模式分析与机器智能
工业和信息化部重点实验室
MIIT Key Laboratory of
Pattern Analysis & Machine Intelligence

ParNeC 模式识别与神经计算研究组
PAttern Recognition and NEural Computing

# Multimodal Models

## 2. Diffusion Model

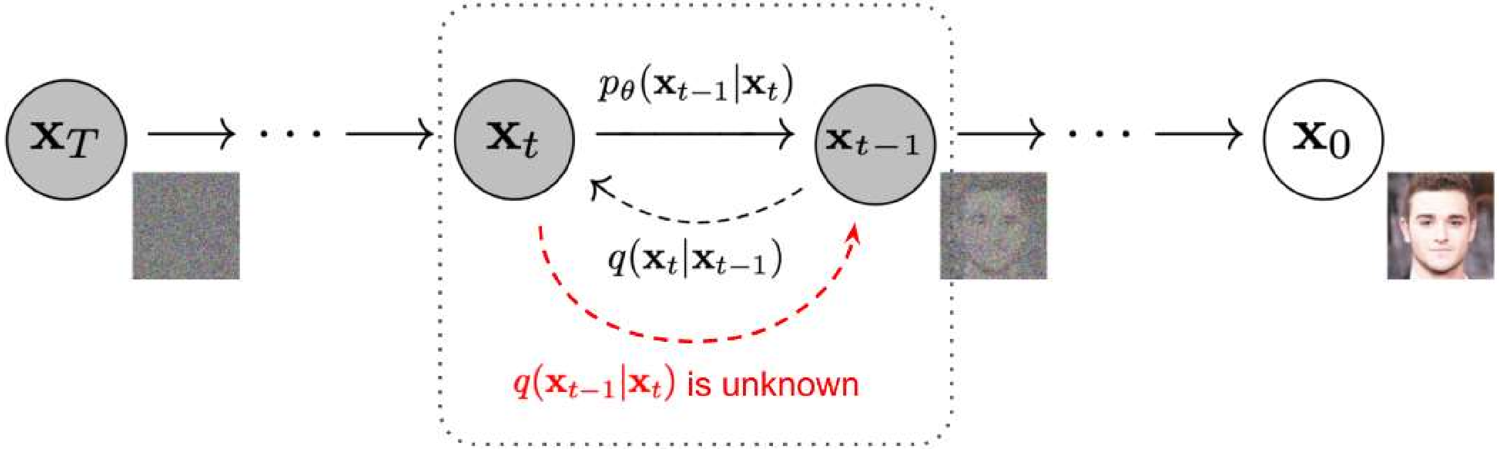
# Background

## Diffusion Models

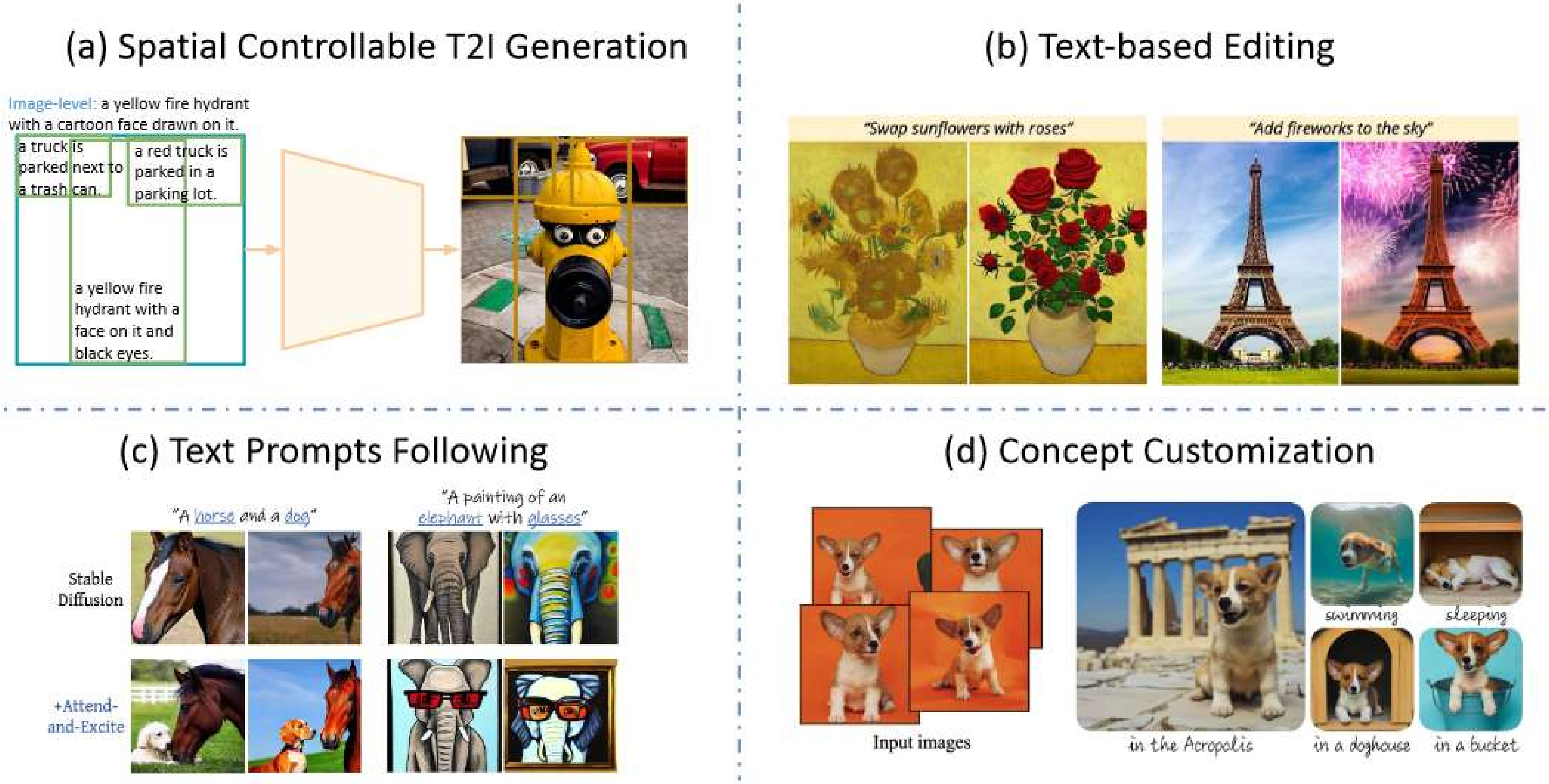
Given a data point sampled from a real data distribution $\mathbf{x}_0 \sim q(\mathbf{x})$, let us define a **forward diffusion process** in which we add small amount of Gaussian noise to the sample in $T$ steps, producing a sequence of noisy samples $\mathbf{x}_1, \dots, \mathbf{x}_T$. The step sizes are controlled by a variance schedule $\{\beta_t \in (0,1)\}_{t=1}^{T}$

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

The data sample $\boldsymbol{x_0}$ gradually loses its distinguishable features as the step $\boldsymbol{t}$ becomes larger. Eventually when $T\to\infty$, $\boldsymbol{x_T}$ is equivalent to an isotropic Gaussian distribution.


$\mathbf{x}_T$
$\mathbf{x}_t$
$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$
$\mathbf{x}_{t-1}$
$q(\mathbf{x}_t|\mathbf{x}_{t-1})$
$\mathbf{x}_0$
$q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ is unknown

# Taxonomy

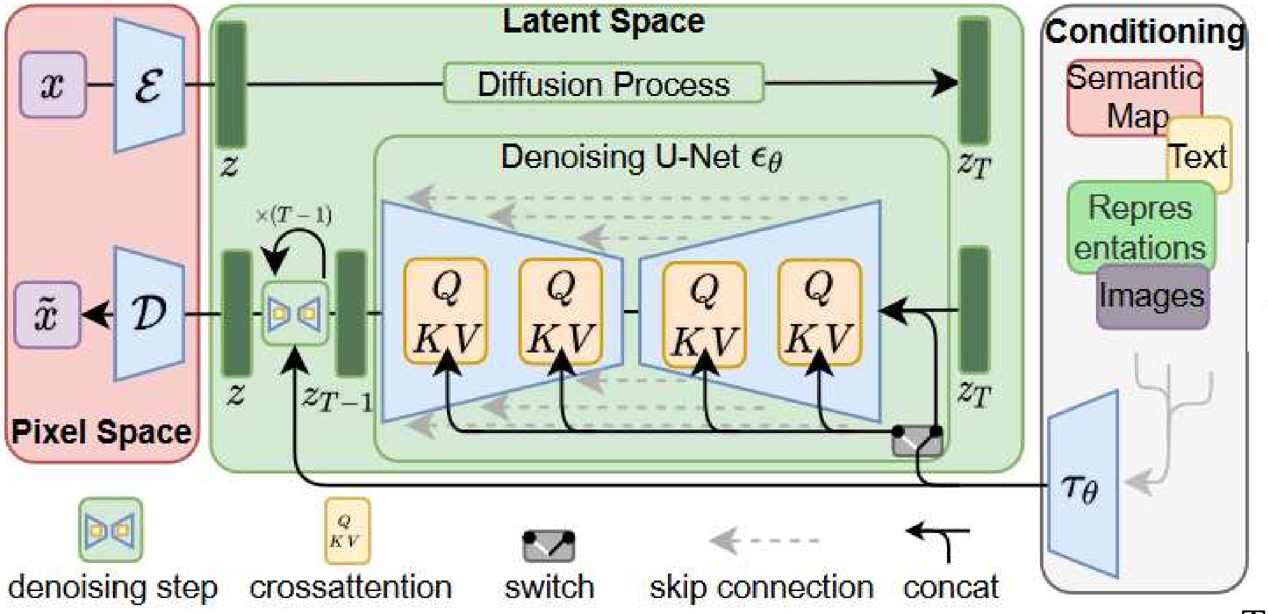

(a) Spatial Controllable T2I Generation
Image-level: a yellow fire hydrant with a cartoon face drawn on it.
a truck is parked next to a trash can.
a red truck is parked in a parking lot.
a yellow fire hydrant with a face on it and black eyes.
(b) Text-based Editing
"Swap sunflowers with roses"
"Add fireworks to the sky"
(c) Text Prompts Following
"A horse and a dog"
"A painting of an elephant with glasses"
Stable Diffusion
+Attend-and-Excite
(d) Concept Customization
Input images
in the Acropolis
swimming
sleeping
in a doghouse
in a bucket

More controllable, more editable, more accurate and more customization

# Background

## Stable Diffusion (2022-04) High-Resolution Image Synthesis with Latent Diffusion Models

**Motivation:** Since previous models typically operate directly in **pixel space**, optimization of powerful DMs often consumes hundreds of GPU days and inference is expensive due to sequential evaluations. To enable DM training on limited computational resources while retaining their quality and flexibility, we apply them in the **latent space** of powerful pretrained autoencoders.



- Before:

$$L_{DM} = \mathbb{E}_{x,\epsilon\sim\mathcal{N}(0,1),t}\Big[\|\epsilon - \epsilon_\theta(x_t,t)\|_2^2\Big]$$

- After:

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

- Conditional image generation by Cross-Attention:

$$\mathrm{Attention}(Q,K,V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)\cdot V$$

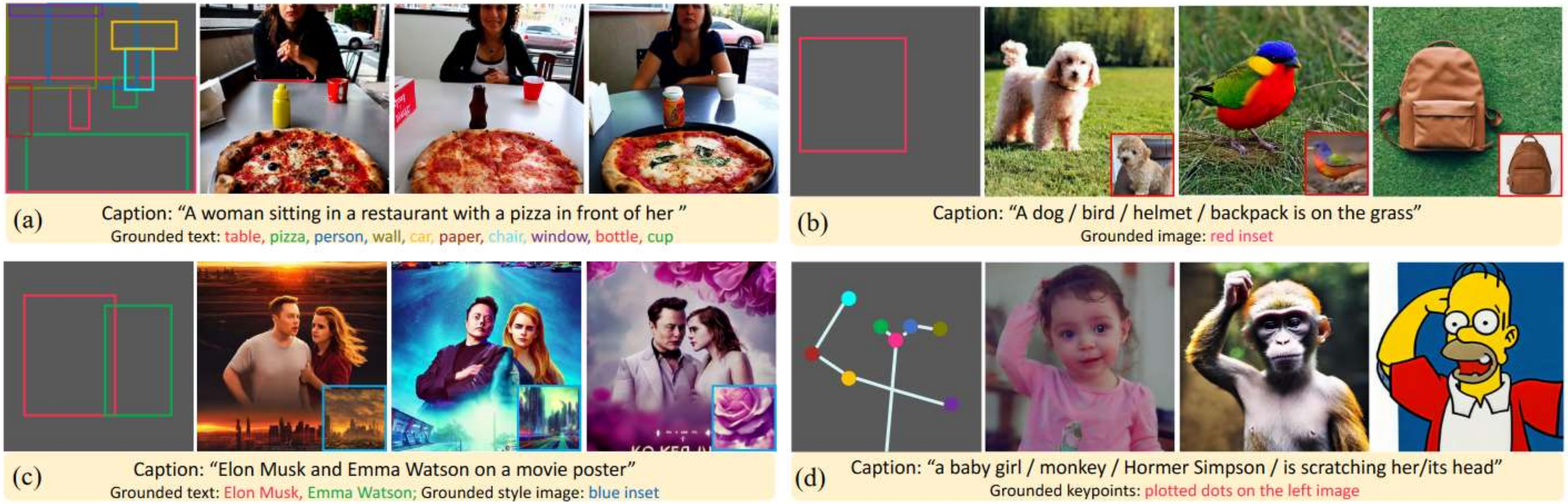
Text embedding as K and V, image embedding as Q

**Basic paradigm of diffusion**

# Methods – Controllable Generation

## GLIGEN (2023-04) GLIGEN: Open-Set Grounded Text-to-Image Generation

**Motivation:** Extend the functionality of existing pre-trained text-to-image diffusion models by enabling them to also be conditioned on grounding inputs.



