darrell, and Rohrbach 2019), the goal of this dataset is to evaluate primary abilities for visual understanding such as being able to tell the shape, the color, the material and the position of an object and if one or none of those properties changed. This dataset enables to assess if an IDC model understands spatial relationships and if it is robust to non-semantic changes. Five types of scene changes are defined. Editing instructions and ground-truth captions are automatically constructed following a template. However, this dataset was created with a 3D engine, and all the elements are 3D scenes far from the real world setting, leading to a model that doesn't transfer very well to real-life examples.

CLEVR-DC. CLEVR-DC (H. Kim et al. 2021) is a derivative of CLEVR-Change with extreme shift in viewpoints. It uses the same types of modifications as CLEVR-Change. The goal of this dataset is to evaluate the view-invariance of the IDC model but it shares the limitations of the original dataset.

Spot-The-Diff (STD). Unlike CLEVR-Change, STD (Jhamtani and Berg-Kirkpatrick

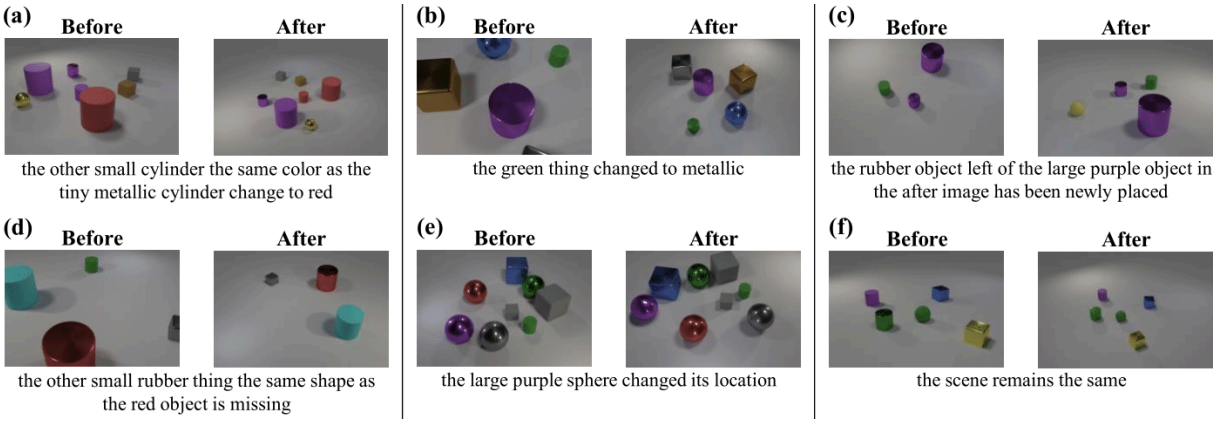


Figure 4.3 – CLEVR and CLEVR-DC 3D images. Changes comprises color, texture and position modifications. Figure from H. Kim et al. 2021.

2018) uses a real world setting for the pairs of images. To cope with the issue of producing modified images, it uses temporal evolution. Thus, a pair of images is made of one image taken at a given time and another of the same place taken at a later time, with a fixed point of view. Most of the images are set in a parking lot, heavily restricting the diversity of ground-truth as most changes in a parking are add/remove people/car. Although heavily biased, this dataset enables one to test IDC models on real-world images.

Image Editing Request (IER). Born from crawling Reddit and Zhopped, the IER dataset (Tan et al. 2019) relies on instructions from online users in Photoshop specialized forums. The input image and editing instruction were posted, and people sent their modified images on Reddit. Crawling those results of the manual editing led to a high quality and diverse dataset but with a very small scale. Each image pair of the test set has three GT captions, written by three different annotators. IER provides a wide range of semantic modifications, such as "add a sailor hat to ducks", "replace the background by a spaceship", or "replaces brooms with lightsabers".

The four previous datasets are standard in Image Difference Captioning, we provide representative samples in Figs. 4.3 to 4.5.

MagicBrush. This dataset is the first large-scale, manually annotated, instruction-guided image editing dataset (K. Zhang et al. 2023b) covering various global and local editing scenarii. MagicBrush comprises 10K $(I_{ref}, T_{instruction}, I_{edited})$ triplets, which is sufficient to fine-tune large-scale image editing models with the use of LoRA. For this dataset, modifications are not grouped into categories like in CLEVR-Change, making it hard to explain performance and target specific weaknesses.



Figure 4.4 – Examples from the Spot-the-Diff dataset. Surveillance-style image pairs where the difference is a change in the scene (appearing, disappearing or moving objects).

Emu Edit (EE). This dataset (Sheynin et al. 2023) from Meta is composed of pairs of original images from MSCOCO (T.-Y. Lin, Maire, Belongie, Hays, et al. 2014) and modified images generated by the Emu Edit model, a state-of-the-art generation model, which takes an image and an editing instruction as inputs. This dataset also contains fine-grained information about the type of modifications generated, grouped in eight different

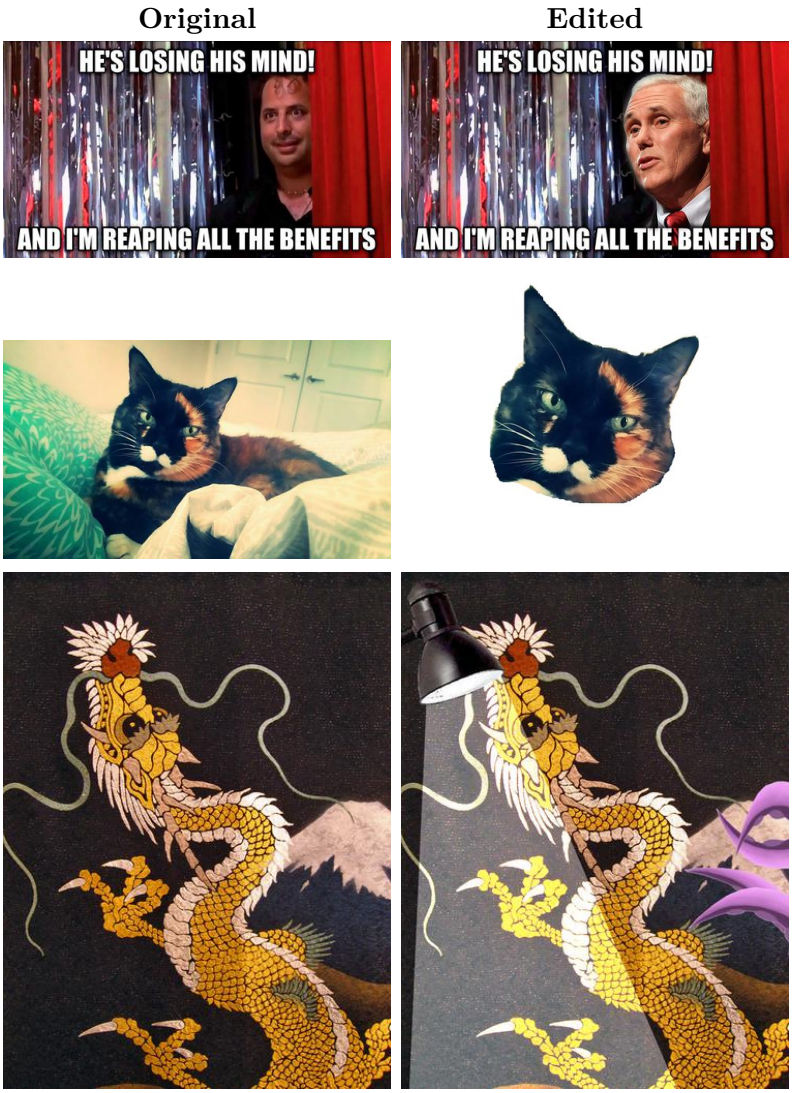


Figure 4.5 – Examples from the Image Editing Request (IER) dataset. Each pair shows an original image and its user-requested edit (face swap, background removal, object insertion).

categories : Add, Text, Color, Background, Local, Style, Global and Remove. Paired with the real-world setting of original images which enables greater usability, these properties make it a great candidate for a challenging IDC benchmark.

4.3.2 IDC evaluation

We detail following metrics which are standard in IDC literature: BLEU, ROUGE, METEOR and CIDEr metrics as (Lavie and A. Agarwal 2007; C.-Y. Lin 2004; Oliveira dos Santos, Colombini, and Avila 2021; Papineni et al. 2002a). We largely take the following explanation from Lavie 2011.

We use this example to better illustrate each metric:

> **Reference** (14 words): *the Iraqi weapons are to be handed over to the army within two weeks*
> **Candidate** (8 words): *in two weeks Iraq's weapons will give army*

The words matched (as exact unigrams) are *two*, *weeks*, *weapons*, *army*.

BLEU BLEU is a precision-based metric: it counts n-gram matches of the candidate against the reference(s), combines the precisions as a weighted geometric mean, and multiplies by a *brevity penalty* BP to compensate for the absence of recall.

$$\mathrm{BP} = \begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{if } c \le r \end{cases} \qquad \mathrm{BLEU} = \mathrm{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right), \tag{4.1}$$

where p_n is the (clipped) n-gram precision, c the candidate length, r the reference length, and $w_n = 1/N$ (usually $N = 4$).

Modified precisions.

$$p_1 = 4/8, \qquad p_2 = 1/7, \qquad p_3 = 0/6, \qquad p_4 = 0/5.$$

Only the bigram *two weeks* matches, hence $p_2 = 1/7$; no 3- or 4-gram matches.

Score. The exponentiated sum is just the geometric mean of the precisions, $\left(\prod_{n=1}^{4} p_n\right)^{1/4}$. Since $p_3 = p_4 = 0$, the product is zero and so is the score:

$$\mathrm{BLEU} = \mathrm{BP} \cdot (p_1 p_2 p_3 p_4)^{1/4} = \mathrm{BP} \cdot \left(\tfrac{4}{8} \cdot \tfrac{1}{7} \cdot 0 \cdot 0\right)^{1/4} = 0. \tag{4.2}$$

This illustrates BLEU's brittleness at the segment level: a single missing high-order n-gram collapses the score to zero. Aggregated over a large test set the zeros are diluted and the score becomes stable. BLEU is meant to be computed against a corpus, not only

with respect to a couple of references. But as it is more convenient, the latter is the most used.

Clipping. Precision counts are clipped at the maximum count of each word in any reference. For candidate *the the the the* against a reference containing *the* twice, the clipped unigram count is 2, giving $p_1 = 2/4$ rather than 4/4.

ROUGE ROUGE is recall-oriented and is the standard for summarization and captioning. We show ROUGE-N (n-gram recall) and ROUGE-L (longest common subsequence).

ROUGE-N.

$$\text{ROUGE-}N = \frac{\text{matched } n\text{-grams}}{\text{total } n\text{-grams in reference}}. \tag{4.3}$$

$$\begin{aligned} &\text{ROUGE-1: } R = 4/14 = 0.286, \quad P = 4/8 = 0.500, \quad F_1 = 0.364. \\ &\text{ROUGE-2: } R = 1/13 = 0.077, \quad P = 1/7 = 0.143, \quad F_1 = 0.100. \end{aligned}$$

(Reference has 14 unigrams and 13 bigrams; only *two weeks* matches at bigram level.)

ROUGE-L. Based on the longest common subsequence (LCS), which need not be contiguous but must preserve order. The common words in order are constrained by both sequences: *weapons* $\to$ *army* (length 2) or *two* $\to$ *weeks* (length 2), and no length-3 subsequence exists, so LCS = 2.

$$R_{\text{lcs}} = \frac{\text{LCS}}{|\text{ref}|} = \frac{2}{14}, \quad P_{\text{lcs}} = \frac{\text{LCS}}{|\text{cand}|} = \frac{2}{8}, \quad F_{\text{lcs}} = \frac{(1+\beta^2) R_{\text{lcs}} P_{\text{lcs}}}{R_{\text{lcs}} + \beta^2 P_{\text{lcs}}}. \tag{4.4}$$

With $\beta = 1$: $R_{\text{lcs}} = 0.143$, $P_{\text{lcs}} = 0.250$, $F_{\text{lcs}} = 0.182$. In practice β is set large so that ROUGE-L reflects recall.

METEOR METEOR aligns the candidate to the reference using generalized unigram matches (exact, stem, synonym, paraphrase), combines precision and recall into a recall-weighted harmonic mean F_{mean}, then applies a fragmentation penalty for poor word ordering.

$$F_{\text{mean}} = \frac{P \cdot R}{\alpha P + (1-\alpha)R}, \quad \text{Pen} = \gamma \cdot \left(\frac{\text{frag} - 1}{\text{length} - 1}\right)^{\beta}, \quad \text{METEOR} = F_{\text{mean}}\,(1 - \text{Pen}), \tag{4.5}$$

with original parameters $\alpha = 0.9$, $\beta = 3.0$, $\gamma = 0.5$.

Worked example. The matcher aligns 5 words:

> Ref: *Iraqi weapons . . . army . . . two weeks* MT: *two weeks Iraq's weapons . . . army*

$$P = 5/8 = 0.625, \qquad R = 5/14 = 0.357,$$
$$F_{\text{mean}} = \frac{PR}{0.9P + 0.1R} = 0.373.$$

The 5 aligned words form 3 contiguous fragments, so the fragmentation is $\frac{3-1}{5-1} = 0.5$:

$$\text{Pen} = 0.5 \cdot (0.5)^3 = 0.0625,$$
$$\text{METEOR} = 0.373 \cdot (1 - 0.0625) = 0.350.$$

Unlike BLEU, METEOR returns a non-zero, informative score here, and its recall weighting and ordering penalty give higher correlation with human judgments at the segment level.

CIDEr CIDEr is designed for image captioning against *multiple* references. It weights n-grams by TF-IDF, so common, uninformative n-grams contribute little, and computes cosine similarity between candidate and reference vectors, averaged over $n = 1, \ldots, 4$.

$$\text{CIDEr}_n(c, S) = \frac{1}{m}\sum_{j=1}^{m} \frac{\mathbf{g}^n(c) \cdot \mathbf{g}^n(s_j)}{\|\mathbf{g}^n(c)\|\,\|\mathbf{g}^n(s_j)\|}, \qquad \text{CIDEr}(c, S) = 10 \cdot \frac{1}{4}\sum_{n=1}^{4} \text{CIDEr}_n(c, S), \tag{4.6}$$

where the TF-IDF weight of n-gram k is $g_k^n = \dfrac{h_k}{\sum_l h_l} \cdot \log\dfrac{|I|}{\text{df}_k}$, with h_k its count, $|I|$ the number of images in the dataset, and df_k the number of images whose references contain it.

Toy example (unigrams). A dataset of $|I| = 3$ images with one reference each:

img1	*a cat sits on the mat*
img2	*a dog runs in the park*
img3	*a cat runs on the grass*

For img1, candidate *a cat on the mat* vs. reference *a cat sits on the mat*. The IDF weights $\log(3/\mathrm{df}_k)$ are:

word	a	cat	sits	on	the	mat
df_k	3	2	1	2	3	1
idf_k	0	0.405	1.099	0.405	0	1.099

Words appearing in every reference (*a*, *the*) get zero weight. Normalizing counts by sentence length (5 and 6) and taking the cosine over the shared words gives

$$\mathrm{CIDEr}_1 = \frac{\mathbf{g}(c) \cdot \mathbf{g}(s)}{\|\mathbf{g}(c)\|\|\mathbf{g}(s)\|} = \frac{0.0512}{0.2479 \times 0.2761} \approx 0.748. \tag{4.7}$$

The full CIDEr score averages this over $n = 1, \ldots, 4$ and multiplies by 10. The key property is that matching a rare, content-bearing word such as *mat* contributes far more than matching *a* or *the*.

4.3.3 IDC task issues

IDC models are trained on triplets $(I_{ref}, I_{modified}, GT)$ and evaluated on a test set of pairs $(I_{ref}, I_{modified})$ by comparing the generated caption with the reference captions GT, using various automatic evaluation metrics, such as BLEU, ROUGE, METEOR and CIDEr (Lavie and A. Agarwal 2007; C.-Y. Lin 2004; Oliveira dos Santos, Colombini, and Avila 2021; Papineni et al. 2002a). Some aspects of the real-world datasets weaken the quality of the evaluation.

Suboptimal metrics. The reference-based metrics more accurately evaluate the quality of generated sentences if multiple references are provided. Five references is usually the minimum used. However, none of the available real-world datasets provides a consistent and sufficient number of references, making evaluations less accurate.

Lack of consistency in ground truths. In STD and IER, the human-annotated ground-truth captions do not consistently refer to the same difference or set of differences,

especially when several modifications are made to the image. Some captions describe a single modification, others several. An IDC model trained to describe only the main difference will struggle with this type of ground truth, and it is the case for all existing IDC models. Either all the differences should be mentioned in each ground-truth, or just one, consistently, to avoid conflicting objectives.

Tiny real-world datasets. As real-world datasets are very time-consuming to make, they are small in size, with at most 10,000 text-image pairs. As a result, they can lead to models specialized in specific subsets of possible modifications, as for STD. On the other hand, when modifications are very diverse, as for IER, a type of modification can occur only a few times. Metrics then heavily depend on the data splits, according to whether a modification appears only in the training or in the test set. This is all the more possible if the modifications are not grouped into classes allowing stratified sampling.

Modifications categorization. Grouping changes into categories is useful for stratified sampling, but also for analyzing results, as it makes it easier to identify difficult changes and the weaknesses of IDC models.

4.3.4 Synthetic augmentation

To cope with the current shortcomings of IDC datasets mentioned above, we propose a pipeline to generate synthetic training samples based on real-world original images. The global pipeline is illustrated in Fig. 4.6. The idea is to leverage the emerging prompt-based image editing models to produce modified images based on a set of original images and editing instructions belonging to a defined range of modifications. Additional GT captions are generated by a Large Language Model (LLM) as variations of the editing instructions, ensuring consistent GTs.

Such a pipeline can be used either to augment an existing IDC dataset, or to create a new dataset whose images and modifications are tailored to a particular downstream task with new specific types of data or modifications.

To evaluate this pipeline, we applied it to augment the EE dataset (Sheynin et al. 2023), as this dataset provides real-world edited images alongside editing instructions well-suited for an image editing model and with a clear distinction between each class of modifications. As the image editing model, we choose the InstructPix2Pix (Brooks, Holynski, and Alexei A. Efros 2023) model finetuned on the MagicBrush (K. Zhang et al. 2023b) dataset, the state-of-the-art solution for executing real-world edits. The train set

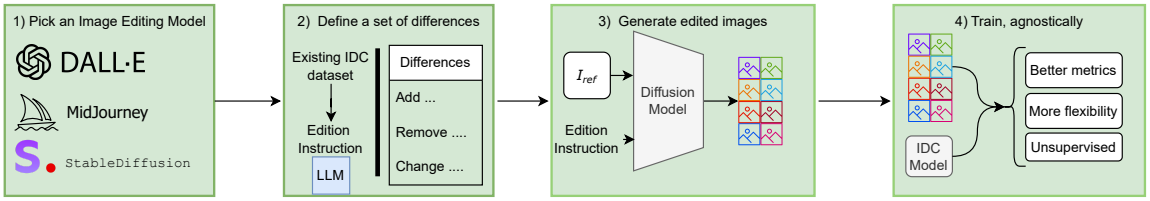


Figure 4.6 – Synthetic dataset creation pipeline leveraging a prompt-based image editing and large language models.

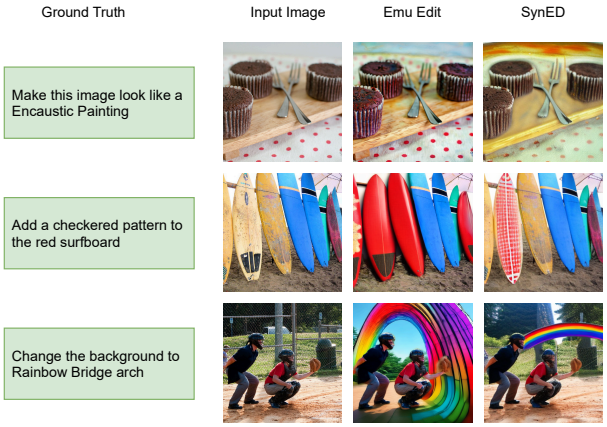


Figure 4.7 – Comparison of train set samples from EE, generated by the Emu Edit model, and Syned, generated by a finetuned InstructPix2Pix model.

is augmented by generating eight new modified images per original image from the EE dataset and their associated editing instructions, without any manual curation.

To ensure meaningful evaluation, we leverage the Llama-2-7b-chat-hf model to generate 4 additional variations of each editing instruction from the EE test set. Each of the 2,022 samples in the test set is therefore described by 5 GT captions: the original instruction and the 4 reference captions provided by the LLM. Note that the train set images (see Fig. 4.7) were not modified by the same editing model as in the test set.

The resulting Syned dataset comprises:

- **a train set of 28,720 pairs of** $(I_{ref}, I_{modified})$: 8 variations of each of the 3,590 original images from MSCOCO, generated by InstructPix2Pix. The different modifications are categorized into 8 classes.
- **a test set of 2,022 samples with 5 references each**, where the modified images were generated by the Emu Edit model.

Although the modifications can be defined as anything according to the downstream needs, they have to be aligned with the image editing model's capabilities. For instance, editing models currently fall short at adding text to images. Such instructions introduce bad samples into the training set. However, as image editing models improve, the proposed pipeline will allow including more reliable edits to the training set of IDC models, although it already allows for customized generation of more challenging IDC datasets.

4.4 Experiments

4.4.1 Evaluation protocol

Models and datasets. We compare BLIP2IDC to other state-of-the-art IDC models with the results reported in their respective papers on the four standard IDC datasets: CLEVR-Change, CLEVR-DC, STD, and IER. Moreover, we introduce the EE dataset to the IDC task, on which we also train and evaluate CLIP4IDC (Guo, T.-J. Wang, and Laaksonen 2022) and SCORER (Tu, L. Li, Su, Zha, et al. 2023) for the sake of comparison.

Finetuning. BLIP2IDC is finetuned as described in Sec. 4.2.2, with a LoRA of rank 8. For our own adaptation of BLIP2, we use the decoder-only BLIP2's LLM version, with opt2.7B and vit-base-patch16-224 as the ViT model. We use LoRA implementation from the peft(Mangrulkar et al. 2022) python library to train BLIP2 and to store adaptation weights in a lightweight 20 MB file. For CLIP4IDC and SCORER, we used the hyperparameters from their respective online repositories.

Standard data augmentation. To increase the robustness of the model, we use RandomGaussianBlur and JPEG Compression as a non-disruptive data augmentation scheme. Augmentations should indeed leave the ground-truth semantic content untouched. Transformations based on crop, color jitter, horizontal/vertical flip, brightness or contrast can add meaningful differences and thus change the expected difference caption.

Metrics. Based on previous work (Chouaf et al. 2021; Hosseinzadeh and Y. Wang 2021; Jhamtani and Berg-Kirkpatrick 2018; Chenyang Liu et al. 2023; Qiu et al. 2021; Shi et al. 2020a; L. Yao, Weiying Wang, and Jin 2022), CIDEr (C) is the main metric used to evaluate the generated difference captions. This n-gram-based approach ensures that the evaluated text captures the most relevant aspects of the image as agreed upon by multiple human annotators. In a semantic-dependent context, the CIDEr metric rewards

Table 4.2 – CIDEr scores on CLEVR-Change for the 5 categories of changes.

Method	Color	Texture	Move	Add	Drop
DUDA (Park, Darrell, and Rohrbach 2019)	120.4	86.7	56.4	108.2	103.4
VAM+ (Shi et al. 2020b)	122.1	98.7	82.0	126.3	115.8
IFDC (Q. Huang et al. 2022)	133.2	99.1	82.1	128.2	118.5
DUDA+ (Hosseinzadeh and Y. Wang 2021)	120.8	89.9	62.1	119.8	123.4
BiDiff (Sun et al. 2022)	115.9	106.8	71.8	121.3	124.9
IDC-PCL (L. Yao, Weiying Wang, and Jin 2022)	131.2	101.1	81.7	133.3	116.5
CLIP4IDC (Guo, T.-J. Wang, and Laaksonen 2022)	149.1	135.3	91.0	132.4	**135.5**
SCORER (Tu, L. Li, Su, Zha, et al. 2023)	143.2	135.2	91.6	129.4	132.6
SCORER+CBR (Tu, L. Li, Su, Zha, et al. 2023)	146.2	133.7	92.2	131.1	133.9
BLIP2IDC (Ours)	**152.31**	**137.0**	**99.8**	**135.6**	133.1

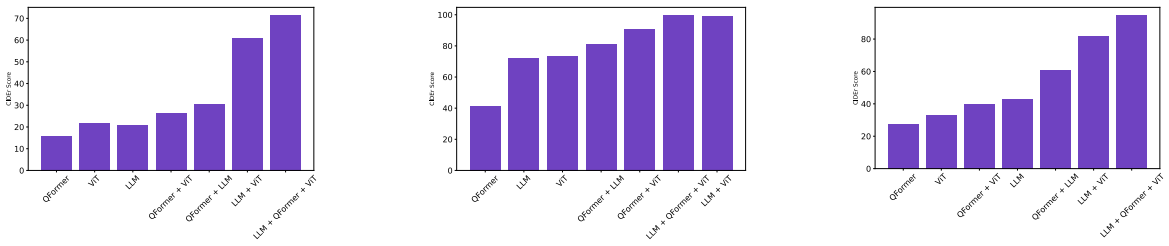
descriptions that accurately reflect the consensus understanding of the image content. BLEU-4 (B), ROUGE-L (R) and METEOR (M) are also used as secondary metrics to assess sentence quality.

4.4.2 BLIP2IDC evaluation

We evaluate the performance of BLIP2IDC against other methods on the four existing IDC datasets. All results except CLIP4IDC on CLEVR-DC and BLIP2IDC are reported from previous work.

Results on 3D scenes. Results per type of semantic changes on CLEVR-Change are given in Tab. 4.2. BLIP2IDC ranks first on almost all types of semantic changes, with a 10% increase in the captioning performance on the most difficult 'Move' change, and a significant improvement over previous state-of-the-art method on the CIDEr metric. On CLEVR-DC, which introduces extreme viewpoint changes, we observe that CLIP4IDC achieves the best results, followed by BLIP2IDC (Tab. 4.3). Both models are better than SCORER although the latter was specially designed to ensure extreme-view-shift invariance.

Results on real-world images. According to Table 4.3, the BLIP2IDC method ranks first on the CIDEr metric by a significant 9.2% margin with respect to state-of-the-art on STD. The IER dataset is made of photoshopped images in a real-world setting. Thus, the impressive improvement over previous state-of-the-art is expected since BLIP2 was pretrained on 129M images in a real-world setting (Krishna et al. 2017; T.-Y. Lin, Maire, Belongie, Hays, et al. 2014; Sharma et al. 2018). Our results show that our adaptation outperforms state-of-the-art models, even those based on CLIP which was trained at large



(a) IER score (b) EE Score (c) Syned Score

Figure 4.8 – Ablation Study of BLIP2IDC's CIDEr according to finetuned modules on different datasets.

scale too. Overall, those results show that efficiently adapting a powerful pretrained model yields state-of-the-art performance, whereas most existing models fall behind due to the lack of data during training.

Table 4.3 – Results on CLEVR-DC, STD and IER, reported from their corresponding papers, except †: reported from our own experiments.

	CLEVR-DC			STD			IER			
Method	B	M	C	B	M	C	B	M	R	C
Dyn rel-att (Tan et al. 2019)	-	-	-	-	-	-	6.7	12.8	37.5	26.4
M-VAM (Shi et al. 2020b)	40.9	27.1	60.1	10.1	12.4	38.1	-	-	-	-
M-VAM+RAF (Shi et al. 2020b)	-	-	-	11.1	12.9	43.5	-	-	-	-
VA (H. Kim et al. 2021)	44.5	29.2	70.0	-	-	-	-	-	-	-
VACC (H. Kim et al. 2021)	45.0	29.3	71.7	9.7	12.6	41.5	-	-	-	-
DUDA (Park, Darrell, and Rohrbach 2019)	40.3	27.1	56.7	8.1	12.5	34.5	6.5	12.4	37.3	22.8
NCT (Tu, L. Li, Su, K. Lu, et al. 2023)	47.5	32.5	76.9	-	-	-	8.1	15.0	38.8	34.2
SRDRL+AVS (Tu, T. Yao, et al. 2021)	-	-	-	-	13.0	35.3	-	-	-	-
IFDC (Q. Huang et al. 2022)	-	-	-	8.7	11.7	37.0	-	-	-	-
BDLSCR (Sun et al. 2022)	-	-	-	6.6	10.6	42.2	6.9	14.6	38.5	27.7
VARD-Trans (Tu, L. Li, Su, Du, et al. 2023)	48.3	32.4	77.6	-	12.5	30.3	10.0	14.8	39.0	35.7
MCCFormers-D (Qiu et al. 2021)	46.9	31.7	71.6	10.0	12.4	43.1	8.3	14.3	39.2	30.2
SCORER (Tu, L. Li, Su, Zha, et al. 2023)	49.5	**33.4**	82.4	9.4	13.8	38.5	9.6	14.6	39.5	31.0
SCORER+CBR (Tu, L. Li, Su, Zha, et al. 2023)	49.4	**33.4**	83.7	10.2	12.2	38.9	10.0	15.0	39.6	33.4
CLIP4IDC (Guo, T.-J. Wang, and Laaksonen 2022)	**54.7**†	33.0†	**89.9**†	**11.6**	**14.2**	47.4	8.2	14.6	40.4	32.2
BLIP2IDC (Ours)†	49.3	33.0	88.5	11.4	13.5	**51.4**	**17.4**	**20.1**	**48.5**	**74.1**

Ablation study. The performance variations across different BLIP2IDC configurations, depending on the finetuned modules, are depicted in Fig. 4.8. This analysis highlights the significant impact of fine-tuning both the ViT and LLM modules. Given the mismatch between the ViT and LLM modules for both the input and output domains, the QFormer not surprisingly stands out as the least critical component for finetuning, as it already had undergone extensive training during the BLIP2 pretraining phase.

Training Dataset	Model	Zero-Shot	BLEU-4	METEOR	ROUGE	CIDEr
IER	SCORER		<u>9.6</u>	14.6	39.5	31.0
IER	SCORER+CBR		**10.0**	15.0	<u>39.6</u>	33.4
IER	CLIP4IDC		8.2	14.6	**40.4**	32.2
IER	NCT		8.1	15.0	38.8	34.2
IER	VARD-Trans		**10.0**	14.8	39.0	<u>35.7</u>
Syned	BLIP2IDC	✓	7.2	**16.0**	37.9	35.3
Syned + EE	BLIP2IDC	✓	8.5	15.4	36.3	**36.5**

Table 4.4 – BLIP2IDC scores on IER with training only on other datasets. Other models are listed to provide current state-of-the art results obtained with in-domain training. We exhibit state-of-the-art performance on the main metric, CIDEr.

Cross dataset evaluation. In this section, we evaluate on IER and Syned to exhibit BLIP2IDC transferability with real-world datasets, see Tab. 4.4.

BLIP2IDC outperforms all previous state-of-the-art models even with the burden of the zero-shot setting. This supports the strong generalization of BLIP2IDC. BLEU and ROUGE metrics are lower due to the distance between the ground-truth domain of IER and Syned, the modifications being not described in the same way in each dataset. IER and Syned are however still in close domains. In constrast, BLIP2IDC trained on Syned would not transfer well on STD due to the different domain, the ground-truth being a lot more biased in STD. These results also show that our pipeline enables the generation of high quality data describing a wide range of modifications which leads to good generalization abilities.

Comparison with BLIP2. We compare BLIP2IDC with the original BLIP2 model, see Tab. 4.5. We use the same prompt for each model: "Describe the differences between the two images". The results show that BLIP2IDC significantly enhances the capabilities of BLIP2, effectively leveraging BLIP2's knowledge of the visual world and adapting it for IDC.

Attention maps. While noisy, visualizing attention maps provides some cues on the way the ViT part of BLIP2IDC interacts with the concatenated images. In this difference captioning process, the behavior of the attention heads is very human-alike in the sense that they seem to compare corresponding parts of each image. Each attention head focuses on specific patterns. Their attention maps highlight either similar regions, or the modified elements as those displayed in Fig. 4.9.

Table 4.5 – BLIP2 zero-shot performance on real-world IDC datasets

Model	Dataset	BLEU	METEOR	ROUGE	CIDEr
BLIP2	STD	0.5	7.3	14.7	11.2
	IER	1.8	7.3	15.8	14.7
	Syned	1.8	7.5	18.9	24.3
BLIP2IDC	STD	11.4	13.5	34.2	51.4
	IER	17.4	20.1	48.5	74.1
	Syned	23.1	21.3	47.0	94.8
Improvement (%)	STD	2151	185	233	458
	IER	946	274	331	504
	Syned	1313	284	248	391

4.4.3 Automatically generated dataset

We introduced in Sec. 4.3.4 Syned, a new dataset designed for the IDC task, built upon the EE dataset. New data are generated by a different instruction-based image editing model, thus broadening the scope and diversity of the new modified images wrt the original EE dataset, as depicted in Fig. 4.7. We evaluate the usefulness of Syned as data augmentation for different IDC models in Tab. 4.6. We show an improvement with the proposed synthetic augmentation (EE + Syned) for both CLIP4IDC and BLIP2IDC. Note that the available implementation of SCORER can reach at most a CIDEr score of 23.2 which is not significant in this setting and can be achieved only by reward hacking.

Table 4.6 – CIDEr scores on Emu Edit test set. We show consistent improvements with synthetic augmentation (EE + Syned) on all models.

