Introduction to Deep Generative Modeling

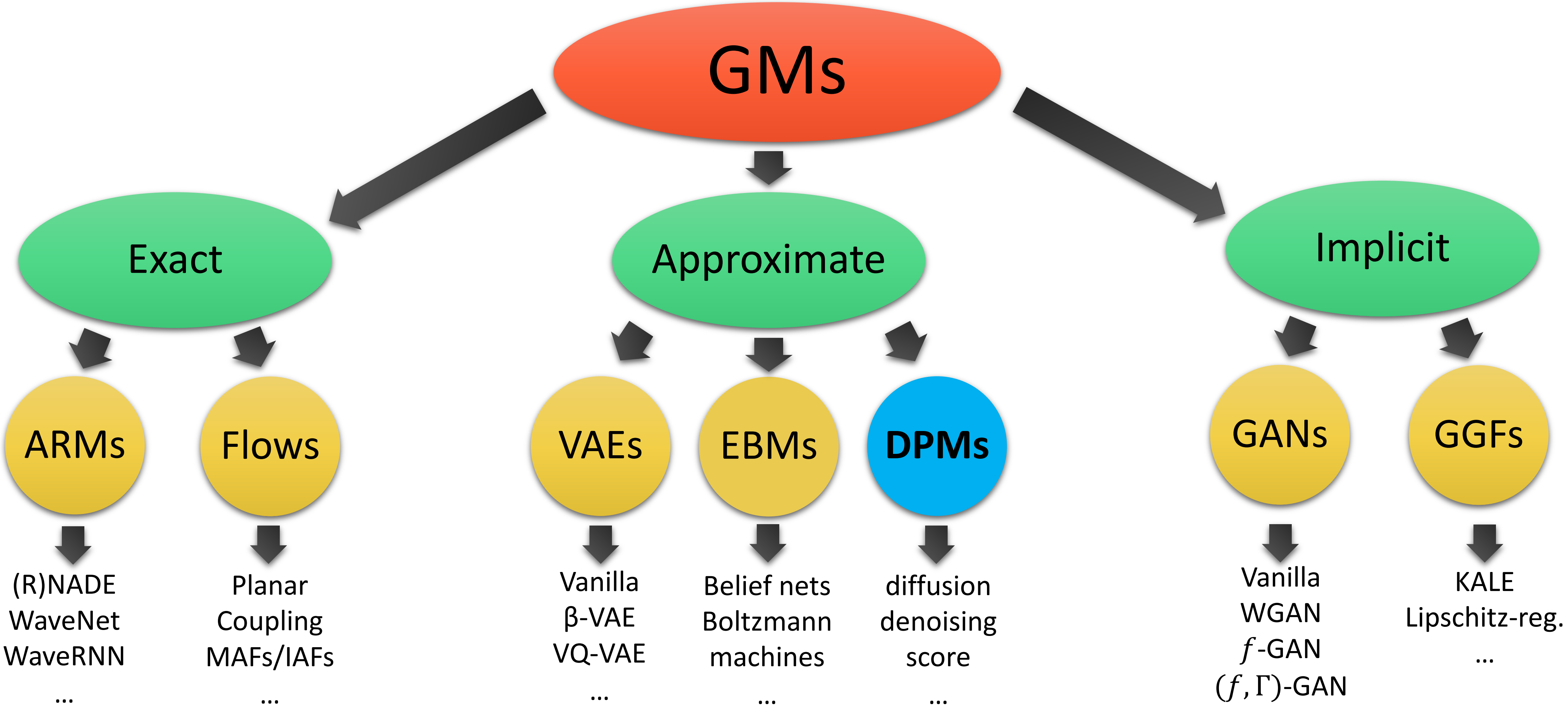
Lecture #15

HY-673 – Computer Science Dep., University of Crete

Professors: Yannis Pantazis, Yannis Stylianou

Teaching Assistant: Michail Raptakis

Taxonomy of GMs

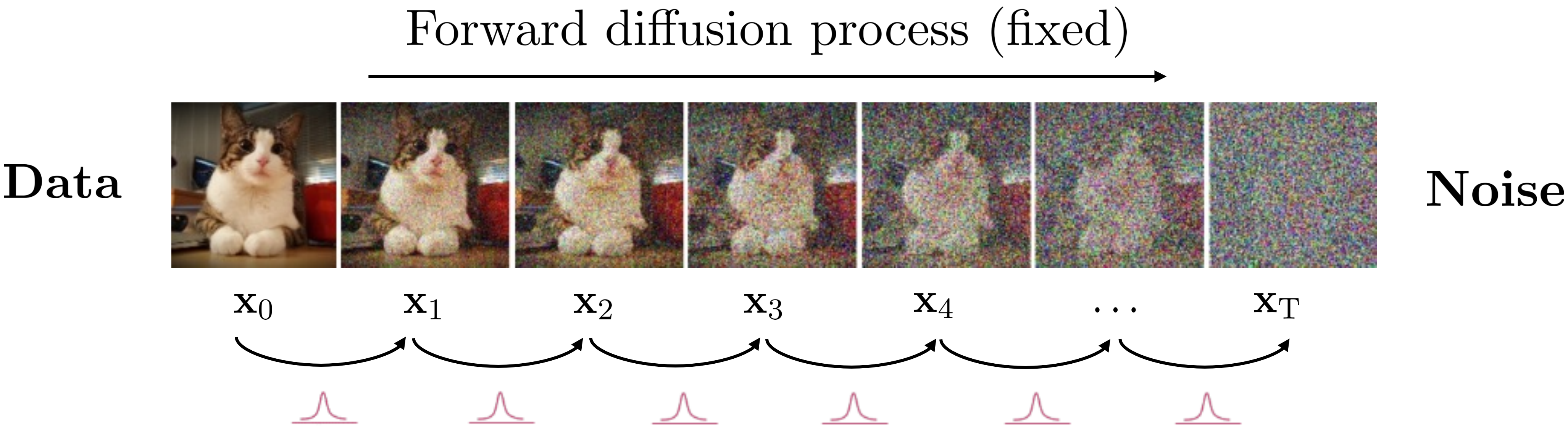


GMs

- Exact
  - ARMs
    - (R)NADE
    - WaveNet
    - WaveRNN
    - …
  - Flows
    - Planar
    - Coupling
    - MAFs/IAFs
    - …
- Approximate
  - VAEs
    - Vanilla
    - β-VAE
    - VQ-VAE
    - …
  - EBMs
    - Belief nets
    - Boltzmann machines
    - …
  - **DPMs**
    - diffusion
    - denoising
    - score
    - …
- Implicit
  - GANs
    - Vanilla
    - WGAN
    - f-GAN
    - (f, Γ)-GAN
  - GGFs
    - KALE
    - Lipschitz-reg.
    - …

Recap: Forward Diffusion Process

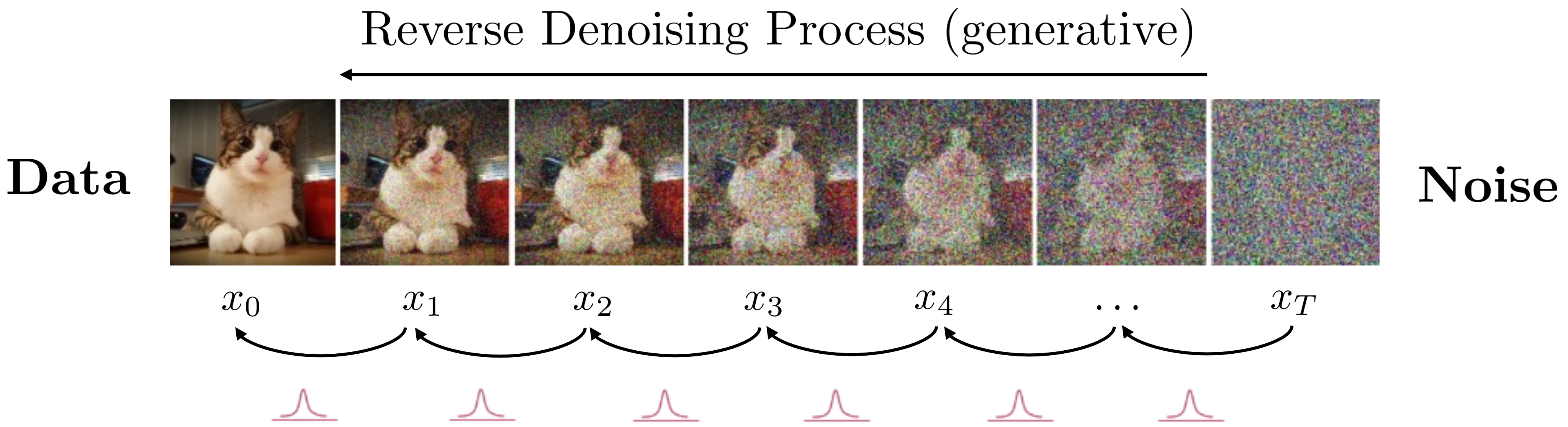
The forward diffusion process:



