

Text to Image

▼ Aperçu

▼ Introduction

Un modèle de génération de texte à image utilise l'apprentissage automatique pour créer des images à partir de descriptions textuelles. Il repose sur des réseaux de neurones profonds pour comprendre les liens entre texte et visuels, produisant des images alignées avec les descriptions.

Applications

- **Industries Créatives** : Génération rapide d'idées visuelles (ex : designs de mode).
- **Marketing** : Création de visuels pour campagnes à partir de descriptions de produits.
- **E-commerce** : Amélioration des listes de produits avec des images générées.
- **Éducation** : Visualisation de concepts complexes pour faciliter l'apprentissage.
- **Expériences Immersives** : Amélioration des interactions avec des chatbots et assistants virtuels.

▼ Évolution des modèles de génération d'images

1. Systèmes Précoces (Années 2000) :

- Axés sur le collage et la récupération d'images basées sur des mots-clés, sans création de contenu nouveau.

2. Ère de l'Apprentissage Profond (Années 2010) :

- **2012** : Les réseaux de neurones convolutifs (CNN) ont révolutionné les tâches visuelles.
- **2015** : alignDRAW a introduit la génération d'images basse résolution à partir de texte.

3. GANs (2016) :

- Ont permis la synthèse d'images réalistes, mais avec des difficultés de cohérence et de haute résolution.

4. Modèles Révolutionnaires (2021-2022) :

- **2021** : DALL-E a combiné génération de texte et d'images.
- **2022** : Stable Diffusion a rendu la génération d'images de haute qualité et open-source largement accessible.

5. Avancées Récentes (2023-Présent) :

- Des modèles comme DALL-E 3 et Imagen ont atteint un niveau quasi-photoréaliste.
- L'émergence de technologies texte-à-véo et multimodales a élargi les applications de l'IA générative.

▼ Modèles populaires de génération de texte à partir d'images

• DALL-E (OpenAI)

- **2021** : Pionnier de la génération de texte à partir d'images avec une compréhension du texte basée sur CLIP.
- **DALL-E 2** : Amélioration de la résolution, du réalisme et de l'alignement du texte.
- **DALL-E 3** : Amélioration de la compréhension du contexte et de la fidélité des messages.

• Diffusion stable (Stability AI)

- **2022** : Modèle open-source basé sur la diffusion, connu pour ses résultats de haute qualité et son accessibilité.
- Excelle dans la génération d'images artistiques et photoréalistes.

• Imagen (Google)

- Combine de puissants encodeurs de texte avec un accent sur le photoréalisme et les détails fins.
- Remarquable pour sa grande fidélité et son alignement sur les invites.















• MidJourney

- Populaire pour la génération d'images stylisées et artistiques avec une gestion intuitive des invites.
- Largement utilisé dans les industries créatives et les médias sociaux.

▼ **Benchmark**

Leaderboard
Image Arena
Personal Arena Leaderboard

TEXT TO IMAGE MODEL LEADERBOARD

CREATOR	NAME	ARENA ELO	ARENA WIN RATE	# SELECTIONS
 Midjourney	Midjourney v6	1171	71%	13129
 Stability.ai	Stable Diffusion 3	1154	68%	10153
 Playground AI	Playground v2.5	1099	61%	11475
 OpenAI	DALLE 3 HD	1094	61%	11349
 OpenAI	DALLE 3	1073	58%	10771
 Stability.ai	Stable Diffusion 3 Turbo	1038	52%	7621
 Stability.ai	Stable Diffusion 3 Medium	1033	53%	2997
 Stability.ai	Stable Diffusion 1.6	1028	51%	7475
 Stability.ai	SDXL Lightning	1009	49%	9033
 Amazon	Amazon Titan G1	1008	49%	4244
 Stability.ai	Stable Diffusion XL 1.0	980	45%	8357
 OpenAI	DALLE 2	819	24%	4440
 Stability.ai	Stable Diffusion 2.1	819	24%	4395
 Stability.ai	Stable Diffusion 1.5	674	16%	829

<https://www.aibase.com/news/9825>

▼ **Pourquoi les modèles de stable diffusion**

1. **Open Source** : Librement accessible, permettant la personnalisation et l'intégration dans diverses applications.
2. **Sorties de haute qualité** : Fournit des images photoréalistes et artistiques avec un excellent alignement sur les invites.
3. **Efficacité** : Fonctionne dans un **espace latent** pour un traitement plus rapide et des coûts de calcul réduits.
4. **Flexibilité** :
 - Prise en charge de différents styles, du photoréalisme à l'art abstrait.
 - Fonctionne avec des outils supplémentaires pour la peinture et l'édition d'images.
5. **Accessibilité** : Fonctionne sur du matériel grand public, démocratisant ainsi la créativité alimentée par l'IA.
6. **Communauté dynamique** : Le développement actif et les contributions de la communauté améliorent les fonctionnalités et la facilité d'utilisation.