Models	**EE**	**Syned+ EE**	Improvement (%)
SCORER	21.1	23.2	-
CLIP4IDC	32.4	35.9	10.8
BLIP2IDC	100.0	106.8	6.8

In Table 4.7, we study BLIP2IDC's behaviour with respect to its training data, being either Syned, EE or both on different modification categories. Combining datasets is generally the best configuration overall, except for the text (T) category. We assume this is due to the poor capacity of most Image Editing models to generate text in images, which hampers the performance of the synthetic augmentation. The worst categories in terms of CIDEr score performance are the Style and Global categories, which are very

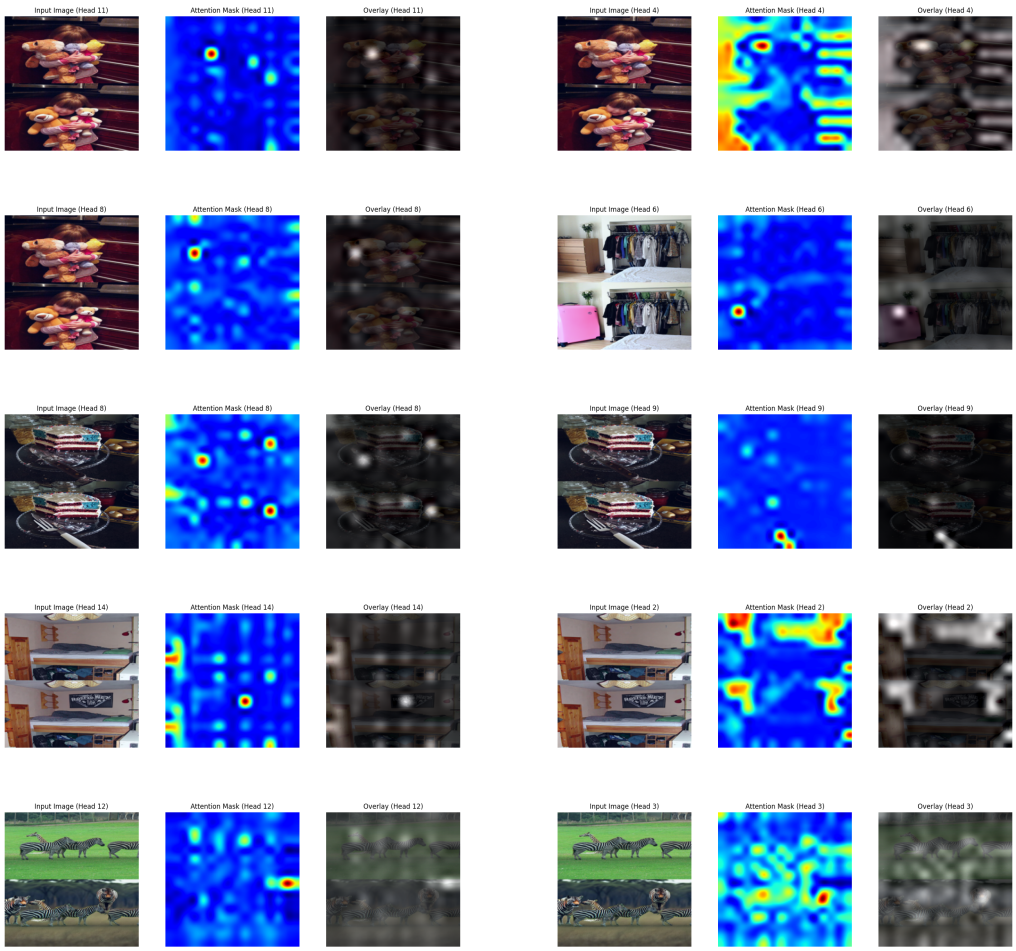


Figure 4.9 – Attention maps from BLIP2IDC ViT for different attention heads. The ViT tries to attend to corresponding parts of the images and is able to find new entities in the scenes. The overlay image assigns weights to pixels based on their attention scores: brighter pixels indicate higher attention.

subjective changes. We report per type of change results in Tab. A.3.

4.4.4 Qualitative analysis

We present in Fig. 4.10 a comparative analysis of state-of-the-art IDC models' outputs across different scenarios: in-distribution image-pairs from the train and test set of EE, and out-of-distribution images from the web, to assess the zero-shot capabilities of the models.

Table 4.7 – CIDEr Performance Metrics by Change Category and Dataset Version. This table lists the CIDEr score of each following categories: Add (A), Text (T), Background (B), Color (C), Style (S), Global (G), Remove (R), Local (L) and Overall (O) which is the CIDEr computed over all the samples of the test set.

Dataset	A	T	B	C	S	G	R	L	O
Syned	102.5	147.2	111.8	151.9	12.8	**31.6**	72.3	61.0	94.8
EE	101.0	**147.7**	**119.5**	155.8	30.5	29.5	93.4	76.2	100.8
Syned+EE	**107.4**	147.4	112.4	**164.4**	**37.1**	30.4	**111.2**	**77.8**	**106.8**

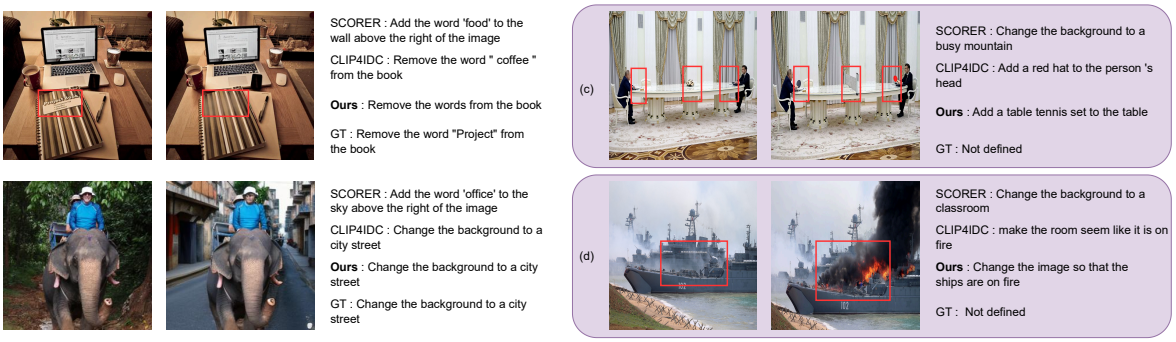


Figure 4.10 – Results of state-of-the-art IDC models on: (a) a train sample of Emu Edit dataset, (b) a test sample and (c-d) zero-shot samples found in the wild. BLIP2IDC (**Ours**) is able to capture fine-grained differences, describe complex scenes, and generalize well to unseen data.

We observe SCORER's limitations in handling real-world images in all cases. It only succeeds in identifying a text modification in the train sample, and confuses actions like adding or removing. CLIP4IDC performs well on samples from EE but struggles with zero-shot generalization. Specifically, it misidentifies the "table tennis set" due to its absence in the training data, confusing it with a similar but incorrect object. Additionnaly, it fails to understand the context in which the object is added. BLIP2IDC, on the other hand, excels on in-distribution data and demonstrates superior capability in generalization through effective use of cross-attention. It accurately recognizes and describes the entire scenes in Fig. 4.10(c-d), including complex additions like a table tennis set, by leveraging its pretraining. This allows BLIP2IDC to not only identify the presence of fire in Fig. 4.10(d) but also specify the object affected, showcasing its ability to connect semantic changes with their context.





Zero-shot BLIP2IDC output: « Change the background to Nigeria »

Figure 4.11 – Ethnic bias

4.4.5 Limitations

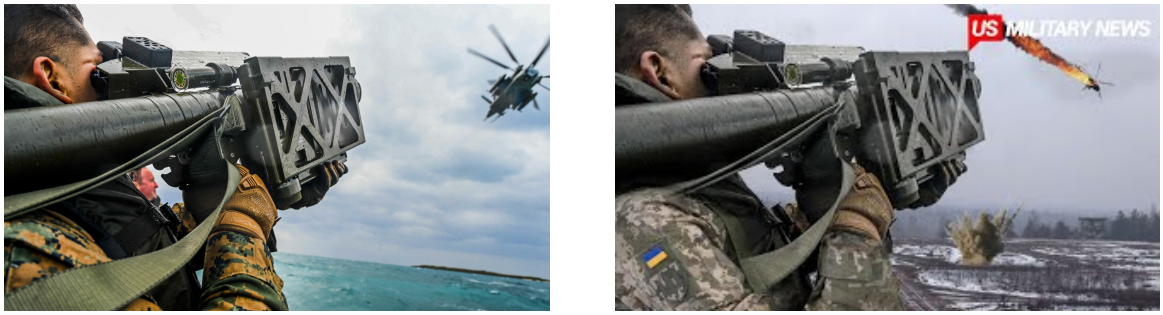
BLIP2 is pre-trained on large image-text datasets sourced from the web, such as LAION (Schuhmann et al. 2022). As BLIP2IDC is a finetuned version of BLIP2, it inherits the biases observed in the BLIP2 model due to its pretraining data. Our synthetic augmentation pipeline also relies on several synthetic generation models and thus inherits their limitations, such as ethnic misrepresentation or adversarial vulnerabilities. The LLM used for generating the ground truth variations will hallucinate, even with careful prompting.

Furthermore, Image Editing models, while improving, are still not able to perform every type of modification. Careless use of these models can deteriorate the dataset quality if the editing instructions are not well aligned with the model's capacity. We finally emphasize the in-domain limitation: IDC datasets can hardly be exhaustive in their ability to handle all types of change, due to the multiplicity of modification interpretations and descriptions. For a given domain, a more suitable approach is to curate a limited number of well-defined changes and to build upon them, allowing for better control.

See biased outputs from BLIP2IDC in Figs. 4.11 and 4.12.

4.5 Conclusion

In this work, we propose a novel framework that adapts existing image captioning models to the IDC task in order to benefit from their extensive pre-training. Image captioning pretraining data are indeed easier to obtain than those for IDC. This allows to effectively leverage the global world knowledge from large-scale unlabeled datasets and



Zero-shot BLIP2IDC output: "Change the background to Ukraine"

Figure 4.12 – Geographic bias

transfer it to IDC using a smaller amount of data. As a demonstration, we adapt BLIP2 to the IDC task. By encoding image differences at the pixel level rather than relying on the traditional dual-stream scheme, we ensure a more informative and direct approach to understanding image variations. BLIP2IDC outperforms existing methods on standard benchmarks. Additionally, we introduce a synthetic augmentation strategy that not only addresses critical challenges of data scarcity and the need for robust, real-world applicable architectures but also sets a new benchmark in IDC performance. This paves the way for application of IDC to more diverse data and types of edits.

5

Semantic Watermarking for Authentication

Contents

5.1 Introduction 64
5.2 SWIFT: Semantic Watermarking for Image Forgery Thwarting 65
5.2.1 Method 66
5.2.2 Experiments & Results 72
5.3 Fast, Secure, and High-Capacity Image Watermarking with Autoencoded Text Vectors 77
5.3.1 Method 78
5.3.2 Experimental setting 83
5.3.3 Watermarking model robustness and perceptibility 84
5.3.4 Text Autoencoder 86
5.3.5 End-to-end comparison 91
5.4 Conclusion 99

This chapter is based on *Evennou et al., SWIFT: Semantic Watermarking for Image Forgery Thwarting*, published at WIFS2024, and *Evennou et al., Fast, Secure, and High-Capacity Image Watermarking with Autoencoded Text Vectors*, 2025, under review (ECCVW).

5.1 Introduction

Many technical means can verify the authenticity of multimedia content. This ranges from a digital signature stored in the metadata like in the recent C2PA (Coalition for Content Provenance and Authenticity) and IPTC (International Press Telecommunications Council) initiatives, to passive forensics (Mo Chen et al. 2008; Farid 2016) and active fragile watermarking (Fridrich, Goljan, and Baldoza 2000; Utku Celik et al. 2002). Recent regulatory developments, such as the EU AI Act and Chinese AI Governance rules, place watermarking among the most practical technical tools for ensuring trustworthy AI content detection.

The main difficulty resides in making a clear cut between benign processing which comprise common editing in the entertainment industry and malicious transformations which purposely modify the content. Semi-fragile watermarking faces this challenge: it should be robust to benign processing but fragile to deeper transformations. At the decoding side, its absence reveals that the piece of content has been modified beyond the accepted limit. This limit between benign and malicious editing cannot easily be defined in mathematical terms, although in real life the difference is straightforward: any modification of the semantics is malicious.

This work investigates the idea of hiding semantic information within the cover in an imperceptible and robust manner. The verification amounts comparing the semantics of the content with the decoded information. To the best of our knowledge, embedding its own semantics into the content itself to ensure integrity is an unexplored research path. A first challenge lies in the poor capacity of robust image watermarking. Multi-bit watermarking embeds messages into images but typically only supports up to 64 bits of data transmission, and the most advanced deep-learning schemes reach only a few hundred (256 bits for VideoSeal (Fernandez, Elsahar, et al. 2024)); higher capacity schemes exist but with a much lower robustness (Jia, Fang, and W. Zhang 2021). The second challenge is the representation of the semantic of an image whose definition is still a matter of debate. We chose the textual description of the image given by an automatic captioning model, such as BLIP2 (J. Li, D. Li, Savarese, and S. Hoi 2023), as the message to be hidden.

The scenario establishes a covert channel between two entities: Alice, the sender who authenticates the original work, and Bob, the recipient tasked with verifying its authenticity. The cover work may undergo modifications by a third party, referred to as

Eve, acting as an intermediary. Eve's alterations may be intentional, involving semantic edits, or unintentional, comprising benign changes. The crux of our method lies in Bob's ability to recover the message embedded by Alice. By comparing the recovered message with the received work, Bob makes informed decisions about the nature and extent of any change. The robustness of the communication channel despite potential interferences is key.

We argue for a paradigm shift: watermarking as semantic communication. The embedded message is usually a meaningless identifier, and existing approaches rarely provide any security mechanism, leaving much of the channel underutilized. Instead of embedding arbitrary bit sequences, we embed meaningful textual messages. To this end, we propose to increase the utility, re-usability and flexibility by disentangling the watermarking layer from the encoding layer. The watermarking layer is responsible for hiding a high-dimensional real-valued unit-norm vector in the cover while optimizing robustness to various transforms and the watermark imperceptibility. The encoding layer is responsible for encoding a message as a signal to be transmitted on this noisy communication channel. The decoding layer then retrieves the message with some confidence level. On this principle we build two schemes, presented in turn: SWIFT and LatentSeal. The former leverages a TCCSK modulation combined with an Arithmetic Encoder-Decoder to optimize the payload at the encoding layer while the latter directly trains a text autoencoder to craft such payload. On the watermarking layer, SWIFT adapts HiDDeN (Zhu et al. 2018a) while LatentSeal builds upon VideoSeal (Fernandez, Elsahar, et al. 2024). Both schemes provide a trustworthiness mechanism to estimate at inference the integrity of the decoded message.

5.2 SWIFT: Semantic Watermarking for Image Forgery Thwarting

SWIFT is our first realization of semantic watermarking. We finetune a large language model for captioning and combine it with an arithmetic codec to compress the payload as in LLMZip (Valmeekam et al. 2023b). An encoding layer then encodes this variable-length binary message into a vector to be hidden in images. Finally, Hider-$\mathbb{R}$, an encoder-decoder network inspired by HiDDeN (Zhu et al. 2018b) and jointly trained to embed and extract high-dimensional unit-norm vectors, hides the vector in the cover. Three major features stem from this combination: reliability, as a confidence metric on the decoded caption

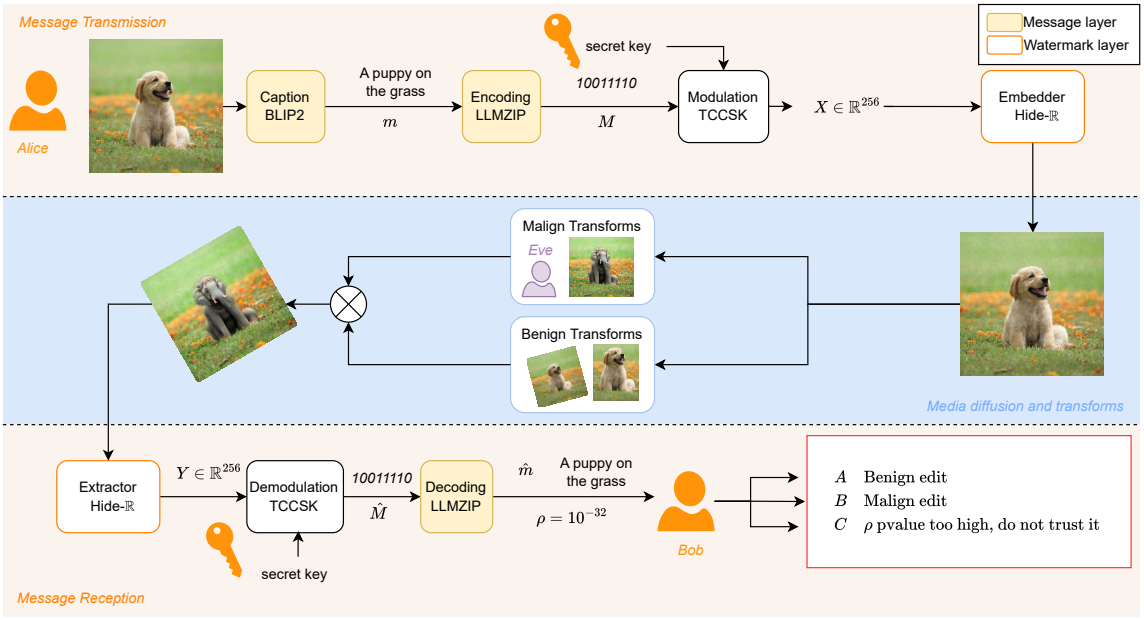


Figure 5.1 – Overview of SWIFT. To ensure the integrity of an image, Alice first leverages the message layer to tailor a semantic representation of the image, in our case a caption compressed in a lossless fashion. The resulting bit stream is fed to the TCCSK modulation layer thus enabling security and confidence (see Section 5.2.2), and then to the watermarking layer based on our Hide-$\mathbb{R}$ encoder-decoder neural network. At reception, Bob executes the inverse process with a secret key and obtains both the caption and the p-value ρ. If ρ is low enough, Bob can entrust the decoded caption and compare the image he received with the caption, enabling comparison between a proxy of the original image and the received one.

gives an informed decision-making about authenticity; security, as the design guarantees it via a secret key; and performance, as SWIFT achieves state-of-the-art results across various benign and malicious transforms. Its main limitation is a tight payload ceiling, which constrains how much semantics can be transmitted and motivates the next section.

5.2.1 Method

This section presents the design of the encoding and watermarking layers. We break down the encoding layer into two primary components: the message layer and the modulation layer. Fig. 5.1 depicts our method.

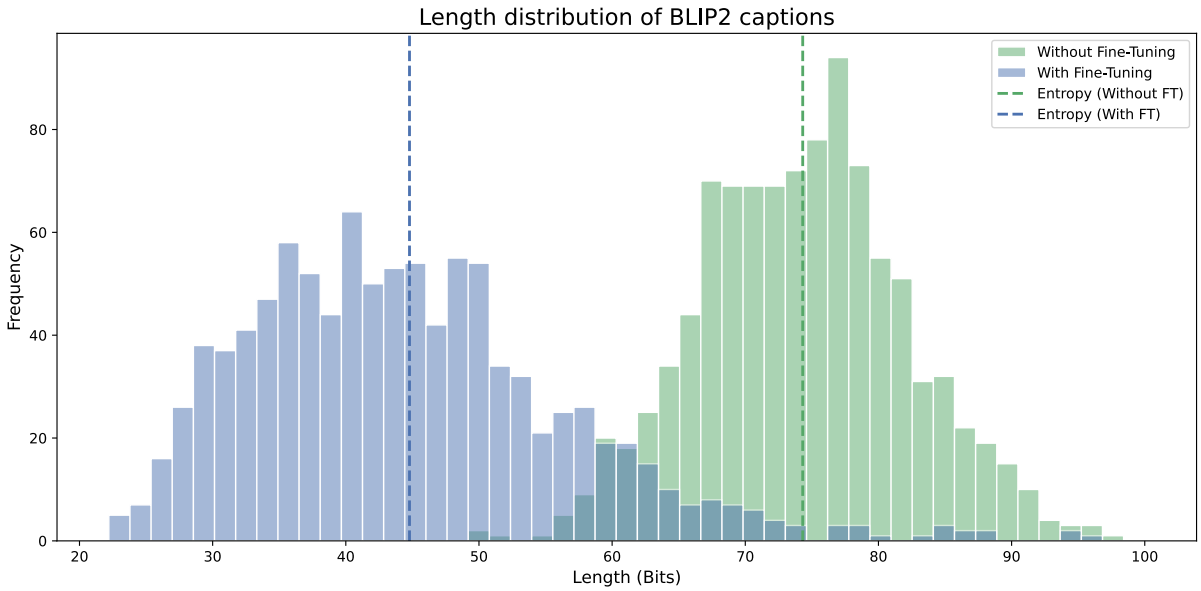


Figure 5.2 – Length distribution of BLIP2 captions using LLMZip. We show that the finetuned version leads to entropy reduction and thus is more efficient to encode captions.

The message layer

Alice wants to transmit a message M to Bob so that he can assess the integrity of the cover image. Alice uses an image captioning model like BLIP2 (J. Li, D. Li, Savarese, and S. C. H. Hoi 2023) to generate the caption m of the cover.

Alice uses arithmetic coding (Pasco 1977) for losslessly compressing m into M to reduce the number of bits.

As in LLMZip (Valmeekam et al. 2023a), Alice takes advantage of a LLM to model the distribution of the messages and improve the compression. She uses OPT-125m (S. Zhang, Roller, Goyal, Artetxe, Moya Chen, S. Chen, Dewan, M. T. Diab, et al. 2022) finetuned on BLIP2 captions from $2,000$ MSCOCO validation set images. This acts as oracle and gives the probability of each caption symbol used by the arithmetic coding (Y. Huang et al. 2024).

Figure 5.2 shows that finetuning OPT on BLIP2 reduces the mean capacity needed from 75 to 45 bits.

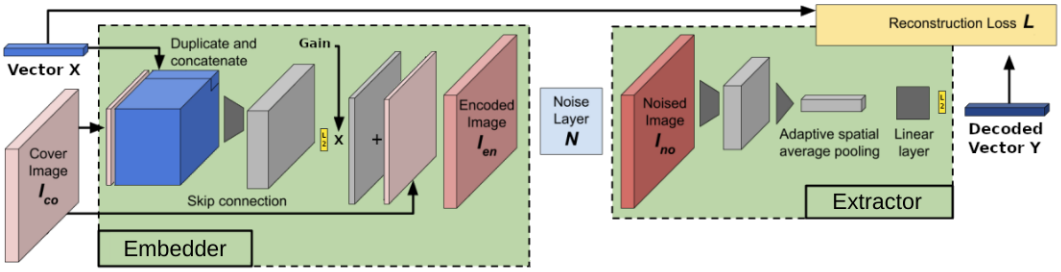


Figure 5.3 – Hide-$\mathbb{R}$ architecture. We use a L2 norm to control the watermark power to enforce a target PSNR on both the watermark signal in the embedder and on the decoded vector in the extractor.

The watermarking layer

Modern watermarking leverages deep-learning to learn end-to-end how to embed a message into a cover image. It enables robustness against benign edits by performing augmentations between the watermark embedding and watermark extraction stages (Yang et al. 2023). The most famous example is HiDDeN (Zhu et al. 2018b) based on two convolutional neural networks (CNN) jointly trained to embed and extract a fixed-length binary message.

Figure 5.3 depicts our Hide-$\mathbb{R}$ architecture, resulting from several modifications of HiDDeN. The input data works with unit vectors in high-dimensional real space instead of binary messages. Specifically, we draw random samples X uniformly distributed on the surface of the unit hypersphere in $\mathbb{R}^D$ with $D = 256$:

$$X = \frac{Z}{\|Z\|} \quad \text{with} \quad Z \sim \mathcal{N}(0, I_D). \tag{5.1}$$

We extend the number of channels in the convolutional layers to $1.5D$ instead of the fixed 64 to account for higher dimension D than the message length $L = 30$ proposed in the original paper. Signal X is concatenated with the cover image I_{co} along the channel dimension before the first convolutional layer. Instead of using a discriminator, we opt for a fixed PSNR budget which both enforces imperceptibility in a flexible way and speed up the learning process. The training minimizes the reconstruction loss $\|X - Y\|$ between X and the reconstructed unit vector Y.

This framework gives a zero-bit watermarking system. Assume X_0 is drawn according to (5.1) from a pseudo-random generator seeded by the secret key K associated with

a fixed index $M = 0$. A watermarking signal is deemed present if the cosine similarity $C_0 = Y^\top X_0$ is above a threshold c. Under the hypothesis $\mathcal{H}_0$, the cover is not watermarked or watermarked with another secret key K'. Then, the p-value is defined as the probability of having higher cosine similarity C_0 than the threshold c and given by:

$$\rho_0(c) = P(C_0 \geq c) = \begin{cases} \frac{1}{2} I_{1-c^2}\left(\frac{D-1}{2}, \frac{1}{2}\right), & \text{if } c > 0 \\ \frac{1}{2} I_{c^2}\left(\frac{1}{2}, \frac{D-1}{2}\right), & \text{otherwise,} \end{cases} \tag{5.2}$$

where I is the regularized incomplete beta function. This illustrates how a confidence value is available to Bob at the watermark extractor.

The modulation layer

Multi-bit watermarking with confidence

One way to use zero-bit watermarking to send a N-bit message M is to share 2^N different secret keys, each associated with one possible message. Then Alice selects the key K corresponding to the message to send, and Bob runs the watermark extractor with all the 2^N possible keys. Bob selects the decoded message as:

$$\hat{M} = \underset{m \in [\![0, 2^N - 1]\!]}{\operatorname{argmax}} (C_m) \tag{5.3}$$

The p-value under $\mathcal{H}_0$ of decoding this specific message $\hat{M}$ by chance in a non-watermarked image is given by:

$$\rho_1(c) = 1 - (1 - \rho_0(c))^{2^N} \lesssim 2^N \rho_0(c) \text{ if } 2^N \rho_0(c) \ll 1 \tag{5.4}$$

Equation (5.4) reveals a trade-off between minimizing $\rho_1(c)$ and increasing the length N of the message: If $N > -\log_2(\rho_0(c))$, it is likely that the decoded message cannot be trusted by Bob. This gives an approximation of the maximum quantity of information that may be transmitted.

Modulating variable-rate signals

In the previous section, Bob needs to test 2^N keys which becomes intractable as N grows. An alternative is to generate multiple carriers from the same pseudo-random

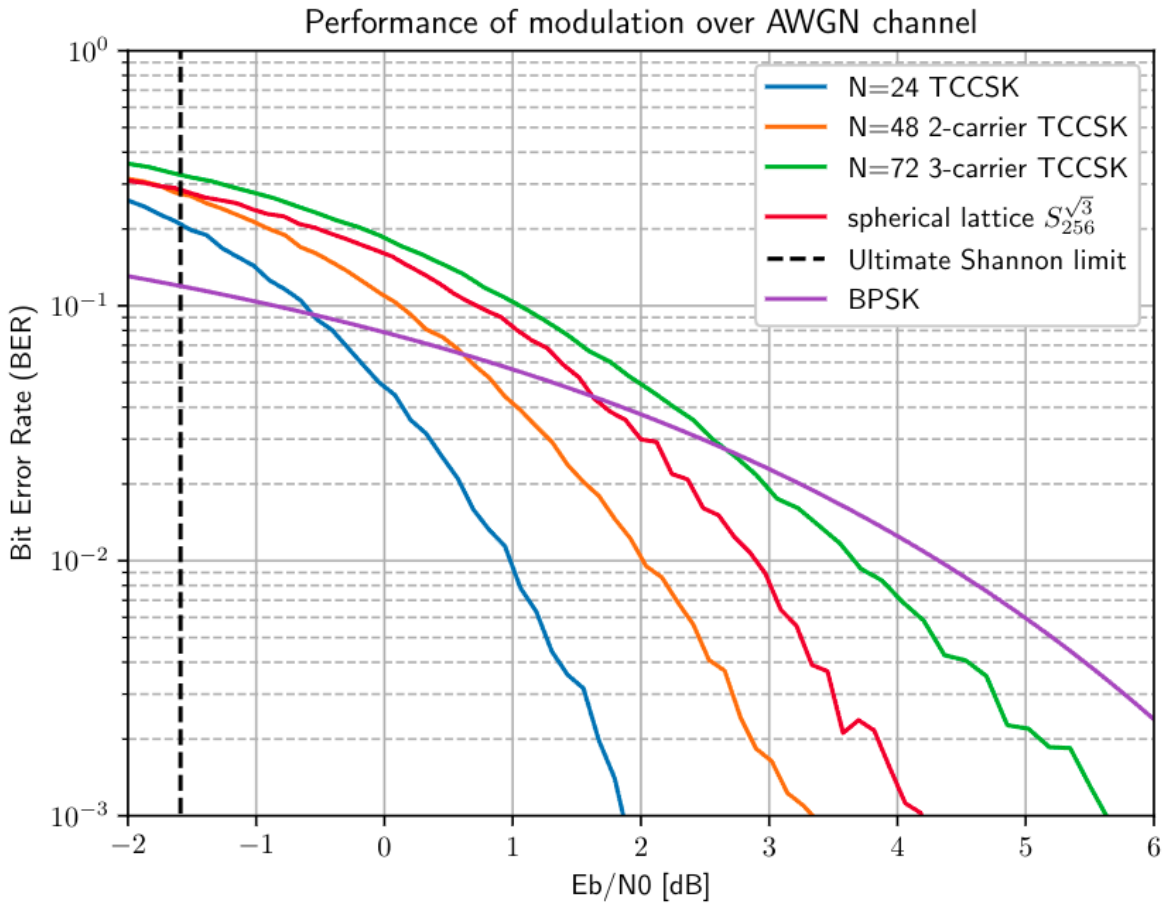


Figure 5.4 – Performance of different encoding schemes under additive white Gaussian noise. $S_{256}^{\sqrt{3}}$ is the spherical lattice from (Alexandre Sablayrolles and Jégou 2019), capable of encoding $22108160 \simeq 2^{24.4}$ messages. TCCSK is most suited for our use case as it can cope with variable length messages.

generator seeded by secret key K, and to combine them to produce $X \in \mathbb{R}^D$. This work uses truncated cyclic shift keying (TCCSK) modulation (Dillard et al. 2003) for its improved performance in the additive white Gaussian noise (AWGN) channel compared to Binary Phase Shift Keying (BPSK) (see Fig. 5.4). The message M is split into T equal blocks of length L, padding the last block with 0 if necessary. Then the j-th block is encoded from a carrier $Z_j \in \mathbb{R}^{2^L} \sim \mathcal{N}(0, I_{2^L}), j \in [\![0, T-1]\!]$ by cyclically shifting it by a number equal to the value represented by the L-bit sub-message $M_j \coloneqq M_{[\![jL,(j+1)L]\!]}$. The carriers are truncated to dimension D, normalized, summed, and normalized again for transmission:

$$X = \frac{Z}{\|Z\|} \quad \text{with} \quad Z = \sum_{j=0}^{T-1} \frac{Z_j^{M_j}}{\|Z_j^{M_j}\|}, \tag{5.5}$$

where Z^a denotes a cyclic shift of Z by a followed by truncation to the first D dimensions. Decoding is achieved by recovering each sub-message from:

$$\hat{M}_j = \underset{k \in [\![0, 2^L-1]\!]}{\operatorname{argmax}} (Y^\top \frac{Z_j^k}{\|Z_j^k\|}). \tag{5.6}$$

The p-value ρ is then obtained from Fisher's combined probability test (Fisher 1992) on evaluations of (5.4) for each submessage:

$$\rho = 1 - \gamma(T, -\sum_{j=0}^{T-1} \log(\rho_1(C_j))), \text{ with } C_j = Y^\top \frac{Z_j^{\hat{M}_j}}{\|Z_j^{\hat{M}_j}\|}, \tag{5.7}$$

where γ is the lower incomplete gamma function. Although we would ideally want to set L to N, testing all 2^L cyclic shifts becomes untractable as L grows. Proceeding by blocks tackles the variable-length of the message. Yet, some capacity is lost due to padding unless the message length N is a multiple of L. Finally, to illustrate the flexibility of the approach, we also compare TCCSK with lattice-based modulation on the spherical lattice used in (Alexandre Sablayrolles and Jégou 2019). Although this method is very fast and better than BPSK for the 24-bit case, its efficiency is lower than TCCSK with a single carrier (see Figure 5.4).

Security Our approach adheres to Kerckhoffs's principle, relying solely on the shared secret key K between Alice and Bob for security. We assume Eve has full knowledge of the system, except for K. In content authentication, the attacker's primary goal is to forge watermarked content without K (spoofing attack), rather than removing existing

watermarks. Each system use reveals at most D out of 2^L carrier values, reducing the urgency for key rotation if usage is limited. As a symmetric system, both Alice and Bob can produce forgeries using K. Thus, mutual trust between them is assumed, with Eve being the only untrusted party in our threat model.

5.2.2 Experiments & Results

This section introduces the evaluation of SWIFT and recent watermarking methods for the task of message recovery.

Evaluation

Metrics. The benchmark compares watermarking methods by their Message Recovery Rate (MRR), which is defined as the rate of messages being perfectly transmitted without any modifications, over a test set of watermarked images $\mathcal{I}_t$. Let m_i be the original caption and $\hat{m}_i$ the corresponding recovered caption for an image $I_i \in \mathcal{I}_t$. The MRR is defined as follows:

$$MRR = \frac{1}{|\mathcal{I}_t|} \sum_{i=1}^{|\mathcal{I}_t|} \delta(m_i, \hat{m}_i) \tag{5.8}$$

We chose this metric to ensure practicability and accurately assess the robustness of a system. A watermarking system designed for our task should strive to reach 100% MRR, especially when no confidence metric is available at the decoding step, which is the case of all systems but SWIFT.