$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I)$$

Recap: Reverse Denoising Process

The formal definition of the reverse process in T steps:



$$p(x_T) = \mathcal{N}(x_T; 0, I_d)$$

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \underbrace{\mu_\theta(x_t, t)}_{\substack{\text{Trainable network} \\ \text{(U-net, Denoising Autoencoder)}}}, \sigma_t^2 I_d) \quad \Rightarrow \quad p_\theta(x_{0:T}) = p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t).$$

$\approx q(x_{t-1}|x_t)$ (true posterior; intractable)

Recap: Training and Sampling

Minimize a simplification of negative ELBO:

$$\mathrm{L}_{\text{simple}} = \mathbb{E}_{x_0 \sim p_d(x_0), \epsilon \sim \mathcal{N}(0, I_d), t \sim \mathcal{U}(1, \mathrm{T})} \left[||\epsilon - \epsilon_\theta(\underbrace{\sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon}_{x_t}, t)||^2 \right].$$

Algorithm 1 Training

```
repeat
    $x_0 \sim p_d(x_0)$
    $t \sim \text{Uniform}(1, \ldots, T)$
    $\epsilon \sim \mathcal{N}(0, I_d)$
    Take gradient descent step on
        $\boxed{\nabla_\theta ||\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon, t)||^2}$
until converged
```

Algorithm 2 Sampling

```
$x_\mathrm{T} \sim \mathcal{N}(0, I_d)$
for $t = T, \ldots, 1$ do
    $z \sim \mathcal{N}(0, I_d)$
    $\boxed{x_{t-1} = \frac{1}{\sqrt{1-\beta_t}}(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(x_t, t)) + \sigma_t z}$
end for
return $x_0$
```

Applications

- There are many successful applications of diffusion models (in constantly growing numbers):
  - **Image generation**, **text-to-image generation**, **controllable generation**.
  - Image editing, image-to-image translation, super-resolution, segmentation, adversarial robustness.
  - Discrete models, 3D generation, medical imaging, video synthesis.
- Key enabler by diffusion models: Perform high-resolution conditional generation!

Conditional Diffusion Models: Include Condition as Input to Reverse Process

- Reverse Process:

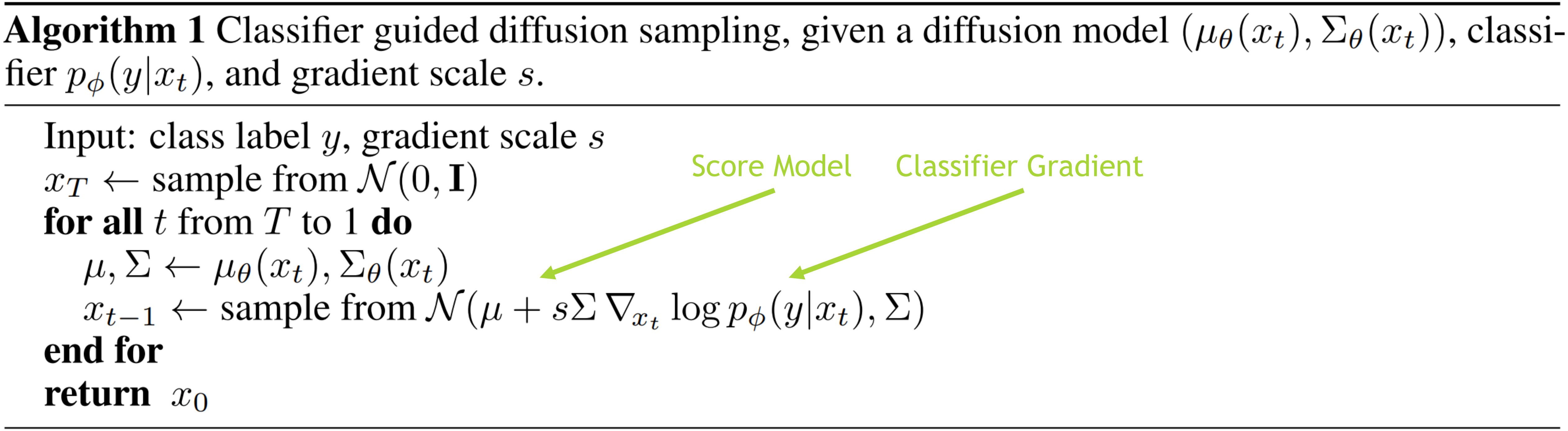
$$p_\theta(x_{0:T}|c) = p(x_T) = \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t, c),\ \ p_\theta(x_{t-1}|x_t, c) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t, c), \Sigma(x_t, t, c)).$$

- Variational Upper Bound:

$$L_\theta(x_0|c) = \mathbb{E}_q\lceil L_T(x_0) + \sum_{t>1} D_{\mathrm{KL}}\left(q(x_{t-1}|x_t, x_0)||p(x_{t-1}|x_t, c)\right) - \log p_\theta(x_0|x_1, c)\rceil.$$

- Incorporate Conditions into U-Net:
  - Scalar conditioning: Encode scalar as a vector embedding, simple spatial addition or adaptive group normalization layers.
  - Image conditioning: Channel-wise concatenation of the conditional image.
  - Text conditioning: Single vector embedding – spatial addition or adaptive group norm / a sequence of vector embeddings - cross-attention.

Classifier-Guided Conditional Diffusion Models: Using the Gradient of a Trained Classifier as Guidance



Algorithm 1 Classifier guided diffusion sampling, given a diffusion model $(\mu_\theta(x_t), \Sigma_\theta(x_t))$, classifier $p_\phi(y|x_t)$, and gradient scale s.

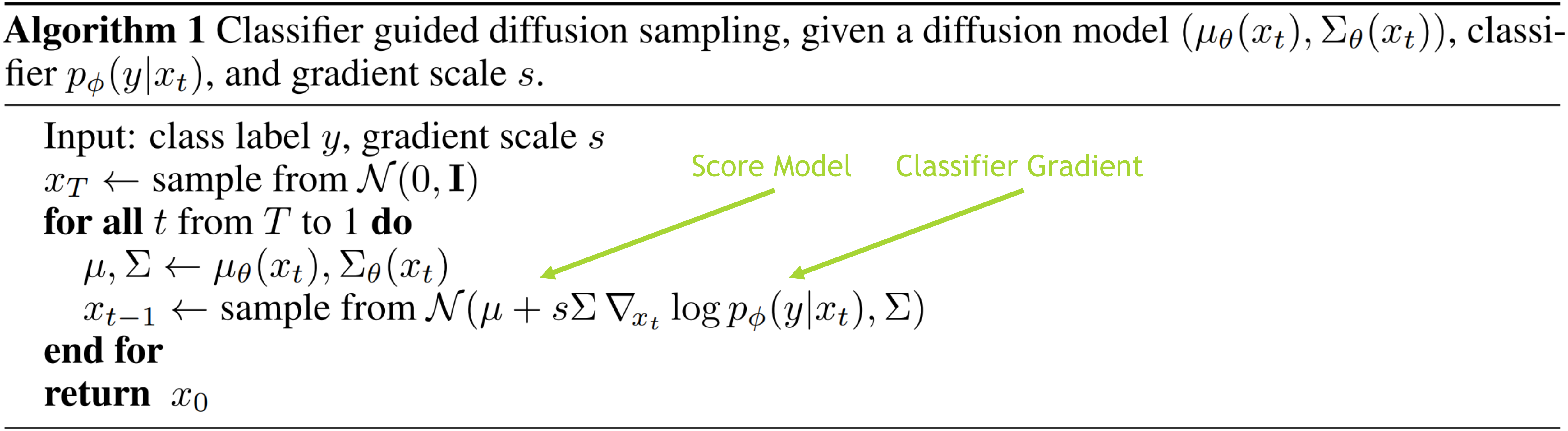
```
Input: class label $y$, gradient scale $s$
$x_T \leftarrow$ sample from $\mathcal{N}(0, \mathbf{I})$
for all $t$ from $T$ to 1 do
    $\mu, \Sigma \leftarrow \mu_\theta(x_t), \Sigma_\theta(x_t)$
    $x_{t-1} \leftarrow$ sample from $\mathcal{N}(\mu + s\Sigma\, \nabla_{x_t} \log p_\phi(y|x_t), \Sigma)$
end for
return $x_0$
```

Score Model

Classifier Gradient

- For class-conditional modeling of $p(x_t|c)$, train an extra classifier $p(c|x_t)$.
- Mix its gradient with the diffusion/score model during sampling.