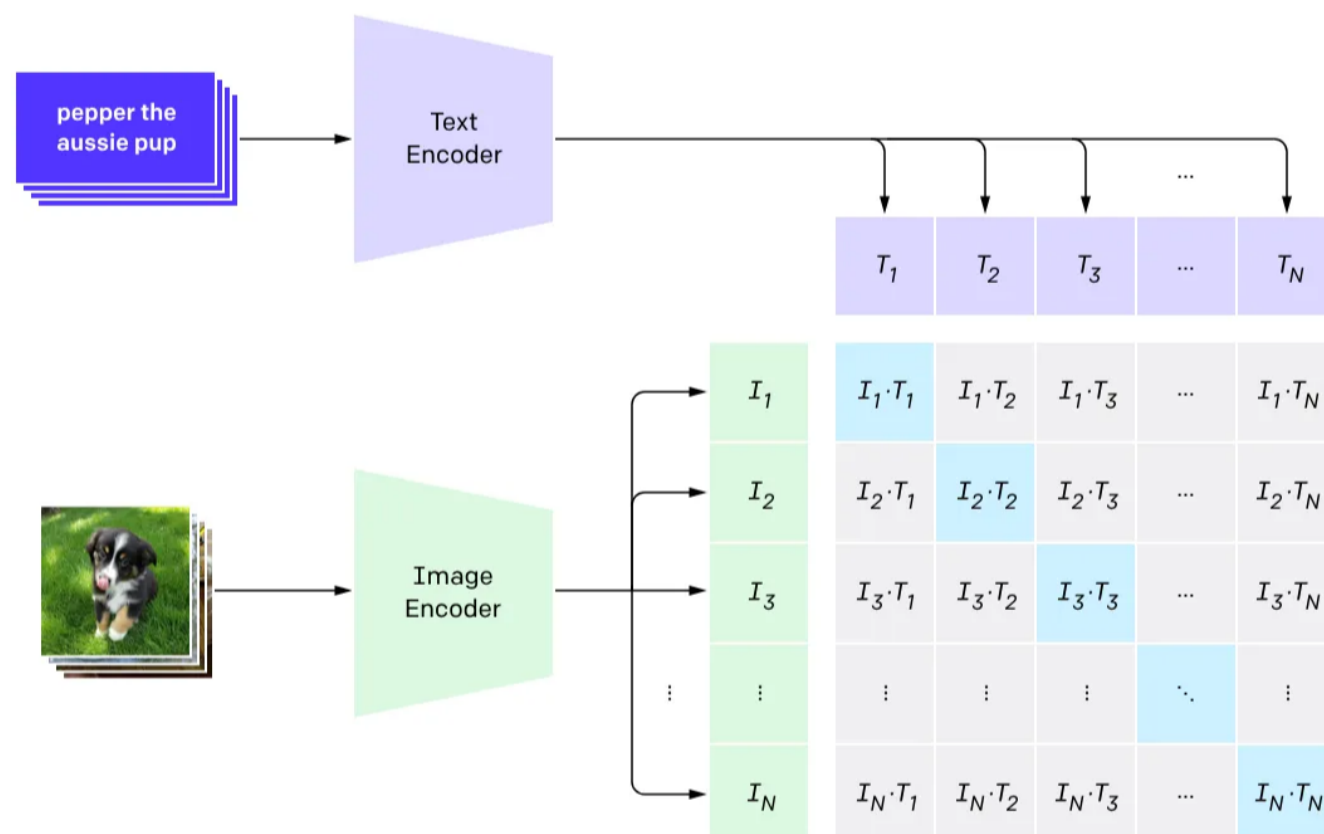
▼ **Comment la génération de texte à partir d'images fonctionne-t-elle grâce à l'architecture de diffusion ?**

▼ **Grandes questions à résoudre**

- Méthode d'apprentissage permettant de générer de nouveaux éléments à partir de nombreux exemples → Forward/Reverse Diffusion
- Méthode de compression des images (pour accélérer la formation et la génération) → Latent Space/VAE
- Moyen de lier le texte et les images → CLIP
- Moyen d'ajouter un biais inductif lié à l'image → U-Net & Cross attention

▼ Étape 1 : Encodage du Texte avec CLIP

- **Entrée** : L'utilisateur fournit un **prompt texte**, par exemple : "**A serene lake surrounded by snow-capped mountains at sunrise**"
 - **Objectif** : Traduire le sens du texte en un format numérique que le modèle peut comprendre et utiliser pour générer une image.
- **Modèle CLIP** :
 - CLIP (Contrastive Language-Image Pre-training) est utilisé pour traiter le **prompt texte**.
 - CLIP a été entraîné sur un grand ensemble de données de paires texte-image, lui permettant de créer un **embedding texte** (une représentation vectorielle) qui capture la signification sémantique du prompt.
- **Pourquoi CLIP est Important** :
 - L'embedding texte guide le processus de génération d'image en conditionnant le modèle sur ce qu'il doit produire.
 - L'objectif est que le modèle génère une image alignée avec le **contenu sémantique** du prompt.
- **Fonctionnement de CLIP** :



CLIP relie le texte et les images par le biais d'encastresments partagés, ce qui améliore la compréhension et permet d'effectuer des tâches telles que la classification sans prise de vue.

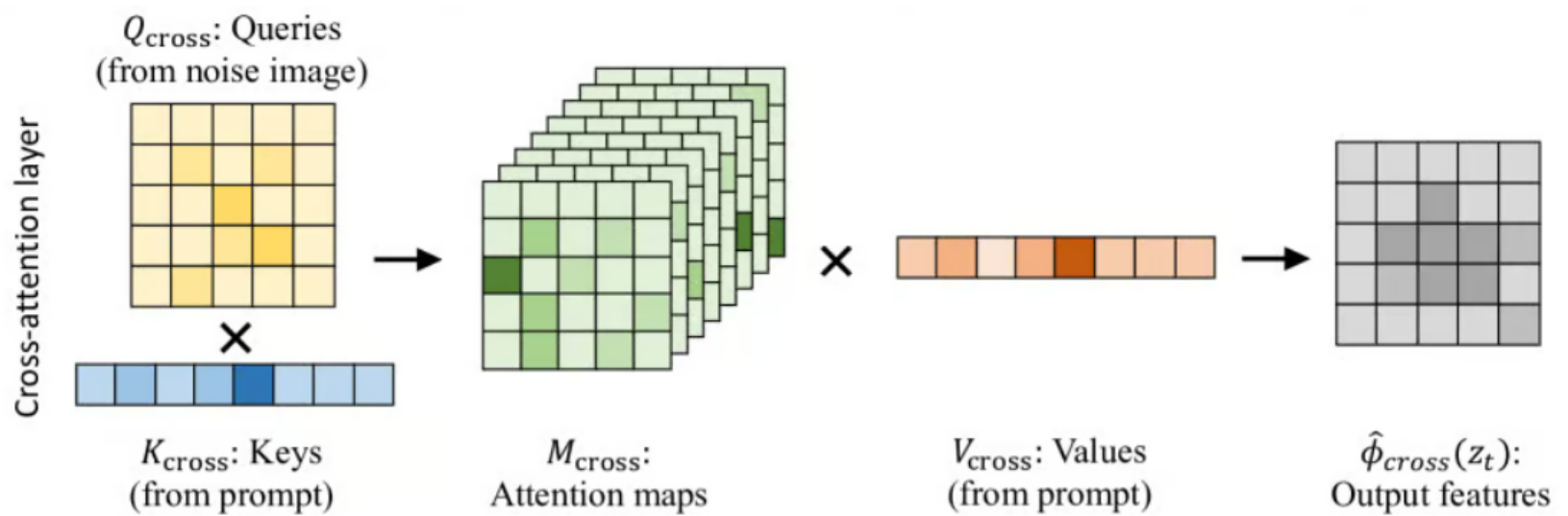
1. Encodeur de Texte :

