Research on Improving Image Editing Performance with GLIGEN Models

김민창

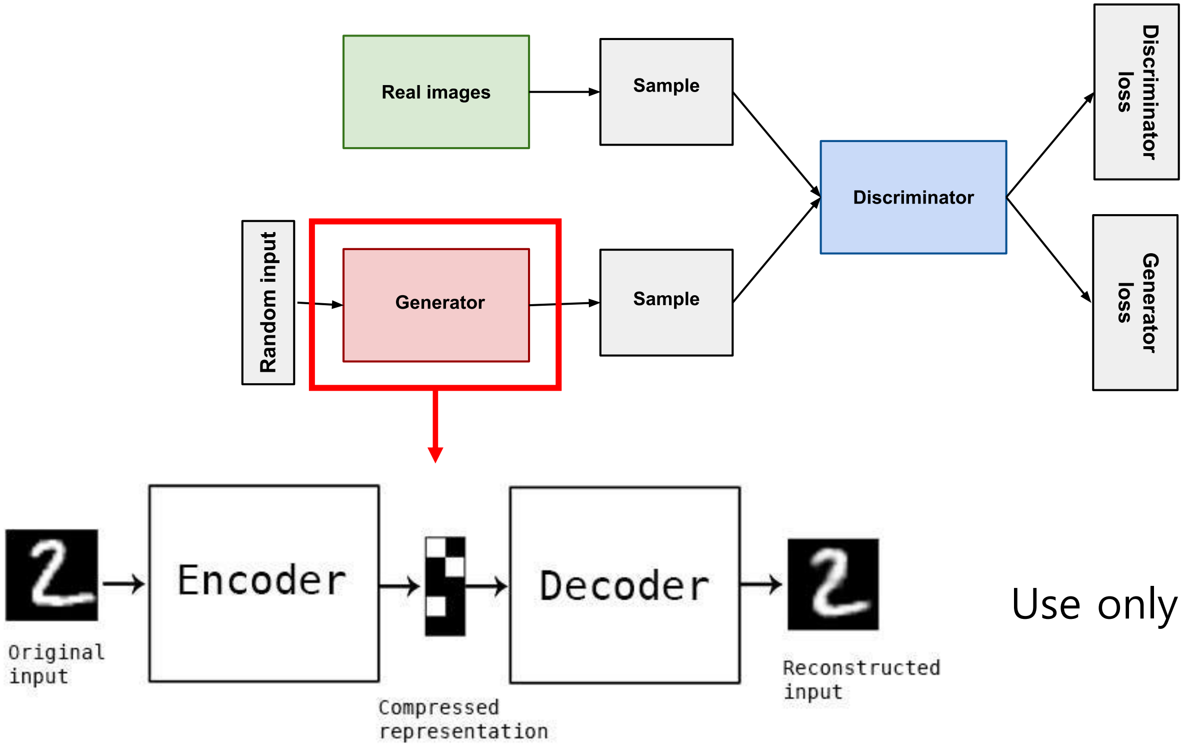
Summary

1. Stable Diffusion
2. GLIGEN
3. DETR

GAN(Generative Adversarial Network)