Classifier-Guided Conditional Diffusion Models: Using the Gradient of a Trained Classifier as Guidance



Algorithm 1 Classifier guided diffusion sampling, given a diffusion model $(\mu_\theta(x_t), \Sigma_\theta(x_t))$, classifier $p_\phi(y|x_t)$, and gradient scale s.

```
Input: class label $y$, gradient scale $s$
$x_T \leftarrow$ sample from $\mathcal{N}(0, \mathbf{I})$
**for all** $t$ from $T$ to 1 **do**
    $\mu, \Sigma \leftarrow \mu_\theta(x_t), \Sigma_\theta(x_t)$
    $x_{t-1} \leftarrow$ sample from $\mathcal{N}(\mu + s\Sigma \nabla_{x_t} \log p_\phi(y|x_t), \Sigma)$
**end for**
**return** $x_0$
```

Score Model

Classifier Gradient

- Sample with a modified score: $\nabla_{x_t} [\log p(x_t|c) + \omega \log p(c|x_t)]$.
- Approximate samples from the distribution $\tilde{p}(x_t|c) \propto p(x_t|c)p(c|x_t)^\omega$.

Classifier-Free Conditional Diffusion Models: Guidance by Bayes' Rule on Conditional Diffusion Models

$$p(c|x_t) \propto \frac{p(x_t|c)}{p(x_t)}.$$

← Conditional Diffusion Model

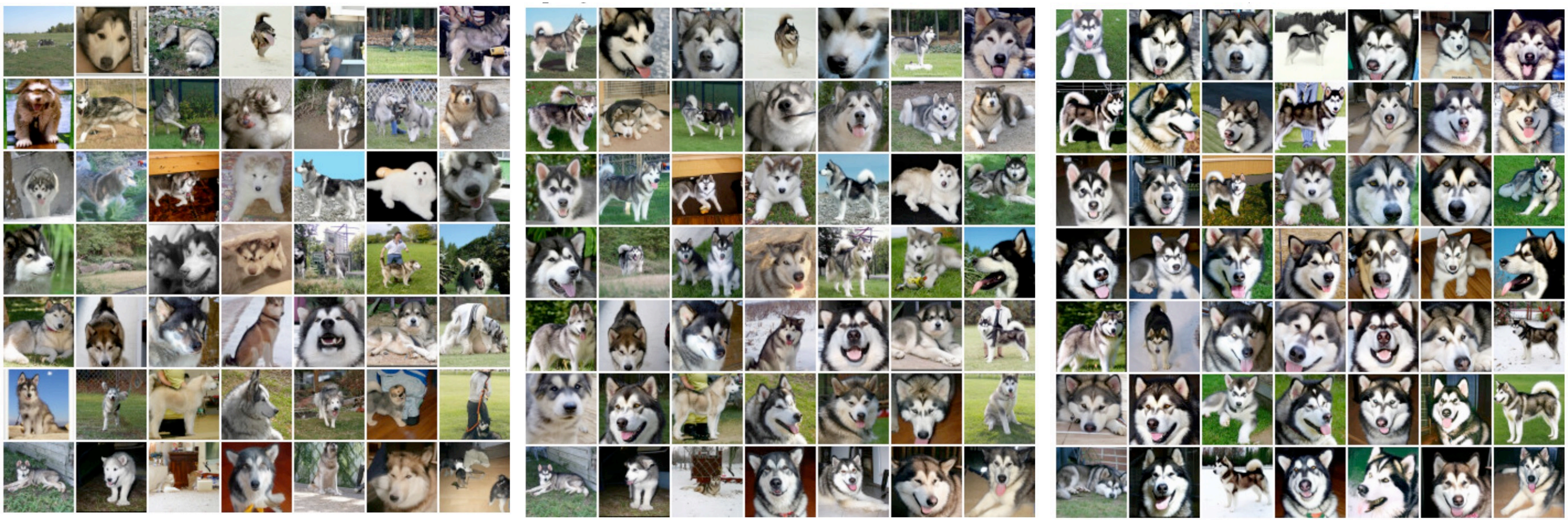
← Unconditional Diffusion Model

- In practice, compute $p(x_t|c)$ and $p(x_t)$ by randomly dropping the condition of the diffusion model at certain chance.

- The modified score with this implicit classifier included is:

$$\begin{aligned}\nabla_{x_t}\left[\log p(x_t|c) + \omega \log p(c|x_t)\right] &= \nabla_{x_t}\left[\log p(x_t|c) + \omega(\log p(x_t|c) - \log p(x_t))\right] \\ &= \nabla_{x_t}\left[(1+\omega)\log p(x_t|c) - \omega \log p(x_t)\right].\end{aligned}$$

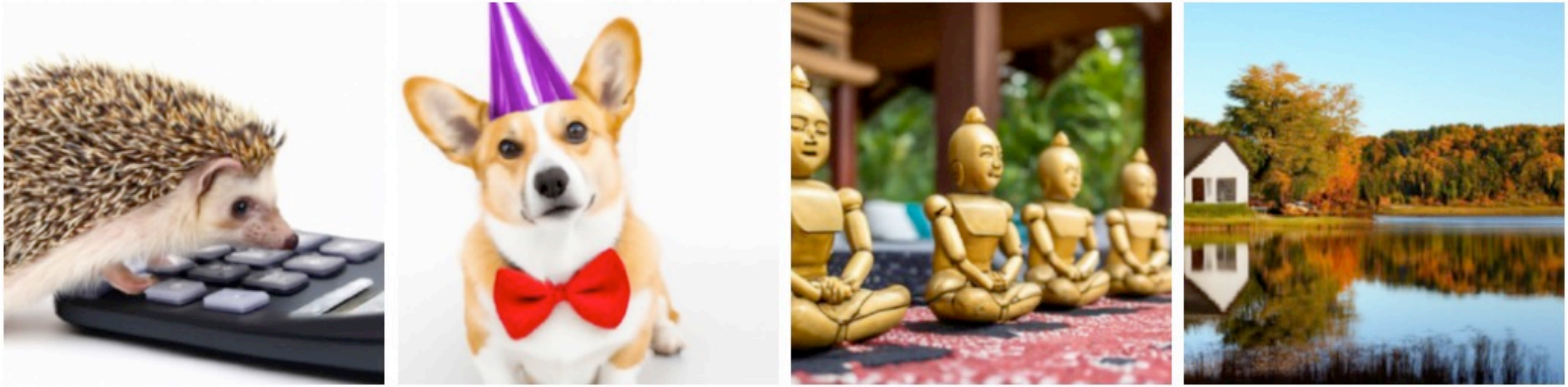
Classifier-Free Conditional Diffusion Models: Trade-Off for Sample Quality and Sample Diversity



Large guidance weight ω usually leads to better individual sample quality but less sample diversity.

GLIDE, OpenAI

- A 64×64 base model + a $64 \times 64 \to 256 \times 256$ super-resolution model.
- Tried classifier-free and CLIP guidance. Classifier-free guidance works better than CLIP guidance.



"a hedgehog using a calculator"

"a corgi wearing a red bowtie and a purple party hat"

"robots meditating in a vipassana retreat"

"a fall landscape with a small cottage next to a lake"

Samples generated with classifier-free guidance (256×256).

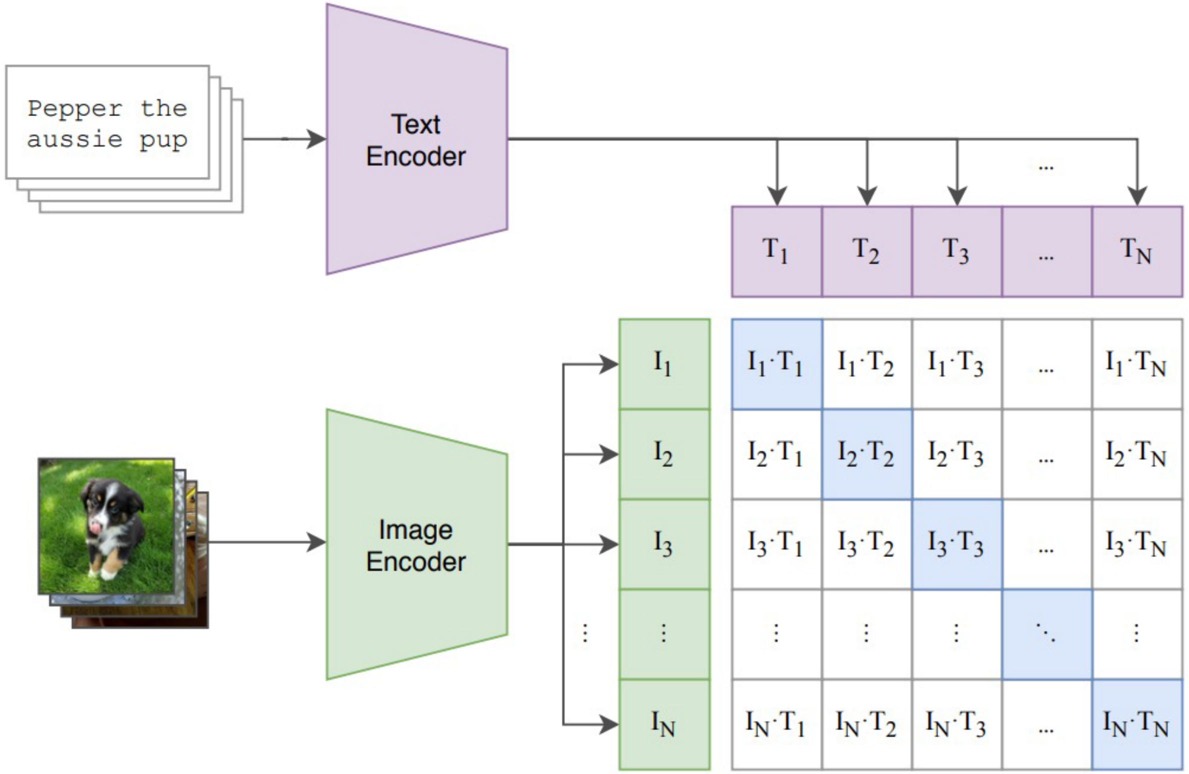
CLIP Guidance: What is a CLIP Model?