(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

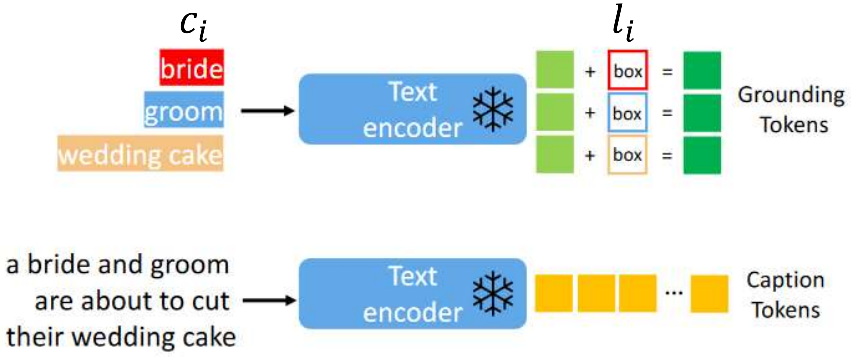
(b) Caption: "A dog / bird / helmet / backpack is on the grass"
Grounded image: red inset

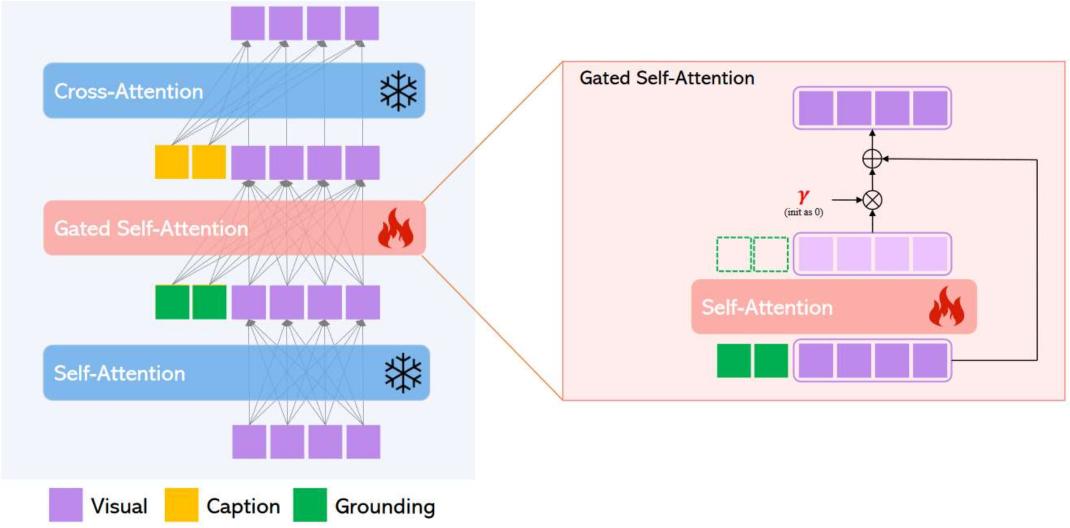
(c) Caption: "Elon Musk and Emma Watson on a movie poster"
Grounded text: Elon Musk, Emma Watson; Grounded style image: blue inset

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

# Methods – Controllable Generation

## GLIGEN (2023-04) GLIGEN: Open-Set Grounded Text-to-Image Generation





Denote the semantic information of the grounding entity as $e$, which can be described either through **text** or an **example image** and as $l$ the grounding spatial configuration.

$$\begin{aligned} \text{Instruction:}\quad & \boldsymbol{y} = (\boldsymbol{c}, \boldsymbol{e}), \quad \text{with} \\ \text{Caption:}\quad & \boldsymbol{c} = [c_1, \cdots, c_L] \\ \text{Grounding:}\quad & \boldsymbol{e} = [(e_1, \boldsymbol{l}_1), \cdots, (e_N, \boldsymbol{l}_N)] \end{aligned}$$

Grounding Tokens

$$h^e = \mathrm{MLP}(f_{\text{text}}(e), \mathrm{Fourier}(\boldsymbol{l}))$$

Also support **Keypoints control**

**Gated Self-Attention.** Insert gated self-attention between SA layers and CA layers

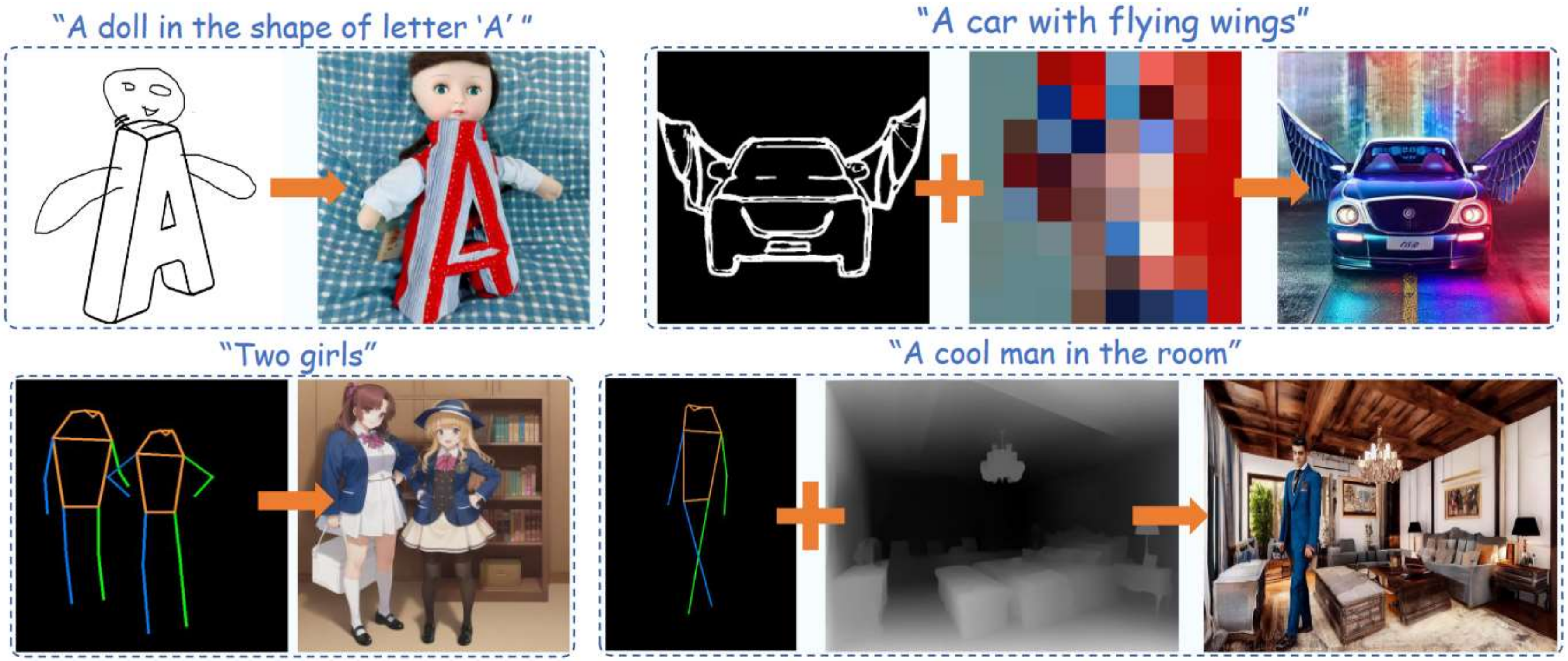
$$\boldsymbol{v} = \boldsymbol{v} + \beta \cdot \tanh(\gamma) \cdot \mathrm{TS}(\mathrm{SelfAttn}([\boldsymbol{v}, \boldsymbol{h}^e]))$$

TS(·) is a token selection operation that considers visual tokens only, β is introduced during inference to improve controllability.

# Methods – Controllable Generation

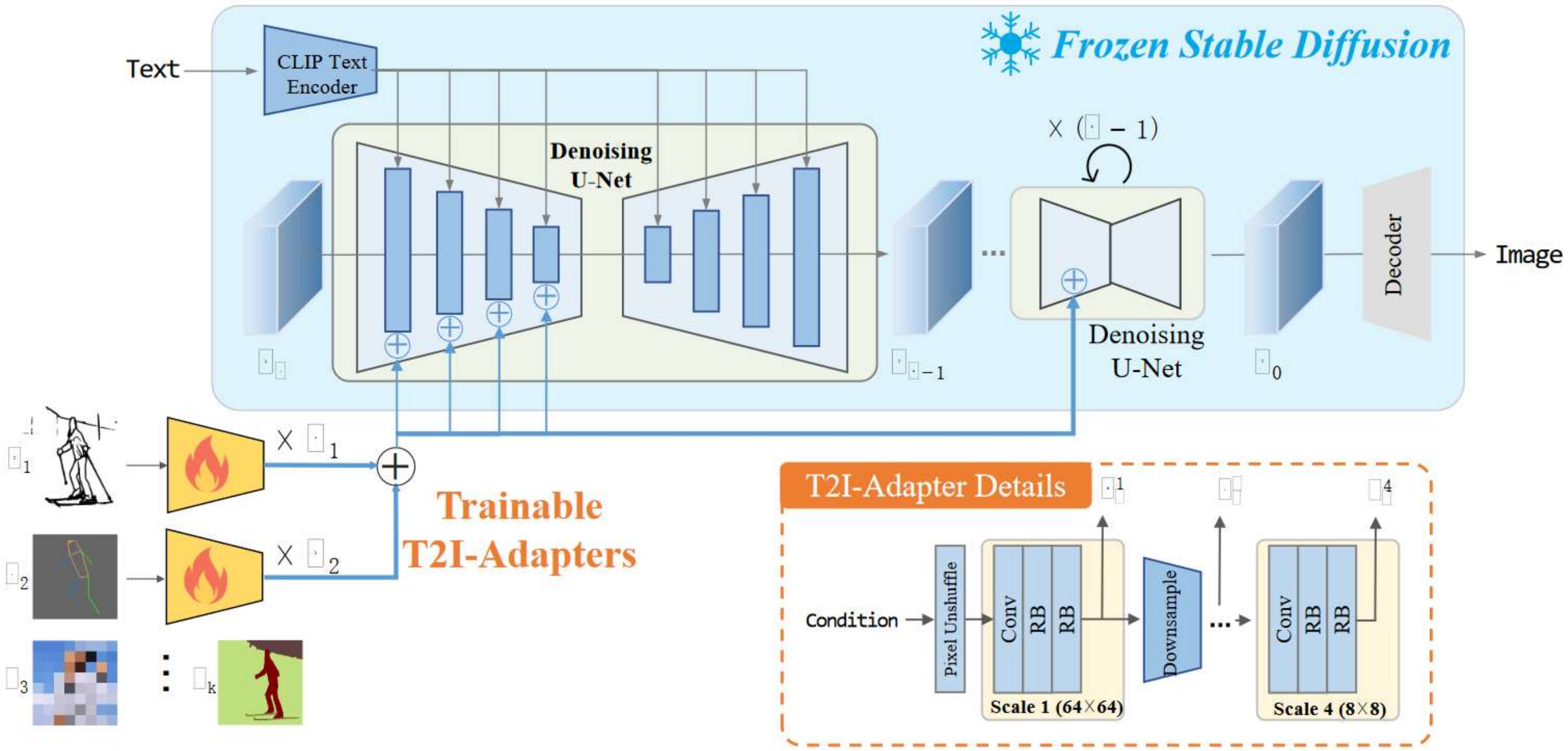
## T2I-Adapter (2023-05) T2I-Adapter: Learning Adapters to Dig out More Controllable Ability for Text-to-Image Diffusion Models

**Motivation:** Relying solely on **text prompts** cannot fully take advantage of the knowledge learned by the model, especially when flexible and accurate controlling (e.g., color and structure) is needed.