- GAN

Composed of constructors and discriminators



Output 1 if it's a real image, 0 if it's fake

Passing generated images through a discriminator to minimize losses

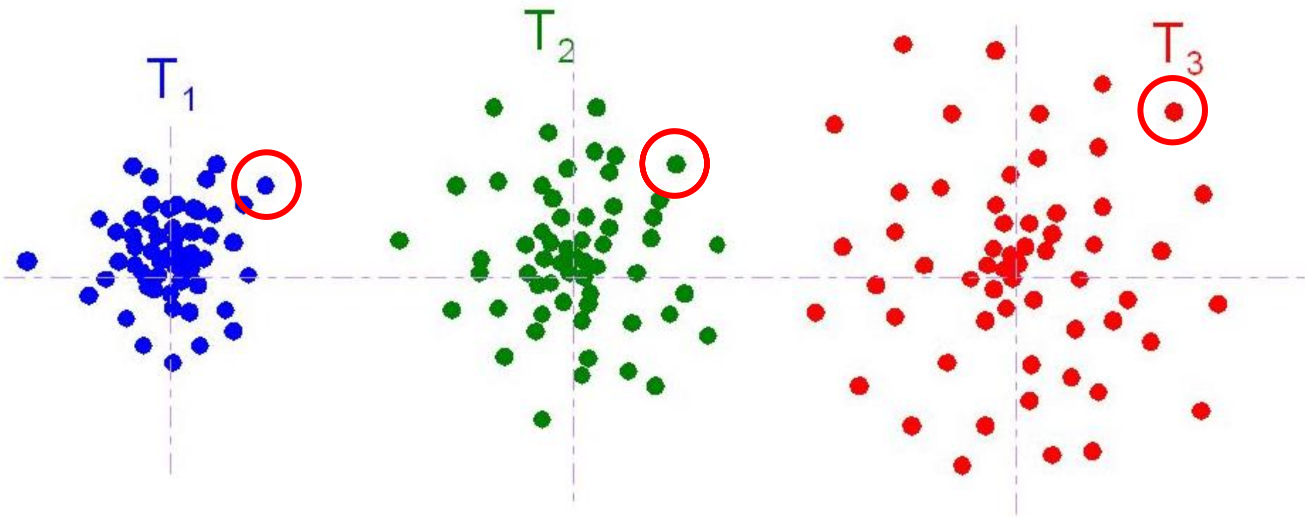
Use only AutoEncoder's Decoder

\>

Diffusion Model

- Diffusion

The movement of the molecules ranges from a Gaussian Distribution

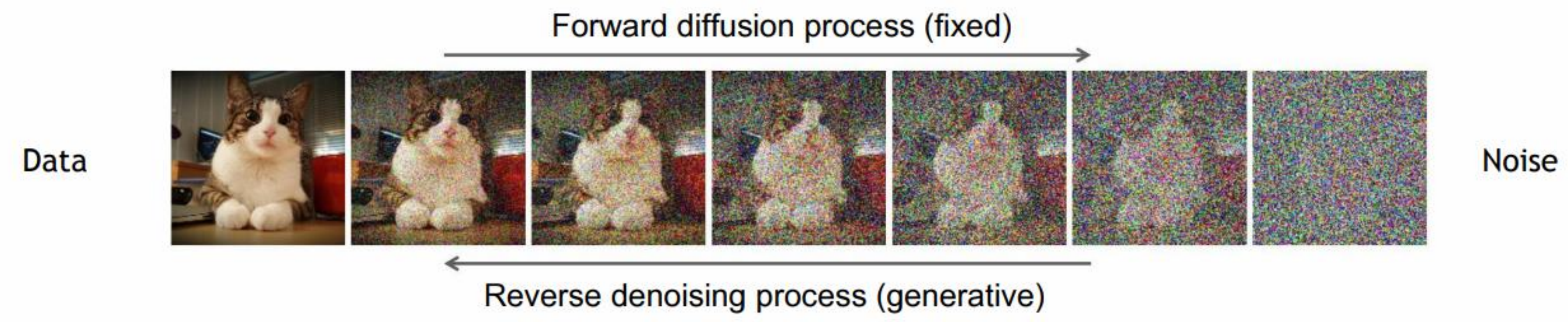


Diffusion Model

- Diffusion Model

  1) Add a Noise value to Pixel values that follows a normal distribution

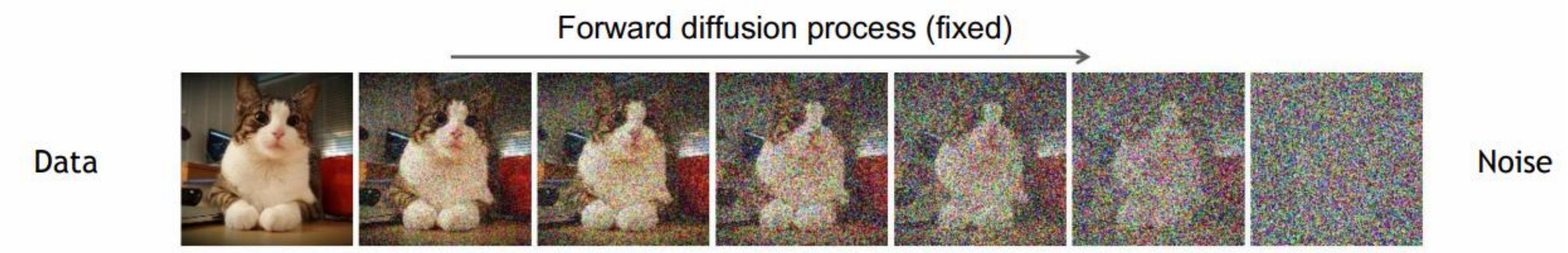
  2) Restoring a Noisy Image to the Original Image