- Trained by contrastive cross-entropy loss:

$$-\log \frac{\exp(f(x_i)\cdot g(c_i)/\tau)}{\sum_k \exp(f(x_i)\cdot g(c_k)/\tau)}$$

- The optimzal value of $f(x)\cdot g(c)$ is:

$$\log \frac{p(x,c)}{p(x)p(c)} = \log p(c|x) - \log p(c).$$

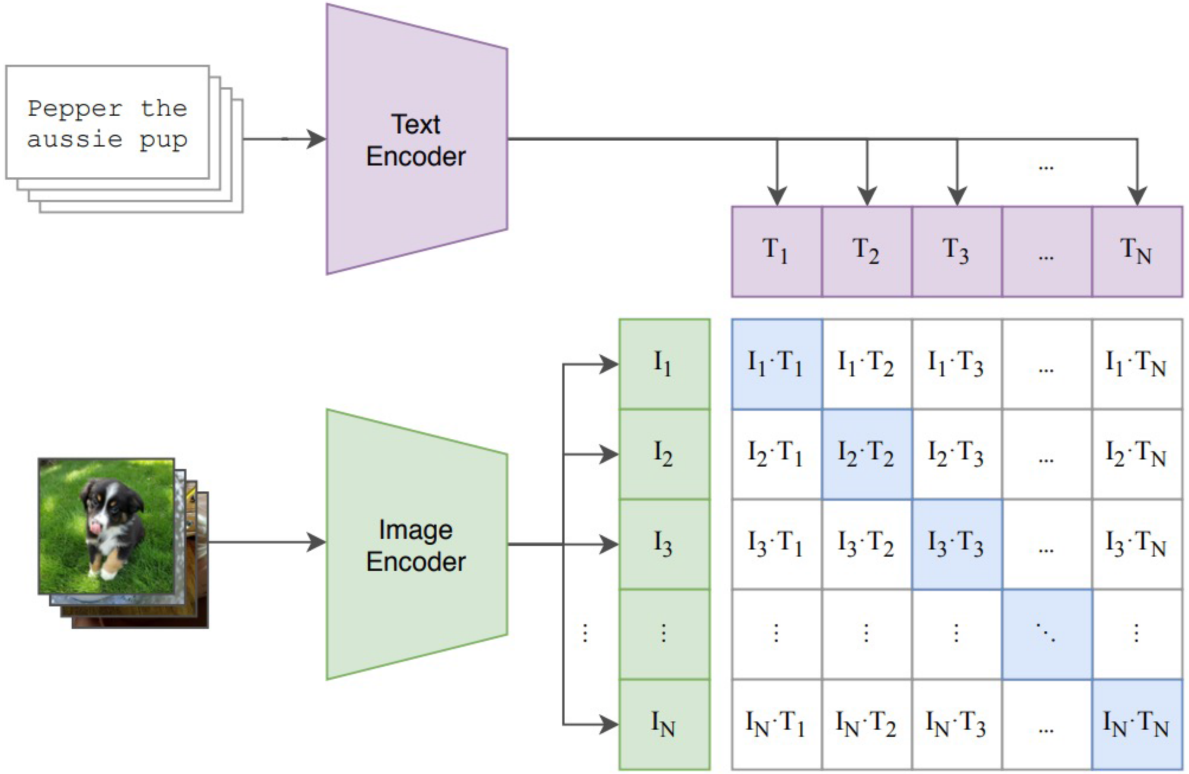


	T_1	T_2	T_3	...	T_N
I_1	$I_1{\cdot}T_1$	$I_1{\cdot}T_2$	$I_1{\cdot}T_3$	...	$I_1{\cdot}T_N$
I_2	$I_2{\cdot}T_1$	$I_2{\cdot}T_2$	$I_2{\cdot}T_3$	...	$I_2{\cdot}T_N$
I_3	$I_3{\cdot}T_1$	$I_3{\cdot}T_2$	$I_3{\cdot}T_3$	...	$I_3{\cdot}T_N$
⋮	⋮	⋮	⋮	⋱	⋮
I_N	$I_N{\cdot}T_1$	$I_N{\cdot}T_2$	$I_N{\cdot}T_3$	...	$I_N{\cdot}T_N$

CLIP Guidance: What is a CLIP Model?

- Sample with a modified score:

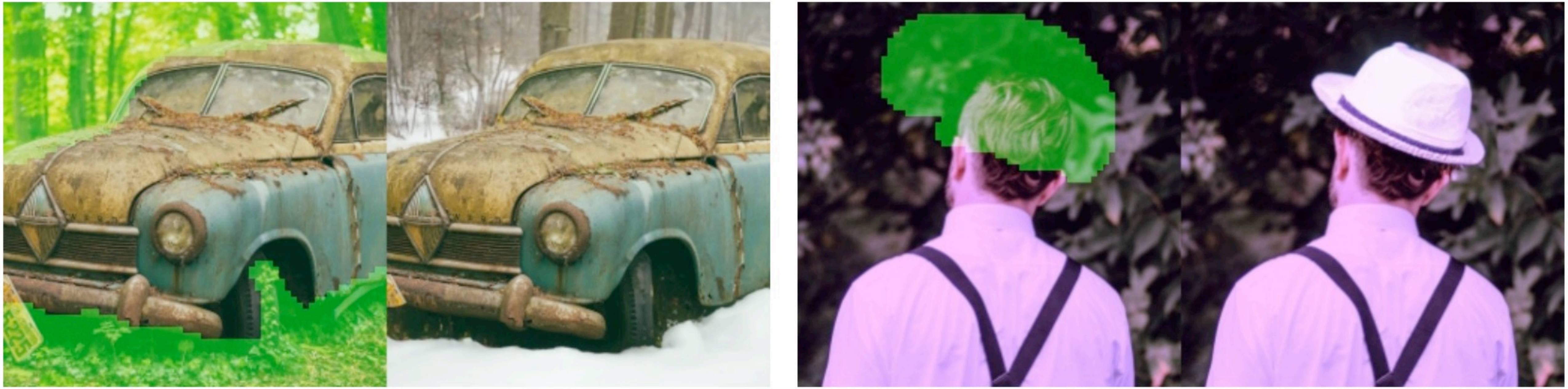
$$
\begin{aligned}
&\nabla_{x_t}\left[\log p(x_t|c) + \omega \log p(c|x_t)\right] \\
&= \nabla_{x_t}\left[\log p(x_t|c) + \omega \left(\underbrace{\log p(c|x_t) - \log p(c)}_{\text{CLIP Model}}\right)\right] \\
&= \nabla_{x_t}\left[\log p(x_t|c) + \omega f(x_t)\cdot g(c)\right]
\end{aligned}
$$



	T₁	T₂	T₃	...	T_N
I₁	I₁·T₁	I₁·T₂	I₁·T₃	...	I₁·T_N
I₂	I₂·T₁	I₂·T₂	I₂·T₃	...	I₂·T_N
I₃	I₃·T₁	I₃·T₂	I₃·T₃	...	I₃·T_N
⋮	⋮	⋮	⋮	⋱	⋮
I_N	I_N·T₁	I_N·T₂	I_N·T₃	...	I_N·T_N

GLIDE, OpenAI

- Fine-tune the model especially for inpainting: feed randomly occluded images with an additional mask channel as the input.