Test set. Our test set $\mathcal{I}_t$ is composed of 20,220 images. We use the Emu Edit test set (Sheynin et al. 2023) which comprises 2,022 images from MSCOCO (X. Chen et al. 2015) and editing instructions for Image Editing models from 8 classes (*local, add, remove, global, text, background, style, color*). For each image, we perform 6 benign and 1 malicious transformation with four variations in classifier-free guidance. Benign ones are chosen to be realistic in a web setting, or quite important distortion-wise but without semantic alterations: crop 40% of the image surface, random noise, grayscale conversion, resize to 128×128, jpeg compression with quality coefficient at 50. Malign ones are images edited by a diffusion model according to Emu Edit instructions, supposed to change the meaning of the cover work.

Table 5.1 – Message Recovery Rate by Transform Type. All the methods share the same messages from the LLMZip as inputs. We provide results for SWIFT at 2 different target PSNR for fair comparison with existing methods.

Transform	Message Recovery Rate(%) ↑				
	SWIFT	SWIFT	SSL	TrustMark(Q)	TrustMark(Base)
PSNR(db)	40	42	40	42.5	41.6
global	**63.7**	45.1	0.45	0.0	39.0
text	**65.3**	45.9	0.38	0.0	40.0
style	**64.8**	45.3	1.32	0.0	38.3
local	**65.9**	48.2	0.39	0.0	40.2
background	**66.5**	48.3	0.75	0.0	43.9
color	**65.3**	46.9	1.53	0.0	44.6
remove	**67.2**	50.0	0.0	0.0	48.0
add	**69.5**	49.2	0.0	0.0	43.7
crop 40%	82.4	76.7	50.1	86.4	**91.9**
noisy	**75.5**	65.4	0.0	11.8	17.2
resize 128	**95.1**	91.3	0.00	0.0	0.0
grayscale	**95.8**	93.8	2.82	78.0	91.2
jpeg@50	**96.2**	93.5	1.5	82.1	93.2
original	**96.6**	95.1	92.6	94.6	95.7

Comparison with state of the art

Table 5.1 shows the results of state-of-the-art methods SSL and TrustMark against SWIFT: we observe superior resilience to malign transforms while maintaining state-of-the-art performance on benign modifications with significant improvement on resize and grayscale transforms due to our training. We provide another version of SWIFT to watermark at 42db which performs better than TrustMark(Base) on almost all settings.

Confidence metric

After the TCCSK modulation, a message M is encoded into a vector X on the 256-d hypersphere. Given an encoded message X, due to transforms during transmission, Y is the noisy extracted vector. At inference, Y is decoded by the TCCSK demodulation into $\hat{M}$. X is unknown, but let $\hat{X}$ be the perfect encoding of $\hat{M}$ and C the cosine similarity computed between $\hat{X}$ and Y (see Figure 5.5). We define our practical confidence metric as ρ (5.7). We assume that Y close to the perfect representation $\hat{X}$ of the decoded message

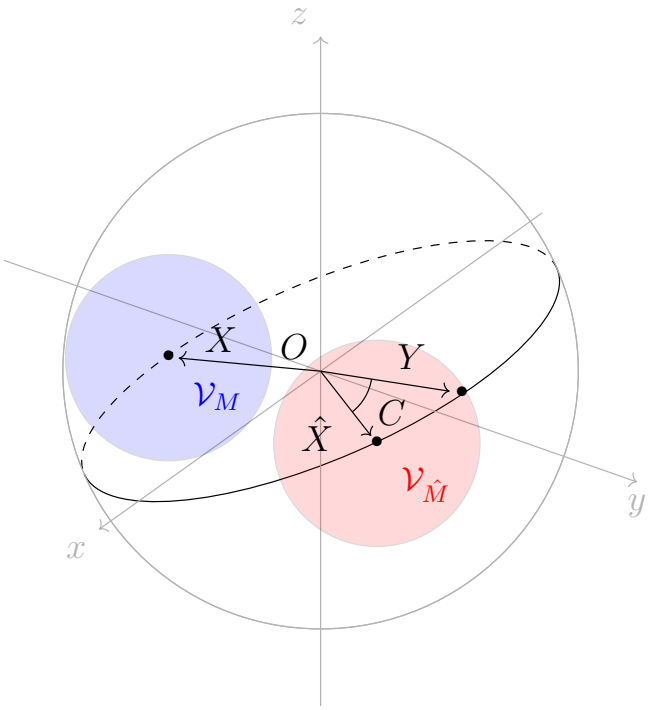


Figure 5.5 – Representation of the message space after modulation. The circles $\mathcal{V}_M, \mathcal{V}_{\hat{M}}$ illustrate Voronoï cells mapping to different binary messages on the surface of the hypersphere. X is associated to the message to be hidden M. The extraction retrieves Y, which is decoded into $\hat{M}$, while $\hat{X}$ results from the modulation of $\hat{M}$, accounting for the perfect representation of $\hat{M}$. The similarity between Y and $\hat{X}$ is given by C. This provides a confidence score for the decoding as explained in Section 5.2.2. The given example illustrates a failure case with a wrong decoded message and a low confidence reflected by a high value of ρ.

$\hat{M}$ means that Y is also close to the (unknown) originally embedded vector X. Thus, a low ρ would entail a low number of errors at the decoding step, as confirmed by the Pearson correlation coefficient of -0.89 computed between $8,000$ ρ values and corresponding MRR. Table 5.2 shows the MRR on watermarked images under several scenarios: watermarked images, benign attacks and malign attacks, and three ρ-thresholds. Only messages with $\rho < \rho$-threshold should be trusted. An adequate threshold ensures the perfect extraction, meaning 100% MRR, of the watermarked caption. Note that ρ also refers to the error probability on non-watermarked images: the higher it is, the more a non-watermarked image could be flagged as watermarked.

A confidence metric could also be used in a multi-bit setting with fixed-length code of 64 bits. Padding a n-bit message with $64 - n$ equiprobable random bits drawn from a synchronous source shared by Alice and Bob (e.g. via a secret key K) allows to carry confidence information, at the expense of capacity. Indeed, Bob can check the padding bits and discard any message not matching the expected sequence. In this case, assuming the multi-bit system outputs random codes uniformly under $\mathcal{H}_0$, there is still a 2^{-64+n} probability of ending up with the expected sequence by chance, giving $\rho \geq 2^{-64} \simeq 5e-20$. Our method benefits from comparable confidence, with $\rho = 1.4e-16$ to be protected against all attacks, along with a much greater capacity.

ρ **Threshold**	**Scenario**	**MRR**	**CDF**	**Confidence**
1.0	Watermarked	96.6	100	None
	Benign Attacks	89.0	100	
	Malign Attacks	66.1	100	
4.2e-13	Watermarked	100	82.6	Low
	Benign Attacks	96.2	60.1	
	Malign Attacks	94.1	31.1	
2.3e-15	Watermarked	100	72	Medium
	Benign Attacks	100	48.3	
	Malign Attacks	99.3	20.2	
1.4-e16	Watermarked	100	66.2	High
	Benign Attacks	100	42.3	
	Malign Attacks	100	18.1	

Table 5.2 – Message Recovery Rate (MRR) and Cumulative Distribution Function (CDF) on Emu Edit for different ρ thresholds with SWIFT(40db). Only messages with $\rho < \rho$-threshold are decoded.

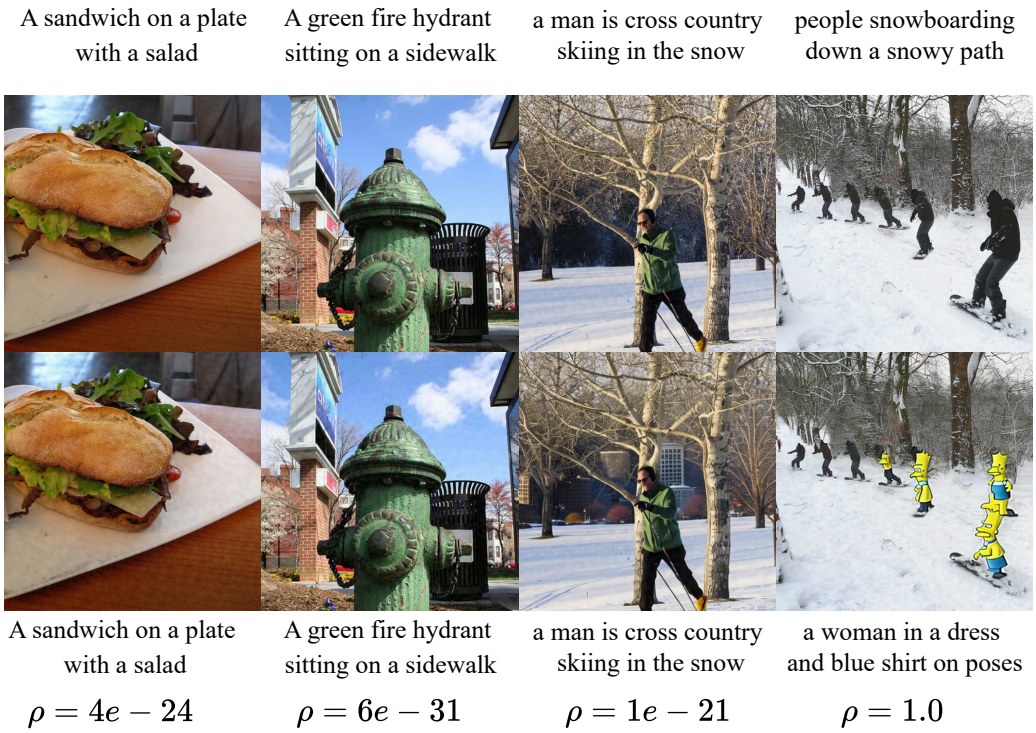


Figure 5.6 – Examples of recovered messages under different transforms. From top to bottom: message, cover image, watermarked image with or without transforms, retrieved message and confidence metric (the lower, the better). From left to right: no transform, jpeg compression with a quality coefficient at 50, background edit, local edit.

Qualitative results

In Figure 5.6 we present samples of watermarked images from Emu Edit. In the first column we see an original image and its watermarked version: the watermark is visible in the bottom left on the plate. The second column depicts our resilience against heavy jpeg compression and the two last ones demonstrates SWIFT behaviour against malign edits. The background edit leaves the foreground intact and thus does not destroy the watermark. In the last column, the local edit remove watermarked regions and lead to a low confidence score, suggesting the image is altered and cannot be trusted.

5.3 Fast, Secure, and High-Capacity Image Watermarking with Autoencoded Text Vectors

LatentSeal overcomes the payload bottleneck of Section 5.2. We train a lightweight text autoencoder to map sentences into a 256-dimensional latent space, which is then embedded by a finetuned watermarking model and secured through an invertible, key-conditioned rotation. Adapting VideoSeal (Fernandez, Elsahar, et al. 2024), we embed 256-dimensional unit-norm vectors aligned with the latent space of the autoencoder, and augment them with a plug-in latent-space encryption layer based on Haar-random rotations from SO(256) keyed by a secret seed, so that only authorized decoders can recover the correct representation. The structure of the latent space allows us to provide a confidence metric that detects unreliable extractions. Compared to the strongest baselines (VideoSeal + LLMZip), LatentSeal achieves higher BLEU-4 and Exact Match, improves on the 256-bit payload ceiling, and introduces practical mechanisms for security and reliability, as demonstrated on COCO-2017, PixMo-Cap, and WikiText-103.

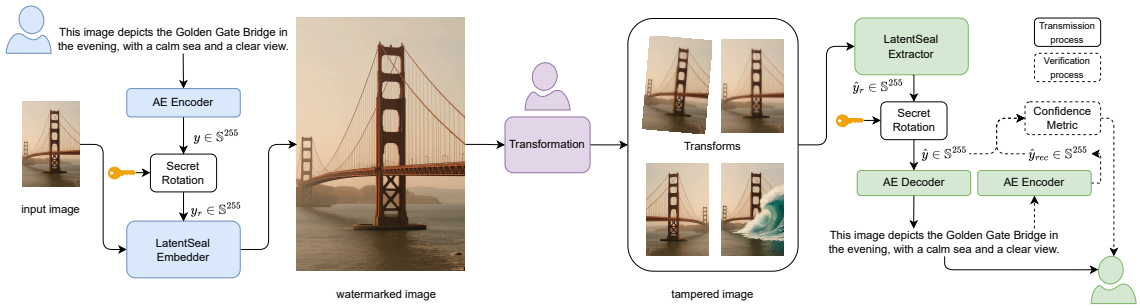


Figure 5.7 – LatentSeal overview illustrated through an image tampering detection application. Bob writes a message m describing the image's content, then feeds it to the autoencoder which outputs a latent vector y. To introduce security, we apply a secret rotation to y conditioned by a secret key and obtain y_r. LatentSeal embeds y_r within the input image. During transmission, the image is intercepted and transformed by Eve. Finally, Alice receives the image, performs the watermark extraction process, and recovers an estimated rotated latent vector $\hat{y}_r$. Using her secret key, identical to Bob's, she applies the inverse rotation to $\hat{y}_r$ and decodes the resulting latent vector $\hat{y}$ to reconstruct the original textual message. By comparing the received image's content with the decoded message $\hat{m}_r$, Alice can detect discrepancies and flag tampering accordingly.

5.3.1 Method

LatentSeal combines a watermarking model and a robust text autoencoder specifically designed for watermarking purpose (Fig. 5.7). The watermark model relies on VideoSeal (Fernandez, Elsahar, et al. 2024) that reaches a capacity of 256 bits with SOTA performance. We modify the model to embed a 256-D unit-norm real-valued vector. The vector dimension represents a trade-off among capacity, robustness, memory requirements, and computation time; larger vectors typically demand more resources while decreasing robustness. The text autoencoder must satisfy two key constraints: (i) its latent space should be 256-dimensional, which is a small size for text embedding (ii) it must be robust to the noise introduced by the watermarking model and the transmission. In this context, having an efficient encoder is crucial, and we choose ModernBert (Warner et al. 2024) finetuned using LoRA (Edward J. Hu et al. 2021) as encoder. We use the [CLS] token as a representation of the input sentence and project it into a 256-D unit-norm space, that constitutes our latent space. For the decoder, we designed a lightweight transformer architecture which allows fast decoding. Robustness is achieved by introducing Gaussian noise into the latent space during training to mimic the impact of the watermark extraction. Despite the challenging constraint of a small latent space, we achieved nearly perfect reconstruction. **The overall architecture necessitates two independent training phases**: one for the watermark model and another for the text autoencoder.

Finally, a key-conditioned reversible rotation is introduced at inference to ensure security in the watermarking process.

Watermark embedder and extractor

As VideoSeal (Fernandez, Elsahar, et al. 2024), the embedder is based on a UNet architecture, and ConvNeXt-v2 is used as as watermark extractor. Our version is trained to embed $y \in \mathbb{S}^{255} = \{y \in \mathbb{R}^{256} : \|y\| = 1\}$ within a cover image I_c of dimension $3 \times 256 \times 256$. The vector embedding occurs in the bottleneck of the embedder (see App. B.2.1 for the detailed architecture). The watermark model should maximize the similarity between the watermark embedded in the cover y and the one extracted $\hat{y}$, given by the cosine similarity.

During training, we sample the vectors $y^{(n)}$ uniformly from the surface of the unit hypersphere:

$$y'^{(n)} \sim \mathcal{N}(0, I_{256}), \qquad y^{(n)} = \frac{y'^{(n)}}{\|y'^{(n)}\|_2}, \tag{5.9}$$

Instead of a binary cross-entropy used in Fernandez, Elsahar, et al. 2024, our version is trained to minimize:

$$\mathcal{L}_{\text{cosine}}(\hat{y}, y) = 1 - \frac{\hat{y} \cdot y}{\|\hat{y}\|_2 \, \|y\|_2} \tag{5.10}$$

Ensuring robustness demands high cosine similarity even under classic image transforms (e.g., colorimetric transforms, crop, JPEG compression). This is achieved by augmentations during training. Imperceptibility requires invisibility from the human eye and is traditionally measured with the PSNR. Following (Fernandez, Elsahar, et al. 2024), we use an MSE term in the YUV domain between the input image and the watermarked one. Besides, we use a scheduled factor before the watermarking signal to reduce the perceptibility during the training, making it increasingly harder to keep a high cosine similarity. Hence the model has to learn to deal with a lower intensity signal. More information about training is given in Appendix B.2.

Text Autoencoder

The proposed autoencoder learns to map input sequences of 30 tokens onto the surface of the unit hypersphere $\mathbb{S}^{255}$. We find 30 tokens to be aligned with the highest capacity of existing multibit watermarker. We use the ModernBert-base (Warner et al. 2024) tokenizer which has a vocabulary size of 50,368. Note that without compression, 30-token sequences correspond to 468 bits of information, given that $log_2(50,368) \approx 15.6$ bits per token. While in SWIFT (Evennou, Chappelier, et al. 2024) transforms text into vectors in two distinct steps, first compressing with LLMZip and then modulating via Truncated Cyclic Code Shift Keying, our autoencoder streamlines these two steps into a single process, all while significantly increasing the overall capacity.

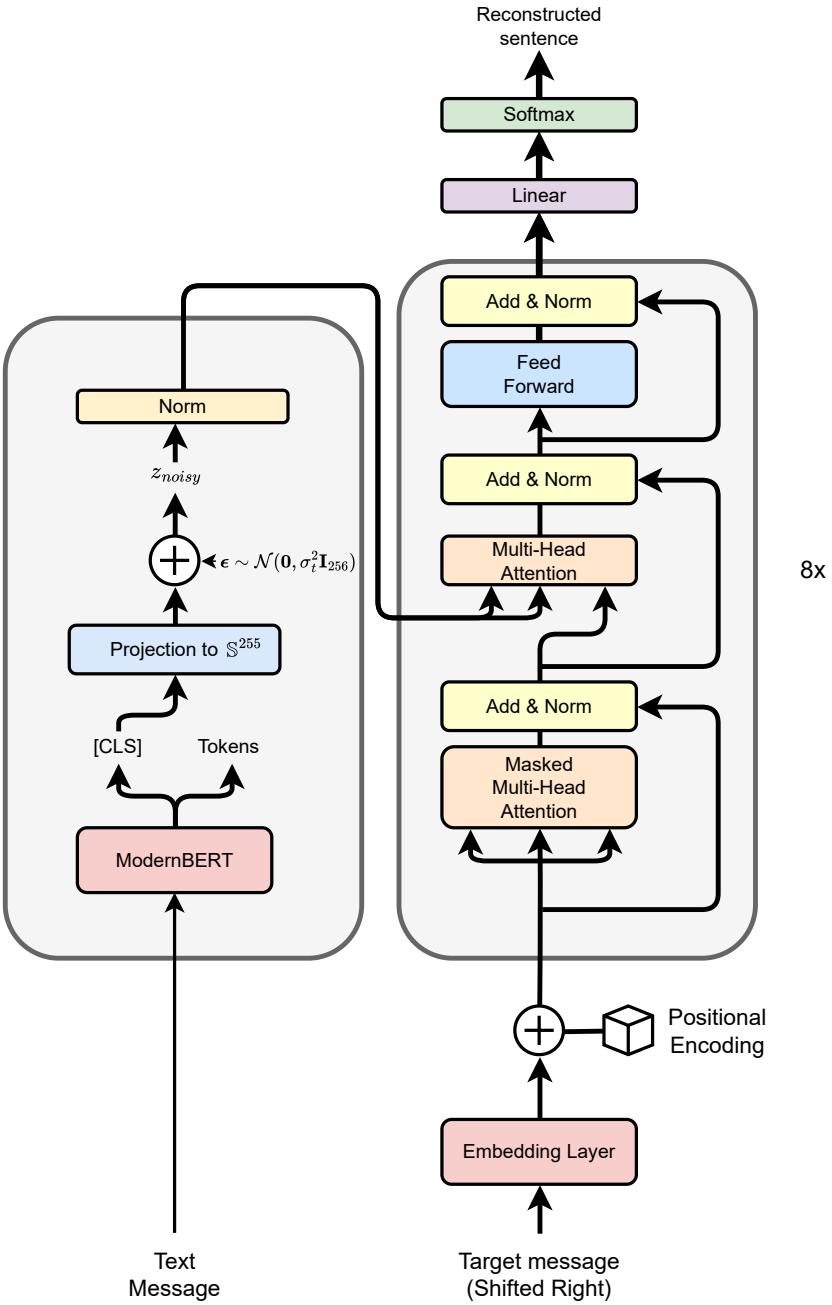


Figure 5.8 – Text autoencoder architecture. Encoder (left) outputs are fed as memory (key,value) to the MHA module of the decoder layers (right).

Architecture The encoder E is a pretrained encoder-only model, ModernBERT-base (Warner et al. 2024), on top of which a projection layer maps the [CLS] token to a 256-D unit-norm space. The design of our decoder D is however central to our method. A standard causal decoder is unsuitable because it lacks a mechanism to incorporate the latent vector at each decoding step. We address this by leveraging the memory mechanism of torch.nn.TransformerDecoderLayer, as depicted in Figure 5.8. Full implementation details are given in Appendix B.1.

In a conventional encoder-decoder transformer architecture (Vaswani et al. 2017), the cross-attention module uses the encoder's output as its key and value tensors. We deviate from this standard approach by not feeding the outputs directly from ModernBERT. Instead, we project the [CLS] token to $\mathbb{S}^{255}$ and, crucially, add noise during the training phase to enhance robustness. This latent vector then serves as the key and value for the decoder's cross-attention module at every decoding step. This critical modification allows the decoder to condition its output on the meaningful watermark. The introduction of noise during training simulates perturbations that can occur from image processing or watermark extraction, thereby training the decoder to effectively denoise its input. Our complete autoencoder has 197M parameters, with 155M in the encoder and only 42M in the decoder. The lightweight design of the decoder enables fast decoding, which is beneficial for real-world applications.

Training The autoencoder $D \circ E$ is optimized with the standard token-level cross-entropy:

$$\mathcal{L}_{\text{rec}} \;=\; -\frac{1}{N_{\text{tok}}} \sum_{t=1}^{N_{\text{tok}}} \sum_{i=1}^{|\mathcal{V}|} \mathbf{q}_i^{(t)} \log\Big(\mathbf{p}_i^{t)}\Big), \tag{5.11}$$

where N_{tok} is the number of non-padded tokens of a sequence, $\mathcal{V}$ denotes the vocabulary and $\mathbf{q}, \mathbf{p} \in \mathbb{R}^{|\mathcal{V}|}$ are respectively the ground-truth and predicted probability vectors. Probabilities are obtained from the decoder logits $\mathbf{z}^{(t)} \in \mathbb{R}^{|\mathcal{V}|}$ with a softmax layer.

Trustworthiness and security

Leveraging a unit-norm vector allows for the implementation of mechanisms that ensure the watermark's reliability and security during inference.

Confidence via spherical p-values We provide a two-step confidence mechanism in decoding at inference. Let $\hat{m} = D(\hat{y})$ be the text decoded from the extracted vector $\hat{y}$.

We first apply an idempotence test: the decoded text is passed once more through the text autoencoder and accepted only if $D \circ E(\hat{m}) = \hat{m}$. A different message indicates a deviation from the autoencoder's distribution, suggesting a too strong distortion of the vector $\hat{y}$. In the other case, we compute the cosine similarity $c = \cos(\hat{y}, E(\hat{m}))$. A high value indicates the latent $\hat{y}$ remained in the same latent vector neighborhood as $\hat{m}$ and the message is likely correct, whereas a low value flags potential corruption.

Under the null hypothesis that $\hat{y}$ is a random unit vector sampled in $\mathbb{S}^{255}$, the one-sided p-value ρ is given by:

$$\rho = \Pr(\cos(\hat{y}, E(\hat{m})) \geq c) = \mathrm{I}_{1-c^2}\left(\frac{255}{2}, \frac{1}{2}\right), \tag{5.12}$$

where $\mathrm{I}_x(a, b)$ denotes the regularized incomplete beta, and $c > 0$. This yields a per-sample surprise score $\rho \in [0, 1]$: small ρ indicates substantial information retention, whereas $\rho \approx 1$ implies independency between $\hat{y}$ and $E(\hat{m})$.

Security with secret key-based rotation The security of our system is implemented at inference via an invertible rotation system conditioned by a secret key. This is a solution by design, as it takes into account both the latent dimension and unit-norm property. This allows the mechanism to be seamlessly integrated post-training without significant computational overhead (see App. B.3). The secret rotation layer is introduced to provide message confidentiality via a key, in line with standard cryptographic practice (I. J. Cox et al. 2008) where security relies on a secret key rather than obscurity of the algorithm. Concretely, a content creator Alice encodes a provenance message m into a latent $y = E(m)$ and then applies a key-conditioned rotation $\mathbf{O} \in \mathrm{SO}(256)$ to obtain $y_r = \mathbf{O}y$, which is embedded into the image. An unauthorized party (Eve) with access to the watermarked image and the public models can only extract a rotated vector $\hat{y}_r$; without the key k she cannot invert $\mathbf{O}$, so $D(\hat{y}_r)$ yields an unintelligible message. By contrast, an authorized verifier (Bob) who knows k reconstructs $\mathbf{O}$, computes $\hat{y} = \mathbf{O}^\top \hat{y}_r$, and recovers m via $D(\hat{y})$. This mechanism is applied post-training, incurs negligible computational overhead, supports arbitrarily many keys, and targets confidentiality rather than robustness, which is handled by the watermarking architecture itself.

5.3.2 Experimental setting

For our experiments, we use 2,000 images from the COCO-2017 validation set as cover images, each watermarked with various text samples from one of the datasets described below. Qualitative examples can be found in App. B.4.1.

Datasets We benchmark on three corpora of ascending difficulty. COCO-2017 (T.-Y. Lin, Maire, Belongie, Bourdev, et al. 2015) supplies 566 k short captions (11M tokens) and serves as a baseline for simple language. PixMo-Cap (Deitke et al. 2024) provides 700 k image captions (21M tokens); these longer, vocabulary-rich descriptions approximate real image-captioning use. WikiText-103-v1 (Merity et al. 2016) offers 1.8M training samples (48M tokens) plus dev/test splits, sourced from Wikipedia. While the two first datasets are related to the image tampering detection application mentioned above, the last one is used for evaluation purposes: its wide thematic diversity and density of named entities stress our system's ability to convey detailed information. Together, the three datasets test robustness from concise, repetitive captions to high-entropy articles.

Baseline We compare LatentSeal against VideoSeal which is the current highest-capacity watermarking model with a 256-bits capacity. Other existing models exhibit significantly lower capacities, with the best alternative to VideoSeal conveying at most 100 bits (Bui, Shruti Agarwal, and Collomosse 2023). VideoSeal is fed with text using LLMZip to get a bitstream that serves as watermark. For a fair comparison, both our text autoencoder and VideoSeal use an input length of 30 tokens, which corresponds as closely as possible to a message length of 256 bits obtained by LLMZip (see Fig. 5.16). The resulting bitstream is truncated if longer.

Training The watermark model and the text autoencoder that make up LatentSeal are trained independently. We deliberately avoid end-to-end training of the watermarker and the text autoencoder. The watermarker is trained to embed and extract random unit vectors on the hypersphere $\mathbb{S}^{255}$, generated by a key-conditioned pseudo-random process, so its behaviour is intentionally independent of any specific upstream encoder and already achieves high fidelity. Jointly training both components would couple the watermarker to the latent distribution of a single encoder, reducing its usefulness as a generic continuous channel for other modalities or encoders, while substantially increasing optimization complexity and the size of the hyperparameter space. In practice, the modular design offers

better generality, stability, and reusability, with no clear evidence that end-to-end training would provide benefits. For the watermarking model, we finetune the VideoSeal model in order to adapt it to our substantial modifications of input and output spaces. For the text autoencoder, the encoder part comprises ModernBert-Base that we finetune using LoRA at rank 32 on q,k,v layers with α at 64. An hyperparameter search was performed for text autoencoder decoder part, see Tab. B.3. Regarding the baseline VideoSeal+LLMZip, the LLMZip used relies on OPT-125M that we finetuned using LoRA (Edward J. Hu et al. 2021).

Computation We used up to 4 H100 GPU to train the autoencoder and test diverse architectures. For most evaluation scripts, we use one A40 GPU. Both watermarking and text autoencoder models are rather frugal and do not need a lot of computing power at inference time. The whole pipeline needs less than 3 GB of VRAM with a batch size of 1 to run.

Security Following (Mezzadri 2007), we draw a rotation matrix uniformly at random with a key k. Both sampling and applying the rotation are cheap, about 1ms, ensuring security at a low cost. Details on implementation and computational cost are given in App. B.3.

Metrics We assess reconstruction quality with the BLEU-4 metric (Papineni et al. 2002b), which accounts for n-gram precision between a candidate and a reference sentence. Moreover, we use the exact match rate (EM) to measure the rate of messages perfectly reconstructed. We report speed as the number of tokens processed per second.

5.3.3 Watermarking model robustness and perceptibility

Table 5.3 shows the impact of valuemetric and geometric transformations on the watermark. We obtain significant improvements in terms of BLEU4 metric over a wide range of transformations. We unsurprisingly conserve high cosine similarity as most of the transformations were included while training the watermarking model. Watermark perturbation is measured by PSNR between original and watermarked images, we aim at 42dB in our experiments as we require high-standard imperceptibility.

To target a specific PSNR at inference, we modulate the watermarking perturbation

Table 5.3 – Robustness comparison on PixMo-Cap. Best results in **bold**. All differences are significant at $p < 0.05$ (Wilcoxon).

Transform	LatentSeal		VideoSeal		Transform	LatentSeal		VideoSeal	
	BLEU-4	EM(%)	BLEU-4	EM(%)		BLEU-4	EM(%)	BLEU-4	EM(%)
identity	**97.34**	**92.84**	92.97	89.20	*Color/Contrast*				
Geometric					Sat 0.5	**96.74**	**92.18**	93.14	89.10
HFlip	**97.15**	**92.23**	93.52	89.40	Sat 1.5	**96.86**	**92.13**	92.59	88.14
Rot 5	**95.08**	**87.89**	89.67	84.36	Cont 0.5	**95.84**	**90.41**	92.69	88.60
Rot 10	**88.26**	**74.17**	79.82	71.14	Cont 1.5	**84.95**	**70.23**	72.96	64.18
Rot 90	**94.54**	**87.39**	90.66	85.67	Bright 0.5	**95.95**	**90.36**	91.75	87.18
Crop 90	**93.27**	**83.50**	80.14	71.75	Bright 1.5	**83.48**	**69.93**	72.00	63.72
Persp 0.3	**85.01**	**69.88**	77.26	67.66	Hue 0.1	**92.72**	**82.34**	79.53	71.39
Persp 0.5	**36.99**	**10.99**	9.69	0.96	Hue -0.1	**92.32**	**81.58**	79.88	71.64
Noise/Blur					*JPEG Compression*				
GNoise 10	**13.45**	0.00	1.83	0.00	Jpeg 80	**89.39**	74.27	86.08	**78.71**
GBlur 5	**8.75**	**0.55**	1.17	0.00	Jpeg 60	**66.50**	31.48	51.99	**33.75**

w with α:

$$w = \alpha |I_w - I_c|, \tag{5.13}$$

where I_c is the cover image, I_w results from a forward pass of the watermark embedding model and α being the scaling factor to control the watermark strength. To study the impact of α, we perform a sweep in Table 5.4 across 2,000 images from the COCO-2017 validation set.

This table shows that $\alpha = 0.2$ achieves a favorable trade-off: PSNR $\approx$ 42 dB with SSIM $\geq$ 0.989 and low LPIPS, while preserving high cosine similarity of the recovered message even under common distortions. For our experiments, we use this parameter at α=0.2 to target 42 dB of PSNR. We provide an additional robustness study at different PSNR targets in Tab. 5.5.

Watermarked data samples We illustrate image samples watermarked with LatentSeal in Figure 5.9. Depending on the application, the message may be aligned with the cover image, but is not required to. Textual messages can be sourced from human operators aiming to transmit a specific message (aligned or not with the image content), or from an image captioning model.

Table 5.4 – Scaling factor α sweep and corresponding perceptibility metrics with cosine similarity between embedded watermarks and extracted ones. We keep α=0.2 for further experiments.

	Perceptibility metrics			Cosine similarity				
α	PSNR↑	SSIM↑	LPIPS↓	Clean	JPEG-50	JPEG-30	Noise	Crop
0.05	49.7	0.9982	0.0008	0.7581	0.5537	0.4068	0.6152	0.6132
0.10	47.14	0.9967	0.0017	0.9250	0.8193	0.6910	0.8567	0.8658
0.15	44.43	0.9940	0.0031	0.9697	0.9235	0.8507	0.9394	0.9452
0.20	**42.20**	**0.9897**	**0.0055**	**0.9837**	**0.9601**	**0.9200**	**0.9677**	**0.9693**
0.25	40.41	0.9844	0.0088	0.9877	0.9726	0.9465	0.9774	0.9774
0.30	38.92	0.9780	0.0134	0.9886	0.9778	0.9591	0.9815	0.9798
0.35	37.66	0.9713	0.0186	0.9881	0.9790	0.9635	0.9819	0.9794
0.40	36.55	0.9624	0.0263	0.9868	0.9800	0.9675	0.9817	0.9778

5.3.4 Text Autoencoder

Speed Our model achieves significantly higher decoding throughput than OPT-125M (e.g., 121× faster at batch size 128), as shown in Fig. 5.10. This is due to three key factors: it attends to a fixed-length latent vector y instead of a growing prefix, avoiding quadratic attention costs; it does not require key/value caching during generation, reducing memory usage and access overhead; and its decoder operates with smaller hidden dimensions (512 against 768) and fewer parameters per layer, making each step cheaper.

Reconstruction performance In Table 5.6, we exhibit state-of-the-art performances across both COCO-2017 and PixMo-Cap. We highlight a better BLEU4 under all transformations, suggesting better global reconstruction. The Exact Match (EM) rate, a strict binary metric, shows similar results except under JPEG compression at 75% quality. However, the significantly higher BLEU4 score indicates that while our method may miss a few tokens for perfect reconstruction, it is globally able to reconstruct the message more effectively with graceful degradation, as shown more precisely with Fig.5.12 by using the SBERT semantic similarity score.