Diffusion Model

- Forward Diffusion Process

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) := \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}), \qquad q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I})$$

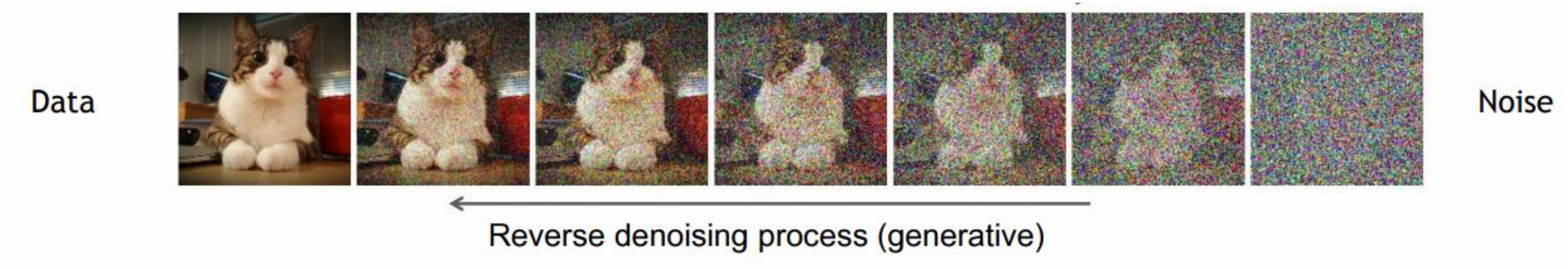


Diffusion Model

- Reverse Denoising Process

$$p_\theta(\mathbf{x}_{0:T}) := p(\mathbf{x}_T)\prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t), \qquad p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \boxed{\boldsymbol{\mu}_\theta(\mathbf{x}_t, t),} \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t))$$