- Tokenise le texte d'entrée et le convertit en embeddings vectoriels de haute dimension à l'aide d'un Transformer.
- Exemple : "forêt sereine avec cascade au lever du soleil" est encodé pour capturer sa signification.

2. Encodeur d'Image :

- Traite les images en patches et les convertit en embeddings à l'aide d'un Vision Transformer (ViT).
- Crée des représentations mathématiques des caractéristiques visuelles.

3. Mécanisme de Cross-Attention :

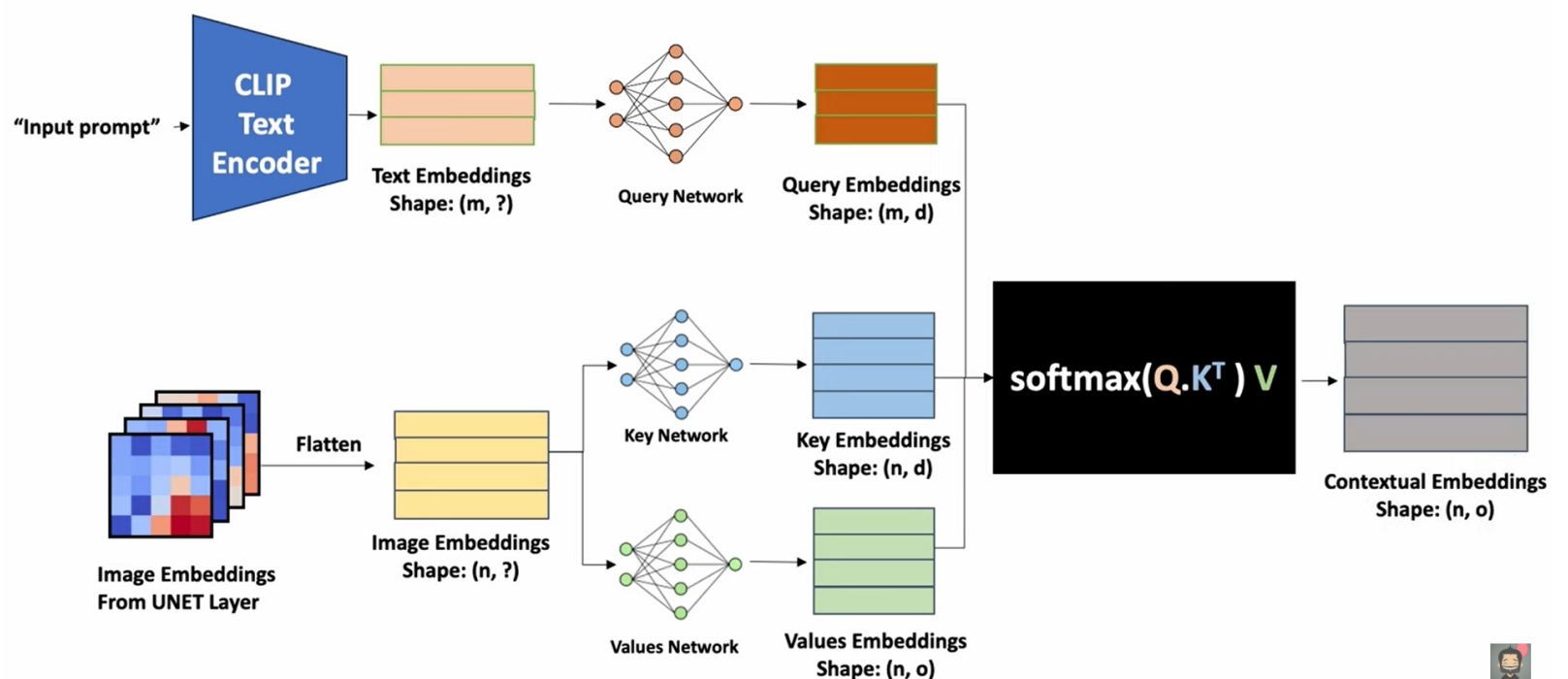


- **Augmentation de la Similarité** : Augmente la similarité pour les paires correspondantes (ex : image de forêt avec "forêt sereine").
- **Réduction du Contraste** : Diminue la similarité pour les paires non correspondantes (ex : skyline urbain avec "forêt sereine").