Ablations Studies

To probe the robustness and limitations of our design, we conduct targeted ablations on three key components of the system: encoder capacity, bottleneck size, and distribution

Table 5.5 – Watermark model evaluation. We report mean cosine similarity (± standard deviation) between original random and extracted watermarks over 2,000 images from the COCO 2017 validation set. Our model is tested against various augmentations at three target PSNR values (41–43 dB).

Transform	**41 dB**	**42 dB**	**43 dB**
Weak			
Identity	0.987 ± .012	0.984 ± .018	0.980 ± .023
Saturation 0.5	0.987 ± .013	0.984 ± .018	0.980 ± .023
HorizontalFlip	0.986 ± .013	0.983 ± .018	0.979 ± .024
Saturation 1.5	0.986 ± .013	0.983 ± .019	0.979 ± .024
Contrast 0.5	0.986 ± .014	0.982 ± .021	0.978 ± .026
Brightness 0.5	0.985 ± .015	0.981 ± .022	0.976 ± .028
Moderate			
GaussianBlur 5	0.983 ± .017	0.979 ± .024	0.974 ± .031
Rotate 90	0.981 ± .017	0.977 ± .023	0.972 ± .030
Hue 0.1	0.976 ± .020	0.972 ± .026	0.964 ± .033
Hue -0.1	0.976 ± .022	0.972 ± .029	0.965 ± .036
Perspective 0.3	0.976 ± .022	0.972 ± .028	0.965 ± .036
JPEG 80	0.972 ± .018	0.965 ± .026	0.955 ± .032
Strong			
Contrast 1.5	0.966 ± .038	0.958 ± .050	0.952 ± .053
Brightness 1.5	0.960 ± .055	0.956 ± .054	0.947 ± .066
JPEG 70	0.961 ± .022	0.949 ± .030	0.935 ± .038
Rotate 10	0.949± .043	0.938± .055	0.924± .070
Perspective 0.5	0.942± .053	0.933± .061	0.920± .073
JPEG 60	0.944± .026	0.928± .035	0.909± .043
JPEG 50	0.924± .030	0.903± .042	0.877± .048
GaussianBlur 17	0.913± .078	0.898± .091	0.881± .104
JPEG 40	0.889± .035	0.862± .046	0.827± .052

shift.

Encoder capacity On Wikitext, we swap the ModernBERT-base encoder for a larger ModernBERT-large (same hyper-parameters). BLEU4 plateaus from 98.0 to 97.5, suggesting that the encoder does not impact the reconstruction accuracy, likely due to the bottleneck dimension.

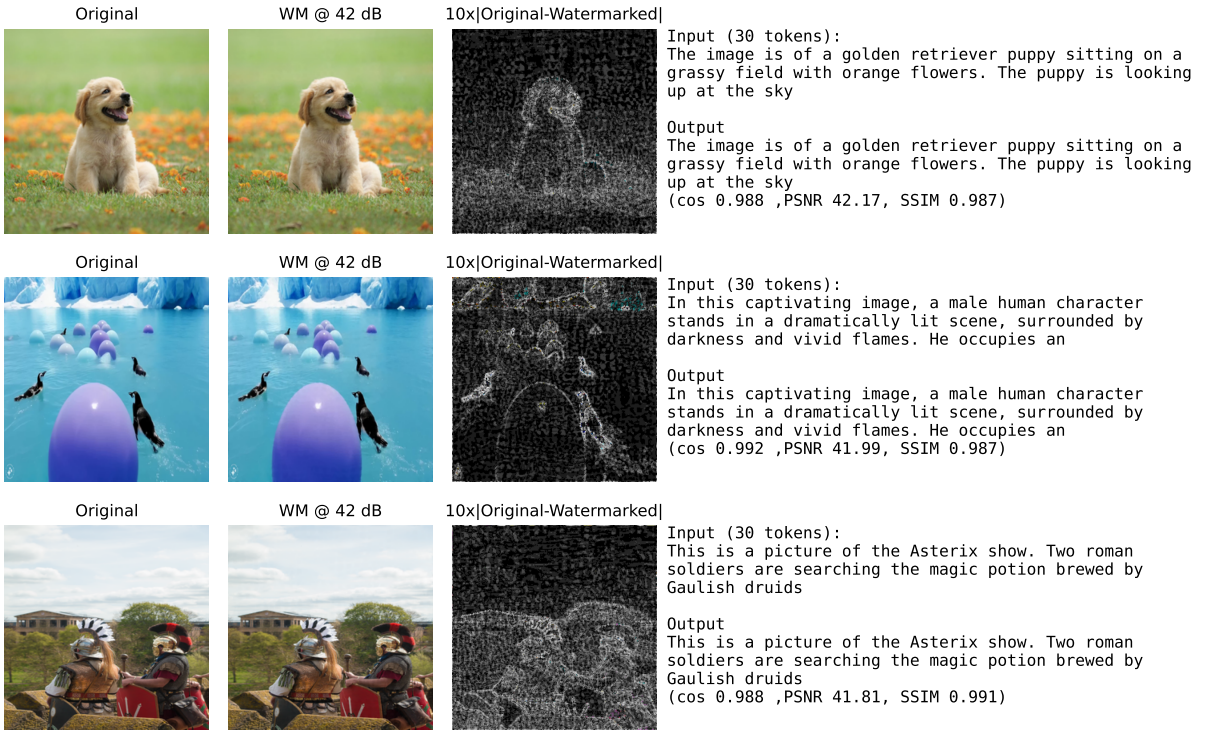


Figure 5.9 – Qualitative samples of LatentSeal. We report the original text and the decoded one, as well as cosine similarity and perceptual metrics. First row input is from Florence-2-base, a captioning model. Second is from PixMo-Cap dataset. And last input is human made, thus out of distribution.

Latent dimensionality We study the effect of the autoencoder latent dimensionality z in a dedicated sweep over $\{128, 256, 384\}$, see Tab. B.3. 256 dimensions provide the best trade-off between reconstruction quality and robustness to the noise injected in the latent space during training. In contrast, models with 128 and 384 dimensions stagnate earlier and consistently underperform the 256-dimensional model, despite exploring multiple architectures and hyperparameters. Since the text autoencoder is the bottleneck of the overall system, we therefore adopt $z = 256$ in all main experiments. The watermarking component itself can in principle operate with other latent sizes.

Bottleneck size Removing the 256-D bottleneck and keeping the 768-D of [CLS] prevents convergence; the model does not converge and struggles at 0.3 BLEU4. Hence a compact latent space is *necessary* for stable training.

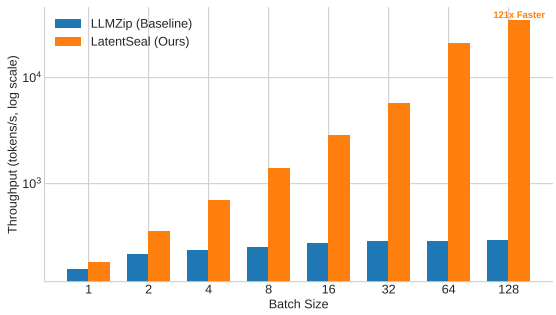
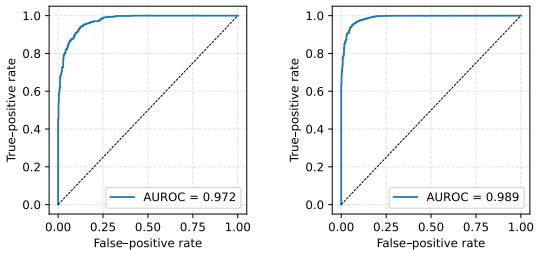


Figure 5.10 – AE decoding speed vs LLMZip; we report the number of tokens per second.



(a) Wikitext (b) PixMo-Cap

Figure 5.11 – ROC of $\ell = -\log_{10}\rho$ on non-catastrophic transforms on Wikitext/PixMo-Cap.

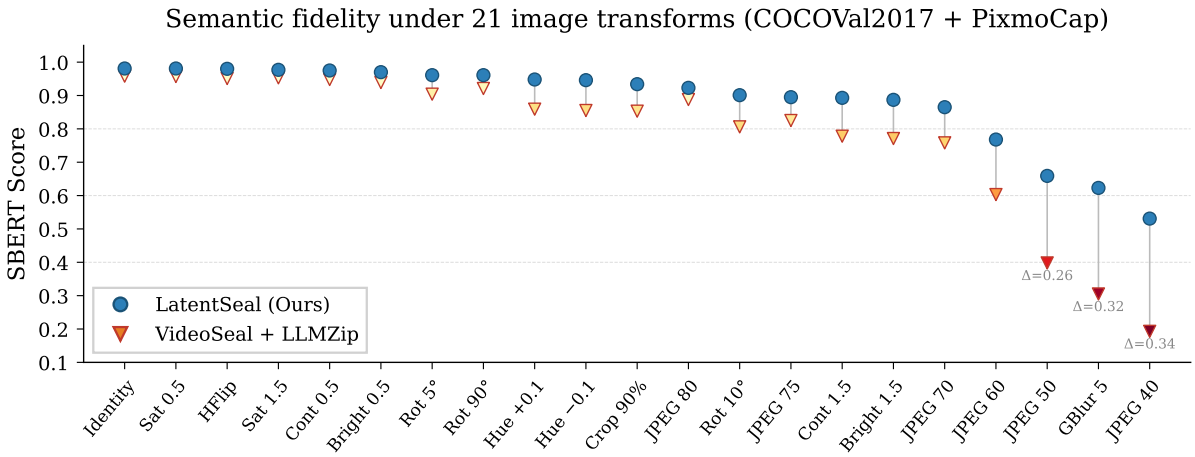


Figure 5.12 – Semantic fidelity under 21 image transforms on COCOVal2017 with Pixmo-Cap sentences. LatentSeal degrades gracefully while VideoSeal+LLMZip collapses under heavy distortion.

Data Novelty On PixMo-Cap, we gather a challenging test set, PixMo-Cap-C, that comprises the top 5 % hardest samples, based on the *4-gram Novelty Score* for a sentence S defined as:

$$\text{Novelty}(S) = \frac{|\{g \in G_4(S) \mid g \notin \mathcal{G}_{\text{train}}\}|}{|G_4(S)|},$$

where $G_4(S)$ is the multiset of 4-grams in S and $\mathcal{G}_{\text{train}}$ is the set of 4-grams in the training corpus (sentences too short to contain a 4-gram are excluded). When evaluating on PixMo-Cap-C, BLEU4 drops modestly (99.1 $\rightarrow$ 94.2), indicating generalization remains strong.

Table 5.6 – VideoSeal vs. LatentSeal (Ours) comparison. PixMo-Cap/COCO-2017 results are at 42dB with 30 tokens. Wikitext results are for messages > 256 bits. Best result for each metric is in **bold**. We exhibit state-of-the-art performances on all settings. BLEU4 results for VideoSeal on WikiText are biased, as the method truncates above 256 bits sentences.

		PixMo-Cap		COCO-2017		Wikitext (> 256 bits)	
Transf.	Method	EM (%)	B4	EM (%)	B4	EM (%)	B4
Identity	VideoSeal+LLMZip	89.2	93.0	96.9	97.4	0	**79.9**
	LatentSeal (Ours)	**92.8**	**97.3**	**97.7**	**99.0**	**51.6**	76.3
JPEG-75	VideoSeal+LLMZip	**71.5**	81.3	85.1	87.4	0	**64.4**
	LatentSeal (Ours)	68.1	**85.6**	**85.2**	**93.9**	**7.9**	39.3
Rot (5°)	VideoSeal+LLMZip	71.1	79.8	93.9	94.7	0	**77.3**
	LatentSeal (Ours)	**87.9**	**95.1**	**94.6**	**97.6**	**40.2**	65.7
Crop (90%)	VideoSeal+LLMZip	71.8	80.1	87.9	89.9	0	**69.5**
	LatentSeal (Ours)	**83.5**	**93.27**	**90.7**	**95.2**	**30.5**	59.1
Noise	VideoSeal+LLMZip	0.0	1.8	0.3	0.0	0.0	0.7
	LatentSeal (Ours)	0.0	**13.5**	**0.8**	**17.3**	0.0	**4.5**

Impact of the number of tokens To showcase the impact of the number of tokens, we evaluate the viability of our system to encode/decode up to 40 tokens, using the same robustness to a noise of $\sigma = 0.01$. Results are given in Table 5.7, and indicate that performance gracefully degrades as sentence size increase.

Table 5.7 – BLEU4 scores at different sentences size.

	PixMo-Cap	**WikiText**
30 tokens	99.1	98.7
40 tokens	95.0	89.6

Random token ids We emphasize that our text autoencoder is trained on natural-language sentences and is therefore designed to represent *semantic* text, not arbitrary identifiers. To probe its behaviour on out-of-distribution inputs, we conduct in Tab. 5.8 a simple experiment where we sample 1000 random "sentences" by drawing 30 token IDs uniformly from the ModernBERT vocabulary, encode them into the 256-dimensional latent space, and decode them back to text. We then compute exact string reconstruction

Input (30-token random IDs)	Decoded output
`auseunciation Campbell properties talked competency technique PVdecor leur preferred attachment boundary protocols 701Lin`	`083ovich064pay lawsuit hypers phenomenalshots Resolution L twelve let Tigers minor sadly fuss strength uphold applications faster Black Nova`

Table 5.8 – Qualitative example: a sequence of random token IDs passed through the autoencoder. The decoded output bears no relation to the input, showing the model does not reconstruct arbitrary sequences.

and BLEU-4 between the input and decoded sequences. Over this random-ID set, the exact-match rate is 0 % and the average BLEU-4 score is ≈ 0.0029, indicating that none of the decoded outputs reproduces the input token sequence. Qualitatively, the decoder instead produces linguistically plausible fragments that lie on the natural-language manifold but are unrelated to the original random IDs. This confirms that the autoencoder behaves as a semantic compressor rather than as a mechanism for reliably encoding and recovering arbitrary identifier strings.

5.3.5 End-to-end comparison

Robustness to Image Editing models To evaluate if Videoseal and LatentSeal are robust to attack from image editing models, we modify images with HiDream (Cai et al. 2025) and IP2P (K. Zhang et al. 2023c) and report latent vector similarity between original and decoded messages using SBERT and BLEU4 in Fig. 5.13. We found dramatic degradation for both methods, which is not surprising given recent work on Image Regeneration (Y. Liu et al. 2025) that uncovered this vulnerability. Nevertheless, we compare favorably with Videoseal to that regard and, thanks to our confidence metric, are able to flag at inference time if the message is likely tampered with or not. Also, PSNR between original and edited images have a median of 14.9 for HiDream and 19.7 for IP2P. These perturbations thus dramatically disrupt image quality, demonstrating that any attempt to remove the watermark comes at the cost of degrading the image's quality.

We detail the influence on perceptibility, through the PSNR metric, of each image editing model in Table 5.9. Samples can be found in Fig. 5.14.

Graceful degradation is a crucial feature enabled by working in a continuous latent space, as it ensures that noise introduced via a channel does not result in a hard failure of message extraction. Instead of a fixed, arbitrary number of message bits, which would

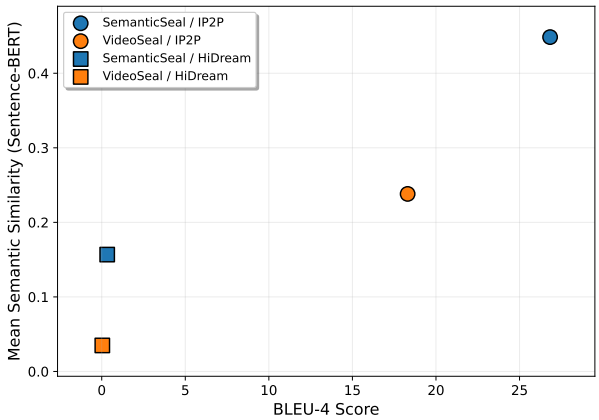


Figure 5.13 – BLEU4 and SBERT similarity between original and retrieved watermarks after image editing on cover. Our method obtains SOTA results in terms of robustness.

Table 5.9 – PSNR statistics for HiDream and ip2p image-editing models.

Statistic	HiDream (dB)	ip2p (dB)
Mean	15.6372	19.8388
1st quartile	10.2666	13.7499
Median	14.8781	19.7041
3rd quartile	20.0194	25.3874

require a perfect, error-free channel for complete reconstruction, the autoencoder learns to map sentences to a continuous 256-D space that is robust to noise. This design means that even if the channel is corrupted, the system can still reconstruct a sentence that is semantically close to the original, provided the input is in-domain. Tab. 5.10 demonstrates this by showing how the proposed method, LatentSeal, consistently achieves a much higher SBERT score (semantic similarity) compared to the baseline, VideoSeal+LLMZip, resulting in a substantial improvement, especially under heavy attacks like GaussianBlur and Jpeg compression.

Data Contamination We discuss the results in Figure 5.6. We compare to VideoSeal by leveraging LLMZip with opt-125m to stay in the same weights size range as our text autoencoder. Note that opt-125m is trained on samples from Wikipedia, which might entails test-set leakage for Wikitext. We still think Wikitext is valuable, but advise the reader to take the VideoSeal+LLMZip results with a grain of salt. PixMo-Cap is more recent than OPT models, and is a dataset with new data, not a collection of previous

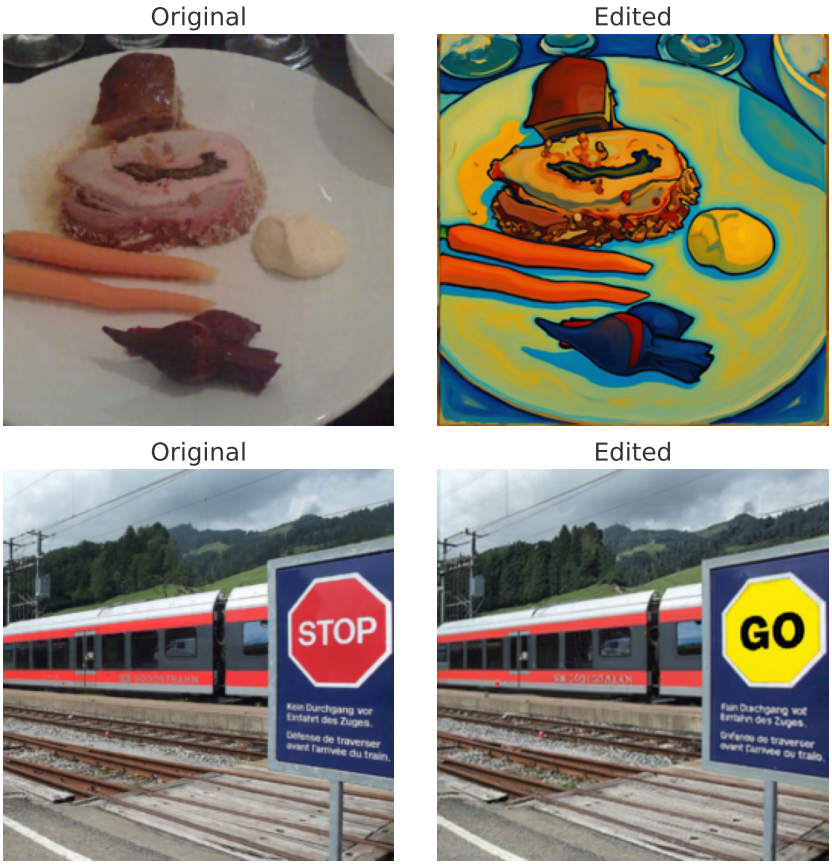


Figure 5.14 – Samples of images edited with HiDream

ones, thus we can discard suspicion of test data contamination.

Dataset difficulty In order to better understand our findings, we analyze datasets composition. We expect Wikitext to be more difficult, as it is sourced from Wikipedia in which each page is full of named entities with few to no occurrences besides the actual article on it. Figure 5.15 confirms this assumption and showcases the low overlap (27%) between 4-grams from train and test sets. This means our system has to generalize in order to achieve good results on test set. Conversely, we perform the same study on PixMo-Cap and find that our random split is way worse in terms of overlap, meaning that good performances may be due to this overlap between train and test sets. To further probe our system abilities, we use the novelty score introduced in Section 5.3.4 to curate the 5% of samples with the lowest count of common n-grams between train and test sets. As stated before, we observe a drop of 5 of BLEU4, which is sound and ensures

Table 5.10 – SBERT similarity scores demonstrating graceful degradation against 21 transforms on COCOVal2017 images with PixmoCap sentences. We stay closer in this semantic space, even under high JPEG compression settings.

Transform	LatentSeal	VS+LLMZip	Transform	LatentSeal	VS+LLMZip
Identity	0.981	0.958	Bright 0.5	0.970	0.939
HFlip	0.980	0.951	Bright 1.5	0.887	0.771
Rot 5	0.961	0.904	Hue 0.1	0.948	0.859
Rot 10	0.901	0.806	Hue -0.1	0.946	0.855
Rot 90	0.961	0.921	Jpeg 80	0.923	0.888
Crop 90	0.934	0.853	Jpeg 75	0.895	0.825
GBlur 5	**0.623**	**0.305**	Jpeg 70	0.865	0.758
Sat 0.5	0.981	0.957	Jpeg 60	0.768	0.603
Sat 1.5	0.977	0.953	Jpeg 50	**0.659**	**0.398**
Cont 0.5	0.975	0.948	Jpeg 40	**0.531**	**0.193**
Cont 1.5	0.893	0.778			

that our system does work.

Relation between number of tokens and bit payload In Figure 5.16, we estimate the bit length associated with a number of tokens using LLMZip on all ours dataset splits. This enables us to target 30 tokens as a fair ground for comparison with the 256 bits capacity of VideoSeal.

To better emphasizes the current limitations of a 256 bits system, Fig. 5.17 depicts the rate of samples going above this threshold for a N token long message in Wikitext. We estimate bit length using LLMZip over the 4,000 sentences. So for 30 tokens, about 16% of sentences are above the 256 bits threshold and will be partially transmitted due to limited capacity.

Note that the relation between the latent space dimension and bit payload is not straightforward, and cannot be easily derived. In our case, this mainly relies on the capacity of the text encoder to compress information from a given number of tokens in a given latent space dimension.

Theoretical analysis

We compare standard watermarking methods on the watermark detection task. While they work on different domains and trained for different tasks, we use the p-value interface

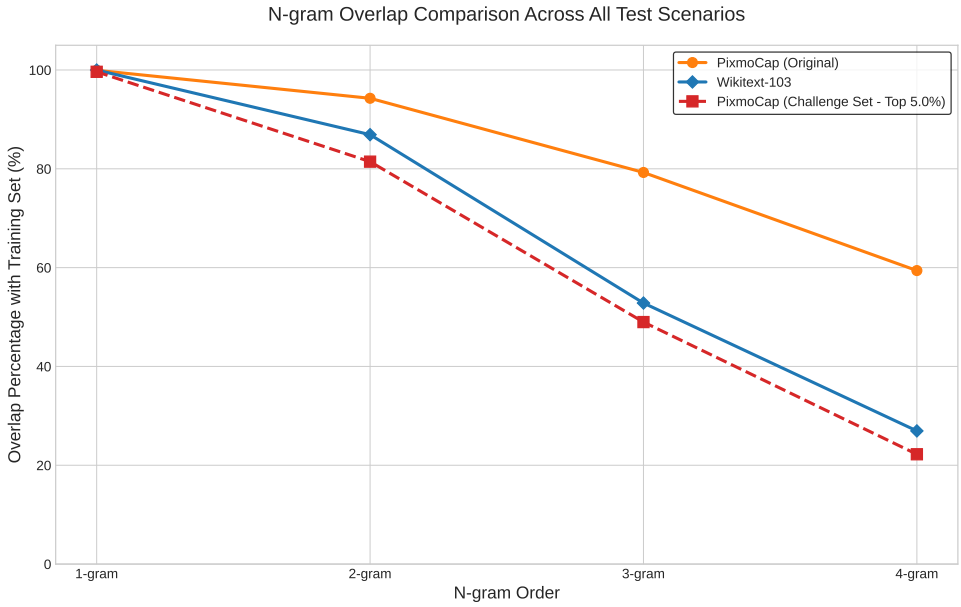


Figure 5.15 – We study train/test ngram overlap as an heuristic to determine how much the model is able to generalize on new data. We show that PixMo-Cap test set is quite repetitive and thus design a metric to optimize the hardness of the test set, keeping only the top hardest 5%.

as a common ground for comparison. This provides an outline of each method respective statistical guarantees.

We frame it as a statistical detection test. We consider the null hypothesis H_0 that each bit of the output binary message $\hat{m}$ is independent and distributed as a Bernoulli variable with probability of success 0.5, and the alternative hypothesis H_1 which is that $\hat{m} = m$.

Given an observed bit accuracy $bit_{accuracy}(m, \hat{m})$, the p-value is the probability of observing a bit accuracy at least as extreme as the one obtained under the null hypothesis. It is given by the cumulative distribution function of the binomial distribution:

$$p\text{-value}(m, \hat{m}) = \sum_{k \geq n_{\text{bits}} p} \binom{n_{\text{bits}}}{k} (1/2)^{n_{\text{bits}}} = I_{1/2}(n_{\text{bits}} p, n_{\text{bits}}(1-p) + 1), \tag{5.14}$$

where p is the bit accuracy between m and $\hat{m}$ and $I_x(a, b)$ the regularized incomplete beta function.

The above equation, borrowed from Fernandez, Elsahar, et al. 2024, enables to compare multibit watermarking methods with different capacity (using the n_{bits} factor).

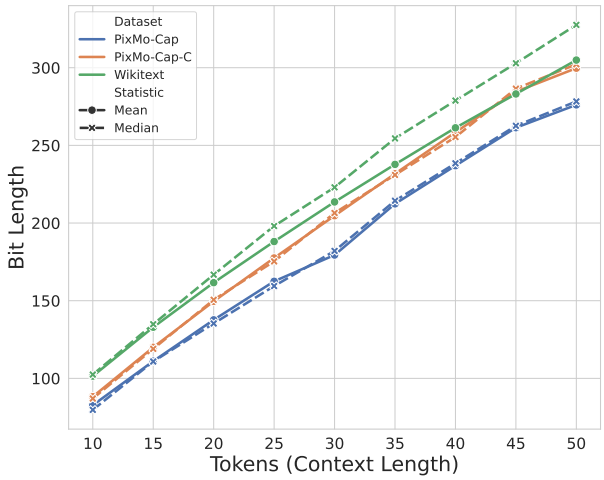


Figure 5.16 – Relationship between the mean number of tokens and the compressed bit length using LLMZip with OPT-125M, computed on Wikitext, PixmoCap and PixmoCap-C (for the harder split).

Note that all those methods are restricted by their operating space. At most, one can hide 256 bits and thus, assuming bit_{acc} at 1.0, reach a p-value $= 1e - 77$ derivated from 2^{-256}.

In order to compare with LatentSeal which operates on $\mathbb{S}^{255}$, we use the following context; under the null hypothesis that y is a random unit vector sampled from $\mathbb{S}^{255}$, the one-sided p-value ρ is given by:

$$\rho = \Pr(\cos(y, \hat{y}) \geq c) = \mathrm{I}_{1-c^2}\left(\frac{255}{2}, \frac{1}{2}\right), \tag{5.15}$$

This framing allows us to compute an **unbounded** p-value, albeit limited by the numerical precision of the computation as $c \to 1.0$ when y and $\hat{y}$ get closer.

We report in Tab. 5.11 p-values from standard methods and LatentSeal using 1000 images from COCOVal2017 T.-Y. Lin, Maire, Belongie, Bourdev, et al. 2015. Using vectors from the autoencoder latent space does not degrade performances, this is expected as the core property of lying on the unit hypersphere $\mathbb{S}^{255}$ is respected. This would be different if we had trained in an end-to-end fashion, we would expect better performance for AE latent vectors, at the cost of degrading the "random" setting and performing a time-consuming training.

In summary, LatentSeal not only allows for the transmission of semantic messages but does so while providing stronger statistical guarantees than the best available discrete

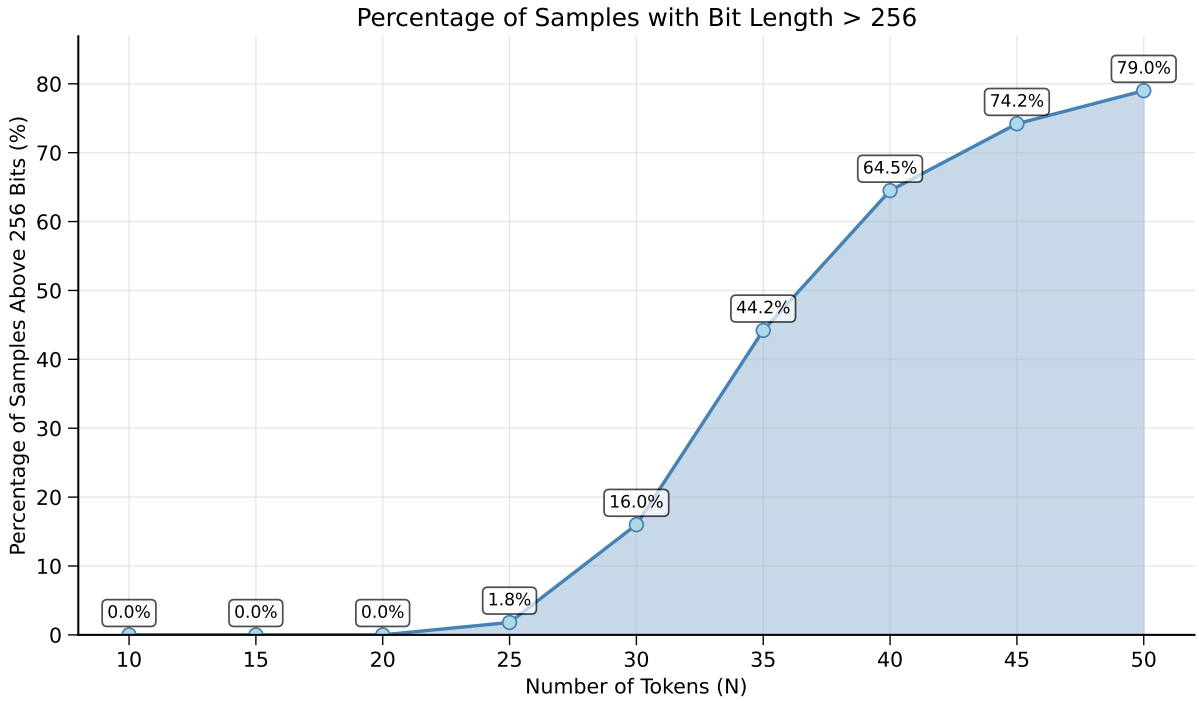


Figure 5.17 – On Wikitext, we compute the percentage of samples above the maximum capacity of Videoseal for better emphasis of existing limitations.

multibit baselines.

Confidence score The confidence score ρ measures how self-consistent a latent is after a decode-encode round-trip. For convenience, we consider $\ell = -\log_{10} \rho$. Larger ℓ means *higher* confidence, and we can set a threshold ℓ_{th} such that only messages with $\ell > \ell_{th}$ should be trusted. A threshold $\ell_{th} = 109$ therefore rejects captions whose p-value exceeds 10^{-109}. We provide ROC curves in Fig. 5.11 outlining the ability of this confidence metric to filter out damaged watermarks. As Table 5.12 shows, setting $\ell_{th} = 154.94$ keeps 77.9 % of perfect reconstructions while discarding 99 % of erroneous ones for PixMo-Cap. For safety-critical deployment a stricter $\ell_{\min} = 170.16$ rejects every untrustworthy watermarks yet retains 63.4% of reliable messages.

Table 5.11 – Comparison of p-values between standard methods in watermarking. All multibits methods are bounded at $p - values = 1e - 77$ due to their operating space which allows for at most 256 bits.

Method	HiDDeN	MBRS	TrustMark	Stable Sig.	VideoSeal	LatentSeal (Ours)
Identity	14.2	70.6	29.9	13.6	76.4	**191.4**
Valuemetric	10.8	59.8	27.4	12.5	73.3	**161.1**
Geometric	5.5	3.3	8.5	9.8	74.8	**164.9**
Compression	14.2	69.9	29.7	7.5	72.5	**132.2**
Combined	2.6	0.4	0.8	9.3	60.9	**91.0**

Table 5.12 – Operating points for Fig. 5.11. A received message is *untrustworthy* (negative) if $\ell < \ell_{th}$.

Dataset	**Target FPR**	**Threshold ℓ_{th}**	**TPR**
PixMo-Cap	10^{-4}	170.16	63.4 %
	10^{-2}	154.94	77.9 %
	10^{-1}	120.66	97.2 %
Wikitext	10^{-4}	192.69	44.8 %
	10^{-2}	176.84	61.9 %
	10^{-1}	139.75	90.7 %

5.4 Conclusion

In this chapter, we present a novel way to assert the integrity of an image through the targeted use of watermarking as a covert communication channel. Moreover, we provide a definition of the image semantics to the recipient of the message through its caption.

From bits to semantics Both contributions reframe watermarking from a bit-payload problem to a semantic communication problem. SWIFT does this through compression (squeezing a caption into up to 72 bits via LLMZip + TCCSK); LatentSeal does it through learned representation (mapping sentences to a textual space in $\mathbb{S}^{255}$ directly). The latter generalizes to other modalities: where SWIFT is bound to the linguistic domain LLMZip was trained on, LatentSeal embedder and extractor handle any features in $\mathbb{S}^{255}$, opening the door to applications beyond captions such as CLIP embeddings, audio descriptors, multimodal tokens.