Trainable network

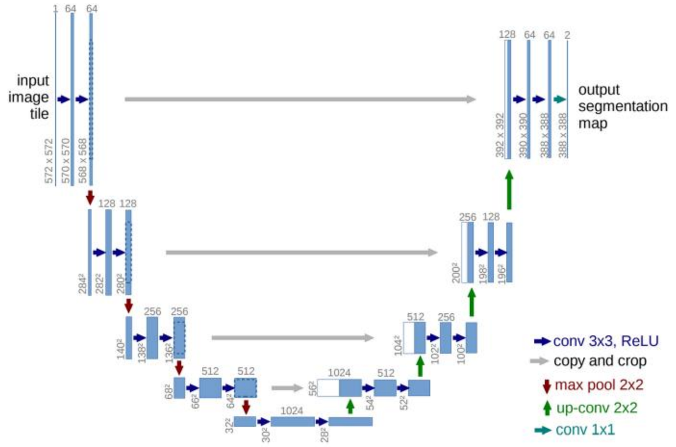


Diffusion Model

U-Net Structure

Image x_t → Diffusion Model → Noisy Image $\varepsilon_\theta(x_t, t)$

t



Skip Connection을 이용

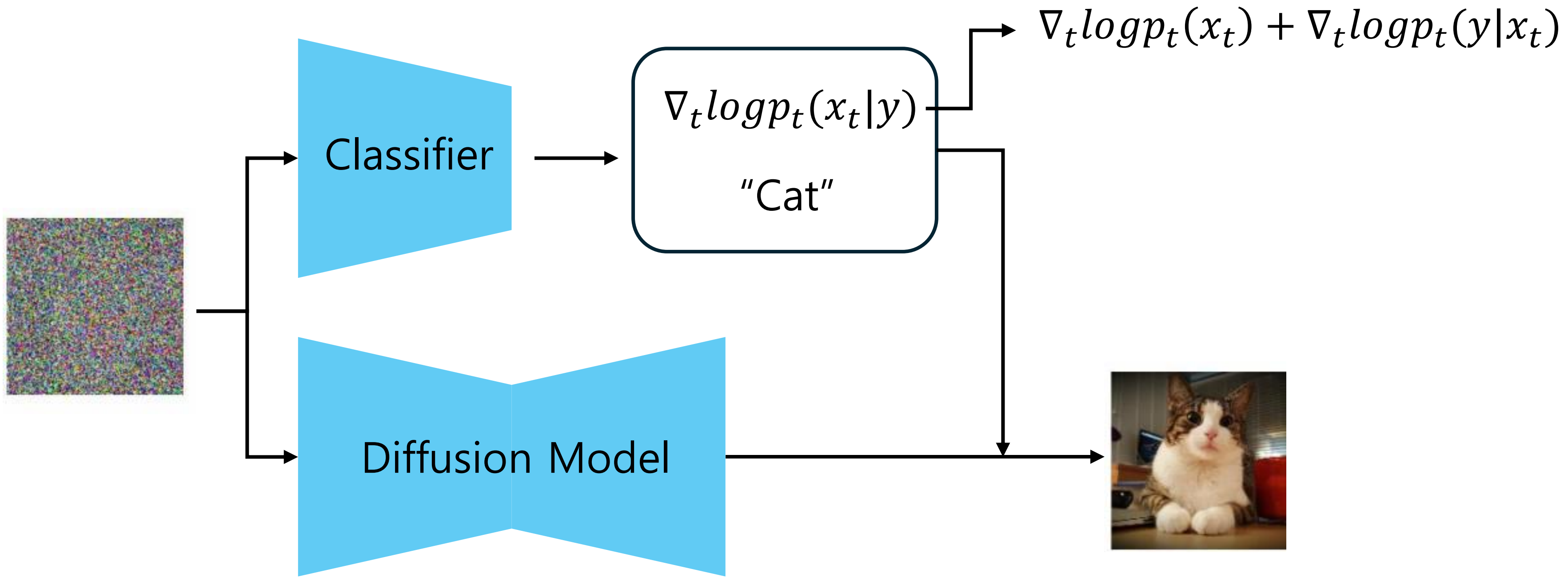
- Loss Function

$$L_{\text{simple}}(\theta) := \mathbb{E}_{t,\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t)\right\|^2\right]$$

Diffusion Model

- Classifier guidance

  Use classifiers to adjust the image generation process



Stable Diffusion

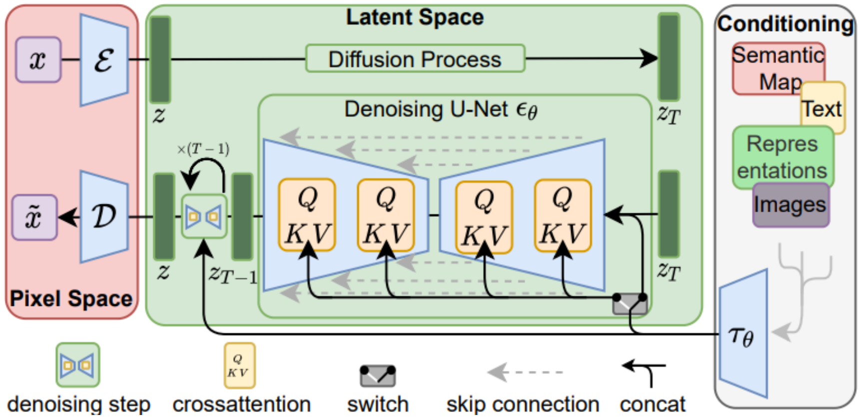
- Stable Diffusion : Models with VAE added to the Diffusion Model