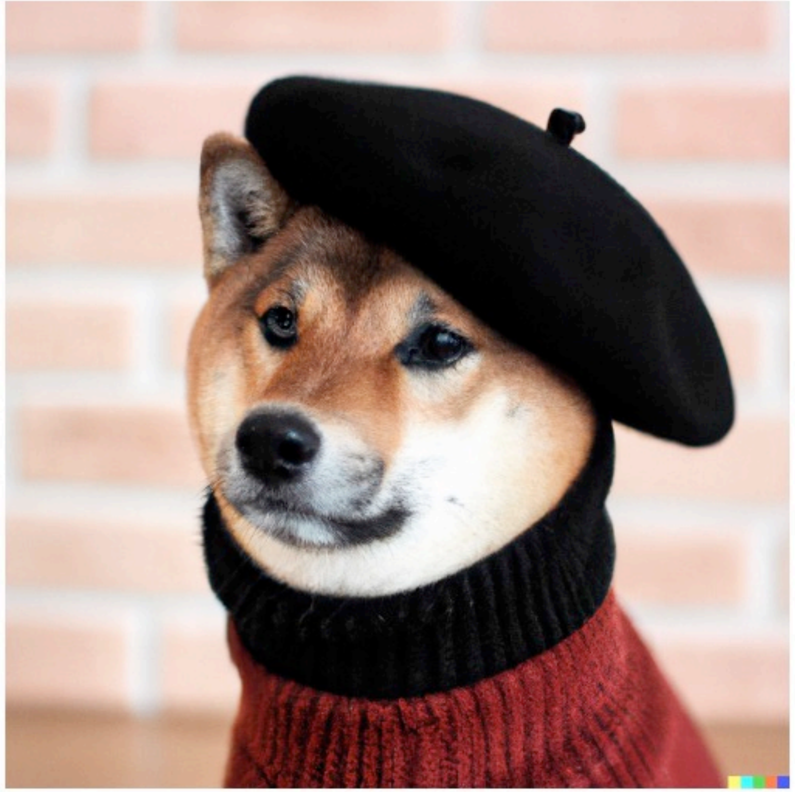


“an old car in a snowy forest”

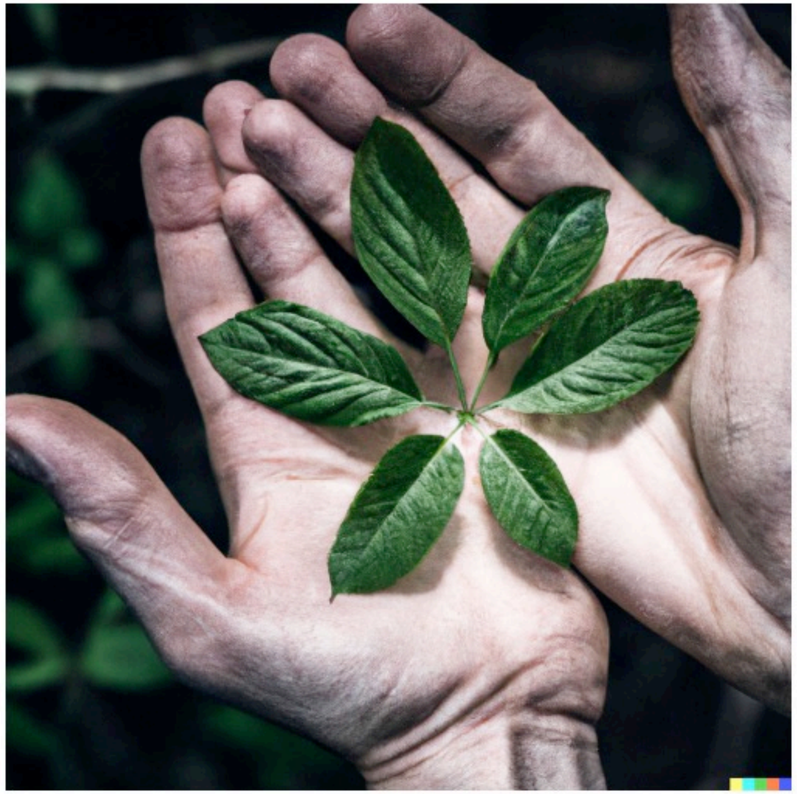
“a man wearing a white hat”

Text-conditional image inpainting examples.

DALL·E 2, OpenAI



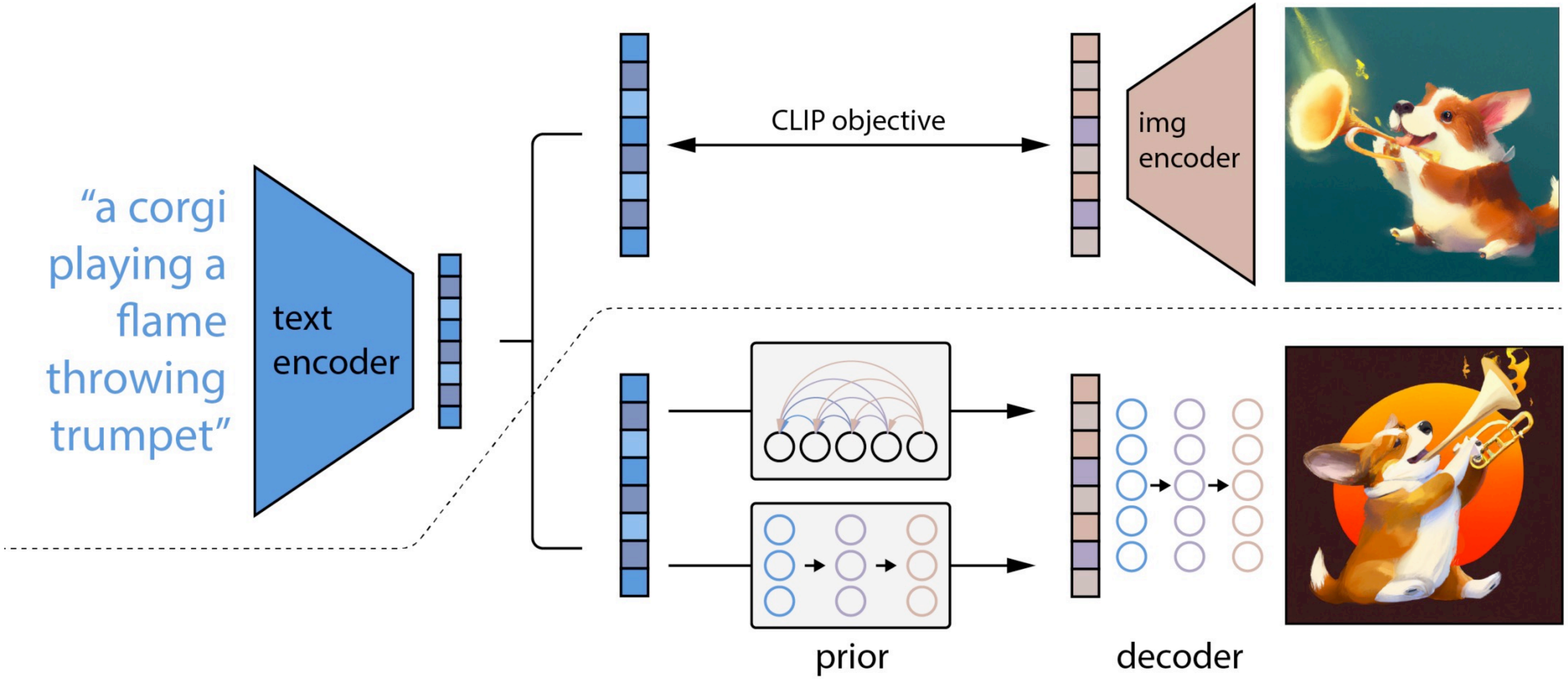
a shiba inu wearing a beret and black turtleneck



a close up of a handpalm with leaves growing from it

$1k \times 1k$ Text-to-Image generation.
Outperforms DALL-E
(autoregressive transformer).