4. Espace d'Embedding Partagé :

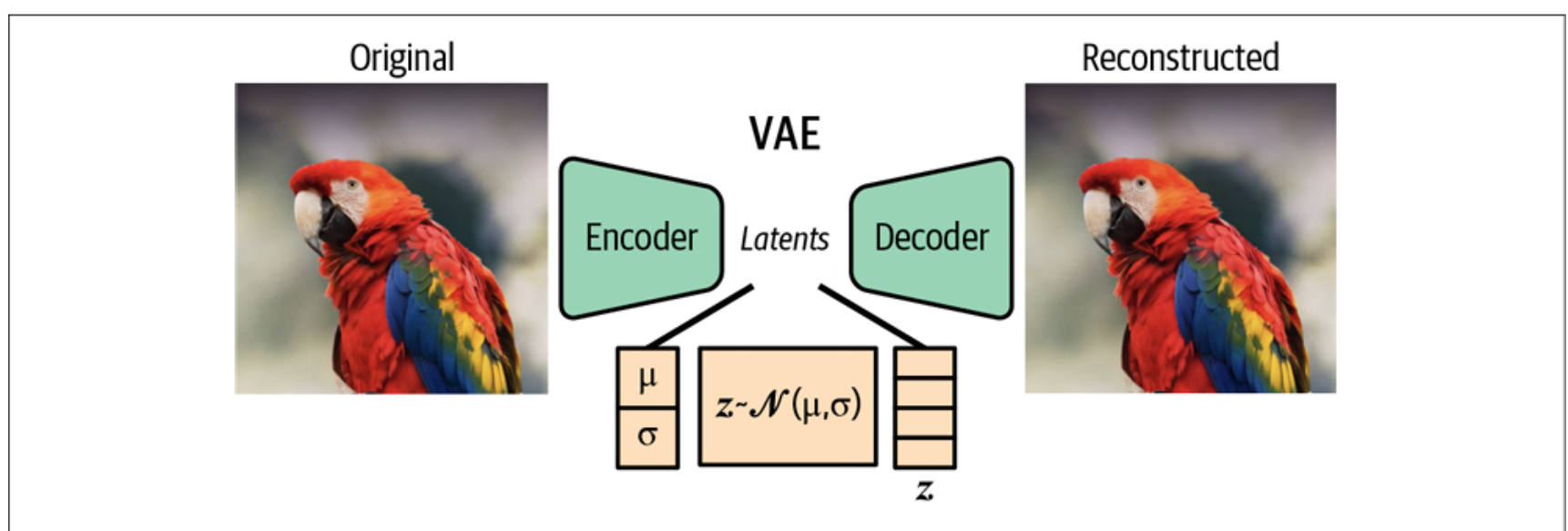
- Les embeddings texte et image sont projetés dans un espace latent commun pour une comparaison directe.

TEXT TO IMAGE ATTENTION LAYER



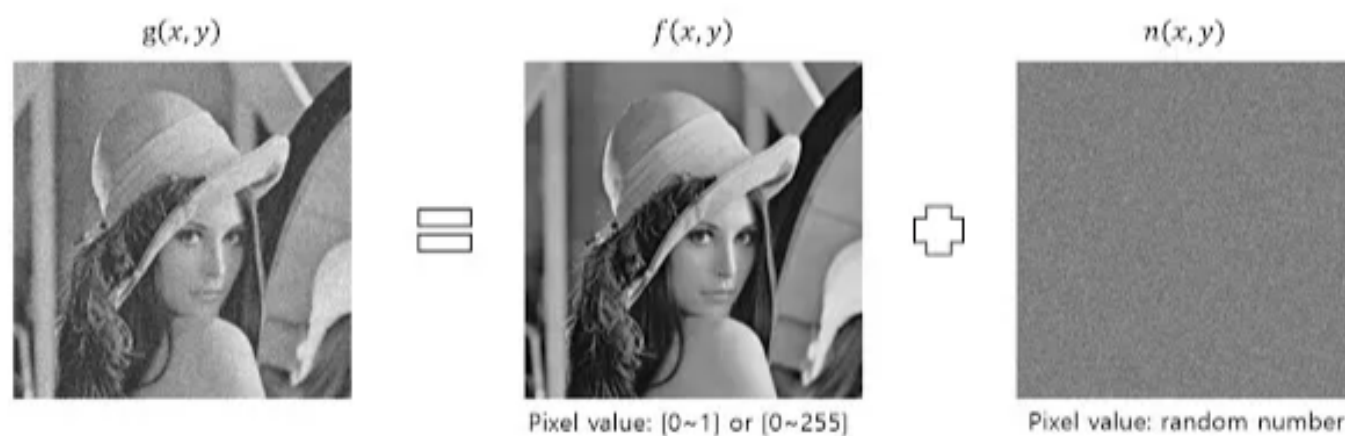
▼ Étape 2 : Initialisation de l'Espace Latent (Bruit Aléatoire)

- Espace Latent :



- Au lieu de travailler directement dans l'espace pixel (ce qui est coûteux en calcul), **Stable Diffusion** opère dans un **espace latent** compressé.
- L'**espace latent** est une représentation de dimension réduite des images qui capture les caractéristiques essentielles tout en omettant les détails superflus.
- **Pourquoi l'Espace Latent :**
 - Travailler dans l'espace latent permet au modèle d'être plus efficace, en se concentrant sur la **structure de haut niveau** de l'image plutôt que sur les détails au niveau des pixels.
 - Cela rend la génération d'images plus rapide et réalisable sur le plan computationnel, en particulier pour les hautes résolutions.
- **Initialisation par Bruit Aléatoire :**