  ✓ Converting Text to Latent Vector with CLIP

  ✓ Adding Random Noise with Gaussian Distribution

  ✓ Remove Random Noise with the Diffusion Process

  ✓ Restoring an Existing Image to Vector with VAE's Decoder



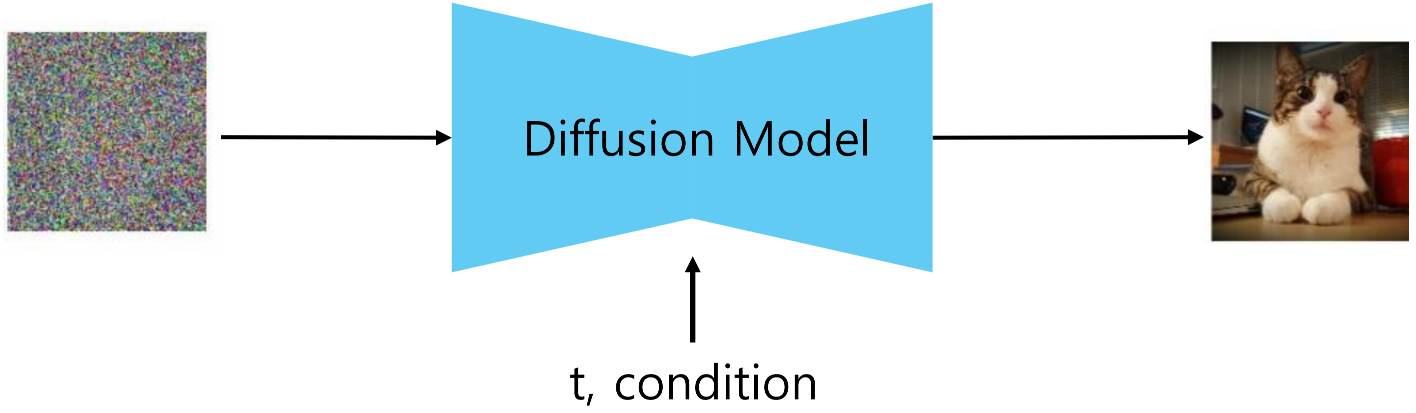
- Loss Function

$$L_{LDM} := \mathbb{E}_{\boxed{\mathcal{E}(x)}, \epsilon \sim \mathcal{N}(0,1), t}\left[\| \epsilon - \epsilon_\theta(z_t, t) \|_2^2 \right]$$

Stable Diffusion

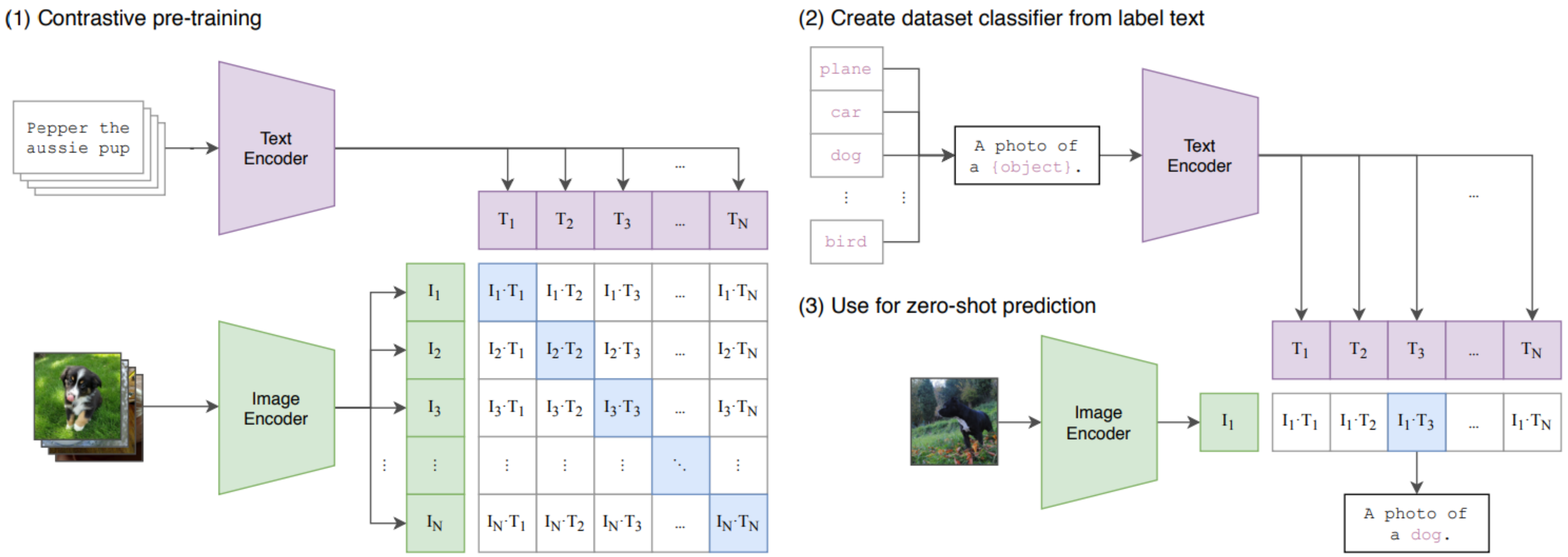
- Classifier free guidance

  Adjust image generation with conditional and unconditional diffusion models



CLIP(Contrastive Language-Image Pre-training)

- Model trained on similarities between text and images



> AE(Auto Encoder)와 VAE(Variational Auto Encoder)의 차이

- Auto Encoder : Iteratively update weights by back-propagating reconstruction losses