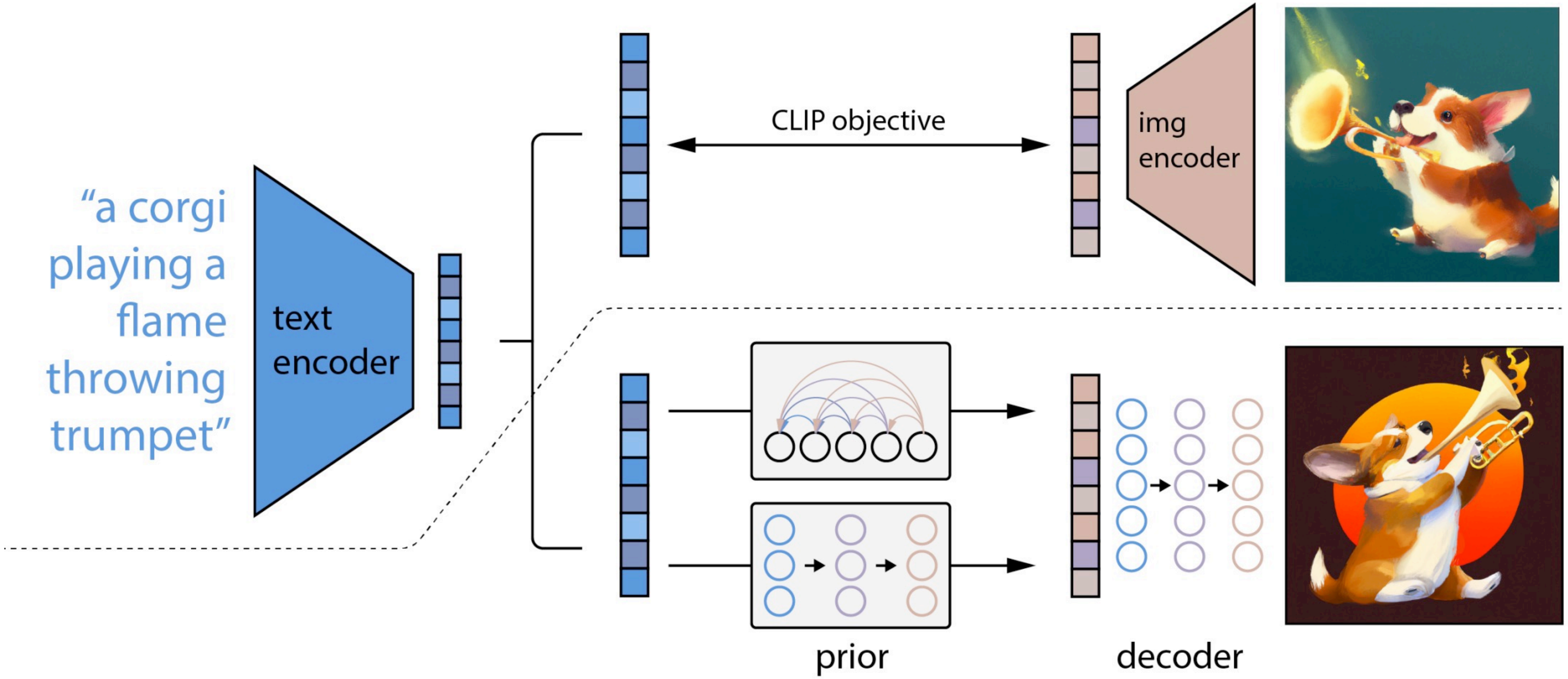
DALL·E 2 Model Components



Prior: Produces CLIP image embeddings conditioned on the caption.

Decoder: Produces images conditioned on CLIP image embeddings and text.

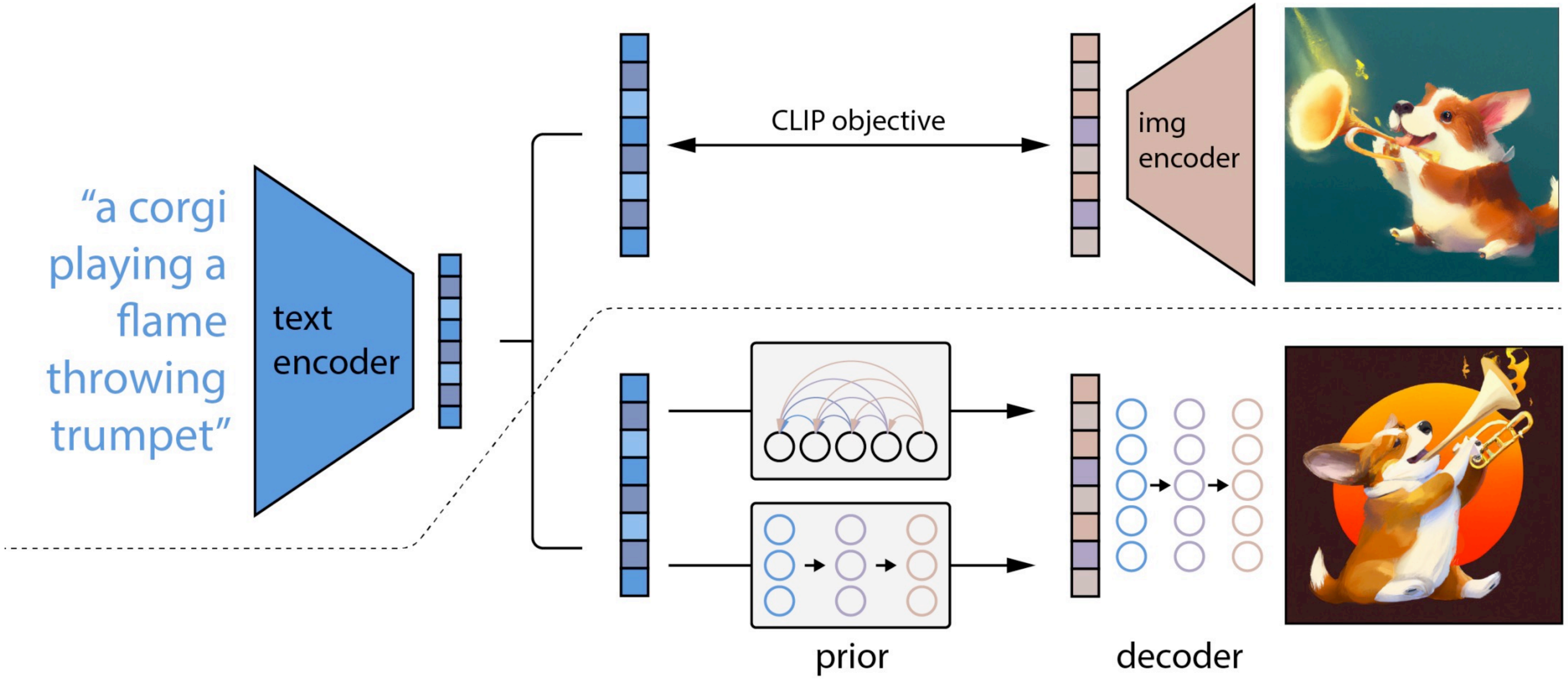
DALL·E 2 Model Components: Prior Model



Why conditional on CLIP image embeddings?

CLIP image embeddings capture high-level semantic meaning; latents in the decoder model take care of the rest. The bipartite latent representation‘ enables several text-guided image manipulation tasks.

DALL·E 2 Model Components: Decoder Model

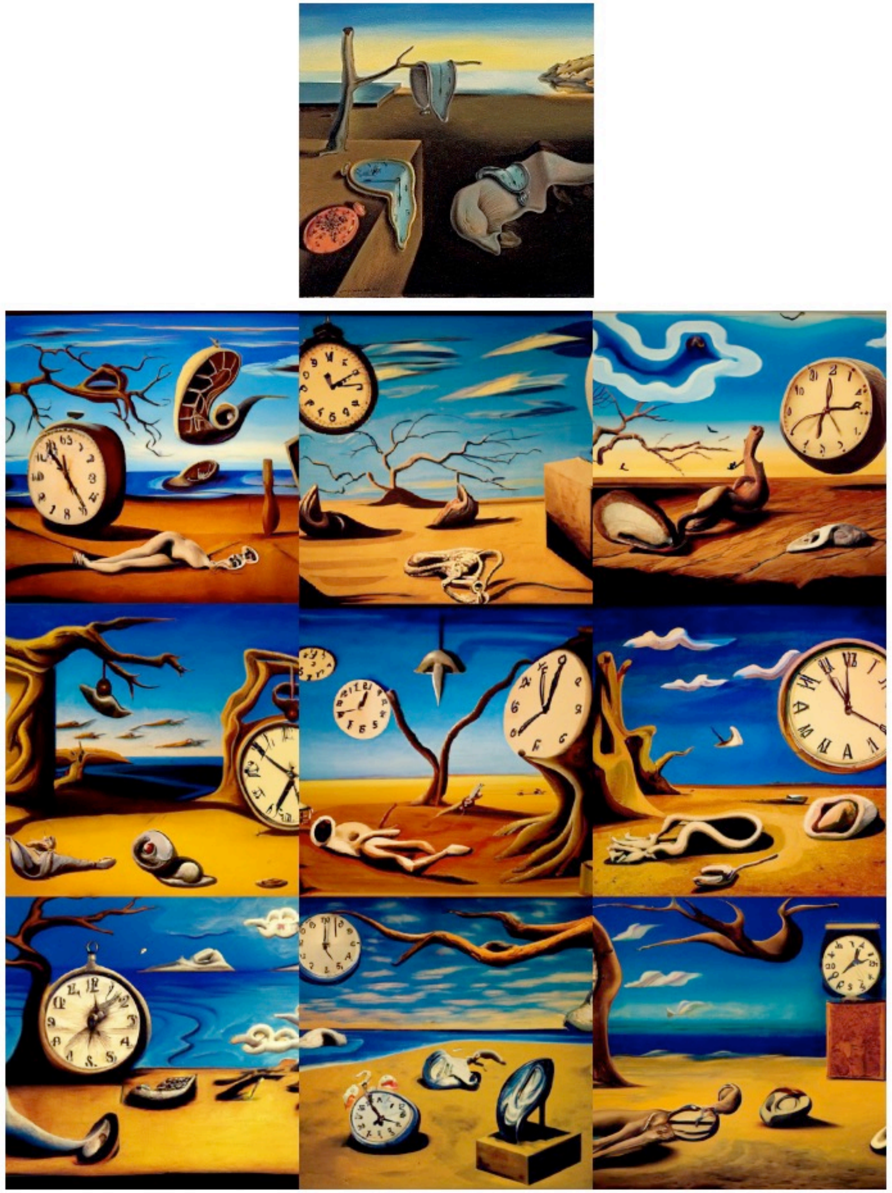


Decoder: produces images conditioned on CLIP image embeddings (and text).