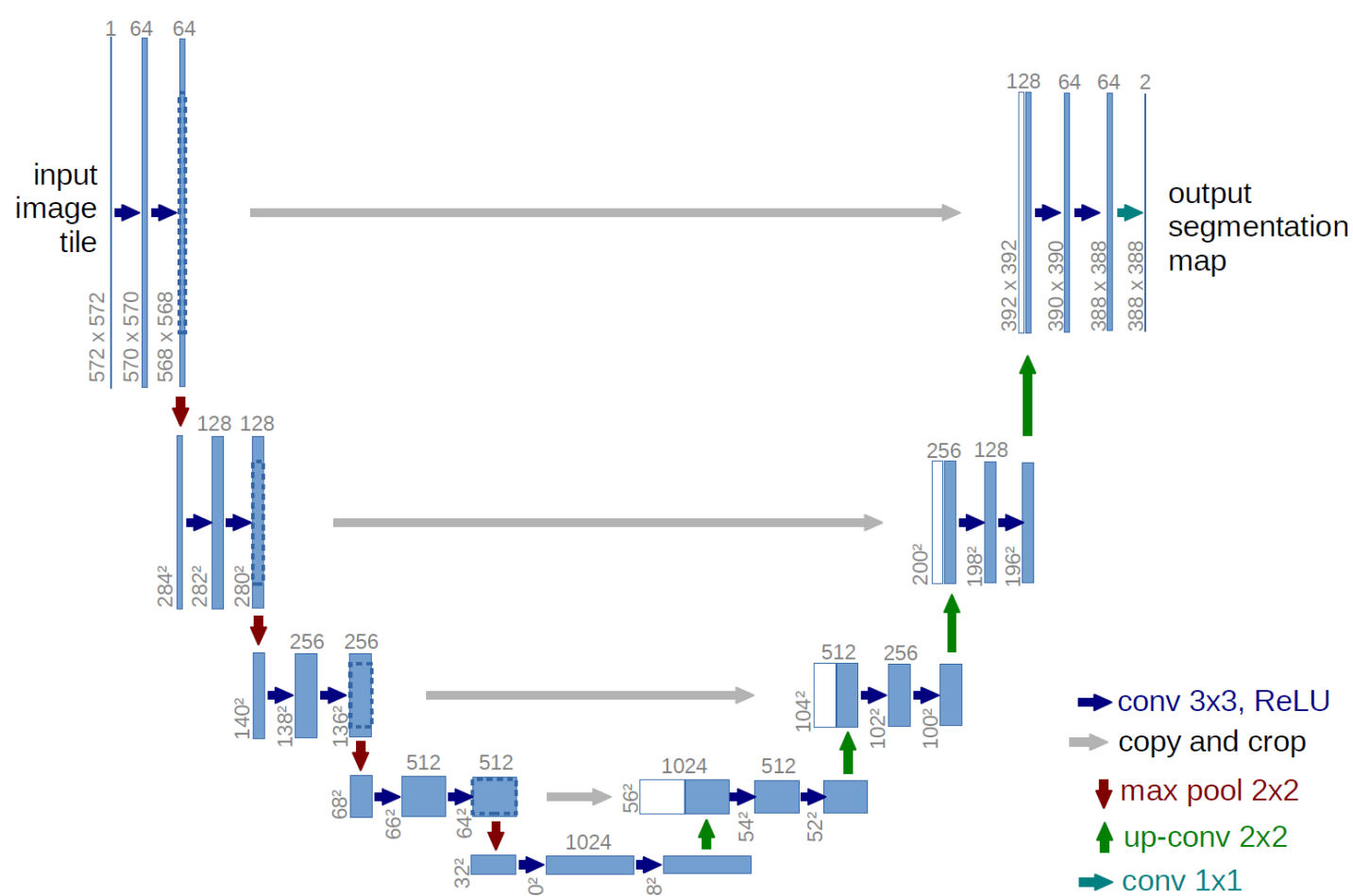
$$g(x, y) = f(x, y) + n(x, y)$$



- Le modèle commence le processus de génération d'image en initialisant une **image latente** avec du **bruit aléatoire**.
- Ce bruit latent sert de point de départ à partir duquel le modèle affine itérativement l'image.

▼ Étape 3 : Dé-bruitage avec U-Net et Cross-Attention (Generation d'image)

U-Net



- **Architecture** : U-Net est un réseau de neurones convolutifs conçu pour la segmentation d'images, avec une architecture en forme de U composée de deux parties principales :

1. Chemin de Contraction (Encodeur) :

- Capture le contexte en réduisant la résolution spatiale de l'image et en augmentant la profondeur des cartes de caractéristiques via des convolutions, des activations ReLU et du max pooling.
- Identifie les caractéristiques pertinentes de l'image, apprenant des représentations de plus en plus abstraites.