"A doll in the shape of letter 'A' "
"A car with flying wings"
"Two girls"
"A cool man in the room"

# Methods – Controllable Generation

## T2I-Adapter (2023-05) T2I-Adapter: Learning Adapters to Dig out More Controllable Ability for Text-to-Image Diffusion Models

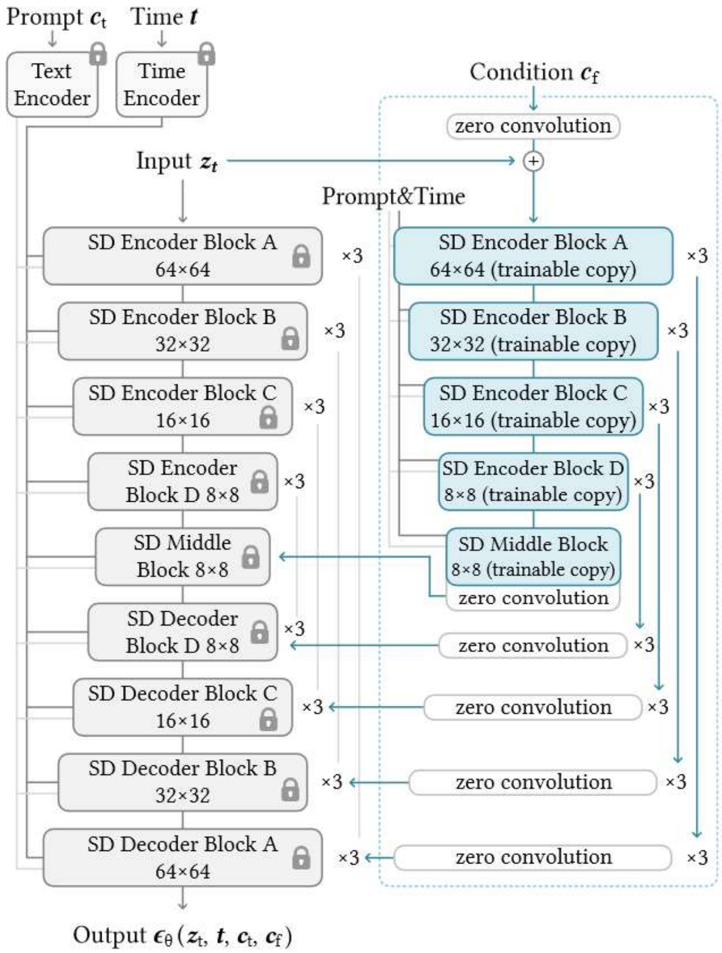


In each scale, one convolution layer and two residual blocks (RB) are utilized to extract the condition feature $\mathbf{F}_{\mathbf{k}}^{\mathbf{c}}$. Finally, multi-scale condition features $\mathbf{F_c} = \{F_{c_1}, F_{c_2}, F_{c_3}, F_{c_4}\}$ are formed.

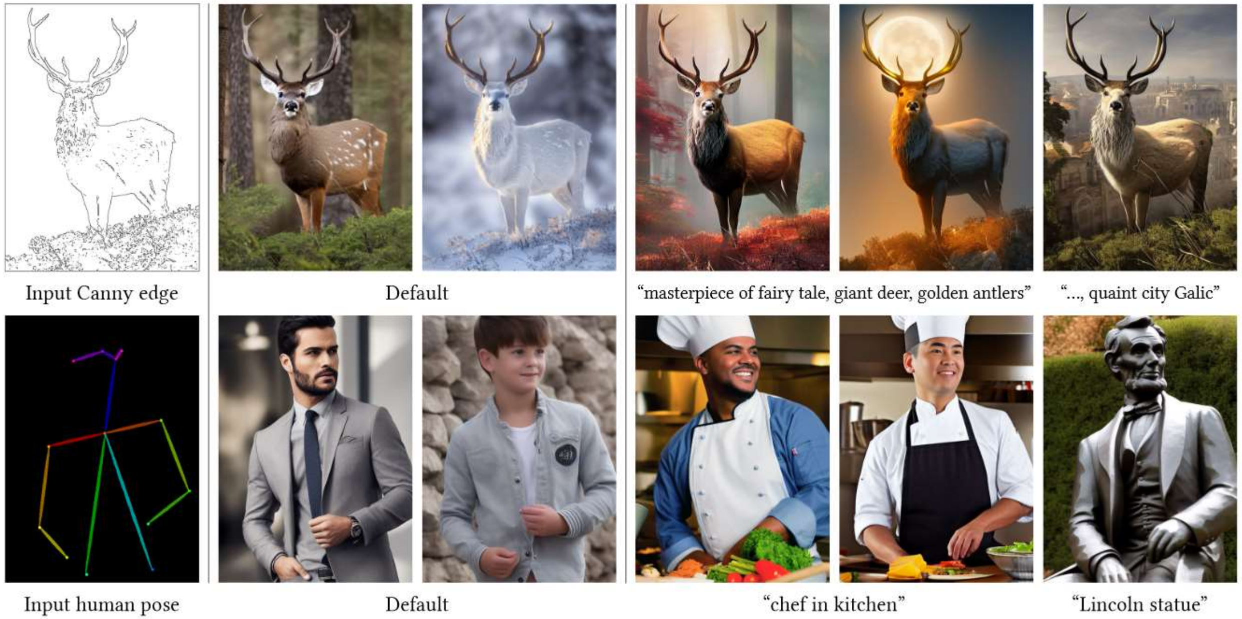
$$\mathbf{F}_c = \mathcal{F}_{AD}(\mathbf{C}) \qquad \hat{\mathbf{F}}^i_{enc} = \mathbf{F}^i_{enc} + \mathbf{F}^i_c,\ i \in \{1, 2, 3, 4\}$$

# Methods – Controllable Generation

## ControlNet (2023-09) Adding Conditional Control to Text-to-Image Diffusion Models



(a) Stable Diffusion (b) ControlNet



A feature map $\boldsymbol{x}$ is transformed into another feature map y as

$$\boldsymbol{y} = \mathcal{F}(\boldsymbol{x}; \Theta).$$

With the introduction of two zero convolutions:

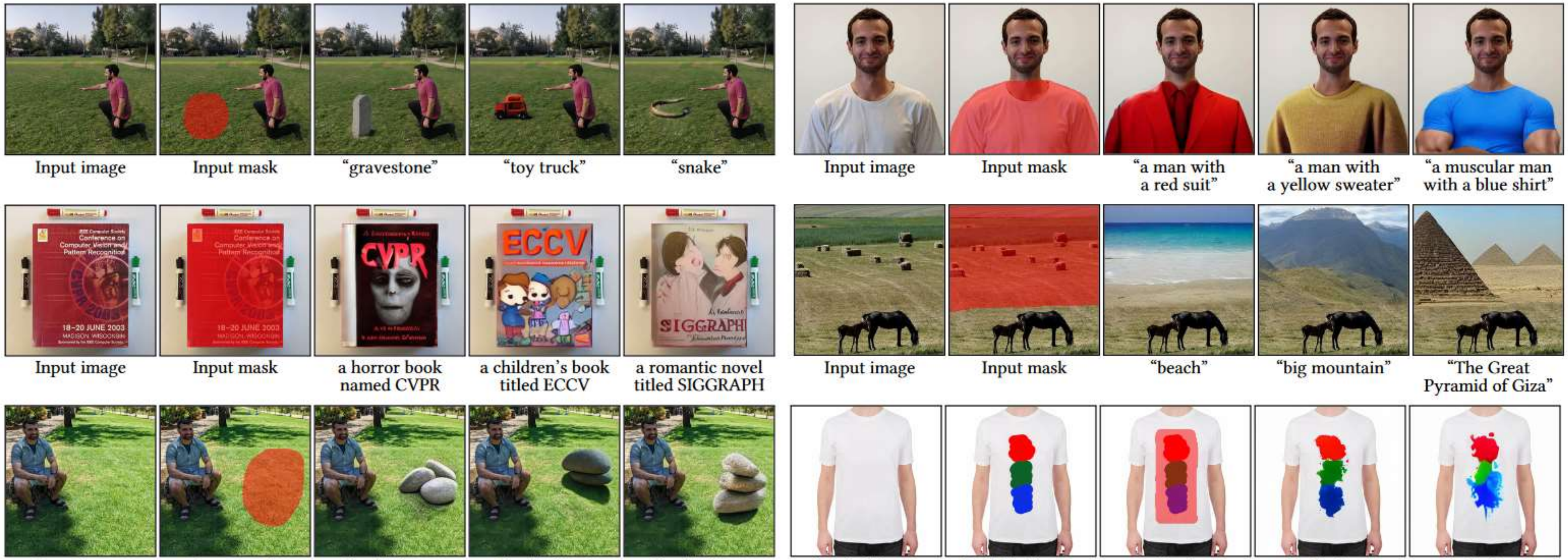
$$\boldsymbol{y}_{\mathrm{c}} = \mathcal{F}(\boldsymbol{x}; \Theta) + \mathcal{Z}(\mathcal{F}(\boldsymbol{x} + \mathcal{Z}(\boldsymbol{c}; \Theta_{\mathrm{z1}}); \Theta_{\mathrm{c}}); \Theta_{\mathrm{z2}})$$

In the first training step, since both the weight and bias parameters of a zero convolution layer are initialized to zero, so $\boldsymbol{y}_{\mathrm{c}} = \boldsymbol{y}$.

# Methods – Editing

## Blended Latent Diffusion (2023-04)

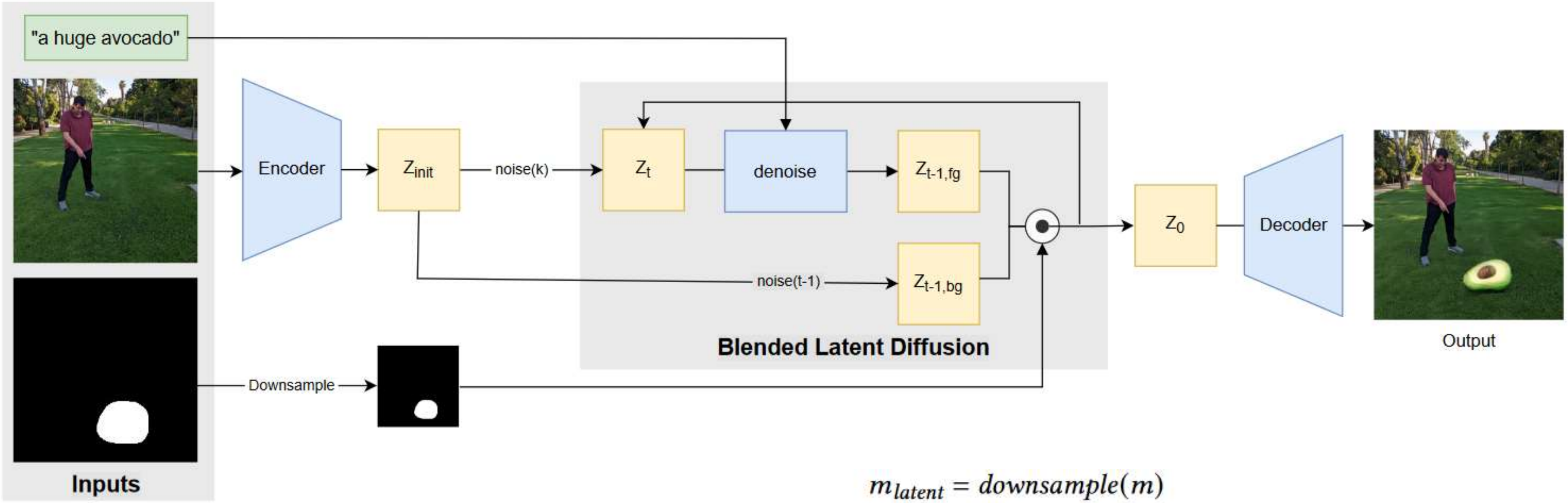
**Motivation:** Using **mask** and **text** to edit original picture.



# Methods – Editing

## Blended Latent Diffusion (2023-04)

Objective: Modify the foreground objects while keeping the remaining parts unchanged.