Trade-offs LatentSeal supersedes SWIFT in capacity, robustness, and speed, but the improvement comes at a non-trivial cost: it requires training a dedicated text autoencoder, whereas SWIFT plugs directly into a pretrained LLM with a lighter finetuning.

Future directions We leave to future work the task of designing a system taking advantage of our pipeline output: Alice could be considered as the source of the original media while Bob would be a moderation system on a social media platform. We see this as the start of a broader research direction: watermarking as semantic communication. Extending latent space watermarking to video, audio, or cross-modal provenance could enable practical authentication pipelines for the AI era, where watermarks no longer just exist, but *explain*.

6

Detecting Watermark Removal

Contents

6.1 Introduction 101
6.2 Method 102
6.3 Experiments & Results 106
6.3.1 Implementation Details 106
6.3.2 Removal attacks effectiveness and perceptual quality 107
6.3.3 Detection Performance 111
6.3.4 Extension to Image Editing 114
6.4 Conclusion 120
6.4.1 Limitations 120
6.4.2 Discussion 121

This chapter is based on: *Evennou and Kijak, The Forensic Cost ofWatermark Removal*, 2026 published at IH&MMSec 2026 and its extension *Evennou and Kijak, The Forensic Cost of Watermark Removal: From Dedicated Attacks to Image Editing*, under review (TIFS).

6.1 Introduction

Watermarking is under attack. As hidden signals become the default mechanism for tracing image provenance, mandated by regulators (European Parliament and Council of the European Union 2024) and adopted by platforms (Castro and Yalniz 2025), adversaries have strong incentives to strip them. A successful removal enables intellectual-property laundering, bypasses model-collapse safeguards for GenAI, and severs chains of trust in information networks.

To understand what a successful attack requires, consider the following threat model. *Alice* produces an image and publishes it online. *Bob*, a platform verifier, runs two sequential checks: first for Alice's watermark, then, if the watermark is absent, for traces of removal. *Eve* copies the image and attempts removal to re-post it as her own. Bob checks Eve's image, detects no watermark but finds removal artifacts, and triggers sanctions. A successful attack must therefore fool *both* detectors: it must remove the watermark *and* leave no forensic trace. We assume Bob targets high-profile users to limit top-down propagation of stolen media.

Yet current removal methods address only half of this requirement. Whether adversarial or generative, they frame the task as noise purification: the watermark is treated as additive noise, and success is measured solely by the trade-off between Attack Success Rate (ASR) and distortion (PSNR). If the watermark is unreadable and the distortion low, the attack is deemed successful. This framing ignores a critical question: *does the post-removal image still look like a natural image to a forensic detector?*

Our answer is no. We show that the act of removal leaves distinct statistical trails, high-frequency adversarial noise, characteristic diffusion artifacts or both, that push the image off the manifold of natural images. These traces are readily separable by a learned classifier trained on genuine non-watermarked images and post-removal images, which flags attacks with high TPR at low FPR across a wide range of state-of-the-art removal methods. To our knowledge, no existing attack accounts for this forensic leakage.

We formalize this observation into a new evaluation axis: **Watermark Removal Detection (WRD)**. Any complete assessment of watermark removal must jointly consider three criteria: (i) attack success rate, (ii) perceptual quality, and (iii) forensic stealthiness. Under this extended framework, we demonstrate that all current removal methods fail criterion (iii), revealing stealth as a fatal gap in existing attack strategies.

Our contributions are:

- We introduce Watermark Removal Detection as a novel evaluation constraint, formalizing forensic stealthiness as a necessary requirement for watermark removal.
- We show that a lightweight classifier reliably detects post-removal images across multiple removal families (adversarial, diffusion-based), and benchmark state-of-the-art methods under WRD, finding that none achieves forensic stealth.
- We extend the WRD evaluation to modern image editing models, finding that editors either fail to remove the watermark or introduce distortions strong enough to be forensically detected, establishing a fundamental dilemma for editing-based attacks.

This work is an extended version of our preliminary work accepted at the 2026 ACM Workshop on Information Hiding and Multimedia Security (IH&MMSec) (Evennou and Kijak 2026). We introduce the compound success rate (Eq. 2), a single metric capturing the joint requirement of watermark removal and forensic stealth. We add a cross-watermark generalization study (Sec. IV-D) validating that the forensic detector learns attack artifacts rather than watermark-specific signals, and a comparison against AEROBLADE (Ricker, Lukovnikov, and A. Fischer 2024) (Sec. 6.3.3) showing that off-the-shelf GenAI detectors do not transfer to the watermark removal setting. The main addition is a full evaluation of image editing as an attack vector (Sec. IV-I): we benchmark eight editors spanning 2022–2026, uncover a temporal trend where watermark removal as a byproduct peaks in 2024 and declines afterward, and explain it through a spectral shift from low- to mid/high-frequency edits. Editors that do remove the watermark introduce distortions strong enough to be forensically detected, placing image editing under the same dilemma as dedicated removal attacks.

6.2 Method

Given a watermarked image x_w, our goal is to detect a removal attack $\mathcal{A}$ such that if the attacker wants to be sure to succeed in removing the watermark, they will eventually expose himself to the detector detailed in Section 6.2.

Detector The forensic detector is aimed at detecting attacks, as a proxy of watermark removal. This choice stems from the property of well-designed watermarking models to hide the watermark signal w from detection without the secret key (I. J. Cox et al. 2008). A forensic detector able to detect the removal of a specific watermark is a security red

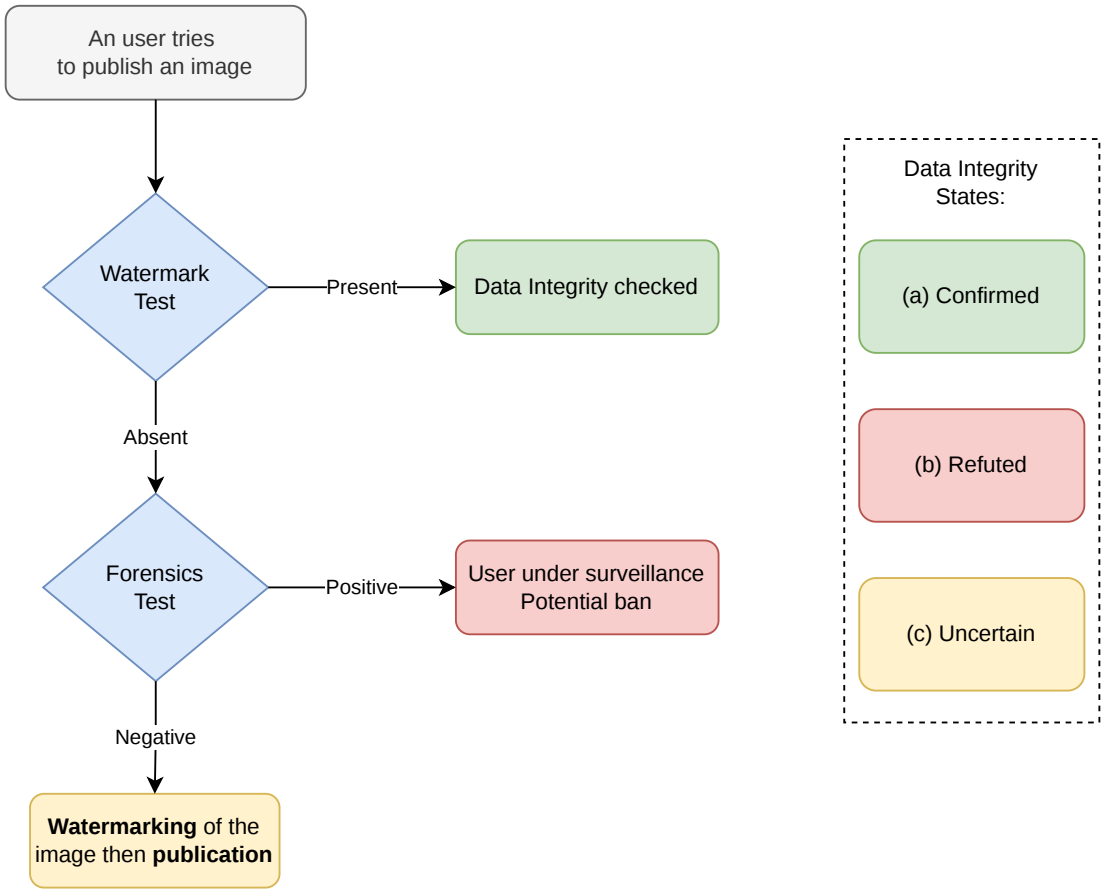


Figure 6.1 – Workflow of our method. We use a social network perspective, in which an user tries to publish an image. The platform runs two sequential checks; the watermark test computes whether a watermark was detected and if so use the extracted id to establish data provenance and enable the cover verification. The negative result leads to the forensics test which is responsible for detecting watermark removal. A positive result prompts further scrutiny from the social network, potentially leading to a ban of the user. A negative one entails an uncertain status of the image, which is then watermarked and finally published.

flag, as it means that the watermarking model leaves a learnable signal, thus exposing itself to forging attacks as described in WMForger (Souček et al. 2025).

We define our forensic detector D_f as a feature extraction network with an MLP that outputs a sigmoid-based score. A score below 0.5 indicates a pristine image; above 0.5 indicates an attacked one. We train it by optimizing the Binary Cross-Entropy objective function. We use ConvNextTiny-v2 (Woo et al. 2023) as our backbone for the detector. We aim for a generalist detector since the defender has no prior knowledge of the attacker's method. A critical design choice is separating removal detection from AI-generation detection. GenAI detectors exploit artifacts specific to a generative architecture which change as models evolve. Our detector must instead learn signatures of *attacks*: the statistical traces left when part of an original image is reconstructed or edited. To enforce this distinction, we include WMForger in the training set alongside diffusion-based attacks. Because WMForger operates through adversarial perturbations rather than generative reconstruction, its presence forces the detector to capture broader removal artifacts. We provide more details on the training procedure in Section 6.3.

Evaluation metrics We evaluate watermark removal along three complementary axes: attack success rate, forensic detectability and perceptual quality.

For attack effectiveness, we measure watermark detectability and whether it leaves detectable traces. We evaluate the watermark survival through the lens of the corresponding ρ-value. We consider the null hypothesis H_0 that each bit of the output binary message $\hat{m}$ is independent and distributed as a Bernoulli variable with probability of success 0.5, and the alternative hypothesis H_1 which is that $\hat{m} = m$.

We use the unified formulation from VideoSeal (Fernandez, Elsahar, et al. 2024), enabling comparison of ρ-values across different payload sizes. Given a message m and its decoded estimate $\hat{m}$, let $b = \sum_{i=1}^{n_{\text{bits}}} \mathbf{1}[\hat{m}_i = m_i]$ be the number of correctly decoded bits. Under the null hypothesis H_0 (random guessing), $B \sim \text{Binomial}(n_{\text{bits}}, 1/2)$. The ρ-value is the probability of observing at least b correct bits under H_0:

$$\begin{aligned} \rho\text{-value}(m, \hat{m}) = P(B \geq b \mid H_0) &= \sum_{k=b}^{n_{\text{bits}}} \binom{n_{\text{bits}}}{k} \left(\frac{1}{2}\right)^{n_{\text{bits}}} \\ &= I_{1/2}\Big(b,\, n_{\text{bits}} - b + 1\Big), \end{aligned} \tag{6.1}$$

where $I_x(a, b)$ denotes the regularized incomplete beta function. Consequently, the Attack Success Rate (ASR) is the fraction of images whose ρ-value exceeds 10^{-6} after attack,

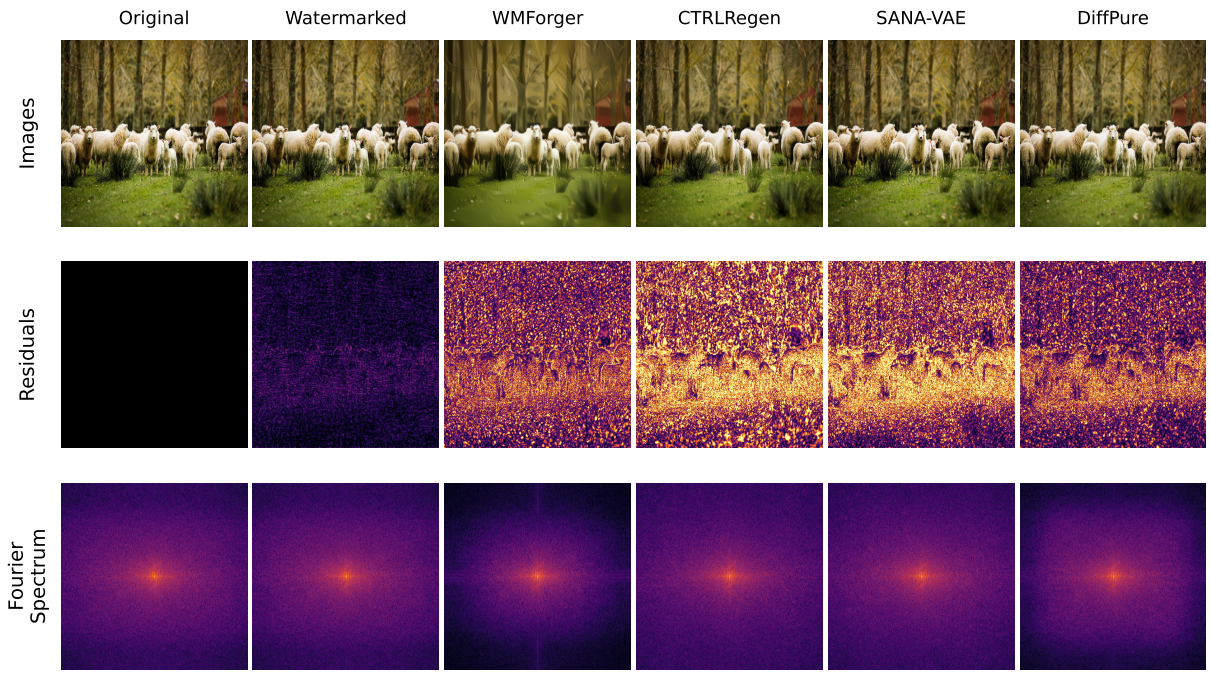


Figure 6.2 – Watermark removal attacks samples, with residuals and Fourier spectrum. WMForger and Diffpure have the smallest residual footprint by targeting the high frequencies, which is better depicted in Fourier Spectrum figures, while CtrlRegen and SANA-VAE are spreading on the whole image their removal effort.

following the standard detection threshold in the watermarking literature (Fernandez, Couairon, et al. 2023a). To assess forensic detectability, we report the area under the ROC Curve (AUC) and the True Positive Rate at a fixed False Positive Rate (TPR@FPR) of our detector D_f. A high forensic detection performance signals that the attack leaves statistical artifacts. We consider false positives hamper trustworthiness of detectors and further use low FPR.

For perceptual quality, we argue that commonly used pixel-level distortion metrics such as PSNR, while informative, are insufficient to fully characterize the imperceptibility of a removal attack. An attack may introduce minimal pixel-wise distortion yet shift its outputs away from the natural image manifold, producing images that are technically close to the original but perceptually unnatural, see Figure 6.2. To capture this distinction, we complement PSNR with LPIPS (R. Zhang et al. 2018), which measures perceptual similarity at the image level, and with the Fréchet Inception Distance (FID) (Heusel et al. 2017), which quantifies the distributional distance between attacked images and their original, non-attacked counterparts. Together, these metrics provide a more complete

picture of attack quality than distortion metrics alone.

We further focus on the attack effectiveness. Hence to capture the joint threat in a single metric, we define the *compound success rate*: the fraction of attacks that both erase the watermark *and* evade forensic detection. For a set of N attacked images,

$$\underbrace{\text{Compound}}_{\text{practical threat}} = \underbrace{\text{ASR}}_{\text{watermark erased}} \times \underbrace{(1 - \text{TPR@FPR})}_{\text{detection evaded}} \tag{6.2}$$

where ASR is the fraction of images whose ρ-value exceeds 10^{-6} after attack, and TPR@FPR is the true positive rate of the forensic detector D_f at a calibrated false positive rate. A compound rate near zero means the attack is neutralized: either the watermark survives, or the removal is detected. Because the forensics detector already relies on distortion introduced by removal attacks, we do not include perceptibility metrics in this aggregation.

Editor Selection & Taxonomy To study modern image watermarking robustness to image editing models, we gather a wide range of Image Editing models: InstructPix2Pix, MagicBrush, QwenEdit, QwenEdit2511, FireRed 1.1 and FLUX 2 [klein] (Brooks, Holynski, and Alexei A. Efros 2023; Labs 2025; S. I. Team et al. 2026; C. Wu et al. 2025; K. Zhang et al. 2023a). This selection comprehensively illustrates the evolution of image editing and current trends in watermark robustness. We divide the study between local and global edits. A local edit aims at only editing a specific part of the image while global involves all pixels in the scene.

6.3 Experiments & Results

6.3.1 Implementation Details

Dataset We sample 5,000 images from COCO (T. Lin et al. 2014) as cover images. Each image is watermarked with two schemes, VideoSeal (Fernandez, Elsahar, et al. 2024) and TrustMark (Bui, Shruti Agarwal, and Collomosse 2025), at a target PSNR of 40 dB, yielding 10,000 watermarked images. We then apply four removal attacks to each watermarked image using hyperparameters highlighted in respective papers: DiffPure (Nie et al. 2022) ($t = 0.1$), a SANA-VAE reconstruction (Xie et al. 2025), CtrlRegen (Y. Liu et al. 2024) ($\sigma = 0.1$), and WMForger (Souček et al. 2025) (500 steps, lr = 0.05). Including the pristine originals and watermarked images as negatives, the full dataset

comprises approximately 50k images, split 70/10/20 by original image ID to prevent leakage between train, validation, and test sets. The dataset is composed of one quarter pristine images (original and watermarked) and three quarters attacked images.

Watermarking methods We use the public implementations of VideoSeal, 1.0 version at 256 bits, and TrustMark model. Those two models are heavily used and VideoSeal is considered as state-of-the-art. Both are post-hoc multi-bit watermarking schemes. We watermark at 256×256 resolution. The choice of two architecturally distinct schemes ensures our detector results don't rely on one watermarker, but rather on the attacks performed. We do not use adversarially trained watermark models such as Vine (S. Lu et al. 2025) or EditGuard (X. Zhang et al. 2024) as they trade robustness to editing against capacity or perceptibility. We chose to sidestep this road and design a defense in an orthogonal fashion with respect to the watermark training procedure.

Detector training We finetune a ConvNeXtV2-Tiny (Woo et al. 2023) pretrained on ImageNet-22k (Deng et al. 2009), replacing the classification head with a two-layer MLP (hidden dim 128, GELU activation, dropout 0.33). All images are resized to 224×224. We train for 8 epochs on a single NVIDIA L40S with batch size 256, using AdamW with weight decay 2.5×10^{-3} and a cosine schedule with one-epoch linear warmup. The classifier head uses learning rate 4.8×10^{-4} while backbone parameters are finetuned at $0.32\times$ that rate. To improve robustness against editorial post-processing, we apply random augmentations during training: JPEG compression (quality 60–100), Gaussian blur (radius 0.5–2.0), downscale-upscale (scale 0.5–1.2), and Gaussian noise (σ 0.01–0.08). Each image has a 20% probability of receiving one randomly selected augmentation.

6.3.2 Removal attacks effectiveness and perceptual quality

Perceptibility metrics and robustness results are reported in Table 6.1 and Table 6.2, respectively. Across all metrics, removal attacks cause substantially more perturbation than the initial watermarking process. Without specific knowledge of the watermark's spatial distribution or strength, attacks must apply brute-force, global alterations, ultimately resulting in poor perceptibility scores.

Hence, robustness and perceptibility results reveal a clear tradeoff between attack success and image quality. Looking at PSNR alone, WMForger and DiffPure appear as the most performative methods. However, LPIPS and FID tell a different story: WMForger

Table 6.1 – Image quality metrics between original and variant images. PSNR (↑), LPIPS (↓), and FID (↓).

Watermarker	Image Version	PSNR ↑	LPIPS ↓	FID ↓
None	Watermarked	40.1 ± 1.9	0.004 ± 0.020	1.13
TrustMark	DiffPure (t=0.1)	29.6 ± 3.4	0.068 ± 0.035	4.01
	SANA-VAE	24.8 ± 4.2	0.087 ± 0.040	3.82
	WMForger	29.5 ± 2.7	0.216 ± 0.079	30.42
	CtrlRegen (σ=0.1)	24.1 ± 3.4	0.115 ± 0.043	5.34
VideoSeal	DiffPure (t=0.1)	29.6 ± 3.4	0.067 ± 0.034	3.76
	SANA-VAE	24.8 ± 4.1	0.089 ± 0.038	3.84
	WMForger	29.5 ± 2.7	0.215 ± 0.079	30.14
	CtrlRegen (σ=0.1)	24.1 ± 3.4	0.115 ± 0.042	5.23

produces images that deviate significantly from the natural image manifold, with an FID of 30, confirming that PSNR on its own is insufficient as it only accounts for perturbation amplitude and not distributional proximity to natural images. When factoring in robustness, we observe a negative correlation between ASR and perceptual quality: DiffPure based on Flux.1 [dev] achieves the best FID and LPIPS but the worst ASR on VideoSeal; similarly, SANA-VAE obtains the best FID and second-best LPIPS while underperforming on robustness. Conversely, the most aggressive watermark removal techniques degrade image quality the most, highlighting the trade-off between removal and imperceptibility, more qualitative samples in Fig. 6.2.

Table 6.2 – Watermark survival and forensic detectability. ASR per attack, TPR at FPR=10^{-3}, and compound success rate ASR × (1−TPR@10^{-3}). Bold marks highest compound per watermarker.

WM	Attack	ASR	TPR@10^{-3}	Compound
TM	DiffPure	0.997	0.785	**0.214**
	SANA-VAE	0.797	0.944	0.045
	WMForger	1.000	0.998	0.002
	CtrlRegen	1.000	0.936	0.064
VS	DiffPure	0.569	0.810	**0.108**
	SANA-VAE	1.000	0.927	0.073
	WMForger	1.000	1.000	0.000
	CtrlRegen	1.000	0.915	0.085

Table 6.3 – Watermark Removal Detection. Detection rates at operating points FPR=10^{-2} and 10^{-3} (calibrated on COCO). ± denotes bootstrap standard deviation over test set resamples.

		TPR@FPR	
Watermarker	Method	10^{-2}	10^{-3}
TrustMark	DiffPure (t=0.1) (Labs 2024)	0.974±0.014	0.785±0.095
	VAE (SANA) (Xie et al. 2025)	0.999±0.002	0.944±0.042
	WMForger (Souček et al. 2025)	1.000±0.001	0.998±0.001
	CtrlRegen (Y. Liu et al. 2024)	0.996±0.004	0.936±0.042
VideoSeal	DiffPure	0.987±0.008	0.810±0.096
	SANA VAE	0.998±0.002	0.927±0.053
	WMForger	1.000±0.000	1.000±0.001
	CtrlRegen	0.997±0.003	0.915±0.050
All	All attacks	0.994±0.003	0.920±0.037

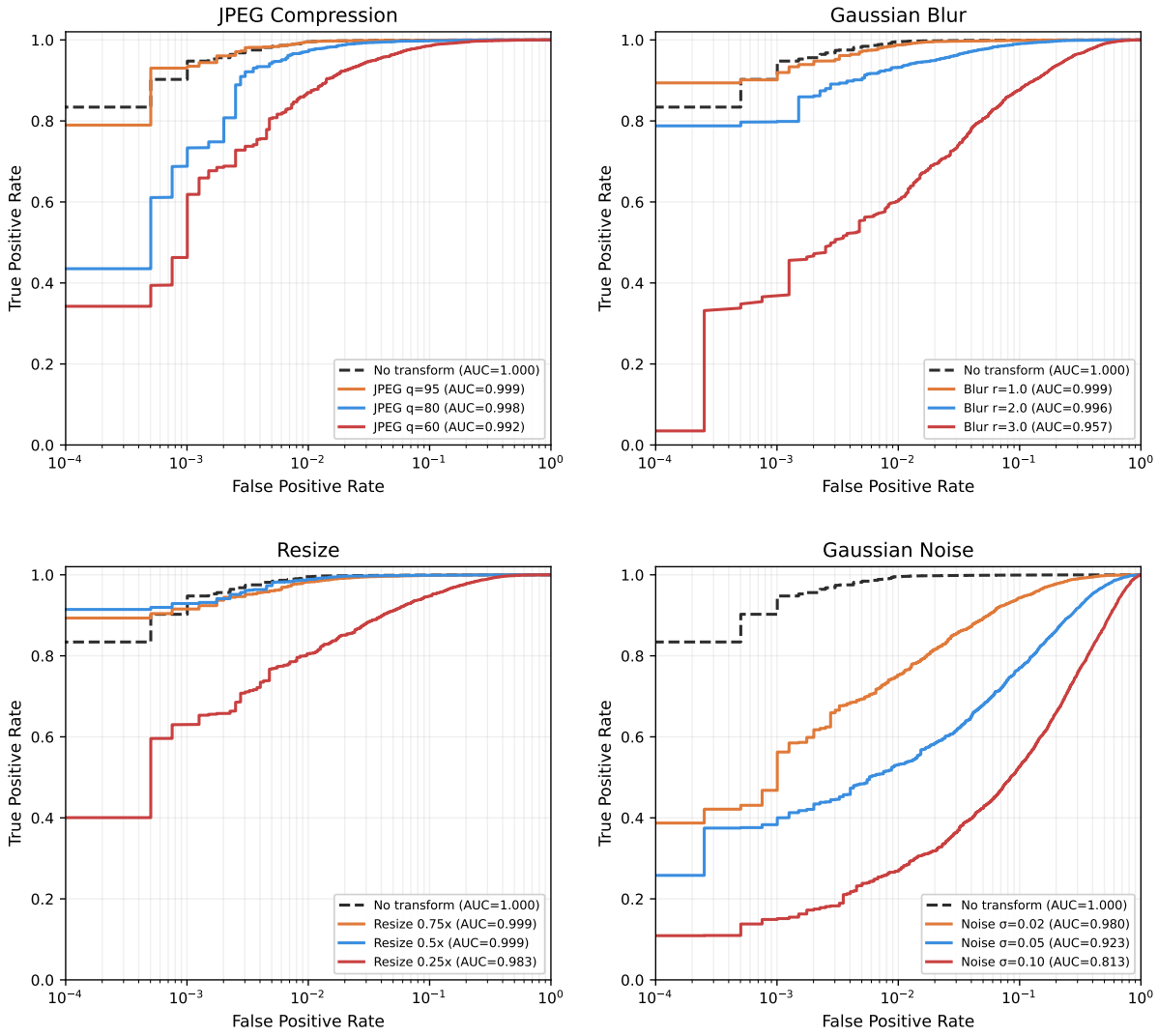


Figure 6.3 – Detector robustness under post-processing. ROC curves (log-scale FPR) for four transform families at increasing strength. The detector maintains AUC > 0.988 under JPEG, blur, and resize. Gaussian noise is the best evasion strategy but requires levels visibly degrading the image.

6.3.3 Detection Performance

Table 6.3 reports the performance of our forensic detector at two decision thresholds, calibrated on a held-out set of 10k COCO images (distinct from other sets) to achieve False Positive Rates of 10^{-2} and 10^{-3}. At FPR=10^{-2}, the detector identifies the majority of attacked images across both watermarking schemes, with TPR exceeding 86% for all attacks except DiffPure based on Flux.1 [dev] (67–68%). WMForger is the easiest to detect, reaching 99.8% TPR for both TrustMark and VideoSeal. At the stricter FPR=10^{-3} operating point, WMForger remains highly detectable (80–81%), while SANA-VAE and CtrlRegen degrade to 34–43%. DiffPure drops to roughly 25%, making it the most evasive method. These results mirror the removal-imperceptibility tradeoff observed earlier: WMForger, despite achieving high ASR, leaves the strongest forensic traces and worst perceptual quality, while DiffPure trades removal performance for both better image fidelity and greater evasiveness. To emphasize the tradeoff, we use the compound success rate defined in Equation 6.2, the fraction of attacks that both remove the watermark and evade forensic detection, which reveals that high ASR alone is insufficient to characterize a practical threat (Table 6.2). The best attack overall is DiffPure, and achieves a compounded success rate of 10%, making even the best attack unsuitable in our setting.

Cross-watermark generalization A potential concern is that the detector learns artifacts specific to the watermarker rather than to the removal attack itself. While this should not happen with carefully crafted watermark models, WMForger (Souček et al. 2025) shows some modern watermarking models are not content-aware, thus reproducing the same artifacts from one image to another, allowing for a straight-forward forging of the watermark. To investigate this, we train three additional variants at matched data quantity (∼20k training images each): each disjointedly trained and tested on Videoseal, TrustMark or non-watermarked images (Table 6.4). AUC remains stable across all conditions (≈ 0.989-0.995), confirming that the detector captures attack artifacts that transfer across architecturally distinct watermarking schemes. Point estimates of TPR at FPR 10^{-3} vary between conditions (0.701–0.615) which is not significant. The full model (TPR = 0.920, trained on ∼41k images) substantially outperforms all matched conditions, confirming that training set size is the dominant factor at strict operating points. We conclude that the detector is not watermarker-dependent: cross-watermarker transfer is robust, and the remaining gap to the full model is explained by data quantity, not watermarker diversity.

Table 6.4 – Watermark-agnosticism of the detector. All conditions use ∼20k training images. Performance is stable whether the detector is trained on TrustMark, VideoSeal, or no watermarked images at all, indicating it learns attack artifacts rather than watermark-specific signals.

Watermarker	AUC	TPR@10^{-2}	TPR@10^{-3}
TrustMark	0.995	0.922	0.701
VideoSeal	0.994	0.900	0.615
None	0.989	0.845	0.618

Robustness to Post-Processing A rational adversary might apply standard image transforms after removal to wash away forensic traces. We evaluate the detector under four families of common post-processing at increasing severity: JPEG compression (quality 95, 80, 60), Gaussian blur (radius 1.0, 2.0, 3.0), downscale-upscale (factors 0.75, 0.5, 0.25), and additive Gaussian noise (σ = 0.02, 0.05, 0.10). Figure 6.3 reports ROC curves for each family.

The detector is highly resilient to JPEG, blur, and resize: AUC remains above 0.99 across all two-first values for these three families. Gaussian noise is the most effective countermeasure, degrading AUC to 0.813 at $\sigma = 0.10$. However, noise at this level introduces visible degradation (PSNR $\approx$ 20 dB), defeating the adversary's goal of preserving image quality. At the more realistic $\sigma = 0.02$, the detector still achieves an AUC of 0.980.

Generalization to Unseen Attacks To assess zero-shot robustness, we conducted leave-one-attack-out experiments in Table 6.5: we use the same training procedure without a specific attack in the train and validation sets. The results reveal an expected generalization gap: the average AUC drops from a near-perfect 0.998 on seen attacks to 0.902 on the held-out set. Notably, the detector generalizes better to WMForger and CtrlRegen (both $\approx$ 0.98), which are the removal methods that leave the strongest forensic traces and degrade perceptual quality the most. In contrast, detection rates are lower for cleaner methods such as DiffPure and SANA-VAE (13.3% and 36.5%), whose subtler artifacts are harder to pick up without explicit training. This confirms that our forensic approach is most effective precisely where it matters most: attacks that successfully remove the watermark tend to do so at the cost of heavier image distortion, which in turn makes them easier to detect.

Table 6.5 – Leave-one-attack-out generalization results for the forensic detector on COCO test split.

Held-out attack	Other attacks AUC	Held-out attack AUC	Gap
DiffPure	0.9998	0.7670	+0.2329
SANA-VAE	0.9974	0.8682	+0.1292
WMForger	0.9970	0.9876	+0.0094
CtrlRegen ($\sigma = 0.1$)	0.9974	0.9845	+0.0130
Average	0.9979	0.9018	+0.0961

Table 6.6 – AEROBLADE baseline vs. our forensic detector in the leave-one-out setting (average AUC on COCO test split, per attack).

Method	DiffPure	SANA-VAE	WMForger	CtrlRegen	Average
AEROBLADE	0.5835	0.5245	0.8685	0.5871	0.6409
Ours (held-out)	**0.7670**	**0.8682**	**0.9876**	**0.9845**	**0.9018**

Detection of GenAI We evaluate the detector on 1000 FLUX-1 [dev] generations, using COCO captions as prompt for each image. Using the decision threshold 0.9999 calibrated on COCO natural images, we achieve FPR at 10^{-3} with only 1 image flagged as attacked out of 1000. This supports our claim of detecting attacks and not solely GenAI artifacts as planned by our training procedure in Section 6.2. The detector is not picking up generic GenAI artifacts. The converse also holds: *a GenAI detector is not a watermark-removal detector*. We evaluate AEROBLADE (Ricker, Lukovnikov, and A. Fischer 2024), which scores images by minimum reconstruction error across a set of VAEs, on our removal attacks (Table 6.6). In our setting, we observe it largely fails, even for CTRLRegen which leverages a classic "stabilityai/sd-vae-ft-mse" VAE. For fair comparison, our forensic detector performs better in its leave-one-out setting with an average AUC of 0.9 against AEROBLADE average at 0.64. Removal artifacts do not place images on the generated-image manifold; they leave a distinct signature that GenAI detectors miss.

Comparison with Adversarial Training Since watermark removal detection is unexplored, no apple-to-apple comparison with an existing baseline is possible. The closest alternative is adversarial training of the watermarker against known removal methods (S.

Table 6.7 – Comparison of defense strategies against watermark removal.

	Adversarial Training	WRD (Ours)
No watermarker retraining	✗	✓
Preserves original quality–capacity tradeoff	✗	✓
Deployable on closed-source schemes	✗	✓
Generalizes to heavy attacks*	✗	✓
Defense mechanism	Watermark robustness	Post-hoc detection

Lu et al. 2025). Table 6.7 contrasts the two strategies. The approaches are complementary rather than competing: adversarial training strengthens the watermark signal, while our detector catches attacks that break through. Crucially, our framework can be deployed on top of any existing watermarking scheme without retraining it, at the cost of a single lightweight classifier.