2. Chemin d'Expansion (Décodeur) :

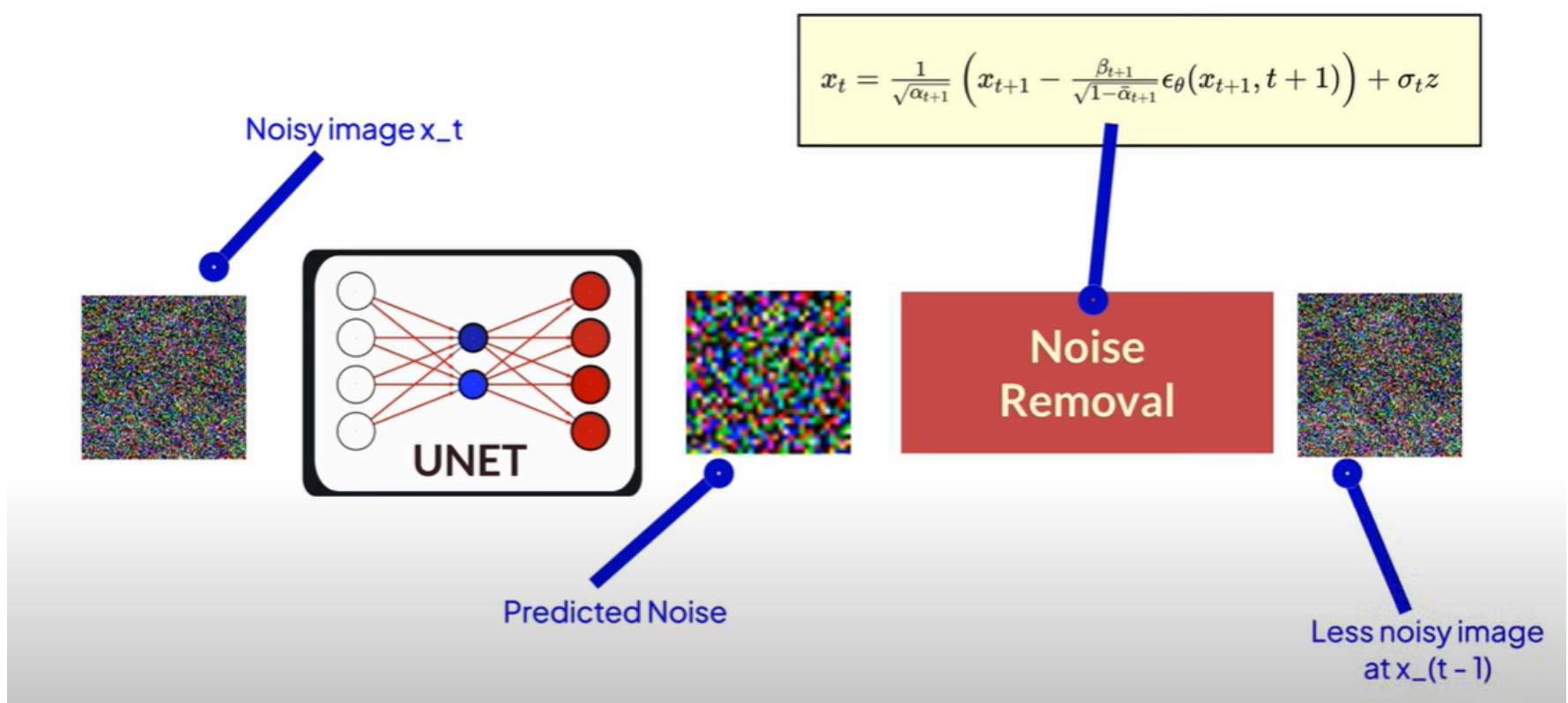
- Rééchantillonne les cartes de caractéristiques pour reconstruire la résolution spatiale perdue, en utilisant des up-convolutions et des connexions skip pour conserver les informations spatiales importantes.
- Les connexions skip combinent les détails de bas niveau de l'encodeur avec le contexte de haut niveau du décodeur, améliorant la précision de localisation.

Rôle de U-Net dans le Débruitage (Diffusion Inverse)



1. **Préservation du Contexte** : Les connexions skip préservent les détails spatiaux pendant le débruitage, aidant à reconstruire les caractéristiques fines de l'image finale.
2. **Prédiction du Bruit** : U-Net prédit le bruit dans l'image latente à chaque pas de temps. L'encodeur capture la structure du bruit, tandis que le décodeur affine l'image.
3. **Raffinement Itératif** : Sur plusieurs étapes, U-Net supprime progressivement le bruit, guidé par l'embedding texte (via cross-attention). L'architecture en U garantit que le contexte global et les détails locaux sont conservés.

Processus de Diffusion Inverse (Débruitage)