Not available for generative models that generate new
data in the event of overfitting

- Variational Auto Encoder : Convert Probability Distribution (Mean, Variance) to Latent Vector

GLIGEN(Grounded-Language-to-Image Generation)

- Add a new Grounding conditional input

Bounding Box

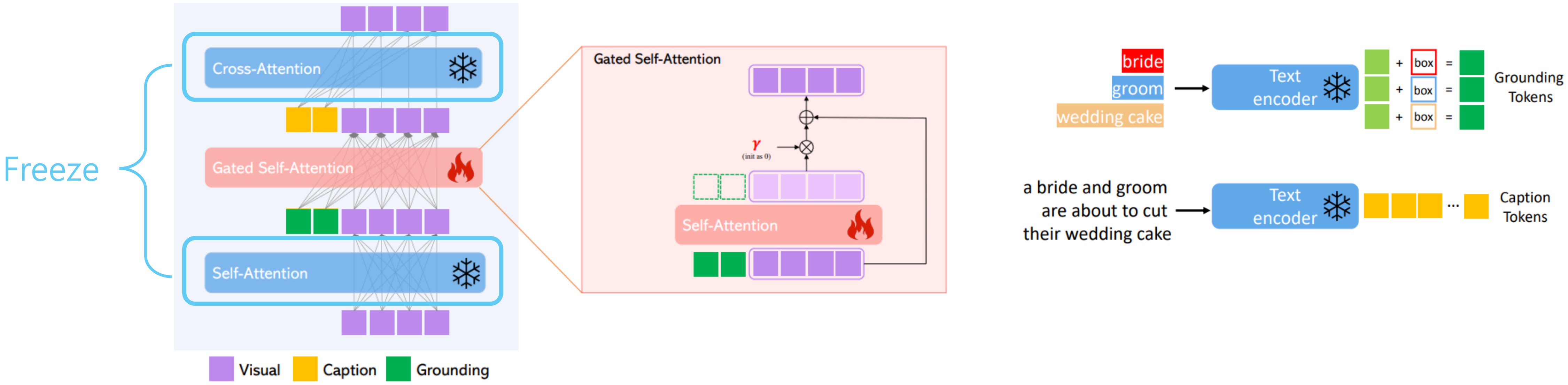
(a) Caption: “A woman sitting in a restaurant with a pizza in front of her ”
Grounded text: table, pizza, person, wall, car, paper, chair, window, bottle, cup

Key Point

(d) Caption: “a baby girl / monkey / Hormer Simpson / is scratching her/its head”
Grounded keypoints: plotted dots on the left image

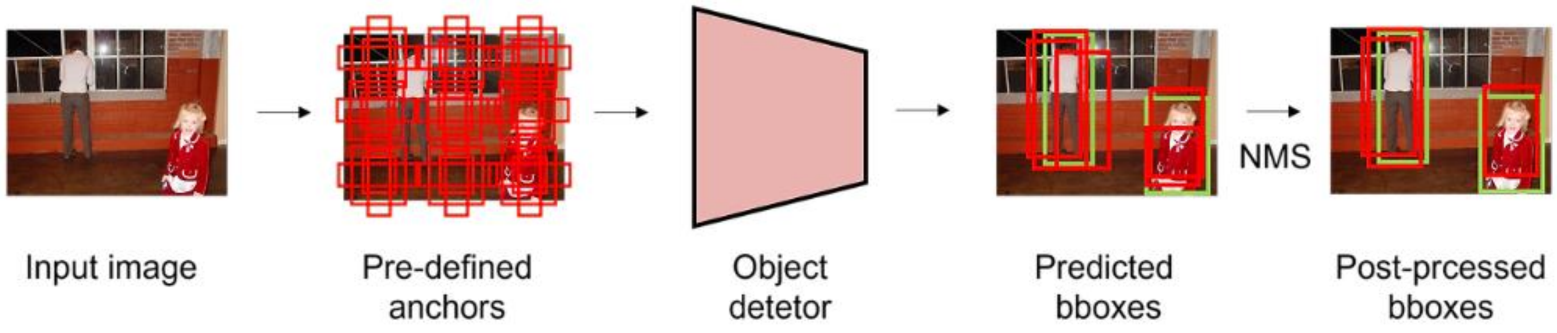
GLIGEN(Grounded-Language-to-Image Generation)