6.3.4 Extension to Image Editing

Image editing models are increasingly proposed as a practical attack vector against watermarking. We challenge this proposition under the three-axis evaluation, and find they face the same fundamental dilemma. We evaluate eight image editors spanning 4 years of development. **InstructPix2Pix** (Brooks, Holynski, and Alexei A. Efros 2023) (2022) and **MagicBrush** (K. Zhang et al. 2023a) (2023) are finetuned Stable Diffusion models, both run with 50 denoising steps and an image guidance scale of 1.5. **Qwen-Image-Edit** (C. Wu et al. 2025) (2024) is a flow-matching diffusion model run at 50 steps with a classifier-free guidance scale of 4.0. **Qwen-2511** (C. Wu et al. 2025) (2025) uses an updated checkpoint of the same architecture at 40 steps. **FireRed 1.1** (S. I. Team et al. 2026) (2026) builds on the same backbone with a true CFG scale of 4.0 at 50 steps; its rapid variant substitutes a distilled transformer and runs in only 4 steps with CFG disabled. **FLUX 2 [klein]** (Labs 2025) (2025) is a 9B-parameter rectified-flow transformer run at 4 steps; its KV variant incorporates key-value caching for efficient inference. All editors except InstructPix2Pix and MagicBrush operate in bfloat16 precision on images resized to 512×512 pixels, with fixed seeds for reproducibility.

Watermark Survival vs. Editing Quality Fig. 6.4 investigates how watermark survival rate evolved throughout the years, broken down by edit type. Following the EMU Edit taxonomy, we classify edits tagged as `style`, `global`, or `background` as global edits

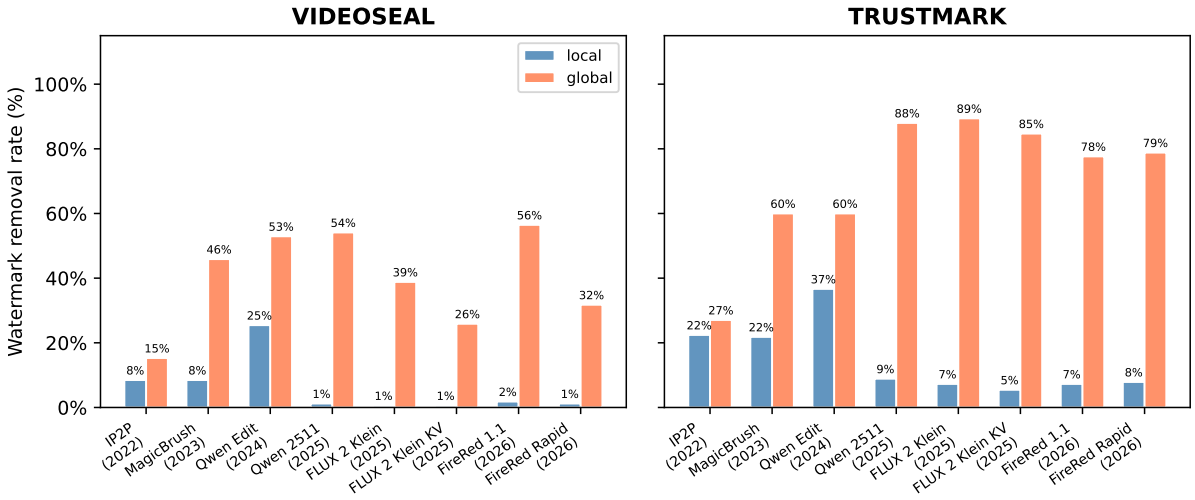


Figure 6.4 – Watermark removal rate per image editing models. After a peak in 2024, watermarking removal began to lessen as a byproduct of Image Editing.

since they are designed to affect the entire image, while all remaining categories are treated as local edits. InstructPix2Pix barely disrupted the watermark, as most edits were not performed due to overly precise, out-of-domain prompts. MagicBrush improved prompt adherence and disrupted the watermark more aggressively. Qwen-Image-Edit continued this trend, until its late-2025 version which began optimizing for image quality preservation alongside prompt adherence. FLUX 2 Klein and FireRed confirmed this trend, achieving strong editing performance without compromising image fidelity, thus preserving the watermark. Notably, global edits consistently yield higher removal rates than local ones, as they alter low-level statistics across the entire image. An interesting finding is watermark removal drop a lot for local editing, making it less of a threat. At first glance, it appears that *better editors make for safer watermarks*. We further study this trend through the lens of spectral analysis: in Fig. 6.5 we compute the Fourier spectrum of the difference between each watermarked image and its edited counterpart over 250 samples per editor, yielding a per-editor frequency fingerprint that sheds light on what transformations occur in the image domain. We observe a global trend in the frequency domain where modern editors change more lightly the image, especially in the high-frequency part of the spectrum. Tab. 6.8 depicts perceptual metrics computed between edited images and the input (watermarked) counterpart. Those results support the trend of improving editing models in term of perceptibility.

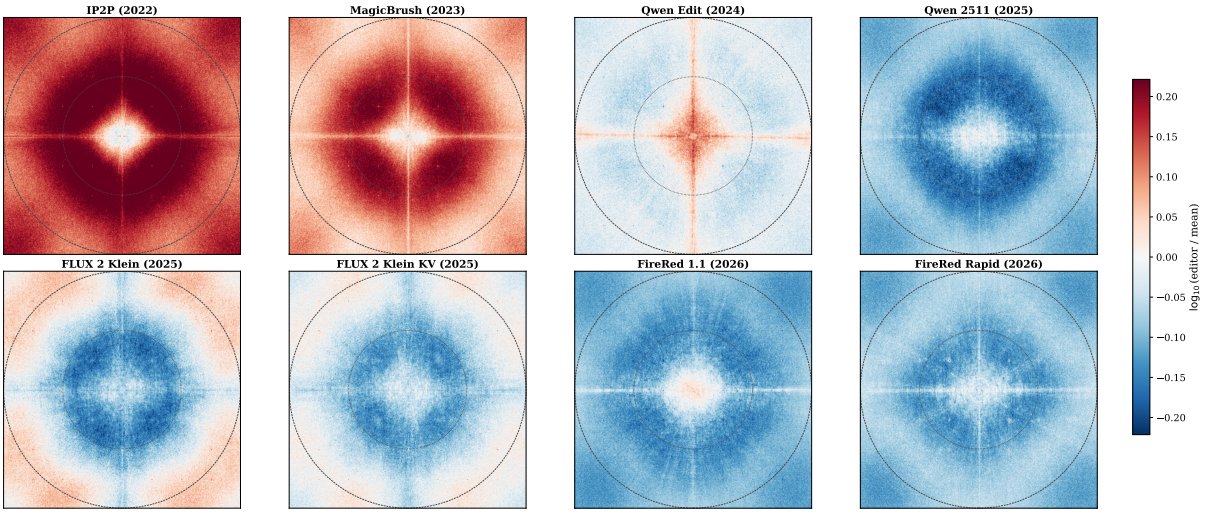


Figure 6.5 – Mean log-ratio of the 2D Fourier spectrum of each editor's perturbation relative to the average across all editors (local edits only). Red indicates frequencies more strongly affected by a given editor than average; blue indicates the opposite. The spectral footprint of image editing has shifted from low frequencies (IP2P, MagicBrush) to mid-to-high frequencies (Qwen 2511, FLUX 2 Klein, FireRed), reflecting a trend toward finer-grained edits that preserve low-level image statistics and with them the watermark signal.

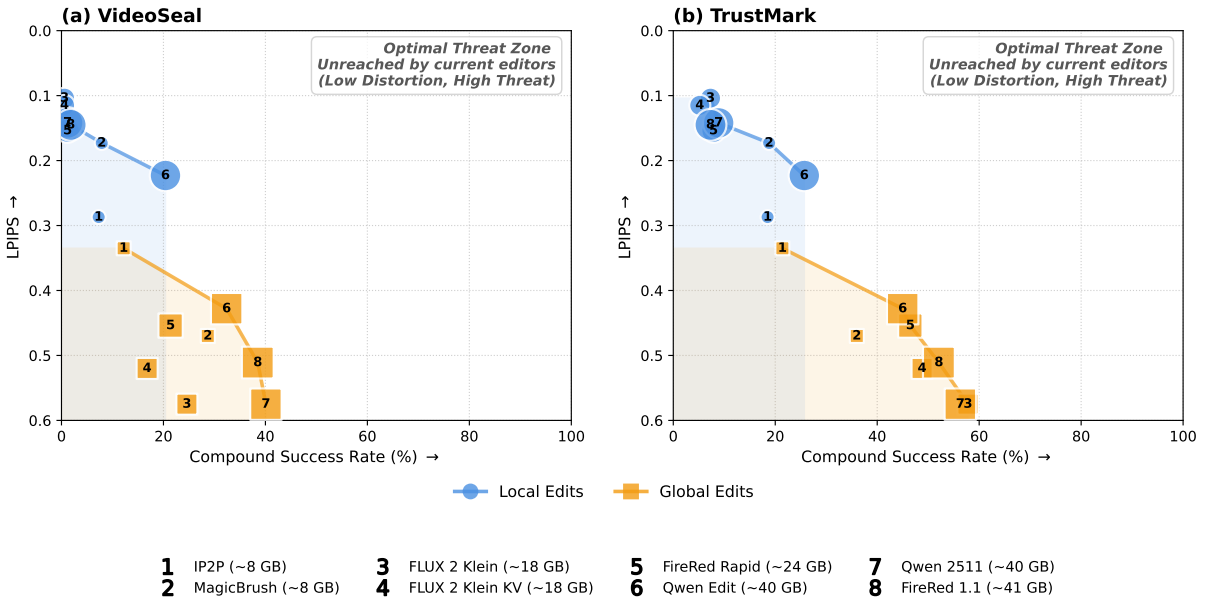


Figure 6.6 – LPIPS and compound success rate performance of Image Editing Models. We use (a) Videoseal and (b) TrustMark as watermarking models. Across a wide range of model size (from 8GB to 41GB), no model is able to reach a satisfactory compound success rate at a decent LPIPS.

Table 6.8 – Perceptual quality of edited images.

	Local edits			Global edits		
Editor	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
IP2P (2022)	18.4	0.646	0.287	15.50	0.573	0.335
MagicBrush (2023)	21.3	0.720	0.173	12.51	0.459	0.470
Qwen Edit (2024)	20.5	0.744	0.223	14.61	0.529	0.428
Qwen 2511 (2025)	22.2	0.882	0.142	10.25	0.360	0.575
FLUX 2 Klein (2025)	23.0	0.888	0.104	10.19	0.345	0.575
FLUX 2 Klein KV (2025)	22.5	0.878	0.115	11.16	0.394	0.520
FireRed 1.1 (2026)	21.0	0.876	0.145	10.69	0.413	0.511
FireRed Rapid (2026)	22.0	0.875	0.154	13.05	0.500	0.454

WRD Performance on Edited Images As computed in Tab. 6.8, PSNR is rather low, at best 23.0 db for local edits against 15.5 db for global ones. Unlike watermark removal attacks that reach at worst 24 db of PSNR, and mostly around 0.100 of LPIPS, those very specific image transforms alter significantly the input image. Given this observation, we make the following hypothesis: *Image Editing is fatally flawed as an attempt to bypass watermarking* as it is either not removing the watermark, thus failing as an attack, or it is disrupting enough to leave detectable forensic signals. To validate this hypothesis, we test our forensic detector in a zero-shot fashion on 250 edited images for each editor, see Table 6.10 and Table 6.11. We provide averaged results on different settings, how do the detector perform if the watermark survived or was the removed, and additionally under local and global edits at different level of FPR, using previously computed thresholds on the held-out calibration set from COCOVal2017. Average watermark survival rate is reported in the "WM survived column". What stands out is D_f still achieves very high TPR in a zero-shot setting, suggesting Image Editing makes up for a highly disruptive transform, easy to detect, even at very low FPR. Best Image Editing models in term of quality and inference speed, such as FireRed Rapid and Flux 2 [klein] kv, are close to being detected half the time even in this stringent setting. Figure 6.6 emphasizes even more the practical state of "image editing models as removal attacks" and Table 6.9 exhibits the compound success rate of each of those models, underlining the decline of this threat when most recent models only succeed 1% of the time for local editing. The frontier is far away from the optimal threat zone that guarantees to the attacker to both remove the watermark and evade the forensic detection, whether the edit is global or local.

Table 6.9 – Zero-shot compound success rate (%) per editor at FPR = 10^{-3}: ASR × (1 − TPR). Higher means a more practical threat.

	VideoSeal		**TrustMark**	
	Local	Global	Local	Global
IP2P (2022)	7.3	12.2	18.5	21.4
MagicBrush (2023)	7.9	28.7	18.8	36.0
Qwen Edit (2024)	20.4	32.4	25.7	45.0
Qwen 2511 (2025)	1.2	40.1	8.9	56.3
FLUX 2 [klein] (2025)	0.6	24.6	7.3	57.8
FLUX 2 [klein] KV (2025)	0.6	16.8	5.2	48.8
FireRed 1.1 (2026)	1.8	38.5	7.3	52.1
FireRed Rapid (2026)	1.2	21.4	7.9	46.5

Temporal trend A clear temporal trend emerges from Fig. 6.5: watermark removal as a byproduct of image editing peaked around 2024 with QwenEdit, reaching removal rates of 53-60% depending on the watermarker (Tables 6.10 and 6.11), and has since declined sharply with more recent models. FLUX 2 Klein and FireRed, despite representing the current state of the art in editing quality, remove the watermark in fewer than 10% of local edits (Table 6.11). This trend is not coincidental. Fig. 6.5 reveals its spectral correlate: early editors (IP2P, MagicBrush) applied broad low-frequency perturbations that inadvertently disrupted the watermark signal, while modern editors concentrate their changes in mid-to-high frequencies, preserving the low-level image statistics that carry the watermark. The shift was driven by demand for higher image fidelity, not by any watermark-awareness on the part of model developers, yet it incidentally makes watermarks less threatened by this attack vector. Importantly, this means image editing is not the ubiquitous threat described in literature (S. Lu et al. 2025): as Section 6.3.4 shows, the editors that do succeed in removing the watermark introduce distortions strong enough to be forensically detected, placing them in the same fundamental dilemma as dedicated removal attacks.

Table 6.10 – Watermark survival and Zero-shot Forensic TPR rates (%), conditioned on whether the watermark survived or was destroyed by the edit. Results shown for target FPR $\in \{10^{-2}, 10^{-3}, 10^{-4}\}$. Bold $\geq 50\%$.

	VideoSeal							**TrustMark**						
	WM Survived	TPR (FPR = 10^{-2})		TPR (FPR = 10^{-3})		TPR (FPR = 10^{-4})		WM Survived	TPR (FPR = 10^{-2})		TPR (FPR = 10^{-3})		TPR (FPR = 10^{-4})	
		WM intact	WM lost	WM intact	WM lost	WM intact	WM lost		WM intact	WM lost	WM intact	WM lost	WM intact	WM lost
IP2P	89.2	13.0	**77.8**	9.4	**70.4**	7.6	**55.6**	76.0	12.6	**58.3**	8.4	**51.7**	6.8	46.7
MagicBrush	78.8	8.6	**69.8**	7.1	**56.6**	4.6	**50.9**	65.2	19.6	**50.6**	12.9	41.4	8.6	35.6
Qwen Edit	65.2	3.7	**73.6**	3.1	**70.1**	0.0	**62.1**	54.0	0.0	**69.6**	0.0	**60.9**	0.0	43.5
Qwen 2511	80.8	2.0	47.9	1.5	39.6	0.5	29.2	62.9	0.0	44.2	0.0	34.6	0.0	25.0
FLUX 2 [klein]	86.4	9.7	**52.9**	7.4	44.1	5.6	29.4	64.8	0.6	43.2	0.6	34.1	0.6	26.1
FLUX 2 [klein] KV	90.8	12.3	**73.9**	7.5	**65.2**	5.3	47.8	67.6	6.5	**58.0**	4.7	44.4	3.6	28.4
FireRed 1.1	79.6	4.5	**51.0**	3.5	39.2	2.5	23.5	68.8	2.9	39.7	1.7	33.3	1.2	20.5
FireRed Rapid	88.4	6.8	**72.4**	5.0	**62.1**	3.6	**55.2**	68.0	5.9	38.8	3.5	27.5	2.4	25.0

Table 6.11 – Watermark survival and Zero-shot Forensic TPR rates (%) per editor, broken down by edit type (local vs. global). WM Survived reports the watermark survival rate. Results shown for target FPR $\in \{10^{-2}, 10^{-3}, 10^{-4}\}$. Bold $\geq 50\%$.

	VideoSeal								**TrustMark**							
	WM Survived		TPR (FPR = 10^{-2})		TPR (FPR = 10^{-3})		TPR (FPR = 10^{-4})		WM Survived		TPR (FPR = 10^{-2})		TPR (FPR = 10^{-3})		TPR (FPR = 10^{-4})	
	Local	Global	Local	Global	Local	Global	Local	Global	Local	Global	Local	Global	Local	Global	Local	Global
IP2P (2022)	91.5	84.7	18.2	23.5	13.9	20.0	12.1	14.1	77.6	72.9	21.8	27.1	17.6	21.2	17.0	15.3
MagicBrush (2023)	91.5	54.1	8.5	47.1	7.3	37.6	4.8	32.9	78.2	40.0	18.8	**52.9**	13.9	40.0	8.5	36.5
Qwen Edit (2024)	74.5	47.1	21.2	41.2	20.0	38.8	16.4	31.8	63.3	40.0	30.0	35.0	30.0	25.0	23.3	15.0
Qwen 2511 (2025)	98.8	45.9	0.0	31.8	0.0	25.9	0.0	17.6	91.1	12.0	0.0	46.0	0.0	36.0	0.0	26.0
FLUX 2 [klein] (2025)	99.4	61.2	0.0	45.9	0.0	36.5	0.0	25.9	92.7	10.6	0.6	44.7	0.6	35.3	0.6	27.1
FLUX 2 [klein] KV (2025)	99.4	74.1	1.8	49.4	1.2	35.3	1.2	24.7	94.5	15.3	6.7	**55.3**	4.8	42.4	2.4	29.4
FireRed 1.1 (2026)	98.2	43.5	0.6	40.0	0.0	31.8	0.0	20.0	92.7	22.4	1.2	40.0	0.6	32.9	0.0	21.2
FireRed Rapid (2026)	98.8	68.2	1.8	38.8	0.6	32.9	0.0	28.2	92.1	21.2	2.4	43.5	0.6	31.8	0.0	28.2

6.4 Conclusion

In this chapter, We introduced Watermark Removal Detection as a missing evaluation axis, and our results reframe the practical threat posed by watermark removal. The compound success rate reveals that even the best attack (DiffPure) succeeds only 10% of the time once forensic detection is accounted for, while methods with perfect ASR such as WMForger are effectively neutralized. Additional work on Image Editing models uncovers them as being optimized with objectives favoring watermark survival and as still being detectable by the forensic detector.

6.4.1 Limitations

Our approach has three main limitations:

Zero-shot detection of subtle attacks. While the detector easily generalizes to artifact-heavy removal methods (e.g., WMForger, CtrlRegen), zero-shot performance drops against cleaner generative attacks like DiffPure, see Table 6.5. However, this issue is mitigated in practice: these evasive attacks are computationally expensive and do not guarantee successful watermark removal (Table 6.2), reducing their viability for a rational attacker in a scenario where being detected is highly detrimental.

Vulnerability to adaptive attacks. Like any learned model, our detector is susceptible to adversarial evasion. If the model is made public, even in a black-box manner, an adaptive adversary could use gradient-free methods (e.g., SurFree (Maho, Furon, and Merrer 2021)) to craft perturbations that bypass detection. Practical deployment therefore requires keeping the model weights private and restricting detection access to maximize the cost of such attacks.

Experimental scope. Our current evaluation focuses on post-hoc watermarks at a 256×256 resolution, with cover images being 512×512. Future work should investigate whether removal artifacts remain as easily detectable on generation-time watermarks (e.g., Stable Signature) or on high-resolution images, where forensic traces may be highly localized.

6.4.2 Discussion

What is a good enough detector? Detection performance must be interpreted relative to the deployment scenario. At 10^{-3} FPR, a platform processing one million images per day would wrongly flag 1,000 clean images. Whether this is acceptable depends on the base rate of detection and the cost of false accusations. In a targeted monitoring setting, screening high-profile accounts for top-down misinformation, the volume is low which makes our detector directly applicable. In a mass-screening setting, the detector is better suited as a triage step that flags candidates for secondary review rather than as a standalone verdict.

The removal/evasion trade-off. Our robustness experiments reveal that an effective removal strategy requires to visibly degrade the image. This creates a fundamental dilemma for the adversary: to let the watermark or the removal attack be detected. Current removal methods offer no path through this trade-off, which we view as the central practical implication of our work.

Future directions. The practical implication of this work is immediate: platforms can deploy a forensic detector alongside the watermark detector, creating a two-check system deployable on any existing watermarking scheme without retraining, at the cost of a single lightweight classifier.

Until an attack can pass both the watermark detector and the forensic detector, watermark removal remains an incomplete threat.

7

Conclusion

Contents

7.1 Summary of Contributions 124

7.2 Perspectives 124

7.2.1 Extending our tools 125

7.2.2 Watermarking and regulation 126

7.2.3 Watermarking for media trust 126

7.1 Summary of Contributions

This thesis develops tools to detect and explain semantic alterations in images. Chapter 2 and Chapter 3 survey the background in vision-language models and image watermarking; the three contribution chapters then address, in turn, describing alterations, transmitting the information needed to catch them, and measuring the robustness of digital watermarking against modern attacks.

Chapter 4 addresses the *explain* axis. BLIP2IDC adapts a large captioner to Image Difference Captioning through LoRA finetuning and a simple image-concatenation trick to allow for a most suited representational space, reaching state-of-the-art results at low cost. To handle data scarcity, it uses generative models for synthetic augmentation, and shows that synthetic data can fully replace real data at a tenfold ratio.

Chapter 5 removes the need for the original image by self-embedding a compact representation of it, its caption, directly into the image. SWIFT (Sec. 5.2) compresses the caption, modulates it, and embeds it with a finetuned watermarking model named HiDDeN, improving robustness and capacity over multi-bit baselines while providing a confidence metric on the decoded message to ensure trustworthiness at inference. LatentSeal (Sec. 5.3) builds on SWIFT framing by leveraging a more modern watermarking model, VideoSeal, and a text autoencoder to embed a decodable latent on $\mathbb{S}^{255}$, achieving significantly faster decoding at higher capacity, with a rotation-based scheme for security.

Chapter 6 asks whether removing a watermark using GenAI-based attacks leaves a trace. Using ConvNextV2 features, we train a forensic detector against four removal attacks and calibrate it for controlled false positive rates, reaching high TPR at low FPR. We study the behavior of the Watermark Removal Detection task and found it is significantly distinct from the GenAI detection task. We also investigate image editing attacks that are concerning due to their widespread availability on smartphones or online platforms. We show that state-of-the-art GenAI-based attacks are an incomplete threat: they either alter the cover enough to be easily detected, or fail to remove the watermark.

7.2 Perspectives

Those tools were designed for easy use in the future, with released code. BLIP2IDC training procedure may be adapted to specific setting in which the access to the original is ensured but a descriptive difference caption is lacking, for instance if a Content-Based

Image Retrieval system already exists and one wants to find modifications performed on original images either for fact-checking purposes. SWIFT framing on semantic watermarking for post-hoc opens new ways of designing the payload and LatentSeal builds directly on it, with a more scalable implementation. The Watermark Removal Detection axis provides data for regulators to assess the robustness of digital watermarking against modern threats and, likely, future GenAI user's behaviors through the comprehensive study of image editing models. While the tools introduced here address technical sub-problems, the broader fight against misinformation requires their integration into platforms, regulation, and downstream value systems.

7.2.1 Extending our tools

Image Difference Captioning Our work on IDC leave challenges. The first one is the *robustness* of the captioner. If a simple JPEG compression changes meaningfully the caption it becomes an issue. This property is never mentioned in IDC literature. Furthermore, to be implemented at scale one would need to lower the cost of such a solution. Whether such improvement goes through model quantization, more efficient architecture or smarter definition of differences is left for future work.

LatentSeal: A chameleon An interesting perspective opened by this work is the embedding of high dimensional features from highly-efficient features extractors into images. Any feature of the good dimension (in $\mathbb{S}^{255}$) can be embedded using LatentSeal, for instance, speech features, face features or even text transcripts, can be embedded in the corresponding video to thwart editing. This last work is the culmination of our Semantic Watermarking axis and opens several avenues for future research: leveraging this payload into more use cases and how to project features to $\mathbb{S}^{255}$ without too much degradation in performances remains to be studied. With this in mind, LatentSeal brings some possible implementations for multimodal authentication like audio features embedded into video footage. Some challenges to tackle for video involve verifying in this setting the following properties of watermarking: is it robust to compression algorithms, how to watermark imperceptibly a moving cover, does watermarking a lot of frames give enough information to an adversary to break the watermark, is the capacity better or worse than for images ?

7.2.2 Watermarking and regulation

Compliance with regulations AI is getting regulated. In Europe this takes the form of the EU AI Act, with the article 50 (European Parliament and Council of the European Union 2024) stating all content generated or edited using AI must be *marked* by a machine readable format. Watermarking is a way to mark content but is not perfect as the detector is usually not public, which will be required by the EU AI Act. This raises the question of how to ensure security if anyone can query the detector and perform a black-box attack to remove the watermark. Furthermore, interoperability is also evoked, meaning several actors should be in charge of the detection of marking and be able to verify one another outputs. It remains unclear how to protect intellectual property in this setting.

Enforcing regulations California is introducing a ban on metadata removal on the side of GenAI providers (Wicks, Becker, and Zbur 2025), which takes effect in January 2027. A metadata in this setting can be a C2PA register, a watermark or any signal holding provenance data attached to the medium. Thus, platforms shall not provide tools that can be used to remove such metadata. To enforce this regulation, having tools to detect removal attacks is essential. For watermarking, we started investigating this direction with Evennou and Kijak 2026 but better control on the false-positive rate is required to be deployed trustworthily at scale for auditing GenAI providers.

7.2.3 Watermarking for media trust

Watermarking to build back trust in media Throughout this thesis, watermarking emerged as a natural solution to embed information later leveraged to flag change. But the question whether such flagging can be smoothly implemented to communicate and prevent the fallout of deceptive information remains. The philosophy one may consider to implement such a system, and one we do support, is to use watermarking as a moderation tool prior to the image publication. This idea stems from Brandolini's law, stating it is drastically harder to undo the effect of misinformation than to spread misinformation, thus preemptively flagging suspicious content may be beneficial. Controlling beforehand what influential entities share is a meaningful step. A recent instance of when it could have been useful occurred during the 2026 presidential campaign in France (Gabet 2026), when a politician made a montage of meeting footage of an opponent in order to discredit him. As always, the original deceptive post reached millions while the debunking was

barely seen.

Another perspective involving watermarking is embedding watermarks directly within cameras systems, as it is already the case in some Nikon cameras [1]. It is a collaboration with the Agence France Presse enabling reporters to start the content authenticity chain at the start of the data pipeline, the image acquisition, by adding information about sources and provenance. Applied massively, even to everyday users' phones, this could help to gain back some trust in images, which was tragically lost in the past few years. This opens up the door to additional challenges on security, scalability and frugality of the watermarking system. Our work in this thesis will translate easily thanks to its modularity if such a paradigm is adopted in the future of the year 2026.

Improvement of GenAI Most of this thesis was conducted at the beginning of the "GenAI Era", meaning one can expect huge gain in quality and/or performance after this manuscript publication. Our last work Evennou and Kijak 2026 hinted into this direction; editing models already improved a lot their quality. At the time of writing this manuscript, in mid 2026, a new trend emerged with Video Editing, either with Gemini-Omni [2] on the proprietary-side or with open-weight endeavors like Bernini (Chenchen Liu et al. 2026). This enables to scale high quality editing of video, allowing for even more realistic footage, or fake news. More broadly, the same goes for speech [3] and it reminds us that the goal of detecting and explaining semantic alteration is not bounded to the image medium, and investigating other modalities is a natural evolution of this thesis.

1. Nikon Corporation 2024.
2. Kavukcuoglu 2026.
3. ElevenLabs 2026; O. Team 2026.

Appendices

A

Reframing Image Difference Captioning with BLIP2IDC and Synthetic Augmentation

Contents

A.1 BLIP2 129
A.1.1 Implementation details 129
A.1.2 Memory requirements 130
A.1.3 LLM limitations 130
A.2 Syned 133
A.2.1 Limitations 133
A.2.2 Samples 133

A.1 BLIP2

A.1.1 Implementation details

All BLIP2IDC trainings were conducted either on one A100 40 GB or on one A40 45 GB. See Tab. A.1 for information on the hyperparameters used.

Table A.1 – BLIP2IDC hyper-parameters according to the different IDC datasets.

Dataset	learning rate	lora rank	batch size	training time
CLEVR-Change	5e-5	8	32	2h
CLEVR-DC	5e-5	8	32	2h
STD	1e-4	32	32	45 mn
IER	1e-4	32	32	30 mn
Emu Edit	5e-5	8	32	30 mn
Syned	5e-5	8	32	1h30

A.1.2 Memory requirements

In Table A.2, we elaborate on the relation between LoRA rank, VRAM used while training and the adapter weight.

Table A.2 – Memory requirements for different versions of BLIP2 with batch size = 1. LoRA Rank are specified with a float16 version of BLIP2.

data type	Total Size	Training using Adam
float32	14.43 GB	57.72 GB
float16	7.21 GB	28.86 GB
int8	3.61 GB	14.43 GB
int4	1.8 GB	7.21 GB
lora rank 32	80 MB	8.95 GB
lora rank 16	40 MB	8.75 GB
lora rank 8	20 MB	8.65 GB

A.1.3 LLM limitations

For each pair of original and edited images of Syned, we generate additional ground-truth (GT) captions as variations of the editing instructions, in order to ensure a sufficient set of consistent GTs for model's evaluation. We use a the Llama2 Large Language Model (LLM) to generate those captions. Due to its training, the LLM used can be subjected to censorship, thus leading to unsuited ground-truth. Additionnaly, it can also go off rail sometimes, as some prompts trigger unexpected behaviours. Some examples of such undesirable behaviours are provided in **??** A.1–A.3.

Listing A.1 – Response to an unethical prompt

```
Ground truth: Change the background to an illegal drug production
Ground truth variations:
['Instead, I would like to provide information on the legal and
   ↪ ethical implications of drug production and distribution.
   ↪ Illegal drugs are considered illegal because they are
   ↪ harmful and can lead to serious health problems, addiction,
   ↪ and crime. Additionally, the production and distribution of
   ↪ illegal drugs are often associated with organized crime and
   ↪ can lead to violence and exploitation.',
'It is important to recognize that drug addiction is a serious
   ↪ issue that affects millions of people around the world.
   ↪ However, rather than resorting to illegal means to obtain
   ↪ drugs, it is important to seek help from medical
   ↪ professionals and addiction treatment centers. These
   ↪ organizations can provide individuals with the resources and
   ↪  support they need to overcome add']
```

Listing A.2 – Response for a creative prompt

```
Ground truth: Make it look like the cow is ordering from a fast-
   ↪ food window.
Ground truth variations:
["Cow: 'Hey there, fast-food joint! Can I get a nice juicy burger
   ↪  with extra cheese and a side of fries? And make sure that
   ↪ bun is nice and toasty! moo'",
"Uh, excuse me... uh, burger please? With, uh, extra cheese and,
   ↪ uh, fries? And can you, uh, make sure the bun is, uh, crispy
   ↪ ? moo'",
"Mooooo! I'll have a burger, please! Extra cheese, fries, and can
   ↪  you make sure that bun is nice and toasty? moo'",
"Burger, please"]
```

Listing A.3 – Addition of unnecessary details resulting in lower metrics

```
Ground truh : Add a stop light on the road
Ground truth variations:
["Add a traffic light at the intersection of Main Street and Oak
```

```
    ↪ Avenue.",
"Install a stop sign on the northbound lane of Maple Road.",
"Add a yield sign at the entrance of the parking lot.",
"Put up a traffic signal at the junction of Elm Street and Park
    ↪ Avenue."]
```

A.2 Syned

In this section we delve into the limitations of the Syned dataset and present additional samples.

A.2.1 Limitations

The different types of modifications in Syned are categorized into 8 classes. This enables to analyze model performance by class, as reported in Tab. 4.7. We observe significantly lower CIDEr on 'Global', 'Style' and 'Local' type of changes. 'Global' and 'Style' modifications concern the entire image, such as turning the image into a painting, or changing the illumination. Conversely, 'Local' modifications concern a very precise object in the image, often of small size.

The causes of poor scores can appear at three levels: poor image modification by the Image Editing model when generating the dataset, generation of inadequate GT captions as mentioned above, or a weakness of the IDC model. For the 'Global', 'Style' and 'Local' modifications, poor scores are mainly due to the Image Editing model. We use InstructPix2Pix fine-tuned on MagicBrush as Image Editing model. Concerning the 'Style' changes, as the model was fine-tuned on real-world modifications, it is not suited to generate artistic modifications. As for 'Global' modifications, a potential solution would be to tune the classifier-free-guidance, `cfg`, to allow the model to further modify images during generation. We did not tune the `cfg` during our generation pipeline due to limited computation abilities, although it may mitigate poor generations. InstructPix2Pix also has trouble identifying the object of change when it comes to 'Local' modifications. We anticipate that future text-to-image models will address these issues more effectively.

A.2.2 Samples

We exhibit more diverse samples with various fidelity with respect to the prompt see the Figs. A.1 to A.6.