**Input:** source image $x$, target text description $d$, input mask $m$, diffusion steps $k$.
**Output:** edited image $\widehat{x}$ that differs from input image $x$ inside area $m$ according to text description $d$

```
$m_{latent} = downsample(m)$
$z_{init} \sim E(x)$
$z_k \sim noise(z_{init}, k)$
for all $t$ from $k$ to $0$ do
    $z_{fg} \sim denoise(z_t, d, t)$
    $z_{bg} \sim noise(z_{init}, t)$
    $z_t \leftarrow z_{fg} \odot m_{latent} + z_{bg} \odot (1 - m_{latent})$
end for
$\widehat{x} = D(z_0)$
return $\widehat{x}$
```

# Methods – Editing

## Imagic (2023-05) Imagic: Text-Based Real Image Editing with Diffusion Models

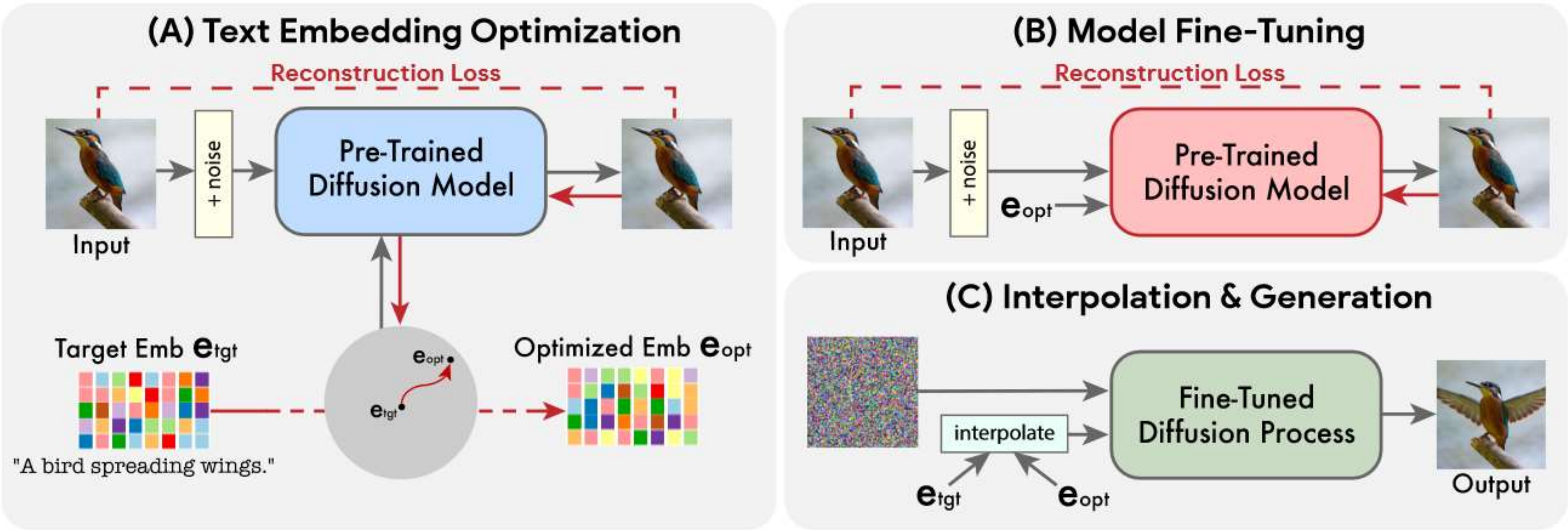

(A) Text Embedding Optimization
Reconstruction Loss
Input
+ noise
Pre-Trained Diffusion Model
Target Emb e tgt
"A bird spreading wings."
e opt
e tgt
Optimized Emb e opt
(B) Model Fine-Tuning
Reconstruction Loss
Input
+ noise
e opt
Pre-Trained Diffusion Model
(C) Interpolation & Generation
interpolate
e tgt
e opt
Fine-Tuned Diffusion Process
Output

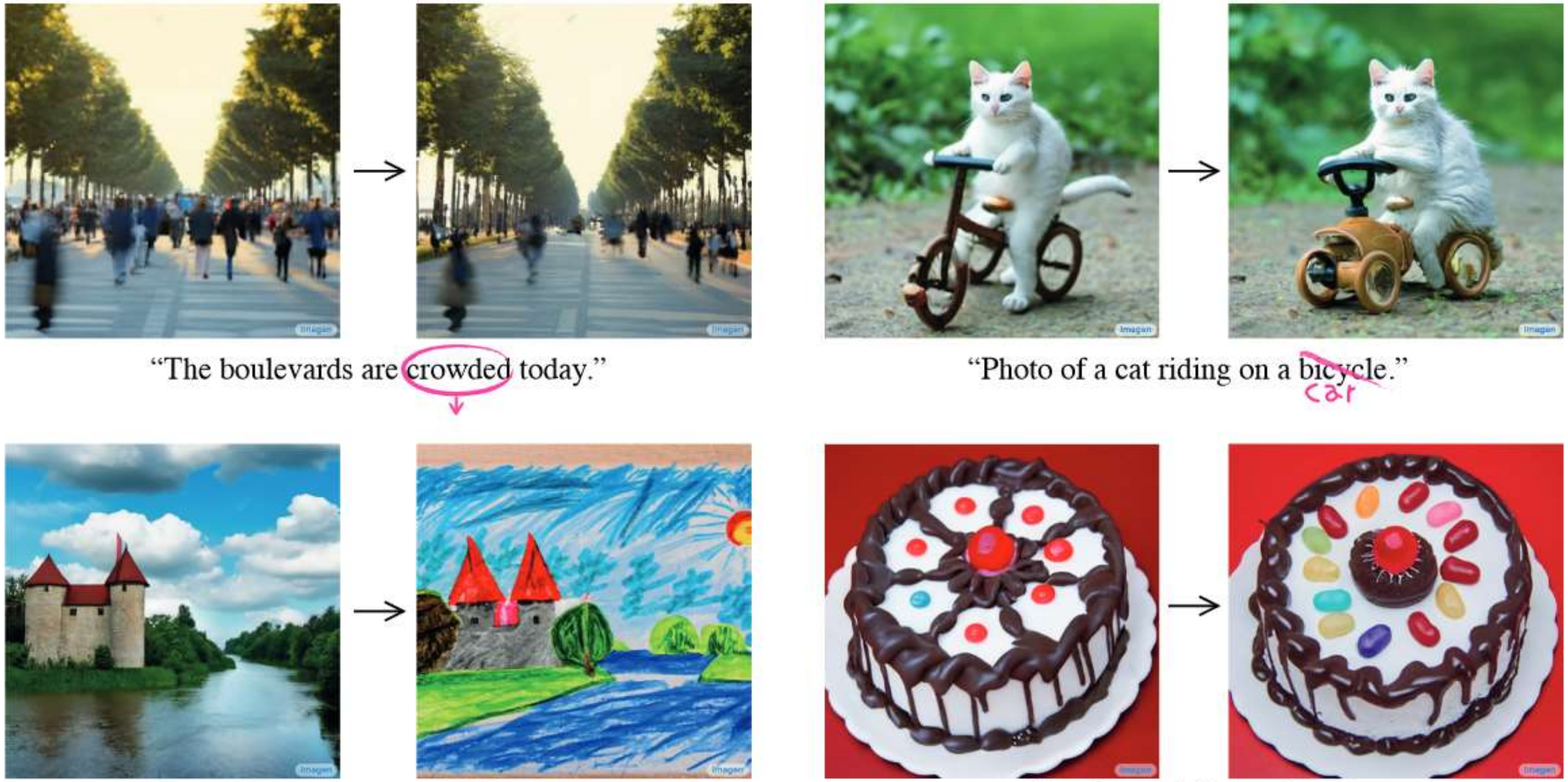
Figure 3. **Schematic description of *Imagic*.** *Given a real image and a target text prompt: (A) We encode the target text and get the initial text embedding* $\mathbf{e}_{tgt}$*, then optimize it to reconstruct the input image, obtaining* $\mathbf{e}_{opt}$*; (B) We then fine-tune the generative model to improve fidelity to the input image while fixing* $\mathbf{e}_{opt}$*; (C) Finally, we interpolate* $\mathbf{e}_{opt}$ *with* $\mathbf{e}_{tgt}$ *to generate the final editing result.*

**Every edit needs to fine-tune pretrained diffusion model and text embedding.**

# Methods – Editing

## Prompt-to-Prompt (2022-08) Prompt-to-Prompt Image Editing with Cross Attention Control

**Motivation:** Pursue an intuitive prompt-to-prompt editing framework, where the edits are controlled by **text only**.



“The boulevards are crowded today.”

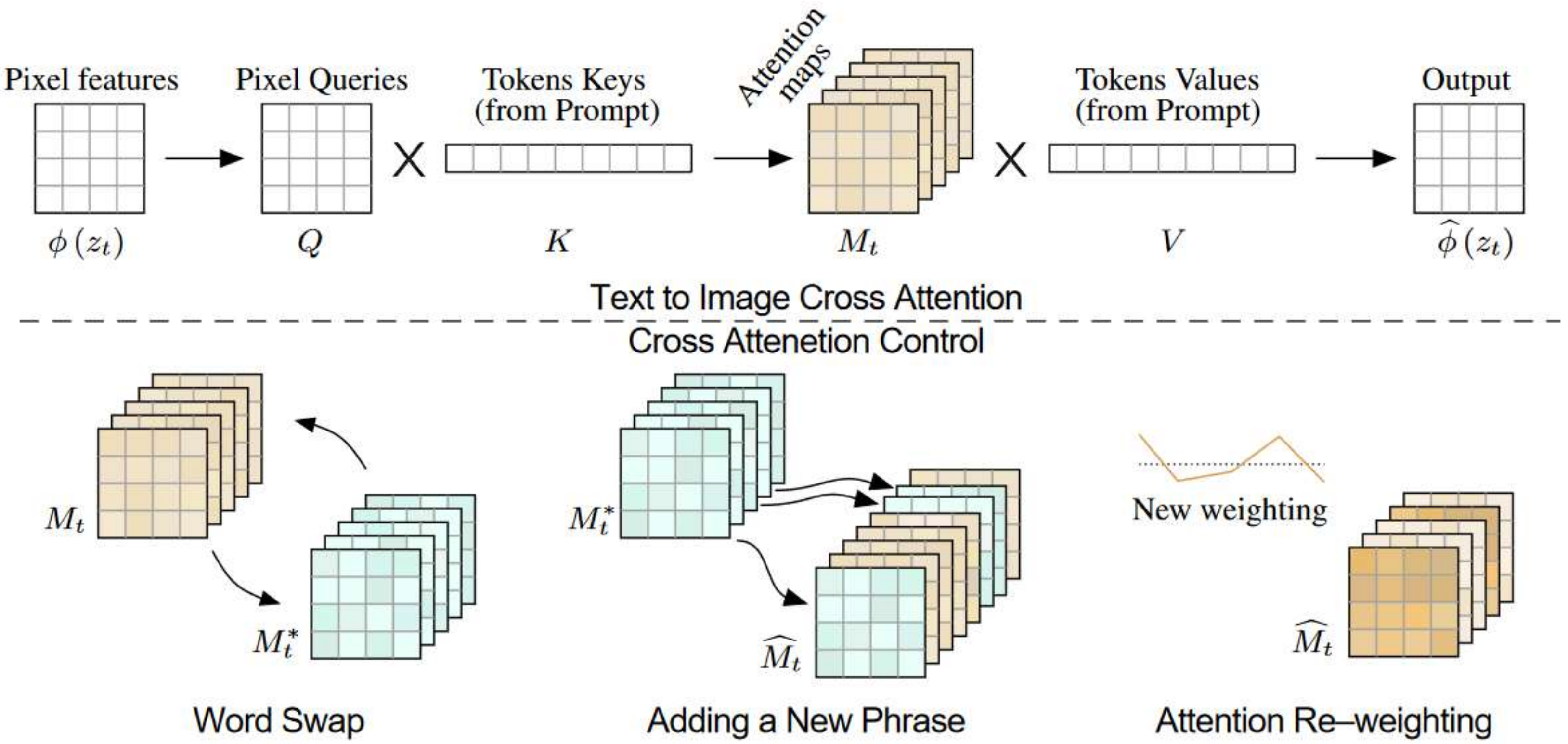
“Photo of a cat riding on a ~~bicycle~~ car.”

“Children drawing of a castle next to a river.”

“a cake with decorations.” jelly beans

# Methods – Editing

**Prompt-to-Prompt (2022-08)** Prompt-to-Prompt Image Editing with Cross Attention Control



A softer attention constrain:

$$Edit(M_t, M_t^*, t) := \begin{cases} M_t^* & \text{if } t < \tau \\ M_t & \text{otherwise.} \end{cases}$$

The composition is determined in the early steps of the diffusion process

Attention Re–weighting

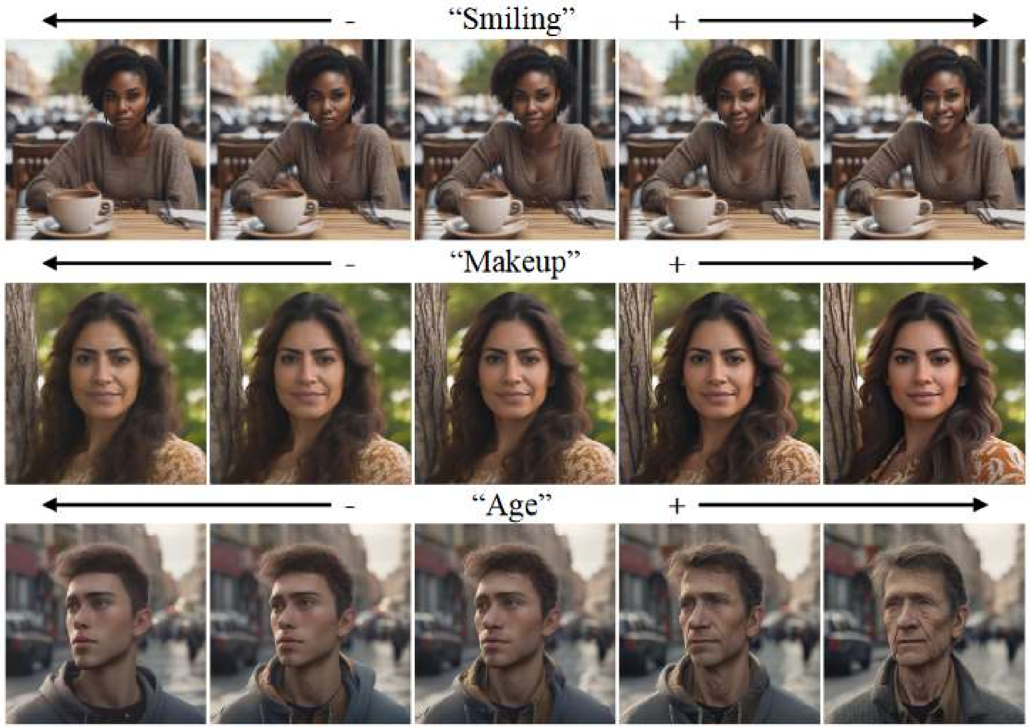
$$(Edit(M_t, M_t^*, t))_{i,j} := \begin{cases} c \cdot (M_t)_{i,j} & \text{if } j = j^* \\ (M_t)_{i,j} & \text{otherwise.} \end{cases}$$