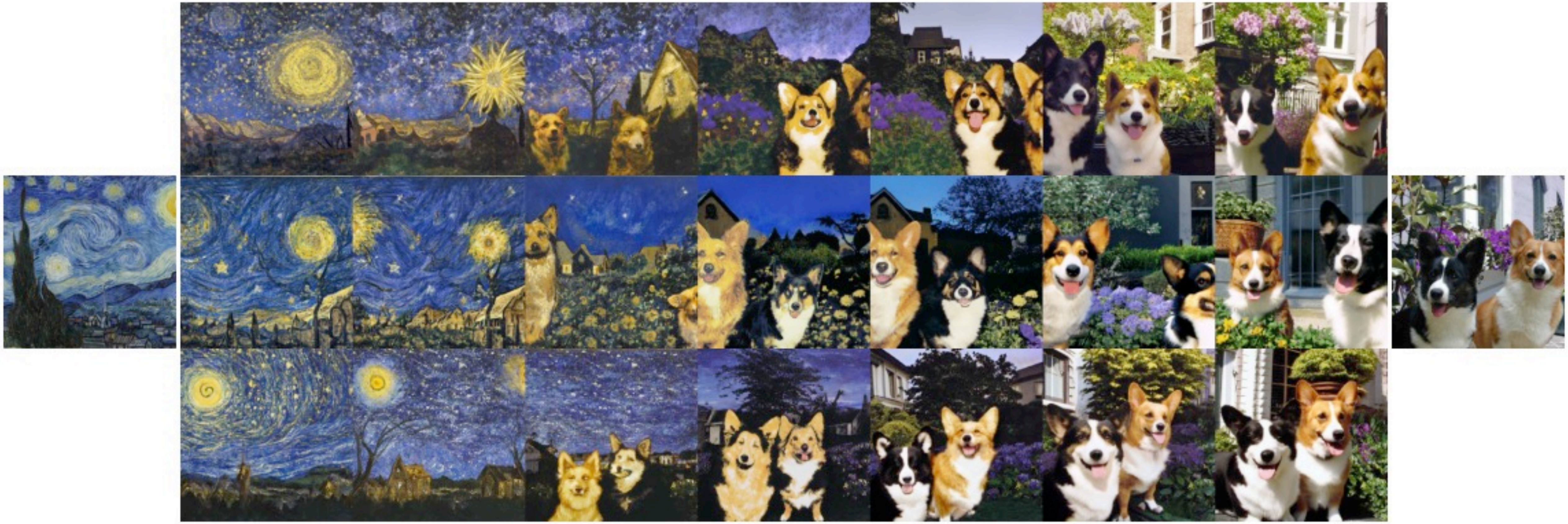
Cascaded diffusion models: 1 base model (64×64), 2 super-resolution models ($64 \times 64 \rightarrow 256 \times 256, 256 \times 256 \rightarrow 1024 \times 1024$). Largest super-resolution model is trained on patches and takes full-res inputs at inference time. Classifier-free guidance & noise conditioning augmentation are important.

DALL·E 2: Image Variations

- Fix the CLIP embedding z.
- Decode using different decoder latents x_T.

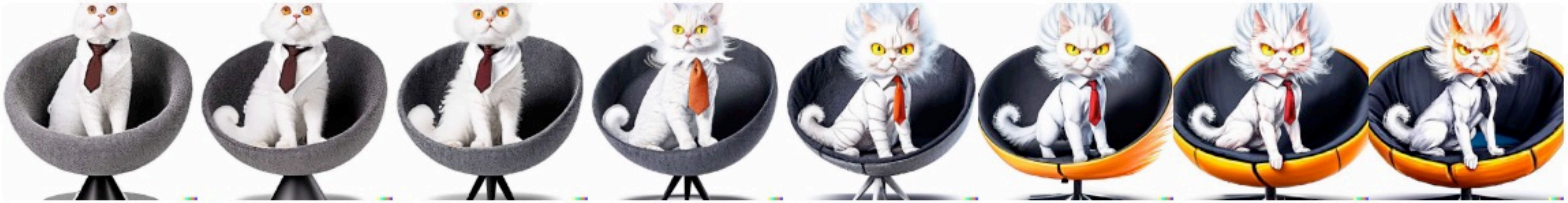


DALL·E 2: Image Interpolation

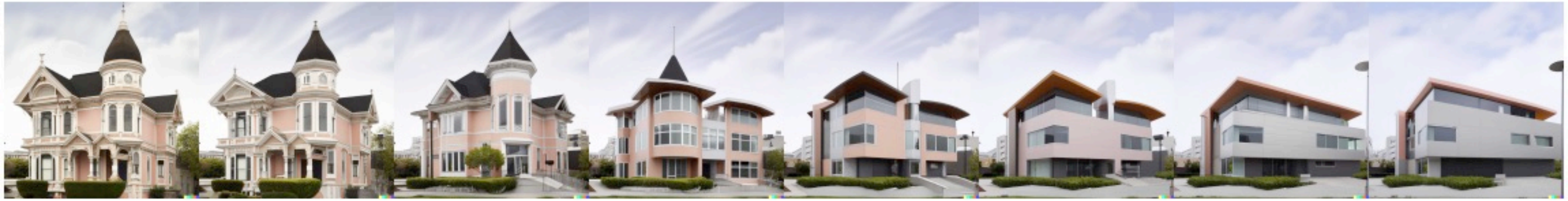


- Interpolate image CLIP embeddings z.
- Use different x_T to get different interpolations trajectories.

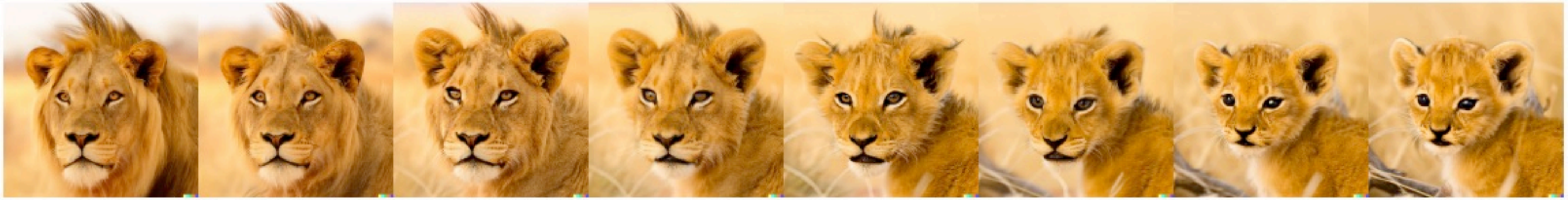
DALL·E 2: Text Diffs



a photo of a cat → an anime drawing of a super saiyan cat, artstation



a photo of a victorian house → a photo of a modern house



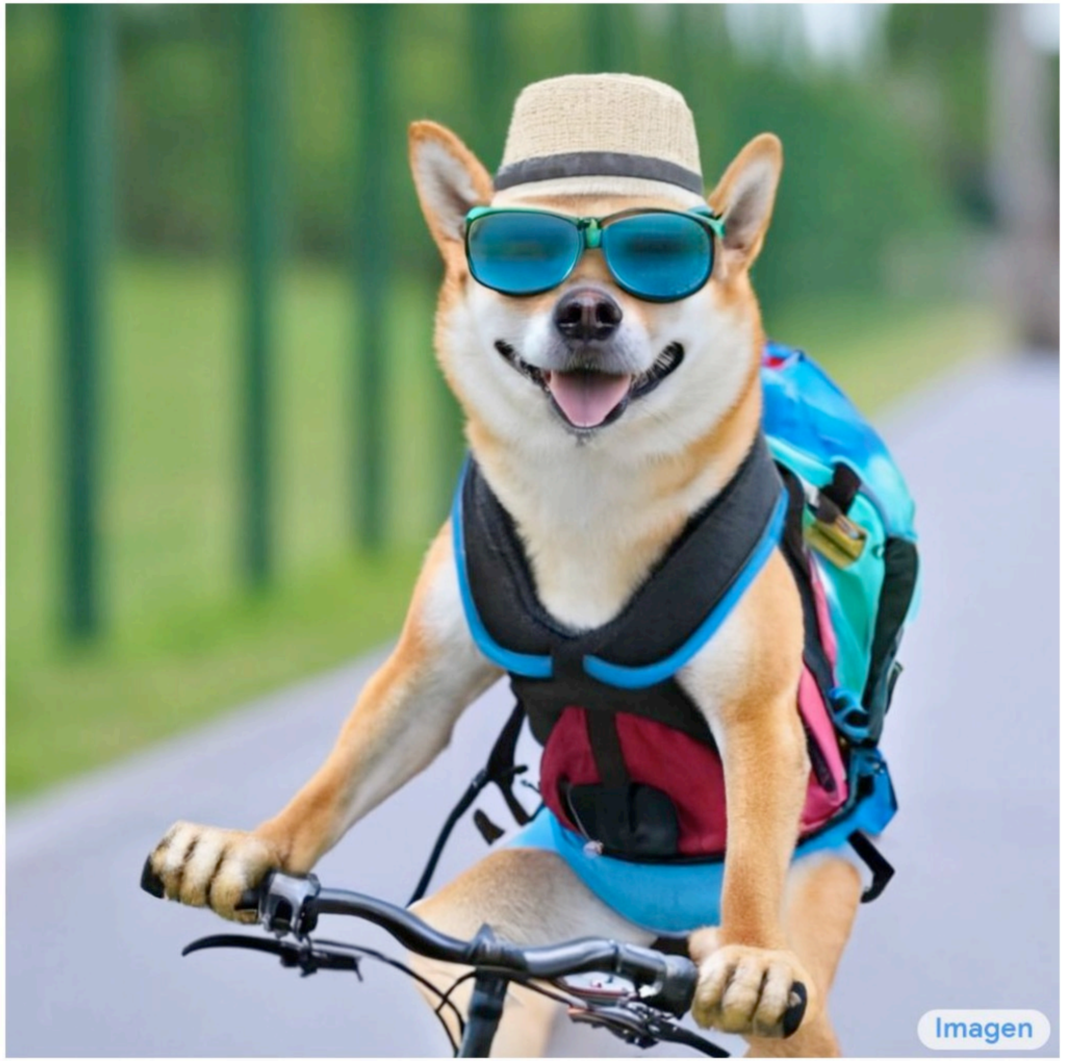
a photo of an adult lion → a photo of lion cub

- Change the image CLIP embedding towards the difference of the text CLIP embeddings of two prompts.
- Decoder latent is kept constant.