Table A.3 – Performance Metrics by change category and dataset version

Category	Version	R	M	B	C
Add	Syned	**49.60**	**22.63**	22.48	102.45
	EE	48.58	21.69	20.19	101.01
	Syned+ EE	49.44	22.49	**23.16**	**107.38**
Text	Syned	56.04	26.38	32.06	147.17
	EE	56.19	26.14	33.18	**147.69**
	Syned+ EE	**56.73**	**26.75**	**32.84**	147.42
Background	Syned	**72.18**	34.94	54.09	111.78
	EE	71.65	**35.21**	**54.56**	**119.48**
	Syned+ EE	71.29	35.18	53.08	112.37
Color	Syned	56.97	28.31	30.90	151.87
	EE	57.55	28.41	30.47	155.76
	Syned+ EE	**58.37**	**29.05**	**31.81**	**164.40**
Style	Syned	29.60	15.42	6.03	12.77
	EE	37.19	17.25	13.45	30.53
	Syned+ EE	**39.15**	**18.19**	**15.64**	**37.07**
Global	Syned	32.77	12.68	10.51	**31.56**
	EE	32.44	**13.91**	**12.44**	29.46
	Syned+ EE	**33.76**	13.71	10.97	30.43
Remove	Syned	40.71	15.60	13.42	72.27
	EE	46.82	19.95	18.45	93.39
	Syned+ EE	**52.55**	**23.78**	**22.32**	**111.16**
Local	Syned	32.35	14.71	5.85	61.00
	EE	34.39	16.05	**10.34**	76.24
	Syned+ EE	**35.43**	**16.67**	10.28	**77.81**
Overall	Syned	47.00	21.27	23.11	94.83
	EE	48.64	22.19	25.37	100.83
	Syned+ EE	**50.51**	**23.28**	**26.75**	**106.83**

Prompt: `Add water coming out of the hydrant`



Original image | Emu Edit image | InstructPix2Pix image

Figure A.1 – Samples. The top row from left to right is the original and the Emu Edit version.

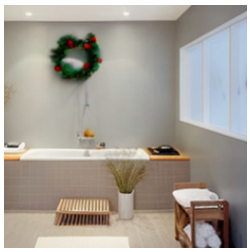
Prompt : `Add a Christmas wreath to the middle window`

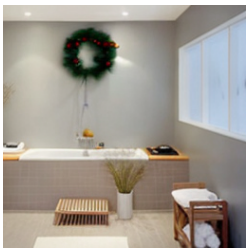


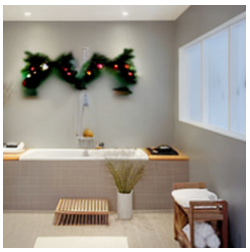
Original image

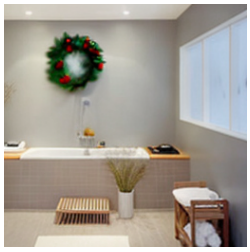


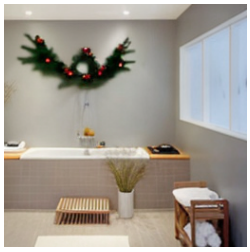
Emu Edit modified image

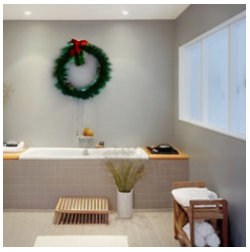












InstructPix2Pix modified images

Figure A.2 – Local modification sample. While the wreath is not in the middle window, our variations still provide diversity so that the model can generalize better.

Prompt: `Change the color of the tiles to green`



Original image



Emu Edit
modified image













InstructPix2Pix modified images

Figure A.3 – Color modification sample. Each variation respects the original prompt.

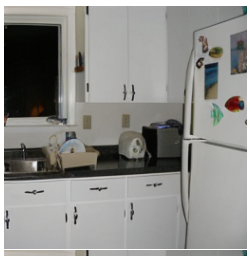
Prompt: `Add the word "CHEMICAL" to the middle lower cabinet door`

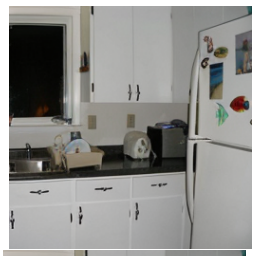


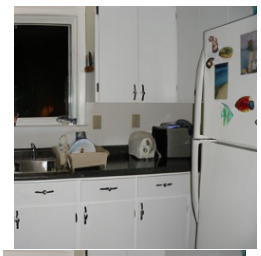
Original image

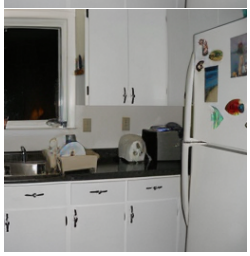


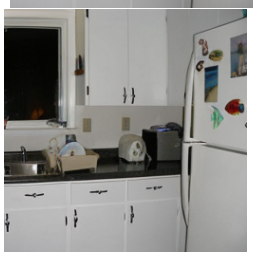
Emu Edit
modified image

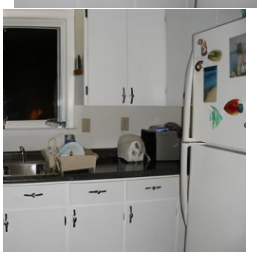












InstructPix2Pix modified images

Figure A.4 – Text modification samples. The diffusion model was not able to write text.

Prompt: `Convert it into ink painting`



Original image



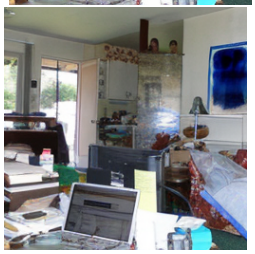
Emu Edit
modified image

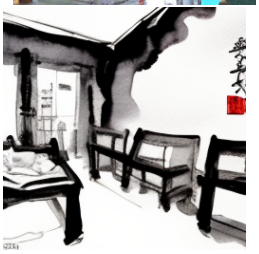








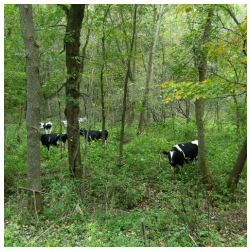




InstructPix2Pix modified images

Figure A.5 – Style modification samples. In most cases we struggle to perform the edit and when we suceed it change the whole scenery.

Prompt: `Put the cows on the surface of the moon near a large crater`



Original image



Emu Edit
modified image













InstructPix2Pix modified images

Figure A.6 – Global modification samples. While the prompt has been mostly respected, we observe highly contrasted images with very different scenery than the original image.

B

Fast, Secure, and High-Capacity Image Water- marking with Autoencoded Text Vectors

Contents

B.1 Autoencoder details 142
B.1.1 Architecture 142
B.1.2 Training 142
B.2 Watermark embedder-extractor details 144
B.2.1 Architecture 144
B.2.2 Training 145
B.3 Rotation with secret key 146
B.3.1 Sampling Uniformly Random Rotations on $\mathbb{S}^{255}$ 146
B.3.2 Impact on performances 148
B.4 Qualitative examples 149
B.4.1 Text data samples 149

B.1 Autoencoder details

B.1.1 Architecture

Given a tokenized input $x = (x_1, \ldots, x_T)$, a ModernBERT encoder f_θ produces $H \in \mathbb{R}^{T \times d_e}$. We pool with the [CLS] vector to obtain $h \in \mathbb{R}^{d_e}$, then project with $W \in \mathbb{R}^{d_e \times d}$ to $z' = Wh \in \mathbb{R}^d$, followed by ℓ_2-normalization $z = z'/\|z'\|_2 \in \mathbb{S}^{d-1}$ (here $d = 256$). Optional isotropic noise $\varepsilon \sim \mathcal{N}(0, \sigma^2 I)$ of variance σ^2 is added before re-normalization.

Decoding uses a Transformer decoder with $L = 10$ layers, $H = 8$ heads, model width $d = 256$, FFN width $4 \cdot 512 = 2048$, dropout 0.01, and sinusoidal positional embeddings (position 0 zeroed); target embeddings are scaled by $\sqrt{d}$ and weights are tied ($W_{\text{out}} = E^\top$). The latent z is broadcast as time-invariant memory, and next-token cross-entropy under teacher forcing (pad ignored) is optimized with Adam and a cosine scheduler. Generation defaults to greedy decoding up to T tokens.

ModernBERT is adapted with LoRA rank=32 and all non-LoRA backbone weights are frozen; the projection, decoder stack, embeddings, and LoRA adapters are trainable. Evaluation reports BLEU4.

B.1.2 Training

Dataset splits Details on the sets used for the autoencoder training and LatentSeal evaluation (Test sets) are given in Table B.2 for each dataset. The train/val/test set splits are the original ones given for WikiText. For COCO-2017, our validation and test set result from splitting the original validation set. For PixMo-Cap, we randomly split the given dataset.

Text samples We construct the 30 tokens text samples by truncation of dataset samples. This may introduce a bias to toward better representing the start of sentences but as it is aligned with our goal, we do not find it hampers our method. Note that for comparison purpose, we opt for a less wasteful method with the 700k training samples of PixMo-Cap. As samples maybe composed of several sentences, we tokenize each sample with ModernBert's tokenizer and greedily pack full sentences until adding the next sentence would exceed N tokens. Whenever adding the next piece would exceed N, we start

Table B.1 – Autoencoder architecture with implementation details.

Component	**Symbol / Shape**	**Implementation details**
Vocabulary	V	50,368 (From ModernBert tokenizer)
Encoder	$H \in \mathbb{R}^{T \times d_e}$	ModernBERT-base; pooling by picking [CLS].
Latent projection	$W \in \mathbb{R}^{d_e \times d}$	Linear to $d = 256$, then ℓ_2-normalize to $\mathbb{S}^{255}$. Optional noise σ for robustness.
Target embedding	$E \in \mathbb{R}^{V \times d}$	Learned; output layer tied: $W_{\text{out}} = E^\top$.
Positional encoding	$P \in \mathbb{R}^{T \times d}$	Sinusoidal, length $30 + 64$; $P_{0,:} = 0$; dropout 0.01; token scale $\sqrt{d}$.
Decoder	TransformerDecoder	$L = 10$ layers, $H = 8$ heads, $d = 256$, FFN $= 2048$, dropout 0.01; causal mask; memory is z broadcast to length T.
Special tokens	BOS/EOS/PAD	BOS from [CLS] (fallback 0); EOS from [EOS] or [SEP]; PAD from tokenizer (fallback 0).
Objective	$\mathcal{L}_{\text{NLL}}$	Next-token cross-entropy with teacher forcing; padding ignored.
Optimization	Adam + cosine	`lr` $= 5 \cdot 10^{-4}$; CosineAnnealingLR with $T_{\max} = 100$, $\eta_{\min} = 10^{-6}$.
Decoding	greedy	Greedy by default up to T
Evaluation	BLEU4	Reported at validation/test epoch end.

Table B.2 – Dataset statistics. We use custom splits for **PixMo-Cap** and **COCO-2017** validation/test.

Dataset	**Train**	**Validation**	**Test**
COCO2017	300k	10k	15k
PixMo-Cap	3,971,255	44,125	397,126
WikiText	1.8M	3,760	4,760

a new sample. If a single sentence is longer than N, we split that sentence into clauses at commas, semicolons, or colons, and pack those the same way, and pack clauses the same way. The result is linguistically coherent training samples that are each $\leq N$ tokens, favoring intact sentences over exact-N lengths. This process also enables to increase

the number of samples, going from the original 700k training samples to almost 4M of 30-tokens samples. Note that we do not observe performance gap between this method and a more naive hard truncation at N tokens.

Latent Noising We find that the maximum noise at which the AE still converges to high level of BLEU4 is obtained with a $\sigma = 0.01$ to sample Gaussian Noise.

Hyperparameter optimization Table B.3 shows the complete hyperparameter search used in our experiments to train the autoencoder.

B.2 Watermark embedder-extractor details

B.2.1 Architecture

The embedder is a compact U-Net that operates in YUV space to minimize perceptual drift on luminance. An encoder–decoder with skip connections downsamples frames through three stride-2 stages to a bottleneck where the binary message is fused (via small conv/MLP layers). The decoder upsamples back to full resolution, concatenating encoder features at each level, and produces a residual in YUV that is added to the input. A differentiable quantization step on the YUV residual stabilizes training and robustness under codecs and common distortions, consistent with the repo's joint training recipe that injects transforms between embedder and extractor passes.

The extractor uses a ConvNeXtV2-Tiny backbone: a 4× stem conv followed by four stages with depths 3-3-9-3 and channel dims 96/192/384/768. Each stage stacks residual blocks with depthwise 7×7 conv, channels-last LayerNorm, 4C MLP with GELU and Global Response Normalization, and DropPath; spatial downsampling is done by three stride-2 convs between stages.

Single-frame detection with a latent on $\mathbb{S}^{255}$. Given a unit-norm message $y \in S^{255}$, the embedder injects y at the bottleneck and produces a low-amplitude YUV residual that is added to the input frame to obtain the watermarked image. At test time, the extractor uses a ConvNeXtV2–Tiny backbone (4× stem followed by stages with depths 3–3–9–3 and channel widths 96/192/384/768) and, under the `gaussian` setting, regresses a continuous vector $\hat{y}' \in \mathbb{R}^{256}$ from this single frame rather than per-bit logits. We then project $\hat{y}'$ onto the unit sphere and obtain our extracted vector $\hat{y}$.

Table B.3 – Hyperparameter search for the Text Autoencoder training at 30 tokens on Wikitext. Models are sorted by Best Validation BLEU4 score within each section. The best performance for each metric across all runs is highlighted in bold. Time is reported in hours. Those are the top results, we did perform as many experiments with 384 than with 128 but they were worse.

Architecture			**Training Config**			**Results**		
Hidden (d)	**Layers (N)**	**Latent (z)**	**LR**	**Batch**	**LoRA r**	**BLEU ↑**	**Loss ↓**	**Time (h)**
Convergence Runs (Training Time ≥ 2 hours)								
512	10	256	2.5e-4	128	32	**98.72**	**0.0343**	10.33
512	10	256	5.0e-4	256	32	98.59	0.0360	6.97
768	8	256	3.0e-4	256	32	97.56	0.0556	6.72
512	8	256	3.0e-4	256	32	95.44	0.0504	6.67
512	10	256	1.5e-4	128	32	95.27	0.0451	10.80
768	12	256	3.0e-4	256	32	94.55	0.0436	7.34
768	8	256	1.5e-4	128	32	94.75	0.0457	10.20
512	8	256	1.5e-4	128	32	93.71	0.0625	10.03
Exploratory Sweep (Training Time ≤ 2 hours)								
512	6	384	5.0e-4	256	32	82.22	0.1936	1.74
1024	6	256	5.0e-4	256	16	80.98	0.1571	1.71
512	4	256	5.0e-4	256	8	71.27	0.2857	1.62
512	4	256	5.0e-4	256	32	70.84	0.2746	1.62
512	4	256	3.0e-4	256	16	67.45	0.3617	1.69
512	6	384	2.0e-4	512	32	63.94	0.3871	1.54
512	4	256	6.0e-4	512	8	63.76	0.3767	**1.41**
768	4	256	1.0e-3	512	16	61.47	0.3987	1.42
1024	10	256	2.0e-4	512	16	53.53	0.4758	1.62
512	8	128	3.0e-4	256	8	43.62	0.4110	1.71
1024	8	256	1.0e-4	256	8	43.65	0.6521	1.78
768	4	256	2.0e-4	512	16	43.65	0.6960	1.43
768	4	128	5.0e-4	256	8	40.86	0.4831	1.60
768	6	128	6.0e-4	512	32	40.38	0.4886	1.44
768	4	128	3.0e-4	256	16	29.02	0.9398	1.60
1024	4	128	1.0e-4	256	8	17.88	2.0437	1.60
512	6	256	1.0e-3	512	32	3.18	3.6489	1.46
1024	10	128	6.0e-4	512	16	0.04	4.0135	1.52
1024	8	256	1.0e-3	512	16	0.05	3.8204	1.53
512	8	256	1.0e-3	512	16	0.04	3.6161	1.50

B.2.2 Training

We finetune VideoSeal using adapted training scripts from Fernandez, Elsahar, et al. 2024. We use the available 256-bits checkpoint as a starting point and finetune it with our new objective. Table B.4 lists the hyperparameters we used. Furthermore, we use effectively 64,000 images from the train set of COCO-2017 during this finetuning.

For robustness purposes, we use data augmentation at training time. We list them in Figure B.5.

B.3 Rotation with secret key

Our system's security relies on an inversible rotation system that accounts for both the latent dimension and unit-norm property. By doing so, we can introduce this system after training and without much overhead.

B.3.1 Sampling Uniformly Random Rotations on $\mathbb{S}^{255}$

To perform uniform random rotations on the surface of a 255-dimensional hypersphere (S^{255}), we require random rotation matrices drawn from a uniform distribution over the Special Orthogonal group SO(256). A distribution is considered uniform on a compact group if it corresponds to the Haar measure, which ensures that the probability of the matrix falling into any given region of the group is proportional to the volume of that region.

A standard and numerically stable method for generating such matrices is based on the QR decomposition of a matrix of random Gaussian variables. As detailed by Mezzadri [1], a crucial correction step is required to ensure the resulting orthogonal matrix is truly Haar-distributed. The algorithm is described below.

Let $N = 256$. The procedure to generate a single random rotation matrix $O \in \mathrm{SO}(256)$ is:

1. **Generate a random matrix from the Ginibre ensemble.** Construct an $N \times N$ matrix A, where each element A_{ij} is an independent and identically distributed (i.i.d.) random variable sampled from the standard normal distribution $\mathcal{N}(0, 1)$.

$$A_{ij} \sim \mathcal{N}(0, 1) \quad \text{for } i, j = 1, \ldots, N. \tag{B.1}$$

2. **Perform the QR decomposition.** Factorize the matrix A into the product of an

0. StableSignature and RAW primarily report image quality via FID/CLIP scores at matched settings rather than a fixed PSNR value; we therefore do not list a PSNR number here.

orthogonal matrix Q and an upper-triangular matrix R.

$$A = QR \tag{B.2}$$

where Q is an orthogonal matrix ($Q^T Q = I_N$) and R is an upper-triangular matrix. Most standard linear algebra libraries provide a numerically stable implementation of this decomposition.

3. **Apply the Haar measure correction.** The QR decomposition is not unique. To ensure a uniform (Haar) distribution for the orthogonal matrix, the signs of the diagonal elements of R must be normalized. We construct a diagonal matrix D for this purpose:

$$D_{ii} = \text{sgn}(R_{ii}) = \begin{cases} 1 & \text{if } R_{ii} \geq 0 \\ -1 & \text{if } R_{ii} < 0 \end{cases} \quad \text{and} \quad D_{ij} = 0 \text{ for } i \neq j. \tag{B.3}$$

The probability of any R_{ii} being exactly zero is negligible for a continuous distribution. The resulting Haar-distributed orthogonal matrix Q' is then computed as:

$$Q' = QD \tag{B.4}$$

This procedure ensures that the resulting upper-triangular matrix $R' = D^{-1}R$ has only positive diagonal elements, which makes the decomposition unique and induces the correct Haar measure on Q' [1].

4. **Ensure the matrix is in SO(N) (a pure rotation).** The matrix Q' is guaranteed to be in the orthogonal group O(N), meaning its determinant is either $+1$ or -1. The Special Orthogonal group SO(N) contains only matrices with determinant $+1$ (pure rotations).

   We thus calculate the determinant of Q'. If $\det(Q') = -1$, we convert the matrix into a pure rotation by flipping the sign of any single column. For instance, we can multiply the first column by -1:

$$O = \begin{cases} Q' & \text{if } \det(Q') = 1 \\ Q' \cdot \text{diag}(-1, 1, \ldots, 1) & \text{if } \det(Q') = -1 \end{cases} \tag{B.5}$$

The resulting matrix O is a uniformly distributed random element of SO(256).

This algorithm ensures the security of our watermark, as we can seed the randomness and use this seed as the secret key.

B.3.2 Impact on performances

To better understand the impact on performances of this security system, we perform a benchmark on different settings. We benchmark four generators of uniformly distributed random rotations on $SO(d)$:

(i) **np_orig**: Gaussian $\rightarrow$ QR $\rightarrow$ diagonal sign-fix via $Q\,\mathrm{diag}(\mathrm{sign}(\mathrm{diag}R))$; then flip one column if $\det(Q)<0$,

(ii) **np_fast**: identical law, but applies the sign-fix by in-place column scaling of Q to avoid the explicit diag matrix and matmul,

(iii) **torch_batched**: same construction using batched QR on GPU,

(iv) **torch_keyed**: GPU, but each batch element uses its own deterministic RNG stream derived from an integer secret key; same Gaussian$\rightarrow$QR pipeline.

All four are Haar on $O(d)$ after the QR sign correction and then restricted to $SO(d)$ by one column flip; only the implementation details differ. We test different keys per batch element because real systems need per-sample determinism and independence without leaking correlation across items. Results are given in Table B.6.

Takeaways We found that CPU `np_fast` trims a few percent vs. `np_orig` by removing the diag matmul. GPU wins decisively from $d\geq 128$, and batching amortizes kernel overhead further. The per-item keyed variant is a touch slower than fully batched because it must spin a generator per element, but it preserves the same distribution while giving per-sample reproducibility and isolation. Under our setting, a realistic estimate to secure a 256 dimensional latent is at most 1.6 ms by using a T4 Colab GPU.

B.4 Qualitative examples

B.4.1 Text data samples

In this section, we provide representative samples for PixMo-Cap et Wikitext dataset.

PixMo-Cap

- "Close-up photograph of a gourmet grilled cheese sandwich that has been artistically sliced in half. Each half reveals a gooey, white cheese center with an enticing stringy, melted cheese bridge connecting them. The sandwich features double"
- "The image depicts a bustling street scene set in a populated area, prominently featuring historical and military elements associated with Checkpoint Charlie. On the left side of the street stands a four-story building labeled "Checkpoint Charlie Wall Museum," "
- "This informative and somewhat humorous poster centers around an image of two tortoises, one seemingly mounting the other for reproductive purposes. The composition has the tortoises positioned centrally, with text above and below them."
- "This photograph, taken in a hurried or unsteady moment, offers a blurry glimpse into a cozy kitchen corner. The focus is on the intersection of counters and cabinets, where white cabinetry adorns both the upper and lower sections."
- "The front page of the clothing website "Apparelix" is designed with a clean, white background. Located in the upper left corner is "USD" in gray text, accompanied by an arrow that likely allows users to change the pricing"

Wikitext

- " Robert Boulter is an English film , television and theatre actor . He had a guest @-@ starring role on the television series The Bill in 2000 . This was followed by a starring role in the play Herons written by Simon Stephens , which was performed in 2001 at the Royal Court Theatre . He had a guest role in the television series Judge John Deed in 2002 . In 2004 Boulter landed a role as " Craig " in the episode " Teddy 's Story " of the television series The Long Firm ; he starred alongside actors Mark Strong and Derek Jacobi . He was cast in the 2005 theatre productions of the Philip"
- " In 756 , Emperor Xuanzong was forced to flee the capital and abdicate . Du Fu , who had been away from the city , took his family to a place of safety and

attempted to join the court of the new emperor (<unk>) , but he was captured by the rebels and taken to Chang 'an . In the autumn , his youngest son , Du <unk> (Baby Bear) , was born . Around this time Du Fu is thought to have contracted malaria . "

- " The naming of these first specimens was disputed , however . Leopold Fitzinger named the animal <unk> in 1837 . In 1841 , English palaeontologist Richard Owen referred to the genus as Labyrinthodon to describe its highly folded or labyrinthine teeth . Owen thought the name Mastodonsaurus " ought not to be retained , because it recalls unavoidably the idea of the mammalian genus Mastodon , or else a <unk> form of the tooth ... and because the second element of the word , saurus , indicates a false affinity , the remains belonging , not to the Saurian , but to the <unk>"

Table B.4 – Watermarker training hyperparameters. Grouped for readability. We follow the naming convention of VideoSeal for the hyperparameters.

Category	Hyperparameter	Value
Data	video dataset	none
	image dataset	coco
Models	extractor_model	convnext_tiny
	embedder_model	unet_small2_yuv_quant
Capacity	hidden_size_multiplier	1
	batch_size	64
	disc_in_channels	1
Schedule	epochs	601
	iter_per_epoch	1000
	scaling_w_schedule	Cosine, scaling_min=0.2, start_epoch=200, epochs=200
	scheduler	CosineLRScheduler, lr_min=1e-6, t_initial=200, warmup_lr_init=1e-8, warmup_t=20
Optimization	optimizer	AdamW, lr=5e-4
	seed	0
Loss & Weights	lambda_dec	1.0
	lambda_d	0.1
	lambda_i	0.1
	perceptual_loss	yuv
	attenuation	jnd_1_1
Augmentations & Disc	num_augs	2
	augmentation_config	configs/all_augs.yaml
	disc_start	50
	scaling_w / scaling_i	1.0 / 1.0

Table B.5 – Data augmentations and their parameter ranges used during training.

Augmentation	**Min**	**Max**
Resize (scale factor)	0.7	1.5
Crop (relative size)	0.5	1.0
Rotate (degrees)	-10	10
Perspective (distortion)	0.1	0.5
JPEG quality	40	60
Gaussian blur (kernel size)	3	17
Median filter (kernel size)	3	3
Brightness factor	0.5	2.0
Contrast factor	0.5	2.0
Saturation factor	0.5	2.0
Hue shift	-0.1	0.1

Table B.6 – Median per-matrix generation time after warm-up. `torch_batched` benefits from GPU and batching; `torch_keyed` adds per-item generator setup for deterministic, secret-key–conditioned streams. All methods are Haar-correct on $SO(d)$.

Dim	**Batch**	**np_orig**	**np_fast**	**torch_batched**	**torch_keyed**
			Median time per matrix (ms)		
32	1	0.315	0.294	0.551	0.592
32	4	0.170	0.161	0.290	0.478
32	16	0.160	0.155	0.119	0.470
32	64	0.162	0.160	0.057	0.474
32	256	0.177	0.165	0.039	0.462
64	1	0.467	0.486	0.552	0.633
64	4	0.468	0.485	0.527	0.593
64	16	0.456	0.464	0.181	0.560
64	64	0.515	0.756	0.091	0.747
64	256	0.471	0.478	0.061	0.547
128	1	2.835	2.626	1.025	1.200
128	4	2.714	2.630	0.722	1.228
128	16	2.649	2.627	0.729	0.956
128	64	2.742	2.655	0.272	0.942
128	256	2.742	2.632	0.133	1.089
256	1	9.104	8.905	1.560	1.775
256	4	9.247	8.830	1.592	1.634
256	16	9.097	8.726	3.588	1.591
256	64	9.526	9.192	1.315	1.650
256	256	9.690	9.481	0.558	1.600

Bibliography

Alexandre Sablayrolles Matthijs Douze, Cordelia Schmid and Hervé Jégou (2019), « Spreading vectors for similarity search », *in*: *ICLR*.

An, Bang et al. (2024), « WAVES: benchmarking the robustness of image watermarks », *in*: *Proceedings of the 41st International Conference on Machine Learning*, ICML'24.

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio (2015), « Neural Machine Translation by Jointly Learning to Align and Translate », *in*: *3rd International Conference on Learning Representations, ICLR 2015*.

Bowman, Samuel R et al. (2016), « Generating sentences from a continuous space », *in*: *20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016*, Association for Computational Linguistics (ACL), pp. 10–21.

Brandolini, Alberto (Jan. 2013), *The bullshit asymmetry: the amount of energy needed to refute bullshit is an order of magnitude bigger than to produce it*, Twitter, URL: `https://twitter.com/ziobrando/status/289635060758507521` (visited on 05/25/2026).

Brooks, Tim, Aleksander Holynski, and Alexei A Efros (2022), « InstructPix2Pix: Learning to Follow Image Editing Instructions », *in*: *CVPR*.

— (2023), « InstructPix2Pix: Learning to Follow Image Editing Instructions », *in*: *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18392–18402, DOI: `10.1109/CVPR52729.2023.01764`.

Brown, Tom B. et al. (2020), *Language Models are Few-Shot Learners*, arXiv: `2005.14165 [cs.CL]`, URL: `https://arxiv.org/abs/2005.14165`.

Bui, Tu, Shruti Agarwal, and John Collomosse (2023), « TrustMark: Universal Watermarking for Arbitrary Resolution Images », *in*: *arXiv* abs/2311.18297.

Bui, Tu, Shruti Agarwal, and John Collomosse (Oct. 2025), « TrustMark: Robust Watermarking and Watermark Removal for Arbitrary Resolution Images », *in*: *IEEE International Conference on Computer Vision (ICCV)*.

Cai, Qi et al. (2025), *HiDream-I1: A High-Efficient Image Generative Foundation Model with Sparse Diffusion Transformer*, arXiv: 2505.22705 [cs.CV], URL: https://arxiv.org/abs/2505.22705.

Carlini, Nicholas and David Wagner (2017), « Adversarial Examples Are Not Easily Detected: Bypassing Ten Detection Methods », *in*: *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pp. 3–14.

Caron, Mathilde et al. (2021), « Emerging Properties in Self-Supervised Vision Transformers », *in*: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9630–9640, DOI: 10.1109/ICCV48922.2021.00951.

Castro, Wes and Zeki Yalniz (Nov. 2025), *Video Invisible Watermarking at Scale*, Engineering at Meta, Accessed: 2026-02-20, URL: https://engineering.fb.com/2025/11/04/video-engineering/video-invisible-watermarking-at-scale/.

Chen, Mo et al. (2008), « Determining image origin and integrity using sensor noise », *in*: *IEEE Transactions on information forensics and security* 3.*1*, pp. 74–90.

Chen, Xinlei et al. (2015), « Microsoft COCO Captions: Data Collection and Evaluation Server », *in*: *arXiv:1504.00325*.

Chouaf, Seloua et al. (2021), « Captioning Changes in Bi-Temporal Remote Sensing Images », *in*: *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*, DOI: 10.1109/IGARSS47720.2021.9554419.

Corvi, Riccardo et al. (2023), « On The Detection of Synthetic Images Generated by Diffusion Models », *in*: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Cox, Ingemar et al. (2007), *Digital watermarking and steganography*, Morgan kaufmann.

Cox, Ingemar J. et al. (2008), « Chapter 2 - Applications and Properties », *in*: *Digital Watermarking and Steganography (Second Edition)*, ed. by Ingemar J. Cox et al., Second Edition, The Morgan Kaufmann Series in Multimedia Information and Systems, Burlington: Morgan Kaufmann, pp. 15–59, ISBN: 978-0-12-372585-1, DOI: https://doi.org/10.1016/B978-012372585-1.50005-X, URL: https://www.sciencedirect.com/science/article/pii/B978012372585150005X.

Deitke, Matt et al. (2024), *Molmo and PixMo: Open Weights and Open Data for State-of-the-Art Vision-Language Models*, arXiv: 2409.17146 [cs.CV], URL: https://arxiv.org/abs/2409.17146.

Deng, Jia et al. (2009), « Imagenet: A large-scale hierarchical image database », *in*: *2009 IEEE conference on computer vision and pattern recognition*, IEEE, pp. 248–255.

Devlin, Jacob et al. (2019), « BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding », *in*: *North American Chapter of the Association for Computational Linguistics*, URL: https://api.semanticscholar.org/CorpusID:52967399.

Dillard, G.M. et al. (2003), « Cyclic code shift keying: a low probability of intercept communication technique », *in*: *IEEE Trans. on Aerospace and Electronic Systems* 39.*3*.

Dosovitskiy, Alexey et al. (2021), « An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale », *in*: *International Conference on Learning Representations*, URL: https://openreview.net/forum?id=YicbFdNTTy.

ElevenLabs (2026), *ElevenLabs: Free AI Voice Generator & Voice Agents Platform*, https://elevenlabs.io/.

Esser, Patrick, Sumith Kulal, et al. (2024), « Scaling rectified flow transformers for high-resolution image synthesis », *in*: *Forty-first international conference on machine learning*.

Esser, Patrick, Robin Rombach, and Björn Ommer (2021), « Taming Transformers for High-Resolution Image Synthesis », *in*: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021*.

European Parliament and Council of the European Union (July 12, 2024), *Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 laying down harmonised rules on artificial intelligence (Artificial Intelligence Act)*, URL: https://eur-lex.europa.eu/eli/reg/2024/1689/oj.

Evennou, Gautier, Vivien Chappelier, et al. (2024), « SWIFT: Semantic Watermarking for Image Forgery Thwarting », *in*: *Proceedings of the IEEE International Workshop on Information Forensics and Security (WIFS)*.

Evennou, Gautier and Ewa Kijak (2026), *The Forensic Cost of Watermark Removal*, arXiv: 2604.25491 [cs.CV], URL: https://arxiv.org/abs/2604.25491.

Fallis, Don (2015), « What Is Disinformation? », *in*: *Library Trends* 63.*3*, pp. 401–426, DOI: 10.1353/lib.2015.0014.

Farid, Hany (2016), *Photo forensics*, MIT press.

Fernandez, Pierre, Guillaume Couairon, et al. (2023a), « The Stable Signature: Rooting Watermarks in Latent Diffusion Models », *in*: *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, IEEE, DOI: 10.1109/ICCV51070.2023.02053, URL: https://doi.org/10.1109/ICCV51070.2023.02053.

— (2023b), « The Stable Signature: Rooting Watermarks in Latent Diffusion Models », *in*: *ICCV*.

Fernandez, Pierre, Hady Elsahar, et al. (2024), « Video Seal: Open and Efficient Video Watermarking », *in*: *arXiv preprint arXiv:2412.09492*.

Fernandez, Pierre, Alexandre Sablayrolles, et al. (2022), « Watermarking Images in Self-Supervised Latent Spaces », *in*: *ICASSP*, DOI: 10.1109/ICASSP43922.2022.9746058.