- Freeze existing layers and only learn Gated Self-Attention

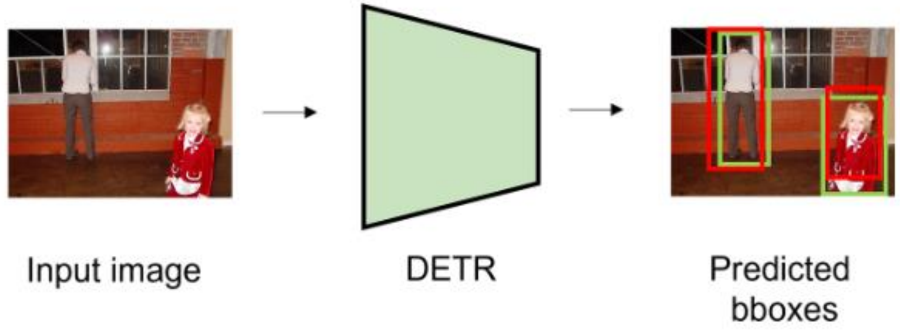


DETR(Detection Transformer)

1) CNN-based Detection



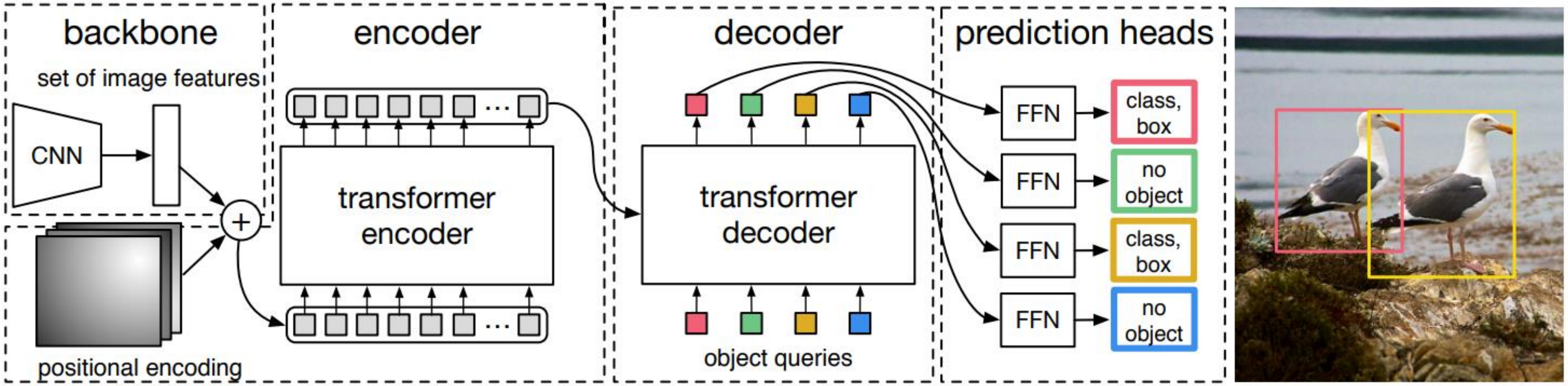
2) Transformer-based Detection



→ Pipelines simplified

DETR(Detection Transformer)

- Solving the set prediction problem with bipartite matching
- Fix the number of outputs, N



DETR(Detection Transformer)

- Matching Loss

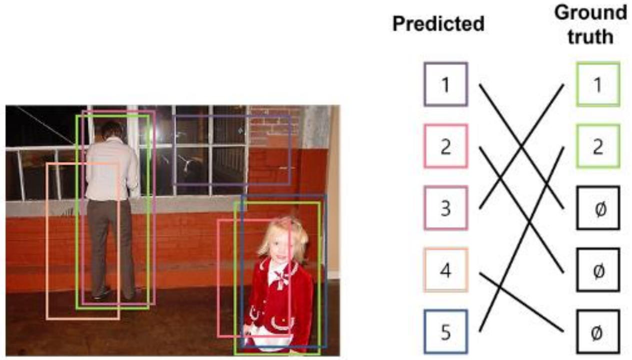
$$\hat{\sigma} = \underset{\sigma \in \mathfrak{S}_N}{\arg\min} \sum_{i}^{N} \mathcal{L}_{\mathrm{match}}(y_i, \hat{y}_{\sigma(i)})$$