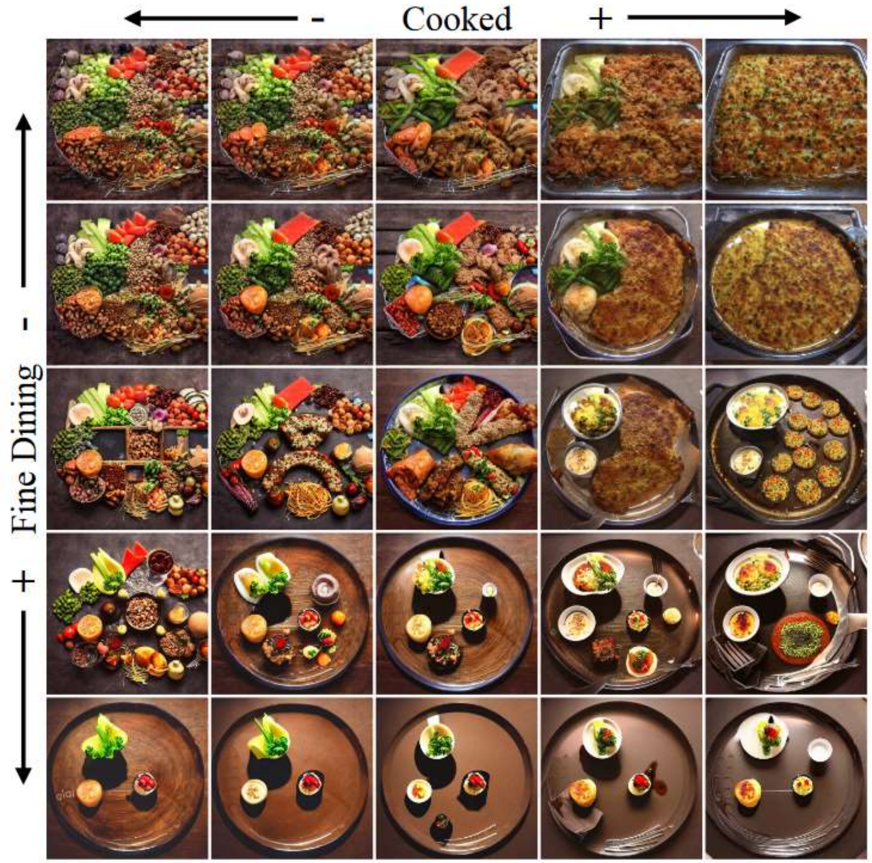
# Methods – Editing

## Concept Sliders (2023-11)

Concept Sliders: LoRA Adaptors for Precise Control in Diffusion Models

**Motivation:** Identify a low-rank parameter direction corresponding to one concept while minimizing interference with other attributes.

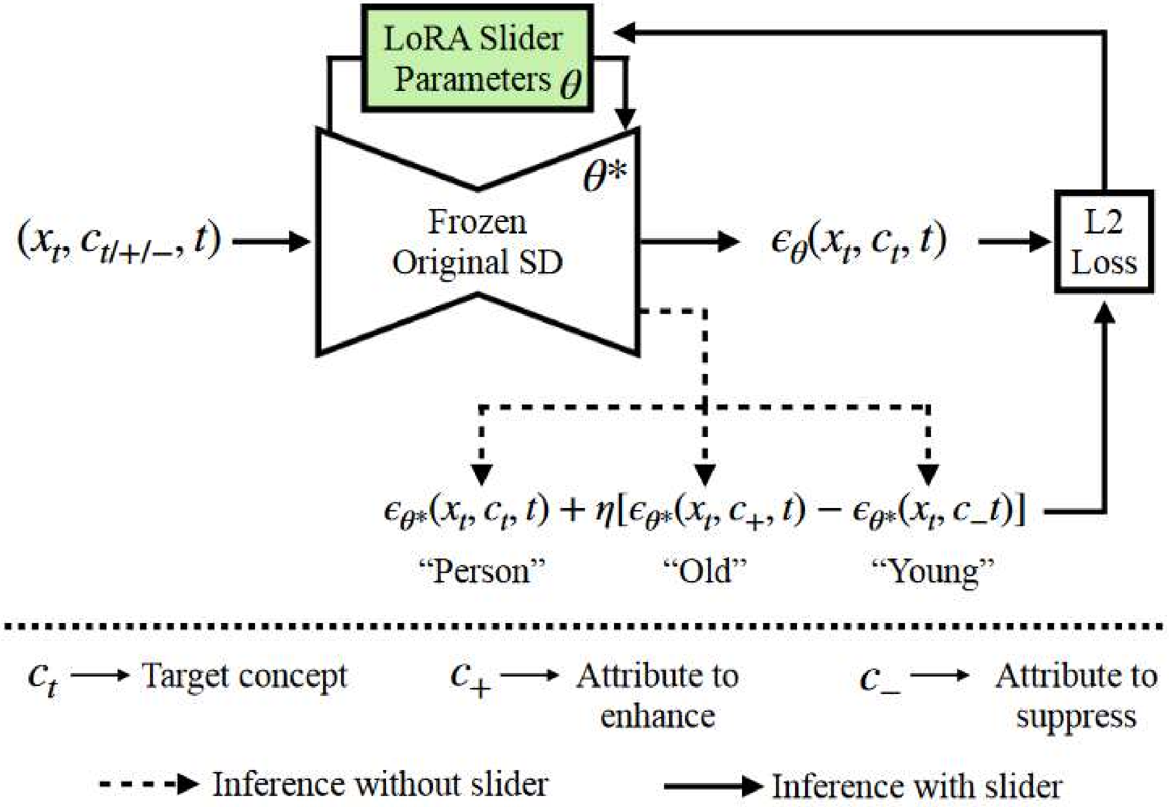




# Methods – Editing

## Concept Sliders (2023-11)

Concept Sliders: LoRA Adaptors for Precise Control in Diffusion Models


LoRA Slider Parameters $\theta$
$\theta*$
Frozen Original SD
$(x_t, c_{t/+/-}, t)$
$\epsilon_\theta(x_t, c_t, t)$
L2 Loss
$\epsilon_{\theta*}(x_t, c_t, t) + \eta[\epsilon_{\theta*}(x_t, c_+, t) - \epsilon_{\theta*}(x_t, c\_t)]$
"Person"
"Old"
"Young"
$c_t$ → Target concept
$c_+$ → Attribute to enhance
$c_-$ → Attribute to suppress
Inference without slider
Inference with slider

The proposed score function shifts the distribution of the target concept ct to exhibit more attributes of $c_+$ and fewer attributes of $c_-$.

$$\epsilon_{\theta^*}(X, c_t, t) \leftarrow \epsilon_\theta(X, c_t, t) + \eta\left(\epsilon_\theta(X, c_+, t) - \epsilon_\theta(X, c_-, t)\right)$$

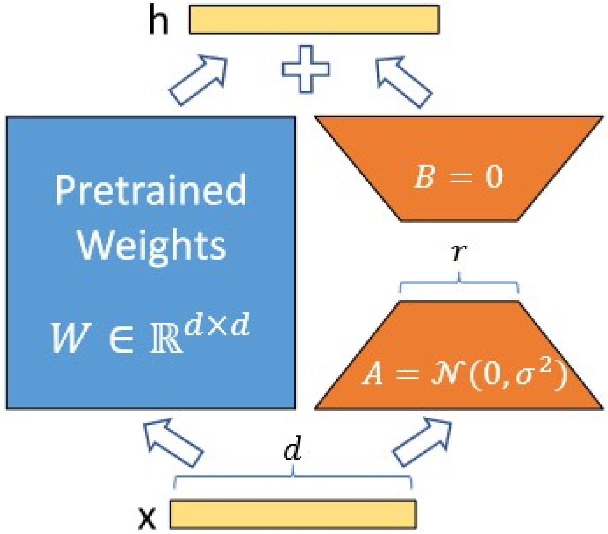
A single prompt pair can sometimes identify a direction that is **entangled** with other **undesired** attributes. We therefore incorporate a set of preservation concepts $p \in$ P (for example, race names while editing age) to constrain the optimization.

$$\epsilon_{\theta^*}(X, c_t, t) \leftarrow \epsilon_\theta(X, c_t, t) + \eta\sum_{p\in\mathcal{P}}\left(\epsilon_\theta(X, (c_+, p), t) - \epsilon_\theta(X, (c_-, p), t)\right)$$

# Aside

**LoRa:** Low-Rank Adaptation

Freezes the pretrained model weights and injects trainable rank decomposition matrices into each layer of the Transformer architecture, greatly reducing the number of trainable parameters for downstream tasks.


h
Pretrained Weights
$W \in \mathbb{R}^{d\times d}$
$B = 0$
$r$
$A = \mathcal{N}(0, \sigma^2)$
$d$
x

For a pre-trained weight matrix $W_0 \in R^{d\times k}$ , we constrain its update by representing the latter with a low-rank decomposition $W_0 + \Delta W = W_0 + BA$, where $B \in R^{d\times r}$, $A \in R^{r\times k}$, and the rank $r \ll \min(d, k)$

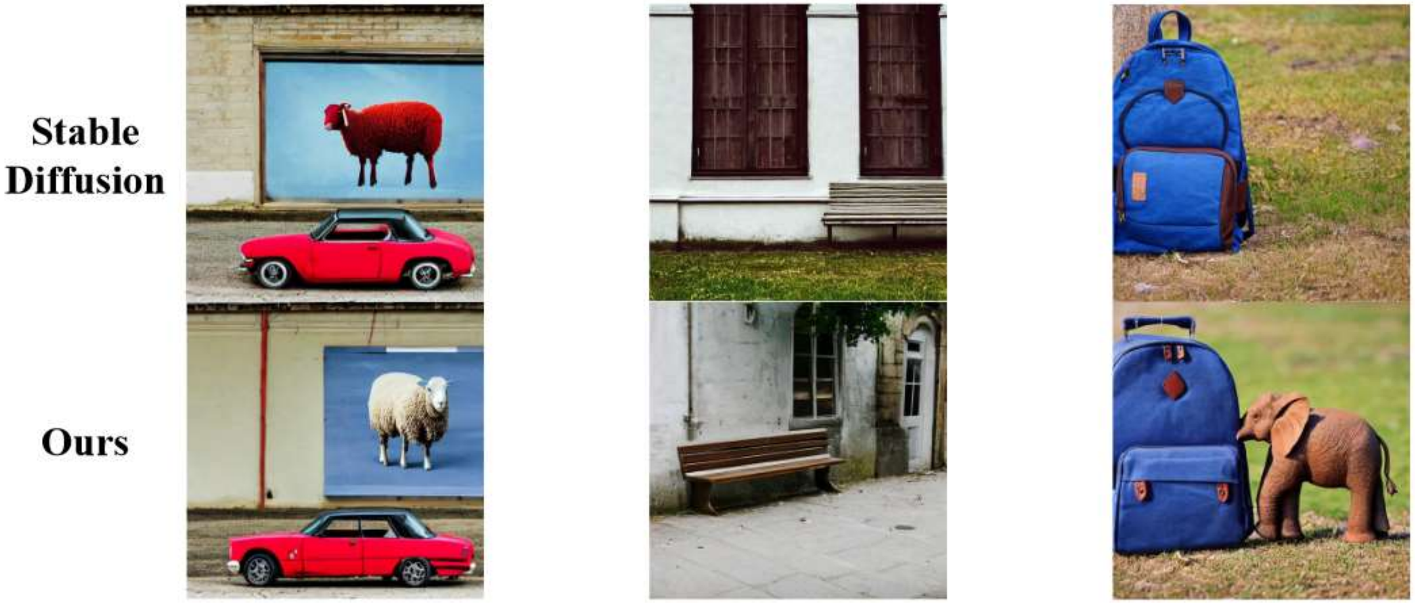
During inference time:

$$h = W_0x + \Delta Wx = W_0x + BAx$$

# Methods – More Faithful to Prompt

## Structured Diffusion (2023-02) Training-Free Structured Diffusion Guidance for Compositional Text-to-Image Synthesis

**Motivation:** Keys and values in cross-attention layers have strong **semantic meanings** associated with object layouts and content. Therefore, by manipulating the cross-attention representations based on linguistic insights, we can better preserve the compositional semantics in the generated image.