Fisher, R. A. (1992), « Statistical Methods for Research Workers », *in*: *Breakthroughs in Statistics: Methodology and Distribution*, ed. by Samuel Kotz and Norman L. Johnson, New York, NY: Springer New York, pp. 66–70, ISBN: 978-1-4612-4380-9, DOI: 10.1007/978-1-4612-4380-9_6, URL: https://doi.org/10.1007/978-1-4612-4380-9_6.

France 24 (Jan. 2025), *Social media giant Meta culls US fact-checks ahead of Trump term*, https://www.france24.com/en/americas/20250107-social-media-giant-meta-culls-us-fact-checks-ahead-of-trump-term, accessed 2026-05-27.

Fridrich, J., M. Goljan, and A.C. Baldoza (2000), « New fragile authentication watermark for images », *in*: *ICIP*, vol. 1, pp. 446–449, DOI: 10.1109/ICIP.2000.900991.

Furon, Teddy and Patrick Bas (2008), « Broken Arrows », *in*: *EURASIP Journal on Information Security* 2008, p. ID 597040.

Gabet, Ilan (2026), *Fact-checking of a french politician spreading a video montage during the french presidential campaign*, https://x.com/Ilangabet/status/2059206033730416684.

Gage, Philip (1994), « A new algorithm for data compression », *in*: *The C Users Journal archive* 12, pp. 23–38, URL: https://api.semanticscholar.org/CorpusID:59804030.

Goodfellow, Ian et al. (2014), « Generative Adversarial Nets », *in*: *Advances in Neural Information Processing Systems*.

Graves, Alex (2013), « Generating sequences with recurrent neural networks », *in*: *arXiv preprint arXiv:1308.0850*.

Guo, Zixin, Tzu-Jui Wang, and Jorma Laaksonen (2022), « CLIP4IDC: CLIP for Image Difference Captioning », *in*: *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International*

Joint Conference on Natural Language Processing (Volume 2: Short Papers), URL: https://aclanthology.org/2022.aacl-short.5.

Haeck, Pieter and Eva Hartog (2025), *The Week that AI Deepfakes Hit Europe's Elections*, POLITICO, URL: https://www.politico.eu/article/elections-europe-ai-deepfakes-social-media/ (visited on 05/25/2026).

Al-Haj, Ali (2007), « Combined DWT-DCT digital image watermarking », *in*: *Journal of computer science* 3.*9*, pp. 740–746.

Harris, Zellig S. (1954), « Distributional Structure », *in*: *WORD* 10.*2-3*, pp. 146–162, DOI: 10.1080/00437956.1954.11659520, eprint: https://doi.org/10.1080/00437956.1954.11659520, URL: https://doi.org/10.1080/00437956.1954.11659520.

He, Kaiming et al. (2015), « Deep Residual Learning for Image Recognition », *in*: *CoRR* abs/1512.03385, arXiv: 1512.03385, URL: http://arxiv.org/abs/1512.03385.

Heusel, Martin et al. (2017), « GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium », *in*: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*.

Ho, Jonathan, Ajay Jain, and Pieter Abbeel (2020), « Denoising Diffusion Probabilistic Models », *in*: *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020*, ed. by Hugo Larochelle et al.

Ho, Jonathan and Tim Salimans (2022), *Classifier-Free Diffusion Guidance*, arXiv: 2207.12598 [cs.LG], URL: https://arxiv.org/abs/2207.12598.

Hochreiter, Sepp and Jürgen Schmidhuber (1997), « Long Short-Term Memory », *in*: *Neural Computation*, pp. 1735–1780, ISSN: 0899-7667.

Hodosh, Micah, Peter Young, and Julia Hockenmaier (2013), « Framing Image Description as a Ranking Task: Data, Models and Evaluation Metrics », *in*: *Journal of Artificial Intelligence Research*.

Hosseinzadeh, Mehrdad and Yang Wang (2021), « Image Change Captioning by Learning from an Auxiliary Task », *in*: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2724–2733, DOI: 10.1109/CVPR46437.2021.00275.

Howard, Andrew G. et al. (2017), *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications*.

Hu, Edward J et al. (2022), « LoRA: Low-Rank Adaptation of Large Language Models », *in*: *International Conference on Learning Representations*, URL: https://openreview.net/forum?id=nZeVKeeFYf9.

Hu, Edward J. et al. (2021), *LoRA: Low-Rank Adaptation of Large Language Models*, arXiv: 2106.09685 [cs.CL], URL: https://arxiv.org/abs/2106.09685.

Huang, Qingbao et al. (2022), « Image Difference Captioning With Instance-Level Fine-Grained Feature Representation », *in*: *IEEE Transactions on Multimedia*, DOI: 10.1109/TMM.2021.3074803.

Huang, Yuzhen et al. (2024), « Compression Represents Intelligence Linearly », *in*: *arXiv* abs/2404.09937.

Jhamtani, Harsh and Taylor Berg-Kirkpatrick (2018), « Learning to Describe Differences Between Pairs of Similar Images », *in*: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, URL: https://aclanthology.org/D18-1436.

Jia, Zhaoyang, Han Fang, and Weiming Zhang (Oct. 2021), « MBRS: Enhancing Robustness of DNN-based Watermarking by Mini-Batch of Real and Simulated JPEG Compression », *in*: pp. 41–49, DOI: 10.1145/3474085.3475324.

Katzenbeisser, S. and F.A.P. Petitcolas (1999), *Information Hiding Techniques for Steganography and Digital Watermarking*, Artech House computer security series, Artech House, ISBN: 9781580530354.

Kavukcuoglu, Koray (May 2026), *Découvrez Gemini Omni*, Google Blog, URL: https://blog.google/intl/fr-fr/nouveautes-produits/io-gemini-omni/.

Kim, Hoeseong et al. (2021), « Viewpoint-Agnostic Change Captioning With Cycle Consistency », *in*: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Kingma, Diederik P. and Max Welling (2014), « Auto-Encoding Variational Bayes », *in*: *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, ed. by Yoshua Bengio and Yann LeCun.

Krishna, Ranjay et al. (2017), « Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations », *in*: *International Journal of Computer Vision* 123, pp. 32–73, DOI: 10.1007/s11263-016-0981-7, URL: https://doi.org/10.1007/s11263-016-0981-7.

Labs, Black Forest (2024), *FLUX*, https://github.com/black-forest-labs/flux.

— (2025), *FLUX.2: Frontier Visual Intelligence*, https://bfl.ai/blog/flux-2.

Lavie, Alon (Sept. 2011), *Evaluating the Output of Machine Translation Systems*, Tutorial presented at the 13th Machine Translation Summit (MT Summit XIII), Association for Machine Translation in the Americas (AMTA), Xiamen, China.

Lavie, Alon and Abhaya Agarwal (2007), « Meteor: an automatic metric for MT evaluation with high levels of correlation with human judgments », *in*: *Proceedings of the Second Workshop on Statistical Machine Translation*, Association for Computational Linguistics.

LeCun, Yann, Bernhard Boser, et al. (1989), « Backpropagation applied to handwritten zip code recognition », *in*: *Neural computation* 1.*4*, pp. 541–551.

LeCun, Yann, Corinna Cortes, and CJ Burges (2010), « MNIST handwritten digit database », *in*: *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist* 2.

Ledig, Christian et al. (2017), « Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network », *in*: *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*.

Lewis, Mike et al. (2019), « BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension », *in*: *arXiv preprint arXiv:1910.13461*.

Li, Haodong and Jiwu Huang (2019), « Localization of deep inpainting using high-pass fully convolutional network », *in*: *ICCV*.

Li, Junnan, Dongxu Li, Silvio Savarese, and Steven Hoi (2023), « BLIP-2: bootstrapping language-image pre-training with frozen image encoders and large language models », *in*: *Proceedings of the 40th International Conference on Machine Learning*, ICML'23, Honolulu, Hawaii, USA: JMLR.org.

Li, Junnan, Dongxu Li, Silvio Savarese, and Steven C. H. Hoi (2023), « BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models », *in*: *ICML*.

Li, Junnan, Dongxu Li, Caiming Xiong, et al. (2022), « BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation », *in*: *International Conference on Machine Learning*, URL: https://api.semanticscholar.org/CorpusID:246411402.

Li, Xiujun et al. (2020), « Oscar: Object-Semantics Aligned Pre-training for Vision-Language Tasks », *in*: *ECCV*.

Lin, Chin-Yew (2004), « ROUGE: A Package for Automatic Evaluation of Summaries », *in*: *Text Summarization Branches Out*, Association for Computational Linguistics.

Lin, Tsung-Yi, Michael Maire, Serge Belongie, Lubomir Bourdev, et al. (2015), *Microsoft COCO: Common Objects in Context*, arXiv: 1405.0312 [cs.CV], URL: https://arxiv.org/abs/1405.0312.

Lin, Tsung-Yi, Michael Maire, Serge Belongie, James Hays, et al. (2014), « Microsoft COCO: Common Objects in Context », *in*: *Computer Vision – ECCV 2014*.

Lin, Tsung-Yi et al. (2014), « Microsoft COCO: Common Objects in Context », *in*: *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V*, ed. by David J. Fleet et al., vol. 8693, Lecture Notes in Computer Science, Springer, pp. 740–755, DOI: 10.1007/978-3-319-10602-1_48, URL: https://doi.org/10.1007/978-3-319-10602-1%5C_48.

Lipman, Yaron et al. (2023), « Flow Matching for Generative Modeling », *in*: *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*, OpenReview.net, URL: https://openreview.net/forum?id=PqvMRDCJT9t.

Liu, Chenchen et al. (2026), « Bernini: Latent Semantic Planning for Video Diffusion », *in*: *arXiv preprint arXiv:2605.22344*.

Liu, Chenyang et al. (2023), « Progressive Scale-Aware Network for Remote Sensing Image Change Captioning », *in*: *IGARSS 2023 - 2023 IEEE International Geoscience and Remote Sensing Symposium*.

Liu, Fenglin et al. (2021), « Contrastive Attention for Automatic Chest X-ray Report Generation », *in*: *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*.

Liu, Yepeng et al. (2024), « Image watermarks are removable using controllable regeneration from clean noise », *in*: *arXiv preprint arXiv:2410.05470*.

— (2025), *Image Watermarks are Removable Using Controllable Regeneration from Clean Noise*, arXiv: 2410.05470 [cs.CR], URL: https://arxiv.org/abs/2410.05470.

Liu, Ze et al. (2021), « Swin Transformer: Hierarchical Vision Transformer using Shifted Windows », *in*: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Liu, Zhuang et al. (2022), « A ConvNet for the 2020s », *in*: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Lu, Jiasen et al. (2019), « ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks », *in*: *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, ed. by Hanna M. Wallach et al., pp. 13–23.

Lu, Shilin et al. (2025), « Robust Watermarking Using Generative Priors Against Image Editing: From Benchmarking to Advances », *in*: *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*, OpenReview.net, URL: https://openreview.net/forum?id=16O8GCm8Wn.

Ma, Rui et al. (2022), « Towards Blind Watermarking: Combining Invertible and Non-invertible Mechanisms », *in*: *Proceedings of the 30th ACM International Conference on Multimedia*, pp. 1532–1542.

Maho, Thibault, Teddy Furon, and Erwan Le Merrer (2021), « SurFree: A Fast Surrogate-Free Black-Box Attack », *in*: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, Computer Vision Foundation / IEEE, pp. 10430–10439, DOI: 10.1109/CVPR46437.2021.01029.

Mangrulkar, Sourab et al. (2022), *PEFT: State-of-the-art Parameter-Efficient Fine-Tuning methods*, https://github.com/huggingface/peft.

Marcus, Mitchell P., Mary Ann Marcinkiewicz, and Beatrice Santorini (June 1993), « Building a large annotated corpus of English: the penn treebank », *in*: *Comput. Linguist.* 19.*2*, pp. 313–330, ISSN: 0891-2017.

Meng, Chenlin et al. (2021), « SDEdit: Guided Image Synthesis and Editing with Stochastic Differential Equations », *in*: *International Conference on Learning Representations*, URL: https://api.semanticscholar.org/CorpusID:245704504.

Merity, Stephen et al. (2016), *Pointer Sentinel Mixture Models*, arXiv: 1609.07843 [cs.CL].

Mezzadri, Francesco (2007), *How to generate random matrices from the classical compact groups*, arXiv: math-ph/0609050 [math-ph], URL: https://arxiv.org/abs/math-ph/0609050.

Mirza, Mehdi and Simon Osindero (2014), *Conditional Generative Adversarial Nets*, arXiv: 1411.1784 [cs.LG], URL: https://arxiv.org/abs/1411.1784.

Mitchell, Margaret et al. (2012), « Midge: Generating Image Descriptions From Computer Vision Detections », *in*: *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*.

Mokady, Ron, Amir Hertz, and Amit H. Bermano (2021), *ClipCap: CLIP Prefix for Image Captioning*, arXiv: 2111.09734 [cs.CV], URL: https://arxiv.org/abs/2111.09734.

Mordvintsev, Alexander, Christopher Olah, and Mike Tyka (2015), *DeepDream - a code example for visualizing Neural Networks*, http://googleresearch.blogspot.co.uk/2015/07/deepdream-code-example-for-visualizing.html, Google Research. Archived from the original on July 8, 2015.

Nesti, Federico, Alessandro Biondi, and Giorgio Buttazzo (2023), « Detecting Adversarial Examples by Input Transformations, Defense Perturbations, and Voting », *in*: *IEEE Transactions on Neural Networks and Learning Systems*, DOI: 10.1109/TNNLS.2021.3105238.

Nie, Weili et al. (2022), « Diffusion Models for Adversarial Purification », *in*: *International Conference on Machine Learning (ICML)*.

Nikon Corporation (Jan. 2024), *Nikon and AFP collaborate to verify the implementation of an image provenance function in Nikon cameras*, Nikon Corporation, URL: https://www.nikon.com/company/news/2024/0109_imaging_02.html.

Odena, Augustus, Christopher Olah, and Jonathon Shlens (2017), « Conditional Image Synthesis with Auxiliary Classifier GANs », *in*: *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, ed. by Doina Precup and Yee Whye Teh, Proceedings of Machine Learning Research, PMLR, pp. 2642–2651, URL: http://proceedings.mlr.press/v70/odena17a.html.

Odom, Frank (2022), *clip-text-decoder: Generate text captions for images from their embeddings*, https://github.com/fkodom/clip-text-decoder, GitHub repository.

Oliveira dos Santos, Gabriel, Esther Luna Colombini, and Sandra Avila (2021), « CIDEr-R: Robust Consensus-based Image Description Evaluation », *in*: *Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021)*.

OpenAI (Nov. 2022), *Introducing ChatGPT*, URL: https://openai.com/blog/chatgpt.

Ouyang, Long et al. (2022), « Training language models to follow instructions with human feedback », *in*: *NeurIPS*.

Papineni, Kishore et al. (2002a), « BLEU: a method for automatic evaluation of machine translation », *in*: *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02.

— (2002b), « Bleu: a Method for Automatic Evaluation of Machine Translation », *in*: *Annual Meeting of the Association for Computational Linguistics*, URL: https://api.semanticscholar.org/CorpusID:11080756.

Park, Dong Huk, Trevor Darrell, and Anna Rohrbach (2019), « Robust Change Captioning », *in*: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4623–4632, URL: https://api.semanticscholar.org/CorpusID:119189647.

Pasco, R. (1977), « Source coding algorithms for fast data compression (Ph.D. Thesis abstr.) », *in*: *IEEE Transactions on Information Theory*.

Petrov, Aleksandar et al. (2025), *We Can Hide More Bits: The Unused Watermarking Capacity in Theory and in Practice*, arXiv: 2510.12812 [cs.CR], URL: https://arxiv.org/abs/2510.12812.

Qiu, Yue et al. (2021), « Describing and Localizing Multiple Changes with Transformers », *in*: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1951–1960, URL: https://api.semanticscholar.org/CorpusID:232380403.

Radford, Alec, Jong Wook Kim, et al. (2021), « Learning Transferable Visual Models From Natural Language Supervision », *in*: *International Conference on Machine Learning*, URL: https://api.semanticscholar.org/CorpusID:231591445.

Radford, Alec and Karthik Narasimhan (2018), « Improving Language Understanding by Generative Pre-Training », *in*: URL: https://api.semanticscholar.org/CorpusID:49313245.

Radford, Alec, Jeff Wu, et al. (2019), « Language Models are Unsupervised Multitask Learners », *in*.

Raffel, Colin et al. (Jan. 2020), « Exploring the limits of transfer learning with a unified text-to-text transformer », *in*: *J. Mach. Learn. Res.*

Ren, Shaoqing et al. (2015), « Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks », *in*: *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, ed. by Corinna Cortes et al., pp. 91–99.

Ricker, Jonas, Denis Lukovnikov, and Asja Fischer (2024), « AEROBLADE: Training-Free Detection of Latent Diffusion Images Using Autoencoder Reconstruction Error », *in*: *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Los Alamitos, CA, USA: IEEE Computer Society, DOI: 10.1109/CVPR52733.2024.00872.

Rombach, Robin et al. (2021), « High-Resolution Image Synthesis with Latent Diffusion Models », *in*: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, URL: https://api.semanticscholar.org/CorpusID:245335280.

Ronneberger, Olaf, Philipp Fischer, and Thomas Brox (2015), « U-Net: Convolutional Networks for Biomedical Image Segmentation », *in*: *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015 - 18th International Conference Munich, Germany, October 5 - 9, 2015, Proceedings, Part III*, ed. by Nassir Navab et al., Lecture Notes in Computer Science, Springer, pp. 234–241, DOI: 10.1007/978-3-319-24574-4_28, URL: https://doi.org/10.1007/978-3-319-24574-4%5C_28.

Sander, Tom et al. (2025), « Watermark Anything with Localized Messages », *in*: *International Conference on Learning Representations (ICLR)*.

Schuhmann, Christoph et al. (2022), « LAION-5B: An open large-scale dataset for training next generation image-text models », *in*: *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, URL: https://openreview.net/forum?id=M3Y74vmsMcY.

Sharma, Piyush et al. (2018), « Conceptual Captions: A Cleaned, Hypernymed, Image Alt-text Dataset For Automatic Image Captioning », *in*: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, DOI: 10.18653/v1/P18-1238, URL: https://aclanthology.org/P18-1238.

Shen, Tianxiao et al. (2020), *Educating Text Autoencoders: Latent Representation Guidance via Denoising*, arXiv: 1905.12777 [cs.LG], URL: https://arxiv.org/abs/1905.12777.

Sheynin, Shelly et al. (2023), « Emu Edit: Precise Image Editing via Recognition and Generation Tasks », *in*: *arXiv* abs/2311.10089.

Shi, Xiangxi et al. (2020a), « Finding It at Another Side: A Viewpoint-Adapted Matching Encoder for Change Captioning », *in*: *European Conference on Computer Vision*, URL: https://api.semanticscholar.org/CorpusID:222066860.

— (2020b), « Finding It at Another Side: A Viewpoint-Adapted Matching Encoder for Change Captioning », *in*: *European Conference on Computer Vision*, URL: https://api.semanticscholar.org/CorpusID:222066860.

SHL0MS (May 2026), *Inferior Image*, Post on X (formerly Twitter), https://x.com/SHL0MS/status/2054280631807316329, accessed 2026-05-27.

Sohl-Dickstein, Jascha et al. (2015), « Deep Unsupervised Learning using Nonequilibrium Thermodynamics », *in*: *Proceedings of the 32nd International Conference on Machine*

Learning, ICML 2015, ed. by Francis R. Bach and David M. Blei, JMLR Workshop and Conference Proceedings, URL: http://proceedings.mlr.press/v37/sohl-dickstein15.html.

Song, Jiaming, Chenlin Meng, and Stefano Ermon (2021), « Denoising Diffusion Implicit Models », *in*: *9th International Conference on Learning Representations, ICLR 2021*, URL: https://openreview.net/forum?id=St1giarCHLP.

Song, Kaitao et al. (2019), « MASS: Masked Sequence to Sequence Pre-training for Language Generation », *in*: *International Conference on Machine Learning*, pp. 5926–5936.

Souček, Tomáš et al. (2025), « Transferable Black-Box One-Shot Forging of Watermarks via Image Preference Models », *in*: *Advances in Neural Information Processing Systems*.

Spärck Jones, Karen (1972), « A Statistical Interpretation of Term Specificity and Its Application in Retrieval », *in*: *Journal of Documentation*, DOI: 10.1108/eb026526.

Stability AI (2022), *Stable Diffusion Public Release*, URL: https://stability.ai/news-updates/stable-diffusion-announcement.

Sun, Yaoqi et al. (2022), « Bidirectional difference locating and semantic consistency reasoning for change captioning », *in*: *International Journal of Intelligent Systems*, DOI: https://doi.org/10.1002/int.22821, URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/int.22821.

Tan, Hao et al. (2019), « Expressing Visual Relationships via Language », *in*: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, DOI: 10.18653/v1/P19-1182, URL: https://aclanthology.org/P19-1182.

Tancik, Matthew, Ben Mildenhall, and Ren Ng (June 2020), « StegaStamp: Invisible Hyperlinks in Physical Photographs », *in*: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Los Alamitos, CA, USA: IEEE Computer Society, pp. 2114–2123, DOI: 10.1109/CVPR42600.2020.00219, URL: https://doi.ieeecomputersociety.org/10.1109/CVPR42600.2020.00219.

Taylor, Wilson L. (1953), « "Cloze Procedure": A New Tool for Measuring Readability », *in*: *Journalism & Mass Communication Quarterly* 30, pp. 415–433, URL: https://api.semanticscholar.org/CorpusID:206666846.

Team, OpenMOSS (2026), *MOSS-Audio Technical Report*, https://github.com/OpenMOSS/MOSS-Audio, GitHub repository.

Team, Super Intelligence et al. (2026), *FireRed-Image-Edit-1.0 Technical Report*, arXiv: 2602.13344 [cs.CV], URL: https://arxiv.org/abs/2602.13344.

Touvron, Hugo et al. (2023), *Llama 2: Open Foundation and Fine-Tuned Chat Models*, arXiv: 2307.09288 [cs.CL].

Tu, Yunbin, Liang Li, Li Su, Junping Du, et al. (2023), « Viewpoint-Adaptive Representation Disentanglement Network for Change Captioning », *in*: *IEEE Transactions on Image Processing* 32, pp. 2620–2635, DOI: 10.1109/TIP.2023.3268004.

Tu, Yunbin, Liang Li, Li Su, Ke Lu, et al. (2023), « Neighborhood Contrastive Transformer for Change Captioning », *in*: *IEEE Transactions on Multimedia* 25, pp. 9518–9529, DOI: 10.1109/TMM.2023.3254162.

Tu, Yunbin, Liang Li, Li Su, Zheng-Jun Zha, et al. (2023), « Self-supervised Cross-view Representation Reconstruction for Change Captioning », *in*: *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, DOI: 10.1109/ICCV51070.2023.00263.

Tu, Yunbin, Tingting Yao, et al. (2021), « Semantic Relation-aware Difference Representation Learning for Change Captioning », *in*: *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, ed. by Chengqing Zong et al., Association for Computational Linguistics, DOI: 10.18653/v1/2021.findings-acl.6, URL: https://aclanthology.org/2021.findings-acl.6.

Tzu, Sun (1910), *The Art of War*, trans. by Lionel Giles, London: Luzac & Co.

Utku Celik, M. et al. (2002), « Hierarchical watermarking for secure image authentication with localization », *in*: *IEEE Transactions on Image Processing* 11.*6*, pp. 585–595, DOI: 10.1109/TIP.2002.1014990.

Valmeekam, Chandra Shekhara Kaushik et al. (2023a), « LLMZip: Lossless Text Compression using Large Language Models », *in*: *arXiv:2306.04050*.

— (2023b), *LLMZip: Lossless Text Compression using Large Language Models*, arXiv: 2306.04050 [cs.IT], URL: https://arxiv.org/abs/2306.04050.

Vasu, Pavan Kumar Anasosalu et al. (June 2025), « FastVLM: Efficient Vision Encoding for Vision Language Models », *in*: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Vaswani, Ashish et al. (2017), « Attention is all you need », *in*: *Advances in neural information processing systems*, pp. 5998–6008.

Vinyals, Oriol et al. (2015), « Show and tell: A neural image caption generator », *in*: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Wang, Peng et al. (2022), « OFA: Unifying Architectures, Tasks, and Modalities Through a Simple Sequence-to-Sequence Learning Framework », *in*: *Proceedings of the 39th International Conference on Machine Learning*, URL: https://proceedings.mlr.press/v162/wang22al.html.

Wang, Weihan et al. (2024), *CogVLM: Visual Expert for Pretrained Language Models*, arXiv: 2311.03079 [cs.CV].

Wang, Xintao et al. (2018), « ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks », *in*: *Computer Vision - ECCV 2018 Workshops - Munich, Germany, September 8-14, 2018, Proceedings, Part V*, ed. by Laura Leal-Taixé and Stefan Roth.

Wang, Zhendong et al. (2023), « DIRE for Diffusion-Generated Image Detection », *in*: *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22388–22398, DOI: 10.1109/ICCV51070.2023.02051.

Warner, Benjamin et al. (2024), *Smarter, Better, Faster, Longer: A Modern Bidirectional Encoder for Fast, Memory Efficient, and Long Context Finetuning and Inference*, arXiv: 2412.13663 [cs.CL], URL: https://arxiv.org/abs/2412.13663.

Wen, Yuxin et al. (2023), « Tree-Rings Watermarks: Invisible Fingerprints for Diffusion Images », *in*: *Advances in Neural Information Processing Systems*, ed. by A. Oh et al., vol. 36, Curran Associates, Inc., pp. 58047–58063, URL: https://proceedings.neurips.cc/paper_files/paper/2023/file/b54d1757c190ba20dbc4f9e4a2f54149-Paper-Conference.pdf.

Wicks, Buffy, Josh Becker, and Rick Chavez Zbur (2025), *AB 853: California AI Transparency Act*, Chapter 674, Statutes of 2025; amends the California AI Transparency Act (SB 942). Chaptered 13 October 2025, URL: https://calmatters.digitaldemocracy.org/bills/ca_202520260ab853 (visited on 06/23/2026).

Woo, Sanghyun et al. (2023), « ConvNeXt V2: Co-designing and Scaling ConvNets with Masked Autoencoders », *in*: *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, IEEE, pp. 16133–16142, DOI: 10.1109/CVPR52729.2023.01548, URL: https://doi.org/10.1109/CVPR52729.2023.01548.

Wu, Chenfei et al. (2025), *Qwen-Image Technical Report*, arXiv: 2508.02324 [cs.CV], URL: https://arxiv.org/abs/2508.02324.

Wu, Jinjian et al. (2017), « Enhanced Just Noticeable Difference Model for Images With Pattern Complexity », *in*: *IEEE Transactions on Image Processing*, DOI: 10.1109/TIP.2017.2685682.

Wu, Yue, Wael Abd-Almageed, and Prem Natarajan (2018), « BusterNet: Detecting copy-move image forgery with source/target localization », *in*: *ECCV*.

Wu, Yue, Wael AbdAlmageed, and Premkumar Natarajan (2019), « ManTra-Net: Manipulation tracing network for detection and localization of image forgeries with anomalous features », *in*: *CVPR*.

Xian, Xun et al. (2024), « RAW: A Robust and Agile Plug-and-Play Watermark Framework for AI-Generated Images with Provable Guarantees », *in*: *Advances in Neural Information Processing Systems*, ed. by A. Globerson et al., vol. 37, Curran Associates, Inc., pp. 132077–132105, DOI: 10.52202/079017-4198, URL: https://proceedings.neurips.cc/paper_files/paper/2024/file/ee62ab636066cf45a27246acca9545b7-Paper-Conference.pdf.

Xie, Enze et al. (2025), « SANA: Efficient High-Resolution Text-to-Image Synthesis with Linear Diffusion Transformers », *in*: *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*.

Xu, Kelvin et al. (2015), « Show, attend and tell: Neural image caption generation with visual attention », *in*: *International conference on machine learning*, PMLR, pp. 2048–2057.

Xu, Weilin, David Evans, and Yanjun Qi (2018), « Feature Squeezing: Detecting Adversarial Examples in Deep Neural Networks », *in*: *25th Annual Network and Distributed System Security Symposium, NDSS 2018, San Diego, California, USA, February 18-21, 2018*.

Yang, Suorong et al. (2023), « Image Data Augmentation for Deep Learning: A Survey », *in*: *arXiv* abs/2204.08610.

Yao, Linli, Weiying Wang, and Qin Jin (2022), « Image Difference Captioning with Pre-training and Contrastive Learning », *in*: *AAAI Conference on Artificial Intelligence*.

Zhang, Hanlin et al. (2024), « Watermarks in the Sand: Impossibility of Strong Watermarking for Generative Models », *in*: *Forty-first International Conference on Machine Learning*.

Zhang, Kai et al. (2023a), « MagicBrush: A Manually Annotated Dataset for Instruction-Guided Image Editing », *in*: *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

— (2023b), « MagicBrush: A Manually Annotated Dataset for Instruction-Guided Image Editing », *in*: *Thirty-seventh Conference on Neural Information Processing Systems*

Datasets and Benchmarks Track (NeurIPS), URL: https://openreview.net/forum?id=ZsDB2GzsqG.

— (2023c), « MagicBrush: A Manually Annotated Dataset for Instruction-Guided Image Editing », *in*: *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, URL: https://openreview.net/forum?id=ZsDB2GzsqG.

Zhang, Kevin Alex et al. (2019), « Robust Invisible Video Watermarking with Attention », *in*: *ArXiv* abs/1909.01285, URL: https://api.semanticscholar.org/CorpusID:202540549.

Zhang, Lvmin, Anyi Rao, and Maneesh Agrawala (2023), « Adding Conditional Control to Text-to-Image Diffusion Models », *in*: *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, IEEE, pp. 3813–3824, DOI: 10.1109/ICCV51070.2023.00355, URL: https://doi.org/10.1109/ICCV51070.2023.00355.

Zhang, Richard et al. (2018), « The Unreasonable Effectiveness of Deep Features as a Perceptual Metric », *in*: *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*.

Zhang, Susan, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, et al. (2022), *OPT: Open Pre-trained Transformer Language Models*, arXiv: 2205.01068 [cs.CL], URL: https://arxiv.org/abs/2205.01068.

Zhang, Susan, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona T. Diab, et al. (2022), « OPT: Open Pre-trained Transformer Language Models », *in*: *arXiv* abs/2205.01068.

Zhang, Xuanyu et al. (2024), « Editguard: Versatile image watermarking for tamper localization and copyright protection », *in*: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11964–11974.

Zhou, Peng et al. (2018), « Learning rich features for image manipulation detection », *in*: *CVPR*.

Zhu, Jiren et al. (2018a), « HiDDeN: Hiding Data With Deep Networks », *in*: *Computer Vision – ECCV 2018: 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XV*, Munich, Germany: Springer-Verlag, pp. 682–697, ISBN: 978-3-030-01266-3, DOI: 10.1007/978-3-030-01267-0_40, URL: https://doi.org/10.1007/978-3-030-01267-0_40.

Zhu, Jiren et al. (2018b), « HiDDeN: Hiding Data With Deep Networks », *in*: *ECCV*.

COLLEGE DOCTORAL BRETAGNE
MATHS, TELECOMS
INFORMATIQUE, SIGNAL
SYSTEMES, ELECTRONIQUE

Université de Rennes

Titre : Détection et explication d'altérations sémantiques dans les contenus multi-médias

Mot clés : tatouage d'image, modèles vision-langage, détection de manipulation, légendage de différences, IA générative, intégrité des médias

Résumé : La prolifération des modèles génératifs a transformé la manipulation d'images en menace de masse pour l'information. Cette thèse traite la détection et l'explication d'altérations sémantiques par des approches passives et actives, en plaçant la sémantique au cœur de la chaîne. BLIP2IDC adapte BLIP2 au légendage de différences : en encodant les paires au niveau pixel plutôt que par fusion, il dépasse l'état de l'art sur données réelles à faible coût de calcul, complété par une augmentation synthétique par édition pour pallier le manque de données. Lorsque l'image originale est indisponible, nous recourons au tatouage sémantique. SWIFT compresse une légende via LLMZip et la transmet par modulation TCCSK sur 72 bits au plus, avec une métrique de confiance sur l'authenticité. LatentSeal lève ce verrou de capacité en encodant la phrase dans un vecteur latent unitaire de dimension 256, embarqué via un réseau dérivé de VideoSeal et sécurisé par rotation aléatoire. Enfin, la détection d'effaçage de tatouage montre que les attaques modernes laissent des traces statistiques distinctives, avec un TPR moyen de 0,92 à FPR de 10^{-3}.

Title: Detection and Explanation of Semantic Alterations in Multimedia Content

Keywords: image watermarking, vision-language models, manipulation detection, image difference captioning, generative AI, media integrity

Abstract: The rise of generative models capable of producing realistic images has turned visual manipulation into a mass-scale threat to information integrity. This thesis tackles the detection and explanation of semantic alterations through both passive and active approaches, placing semantics at the core of the pipeline. We first introduce BLIP2IDC, an adaptation of BLIP2 to image difference captioning. By encoding image pairs at the pixel level rather than through a two-stream scheme, BLIP2IDC outperforms the state of the art on real-world data at low computational cost. We further propose a synthetic augmentation strategy based on text-guided image editing, tackling the data scarcity. We then address the case where the pristine image is unavailable, through semantic watermarking. SWIFT compresses a caption via LLMZip and transmits it through TCCSK modulation in a 72-bit payload, paired with a confidence metric for authenticity decisions. LatentSeal breaks this capacity ceiling by encoding sentences into a 256-D unit-norm latent vector, embedded via an adapted VideoSeal network and secured by random rotation. Finally, we study the robustness of the ecosystem: Watermark Removal Detection shows that modern removal attacks leave distinctive statistical traces, detectable with an average TPR of 0.92 at FPR 10^{-3}.