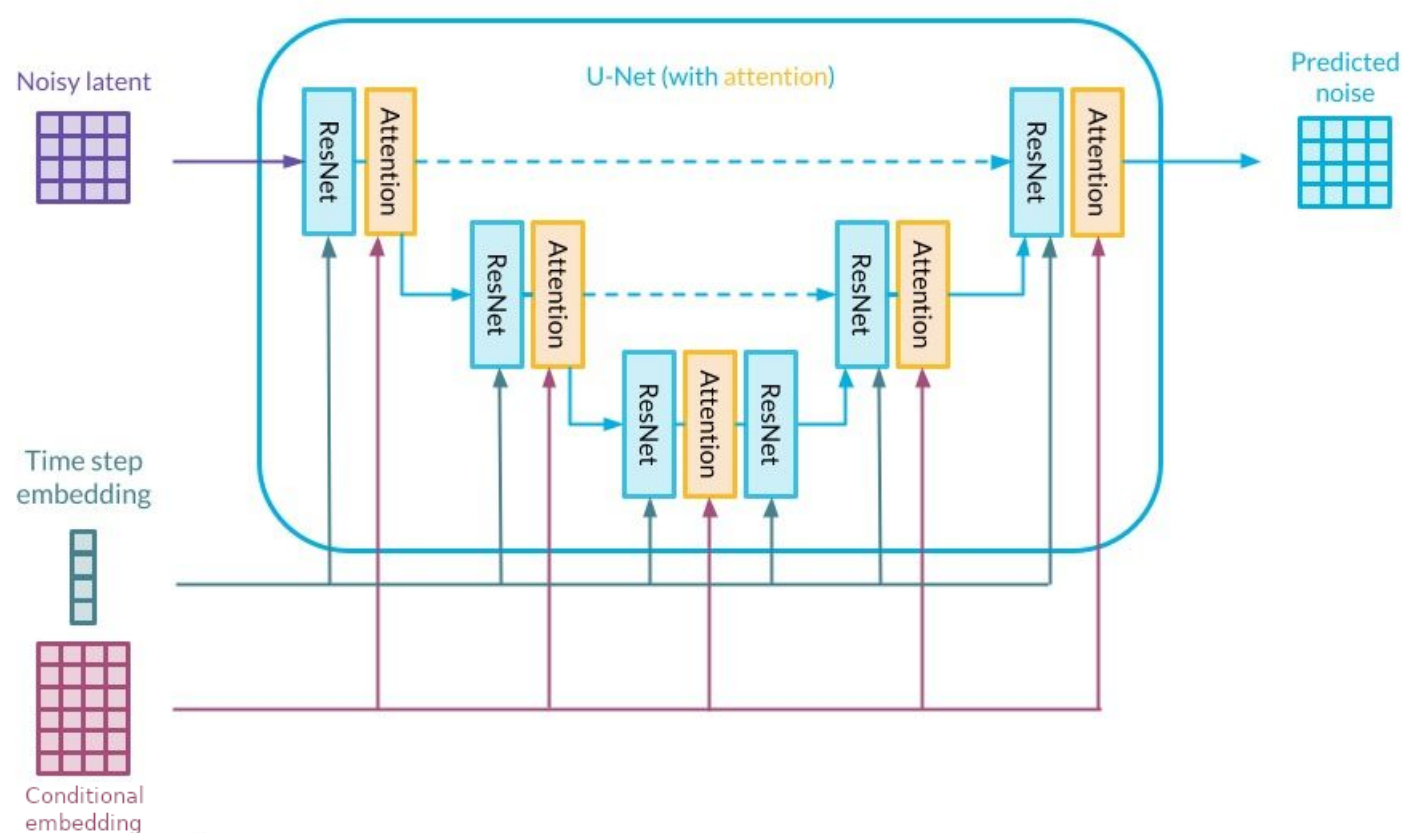


- **Objectif** : Transformer une image latente bruitée en une image claire sur **T étapes**.
- **Rôle de U-Net** : Prédit et supprime le bruit de manière itérative, affinant l'image à chaque étape.
- **Résultat** : Chaque étape rapproche l'image d'un résultat de haute qualité correspondant au prompt texte.

Mécanisme de Cross-Attention

- **Embedding Texte** : Le prompt (ex : "cascade") est converti en une forme numérique (via CLIP) pour guider le débruitage.
- **Query, Key, Value** :
 - **Query** : État actuel de l'image bruitée.
 - **Key/Value** : Embedding texte, se concentrant sur les caractéristiques pertinentes au prompt.
- **Fonction** : Priorise les zones de l'image alignées avec le texte, améliorant les détails comme "cascade" ou "forêt".

Processus Itératif de Diffusion Inverse



1. **Point de Départ** : Commence avec du bruit aléatoire dans l'espace latent.
2. **Première Itération** :
 - U-Net prédit et supprime le bruit.
 - L'embedding texte guide le raffinement en fonction du prompt.
3. **Itérations Suivantes** :
 - Le modèle affine l'image sur plusieurs étapes.
 - La cross-attention assure l'alignement avec le prompt.
4. **Itération Finale** : L'image latente devient une représentation claire du prompt.



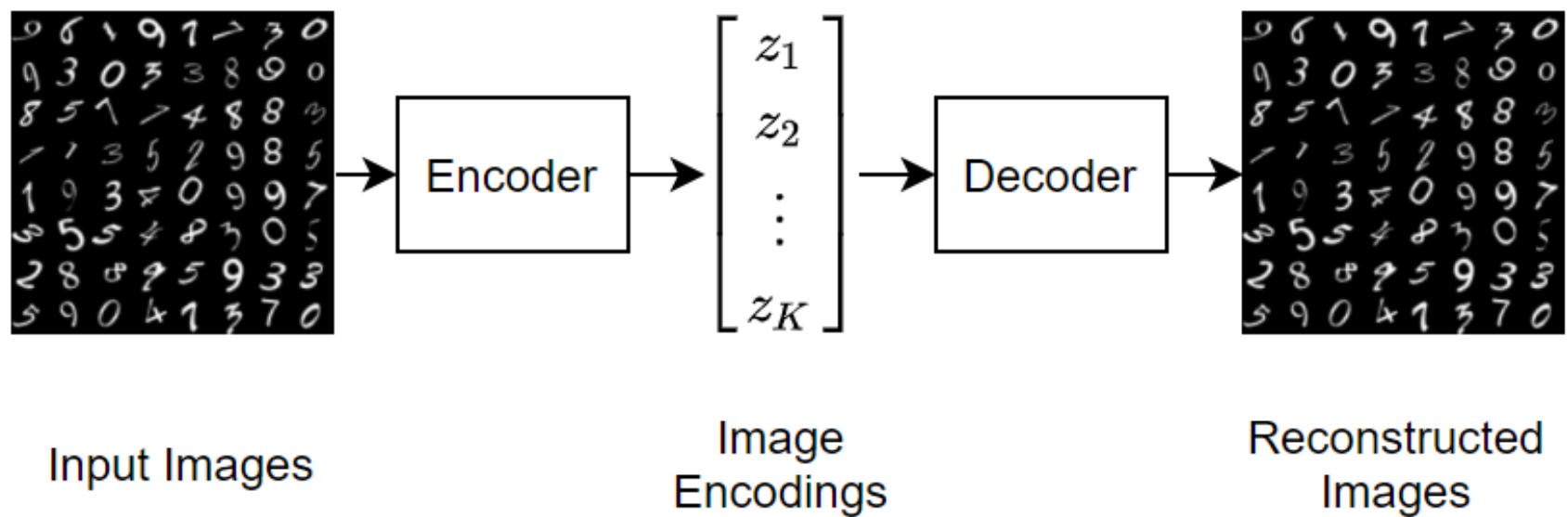
Ce qui se Passe

- **Raffinement Itératif** : Le bruit est progressivement supprimé, et les caractéristiques spécifiques au prompt deviennent visibles.

- **Résultat** : Une image détaillée et de haute qualité correspondant à la description textuelle.

▼ Étape 4 : Décodage de l'Image avec le Décodeur VAE (Étape Finale)

- Une fois le vecteur latent débruité final produit, il doit être transformé en une **image en pleine résolution**.
- C'est ici que le **décodeur VAE** intervient.