*A red car and a white sheep.*

Attribute leakage

*A brown bench sits in front of an old white building*

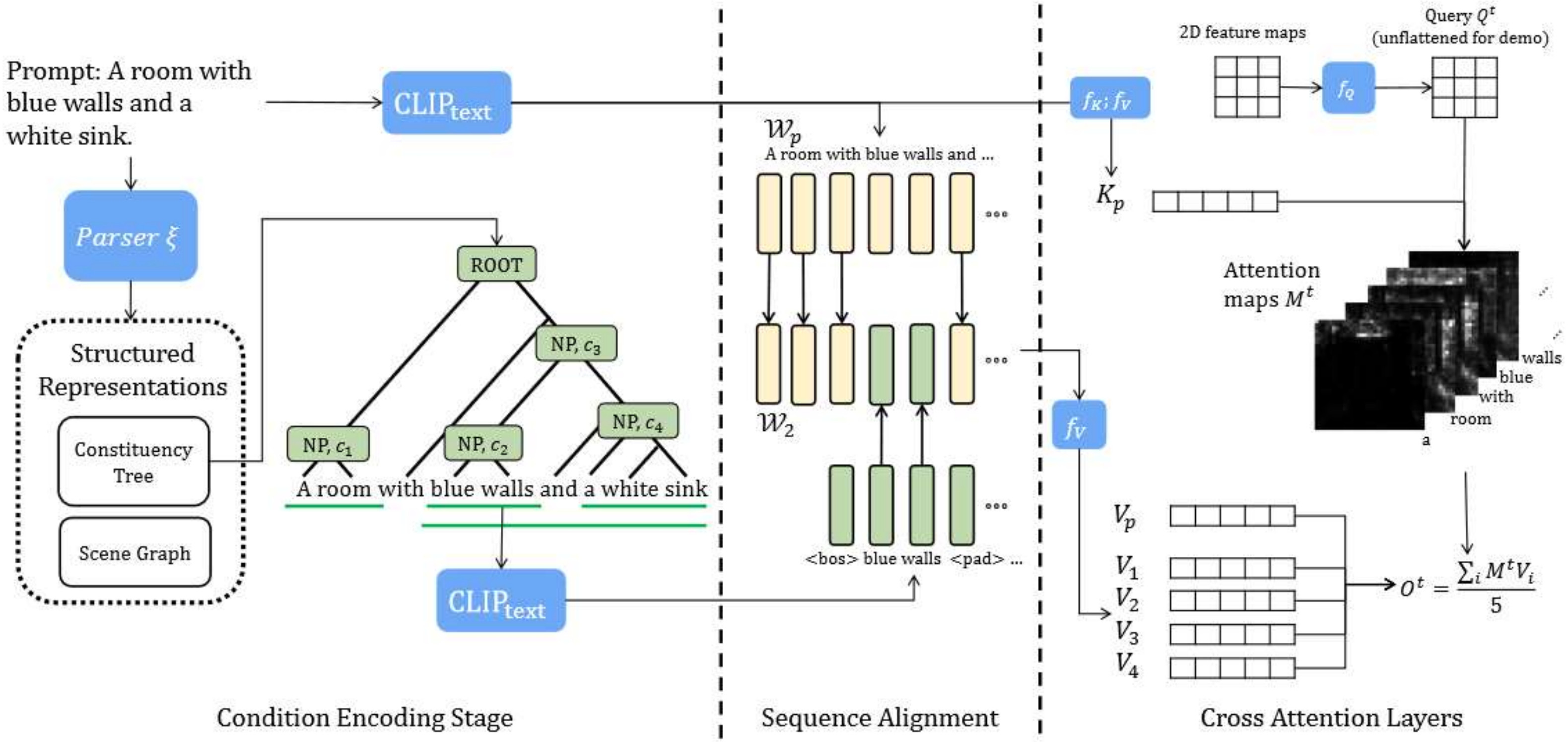
Interchanged attributes

*A blue backpack and a brown elephant*

Missing objects

# Methods – More Faithful to Prompt

## Structured Diffusion (2023-02) Training-Free Structured Diffusion Guidance for Compositional Text-to-Image Synthesis



Extract a collection of concepts from all hierarchical levels, and encode with CLIP text encoder

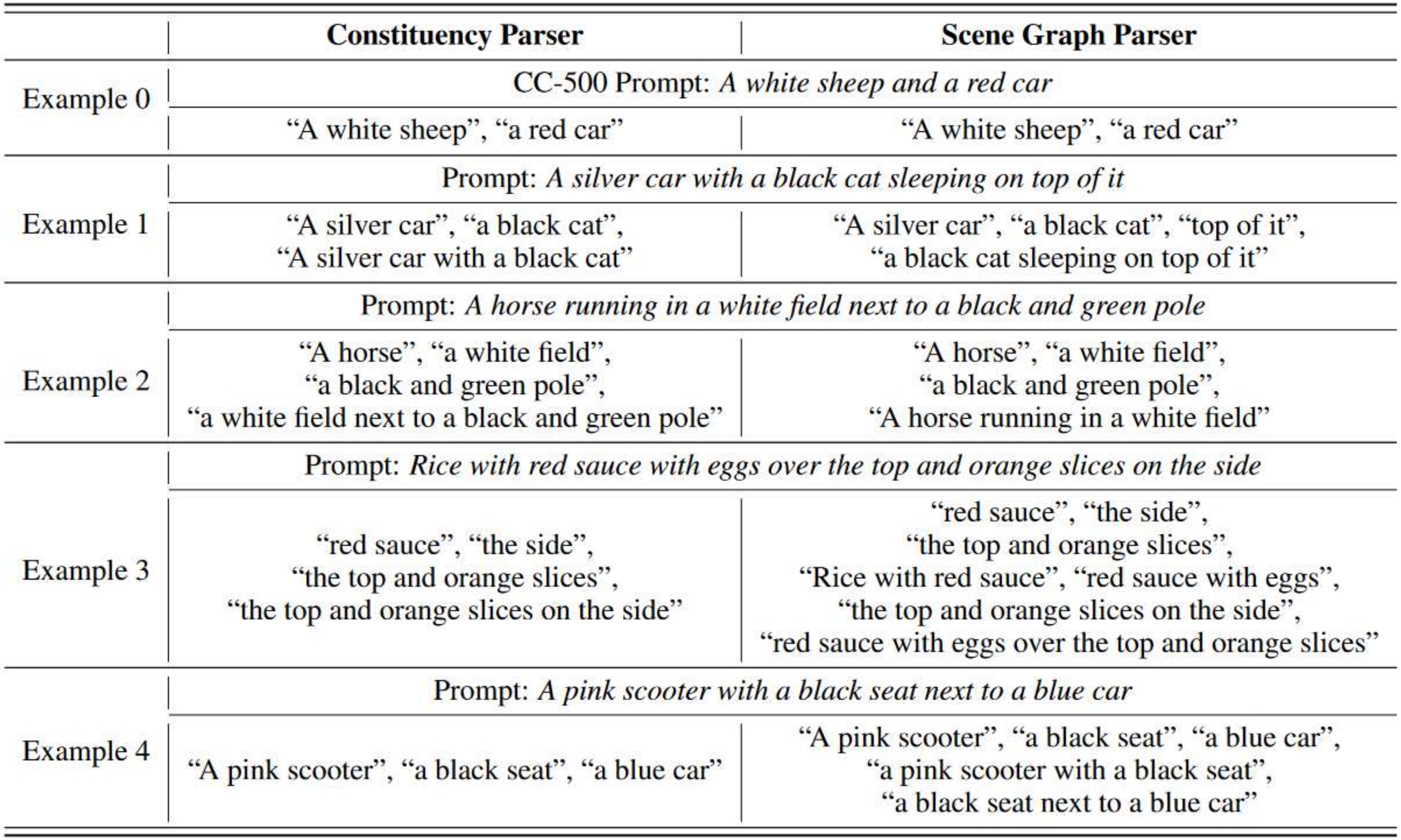
$$\mathbb{W} = [\mathcal{W}_{\text{p}}, \mathcal{W}_1, \mathcal{W}_2, \ldots, \mathcal{W}_k],\ \mathcal{W}_i = \text{CLIP}_{\text{text}}(c_i),\ i = 1, \ldots k.$$

Embeddings between⟨$bos$⟩and⟨$pad$⟩are inserted into $W_p$ to create a new sequence, denoted as $\overline{W_i}$ .

$$\mathbb{V} = [f_V(\mathcal{W}_{\text{p}}), f_V(\overline{\mathcal{W}}_1), \ldots, f_V(\overline{\mathcal{W}}_k)] = [V_{\text{p}}, V_1, \ldots, V_k]. \qquad O^t = \frac{1}{(k+1)} \sum_i (M^t V_i), i = \text{p}, 1, 2, \ldots, k.$$

# Methods – More Faithful to Prompt

## Structured Diffusion (2023-02) Training-Free Structured Diffusion Guidance for Compositional Text-to-Image Synthesis

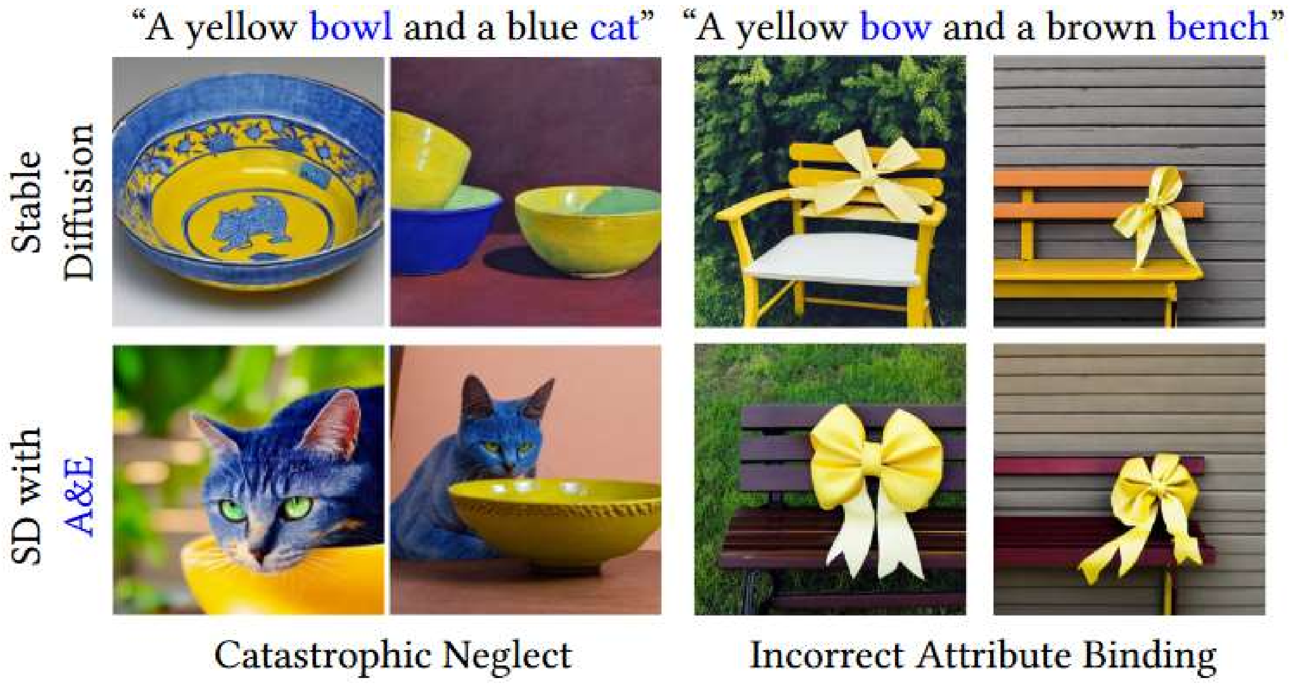


| | Constituency Parser | Scene Graph Parser |
|---|---|---|
| Example 0 | CC-500 Prompt: *A white sheep and a red car* | |
| | "A white sheep", "a red car" | "A white sheep", "a red car" |
| Example 1 | Prompt: *A silver car with a black cat sleeping on top of it* | |
| | "A silver car", "a black cat",<br>"A silver car with a black cat" | "A silver car", "a black cat", "top of it",<br>"a black cat sleeping on top of it" |
| Example 2 | Prompt: *A horse running in a white field next to a black and green pole* | |
| | "A horse", "a white field",<br>"a black and green pole",<br>"a white field next to a black and green pole" | "A horse", "a white field",<br>"a black and green pole",<br>"A horse running in a white field" |
| Example 3 | Prompt: *Rice with red sauce with eggs over the top and orange slices on the side* | |
| | "red sauce", "the side",<br>"the top and orange slices",<br>"the top and orange slices on the side" | "red sauce", "the side",<br>"the top and orange slices",<br>"Rice with red sauce", "red sauce with eggs",<br>"the top and orange slices on the side",<br>"red sauce with eggs over the top and orange slices" |
| Example 4 | Prompt: *A pink scooter with a black seat next to a blue car* | |
| | "A pink scooter", "a black seat", "a blue car" | "A pink scooter", "a black seat", "a blue car",<br>"a pink scooter with a black seat",<br>"a black seat next to a blue car" |