- Hungarian Loss

$$\mathcal{L}_{\mathrm{Hungarian}}(y, \hat{y}) = \sum_{i=1}^{N} \left[-\log \hat{p}_{\hat{\sigma}(i)}(c_i) + \mathbb{1}_{\{c_i \neq \varnothing\}} \mathcal{L}_{\mathrm{box}}(b_i, \hat{b}_{\hat{\sigma}}(i)) \right]$$

- Bounding box Loss

$$\lambda_{\mathrm{iou}} \mathcal{L}_{\mathrm{iou}}(b_i, \hat{b}_{\sigma(i)}) + \lambda_{\mathrm{L1}} ||b_i - \hat{b}_{\sigma(i)}||_1$$

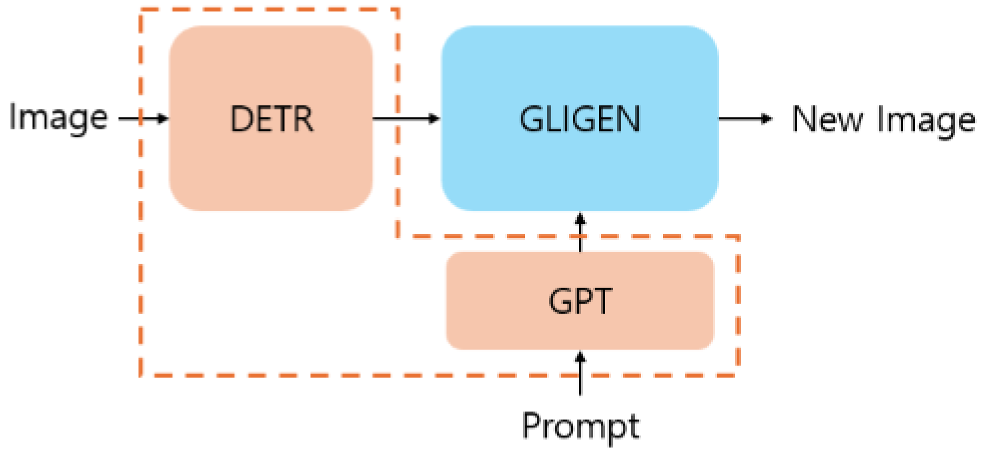


	1	2	ø	ø	ø
1	12	11	1	1	1
2	4	2	8	5	9
3	1	3	5	7	8
4	2	5	6	7	4
5	2	1	9	10	6

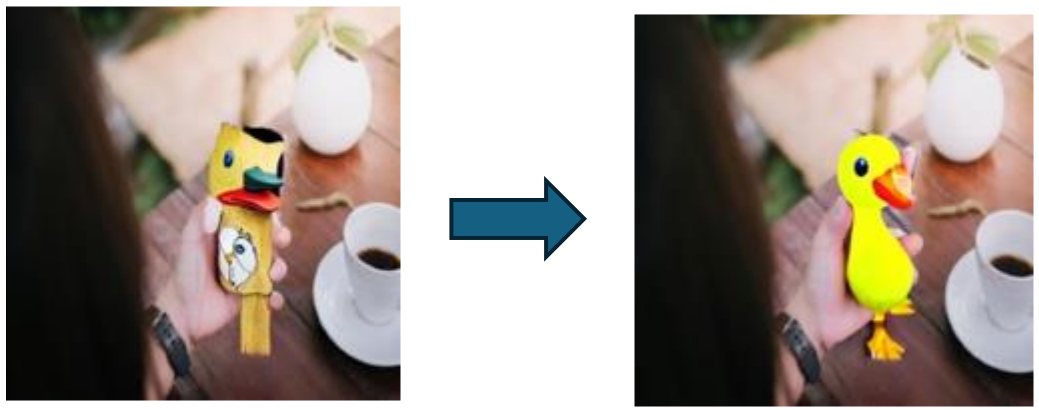
Permutation = [3, 4, 1, 5, 2]
Matching score = 1 + 5 + 1 + 4 + 1 = 12

Model Structure & Results

- Model Structure



- Results



> Dataset & Evaluate

Dataset	내 용	평가 방법
COCO2017	Used in computer vision for various computer vision tasks such as object recognition, segmentation, keypoint detection, etc.	Evaluate by randomly masking any object among the original image objects and then performing Inpainting on it

Model	FID Score
GLIGEN	28.94
A novel approach to modeling	26.13