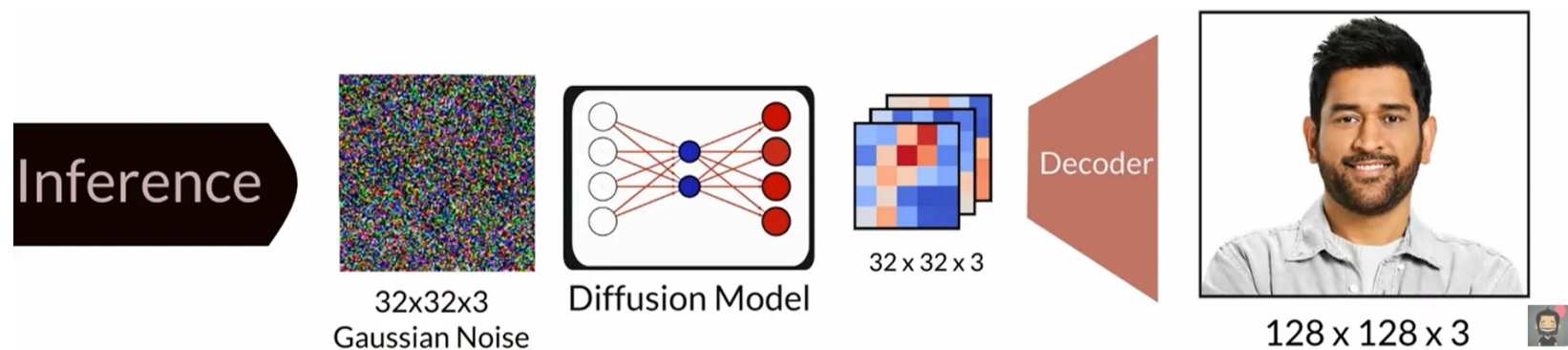


• Pourquoi le Décodeur VAE :

- Le décodeur VAE garantit que l'image latente est **convertie dans un format** que nous pouvons visualiser et utiliser. Il transforme les **informations compressées** en un **domaine visuel**.

• Fonctionnement du Décodeur VAE :

- Le **décodeur VAE** est responsable du **décodage de l'espace latent** en une image dans l'**espace pixel**.



- Il prend la représentation latente débruitée et la reconstruit en une image haute résolution, transformant les informations abstraites et compressées en une **image visuellement significative**.

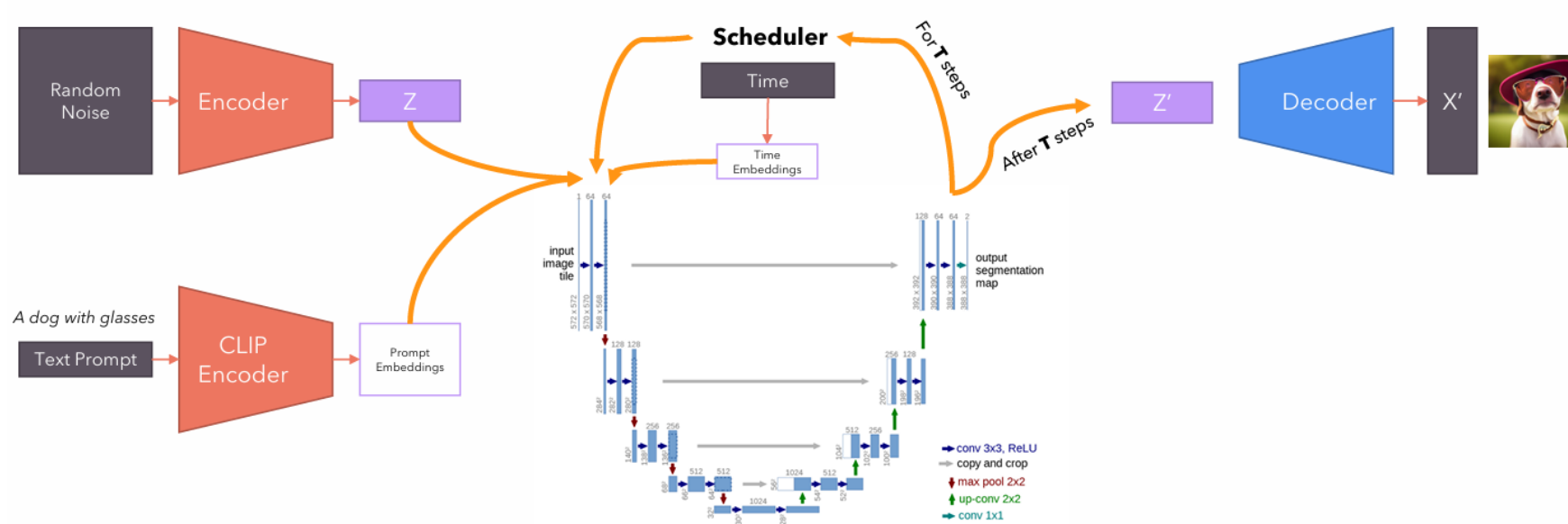
▼ Résultat Final

- Après l'étape finale de débruitage et de décodage, le résultat est une **image haute résolution** qui correspond étroitement au prompt texte.
- Par exemple, le prompt "**A serene lake surrounded by snow-capped mountains at sunrise**" générerait une **image réaliste** d'un lac entouré de montagnes, illuminé par un lever de soleil.



▼ Architecture Finale de Génération de Texte à Image

Architecture (Text-To-Image)



1. **Entrée** : Un prompt texte (ex : "Un chien avec des lunettes") est fourni.
2. **Encodage du Texte** : Le prompt est converti en une représentation numérique (embedding) à l'aide de **CLIP**.
3. **Initialisation du Bruit** : Une image de bruit aléatoire est générée dans l'espace latent.
4. **Encodage Temporel** : Des informations sur les pas de temps sont ajoutées pour guider le processus de débruitage.
5. **Débruitage (U-Net)** :
 - L'**encodeur** traite l'image bruitée et l'embedding texte.
 - Le **décodeur** affine l'image de manière itérative, en supprimant le bruit étape par étape.

6. **Planificateur** : Contrôle le processus de suppression du bruit sur plusieurs pas de temps.

7. **Sortie** : Une image claire et de haute qualité correspondant au prompt texte (ex : "Un chien avec des lunettes").