# Methods – More Faithful to Prompt

## Attend-and-Excite (2023-05) Attend-and-Excite: Attention-Based Semantic Guidance for Text-to-Image Diffusion Models

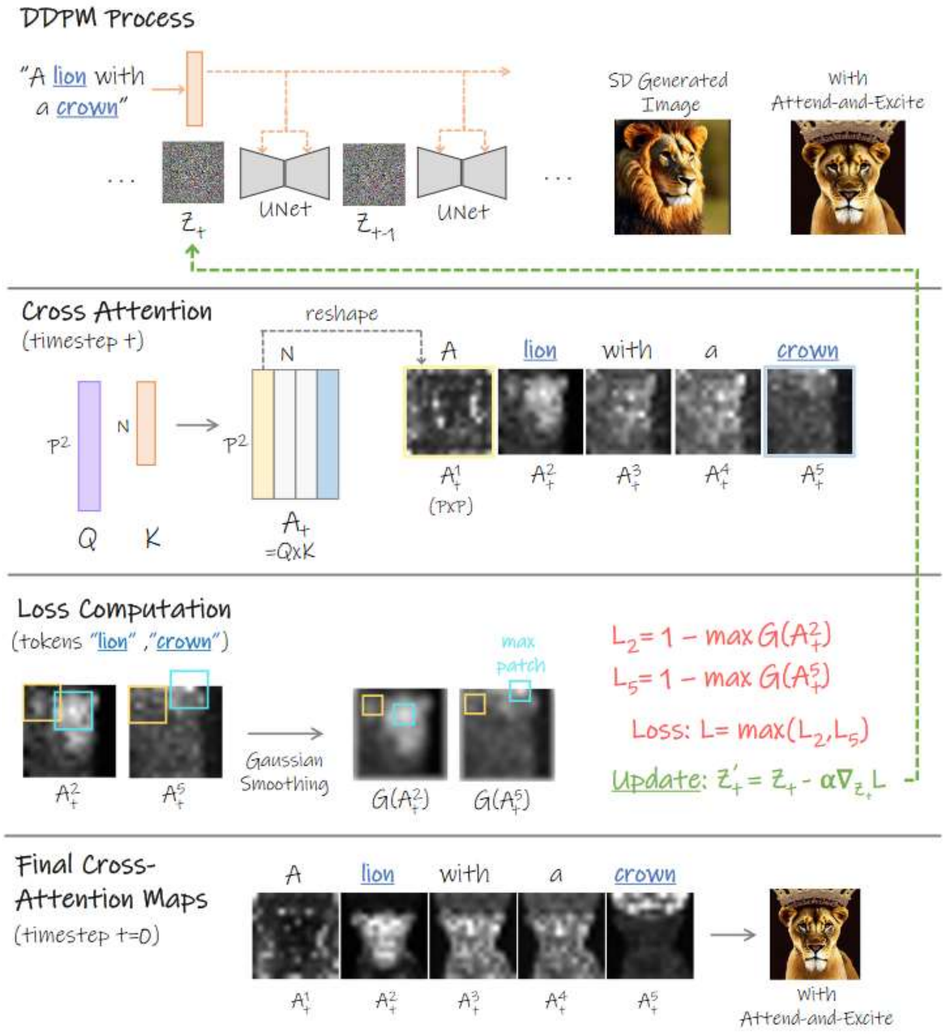
**Motivation:** Current state-of-the-art diffusion models may still fail in generating images that fully convey the semantics in the given text prompt. These models mainly have two failures: **Catastrophic Neglect** and **Incorrect Attribute Binding**


"A yellow bowl and a blue cat"
"A yellow bow and a brown bench"
Stable Diffusion
SD with A&E
Catastrophic Neglect
Incorrect Attribute Binding

# Methods – More Faithful to Prompt

## Attend-and-Excite (2023-05)

Attend-and-Excite: Attention-Based Semantic Guidance for Text-to-Image Diffusion Models



### Extracting the Cross-Attention Maps

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right) \cdot V$$

The resulting aggregated map $A_t$ contains $N$ spatial attention maps, one for each of the tokens of $\mathcal{P}$

### Obtaining Smooth Attention Maps

The model may **not generate the full** subject, but rather a patch that resembles some part of the subject. We apply a Gaussian filter, so that the attention value of the maximally-activated patch is dependent on its neighboring patches

$$A_t^s \leftarrow \text{Gaussian}(A_t^s)$$

### Performing On the Fly Optimization

For each subject token in $S$, our optimization encourages the existence of at least one patch of $A_t^s$ with a high activation value.

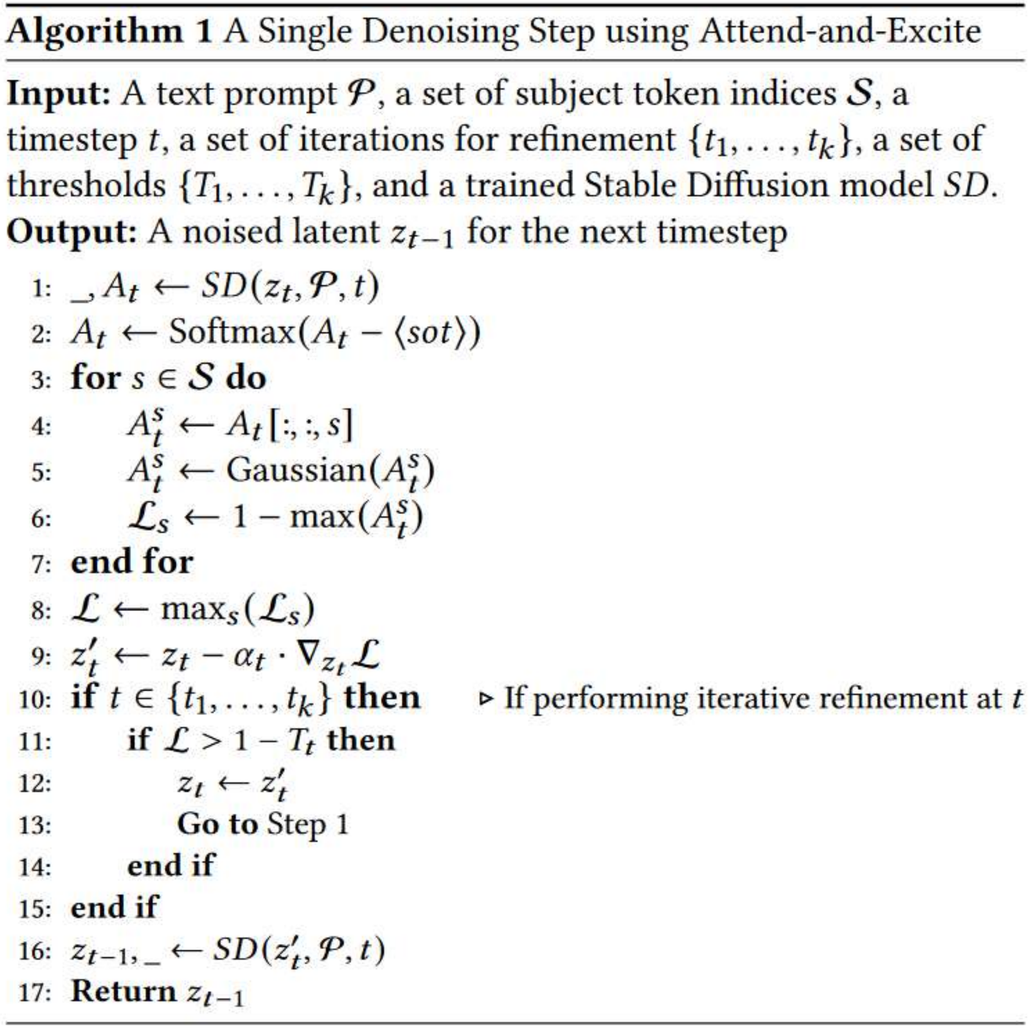
$$\mathcal{L} = \max_{s \in S} \mathcal{L}_s \qquad \text{where} \qquad \mathcal{L}_s = 1 - \max(A_t^s).$$

Shift the current latent $z_t$ by

$$z_t' \leftarrow z_t - \alpha_t \cdot \nabla_{z_t} \mathcal{L},$$

# Methods – More Faithful to Prompt

## Attend-and-Excite (2023-05) Attend-and-Excite: Attention-Based Semantic Guidance for Text-to-Image Diffusion Models



**Algorithm 1** A Single Denoising Step using Attend-and-Excite

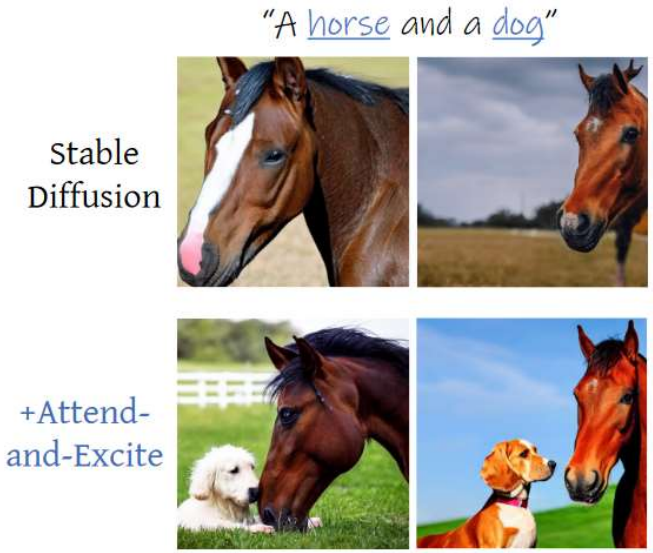
**Input:** A text prompt $\mathcal{P}$, a set of subject token indices $\mathcal{S}$, a timestep $t$, a set of iterations for refinement $\{t_1, \dots, t_k\}$, a set of thresholds $\{T_1, \dots, T_k\}$, and a trained Stable Diffusion model $SD$.

**Output:** A noised latent $z_{t-1}$ for the next timestep

```
1:  $\_, A_t \leftarrow SD(z_t, \mathcal{P}, t)$
2:  $A_t \leftarrow \text{Softmax}(A_t - \langle sot \rangle)$
3:  for $s \in \mathcal{S}$ do
4:      $A_t^s \leftarrow A_t[:, :, s]$
5:      $A_t^s \leftarrow \text{Gaussian}(A_t^s)$
6:      $\mathcal{L}_s \leftarrow 1 - \max(A_t^s)$
7:  end for
8:  $\mathcal{L} \leftarrow \max_s(\mathcal{L}_s)$
9:  $z_t' \leftarrow z_t - \alpha_t \cdot \nabla_{z_t} \mathcal{L}$
10: if $t \in \{t_1, \dots, t_k\}$ then        ▷ If performing iterative refinement at $t$
11:     if $\mathcal{L} > 1 - T_t$ then
12:         $z_t \leftarrow z_t'$
13:         Go to Step 1
14:     end if
15: end if
16: $z_{t-1}, \_ \leftarrow SD(z_t', \mathcal{P}, t)$
17: Return $z_{t-1}$
```

If the attention values of a token do not **reach a certain value** in the **early** denoising stages, the corresponding object will not be generated.

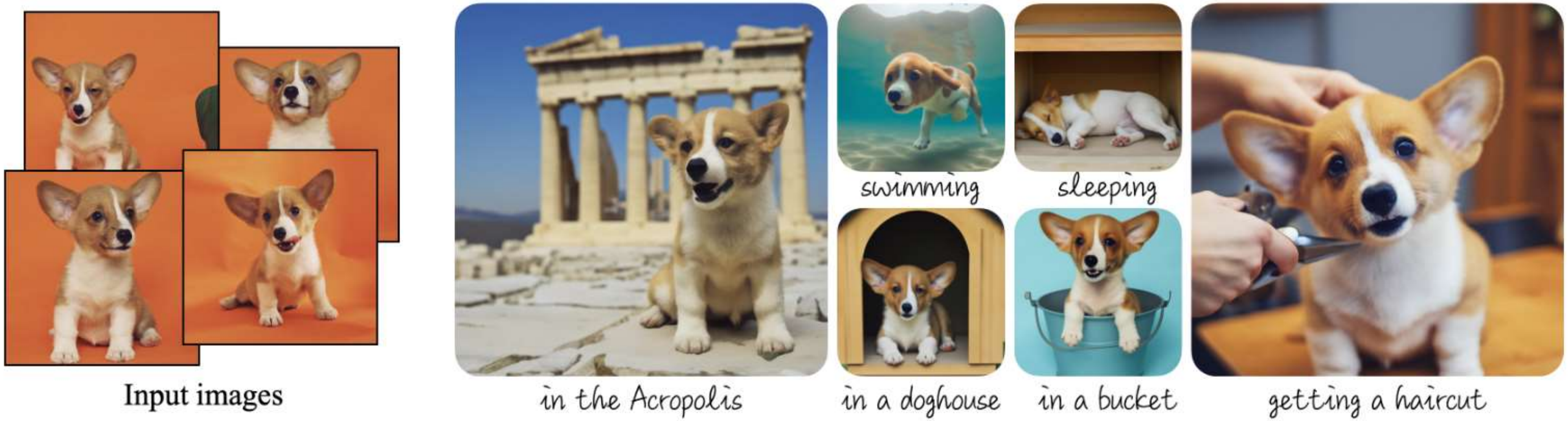
We iteratively update $z_t$ until a pre-defined **minimum attention value** is achieved for all subject tokens.



# Methods – Customization

## DreamBooth (2022-08) DreamBooth: Fine Tuning Text-to-Image Diffusion Models for Subject-Driven Generation

**Motivation:** These diffusion models lack the ability to mimic the appearance of subjects in a given reference set and synthesize novel renditions of them in different contexts.

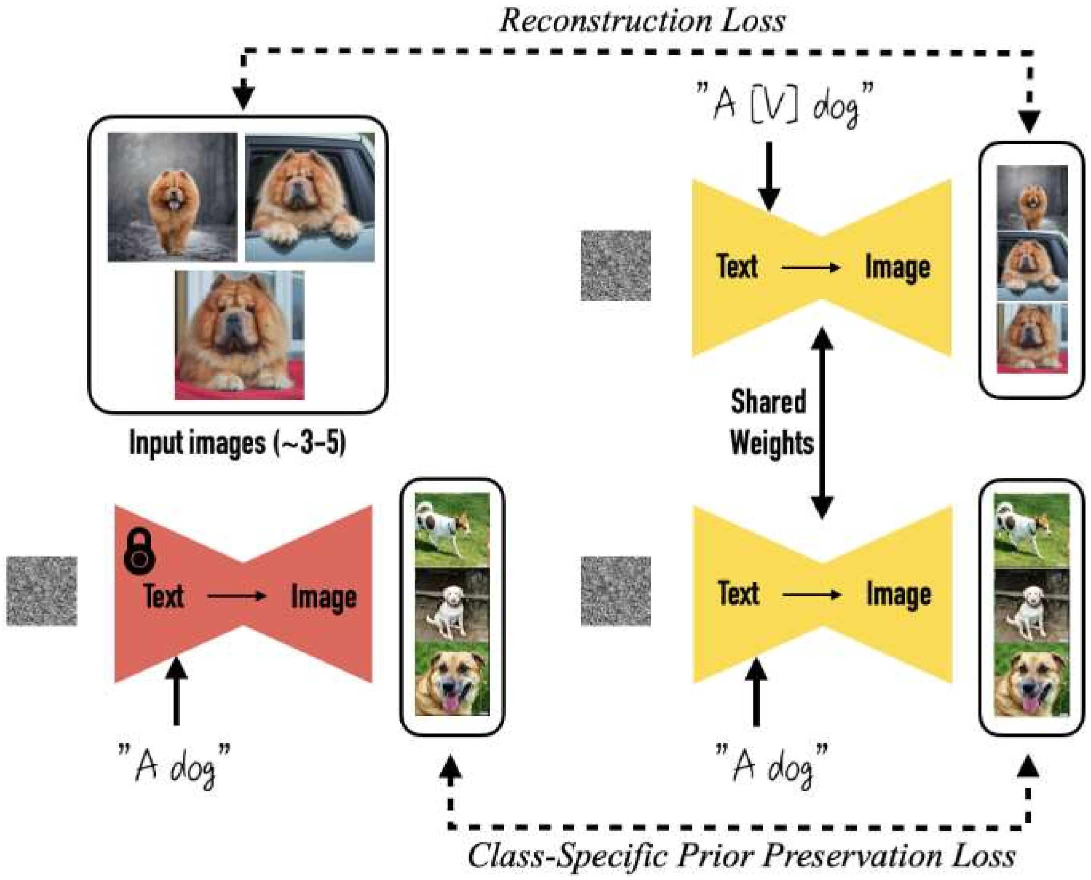


I want **[that]** in different contexts…

# Methods – Customization

## DreamBooth (2022-08)

DreamBooth: Fine Tuning Text-to-Image Diffusion Models for Subject-Driven Generation



We use a simple structure to refer a **customized** concept or a special object

"a [identifier] [class noun]"

This [identifier] has to be **rare.**

Find rare tokens in the vocabulary, and then invert these tokens into text space, in order to **minimize** the **probability** of the identifier having a **strong prior.**

### Class-specific Prior Preservation Loss

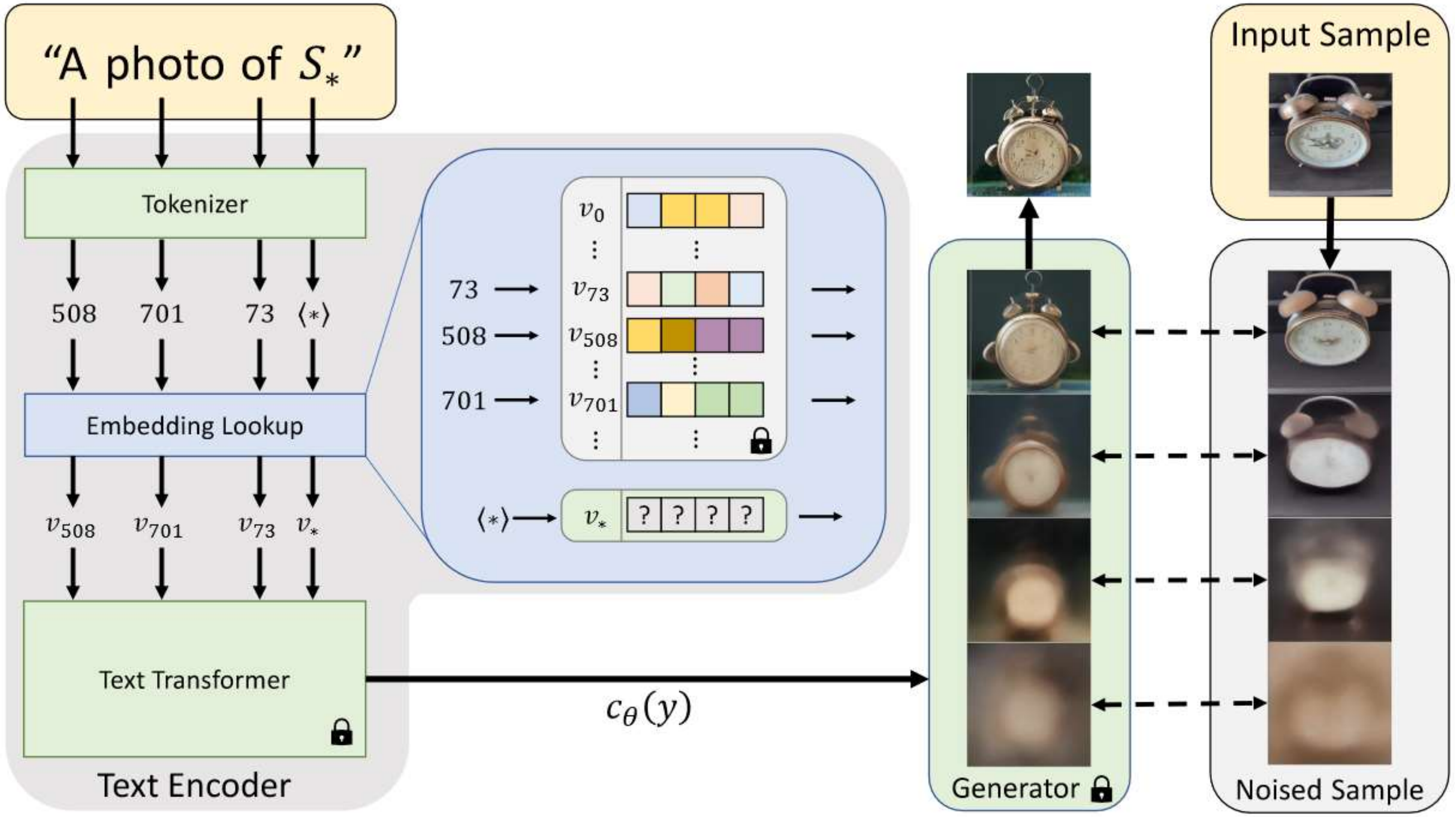
**Language drift**: model slowly forgets how to generate subjects of the same class as the target subject.

$$\mathbb{E}_{\mathbf{x},\mathbf{c},\boldsymbol{\epsilon},\boldsymbol{\epsilon}',t}[w_t\|\hat{\mathbf{x}}_\theta(\alpha_t\mathbf{x}+\sigma_t\boldsymbol{\epsilon},\mathbf{c})-\mathbf{x}\|_2^2+ \lambda w_{t'}\|\hat{\mathbf{x}}_\theta(\alpha_{t'}\mathbf{x}_{\text{pr}}+\sigma_{t'}\boldsymbol{\epsilon}',\mathbf{c}_{\text{pr}})-\mathbf{x}_{\text{pr}}\|_2^2],$$

Where $\mathbf{c}_{\text{pr}} := \Gamma(f(\text{"a [class noun]"}))$

# Methods – Customization

**Textual Inversion (2022-08)** An Image is Worth One Word: Personalizing Text-to-Image Generation using Textual Inversion

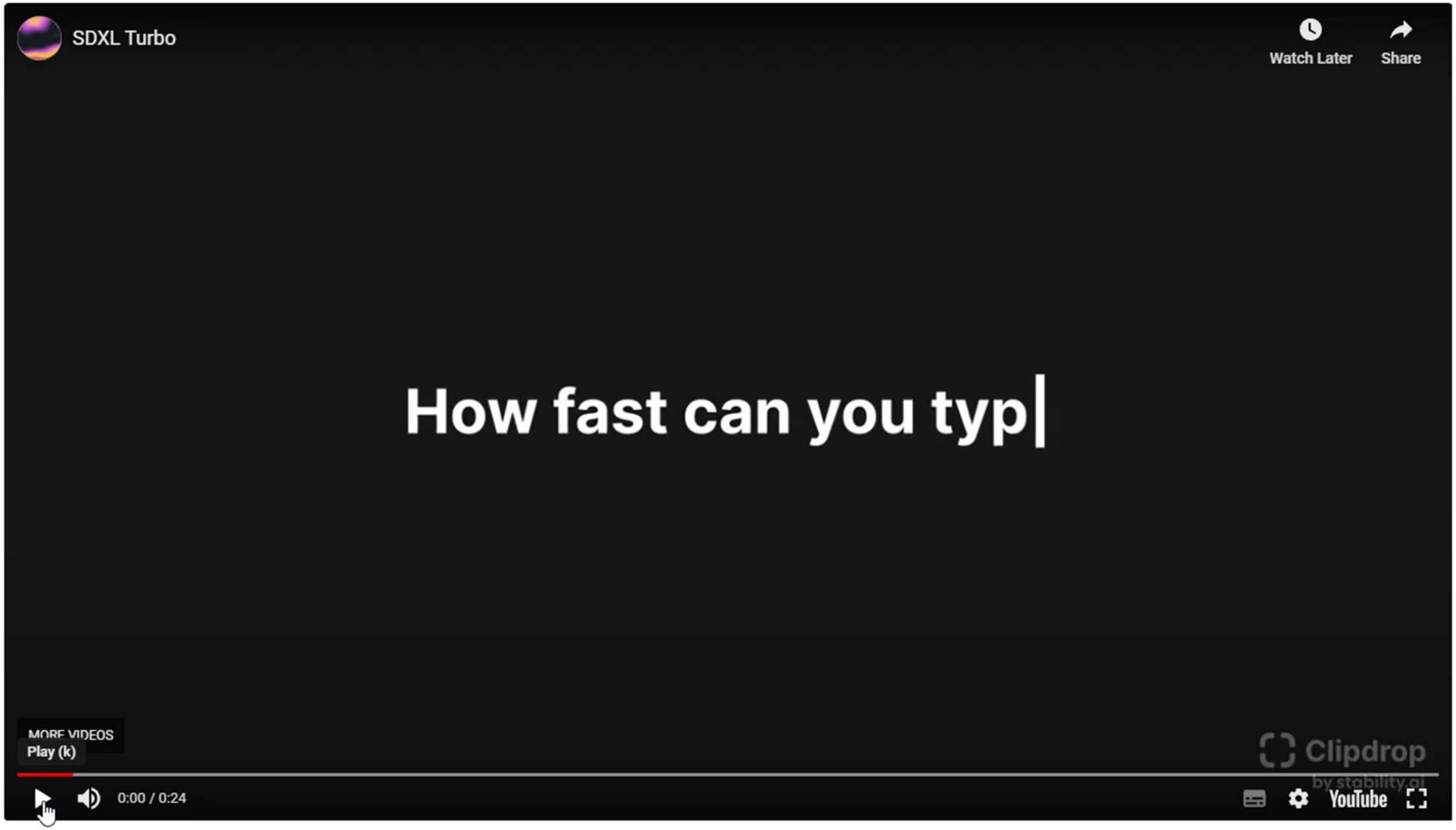


Rather than fine-tuning the whole diffusion model, Textual Inversion only learn a special text embedding

$$v_* = \arg\min_v \mathbb{E}_{z\sim\mathcal{E}(x),y,\epsilon\sim\mathcal{N}(0,1),t}\left[\|\epsilon - \epsilon_\theta(z_t, t, c_\theta(y))\|_2^2\right],$$

# Methods – Real Time Text2img Generation

## Adversarial Diffusion Distillation (2023-11-28)



How fast can you typ

# Thanks