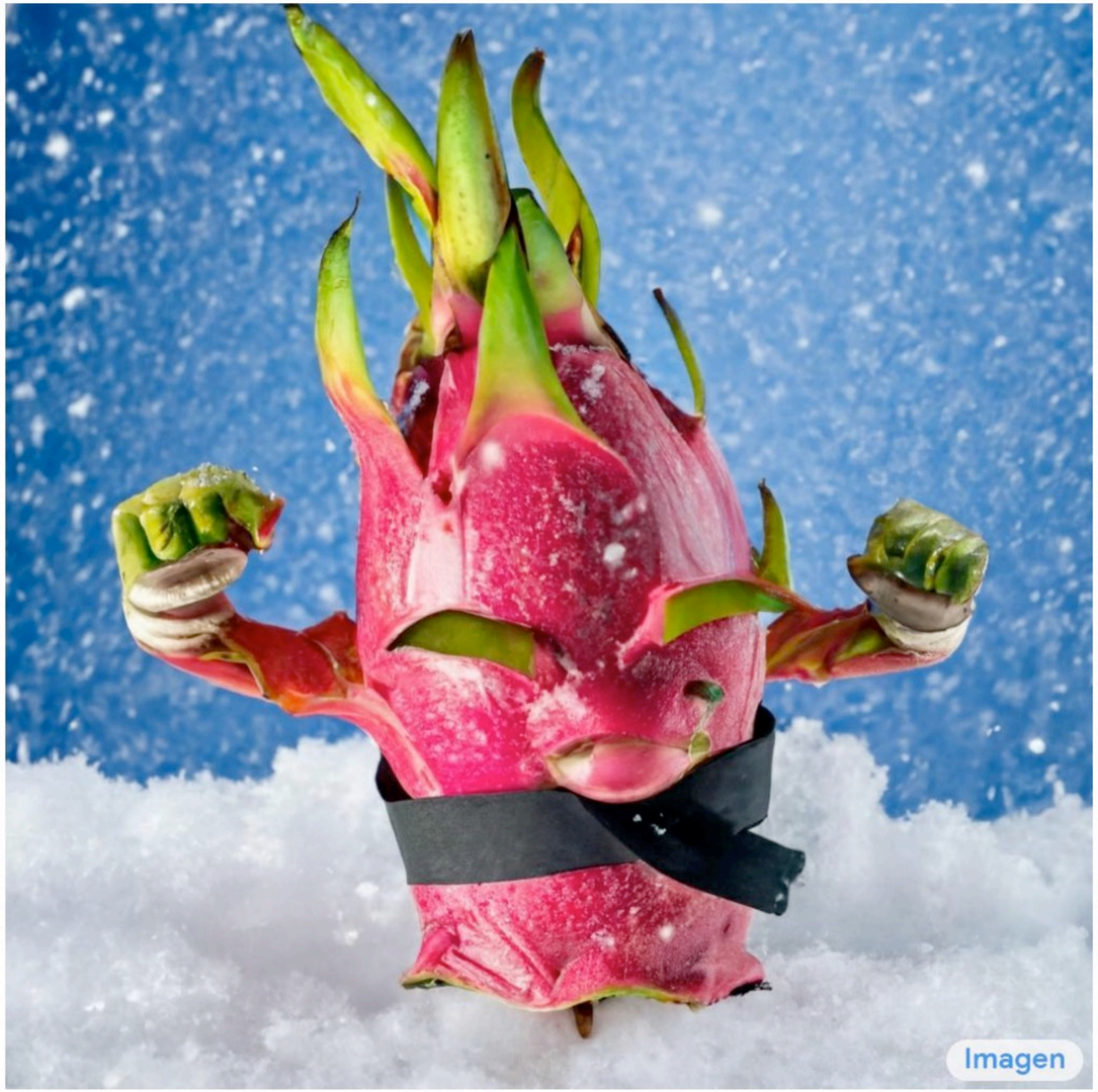
Imagen: Google Research

- Input: text, Output: 1k × 1k images.
- An unprecedented degree of photorealism.
  - SOTA automatic scores & human ratings.
- A deep level of language understanding.
- Extremely simple.
  - No latent space, no quantization.

Imagen: Google Research

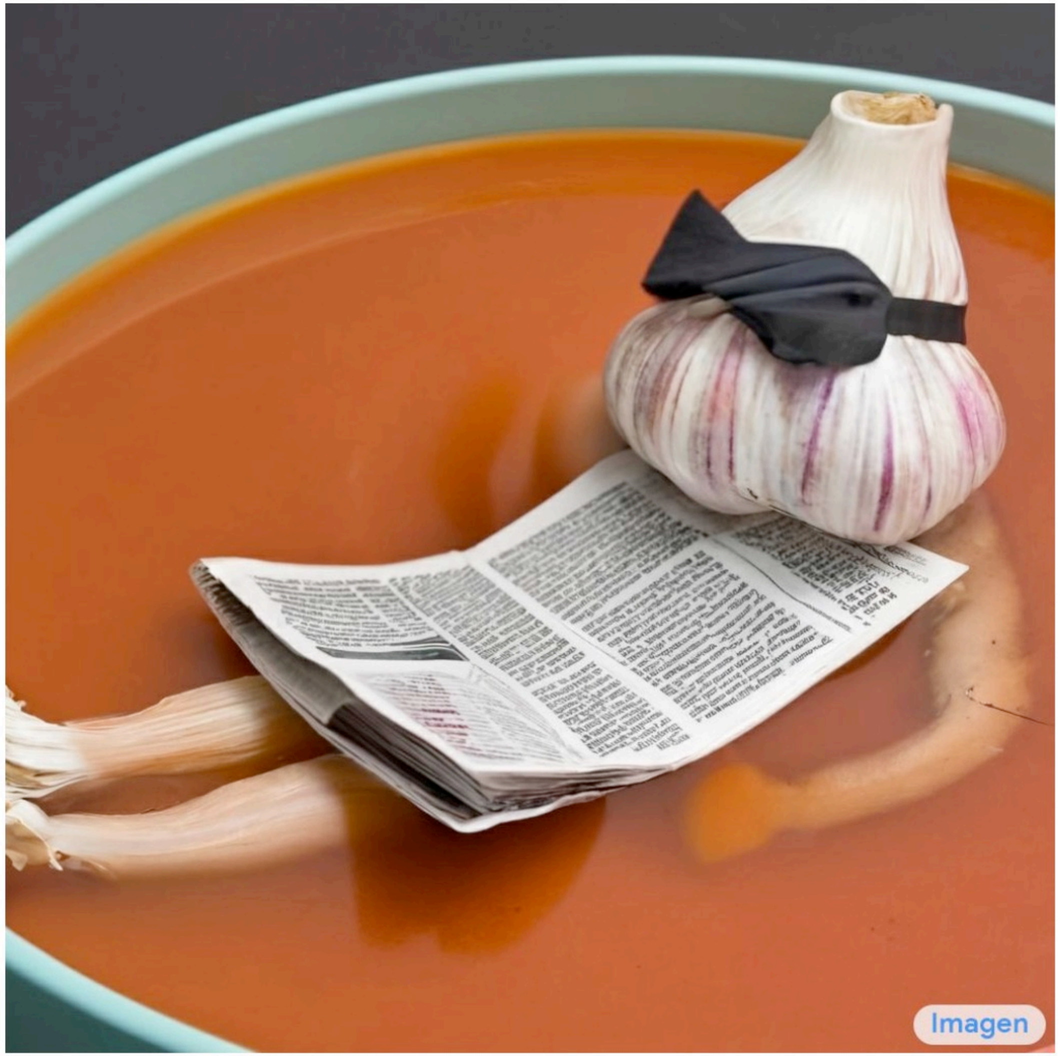


A photo of a Shiba Inu dog with a backpack riding a bike. It is wearing sunglasses and a beach hat.

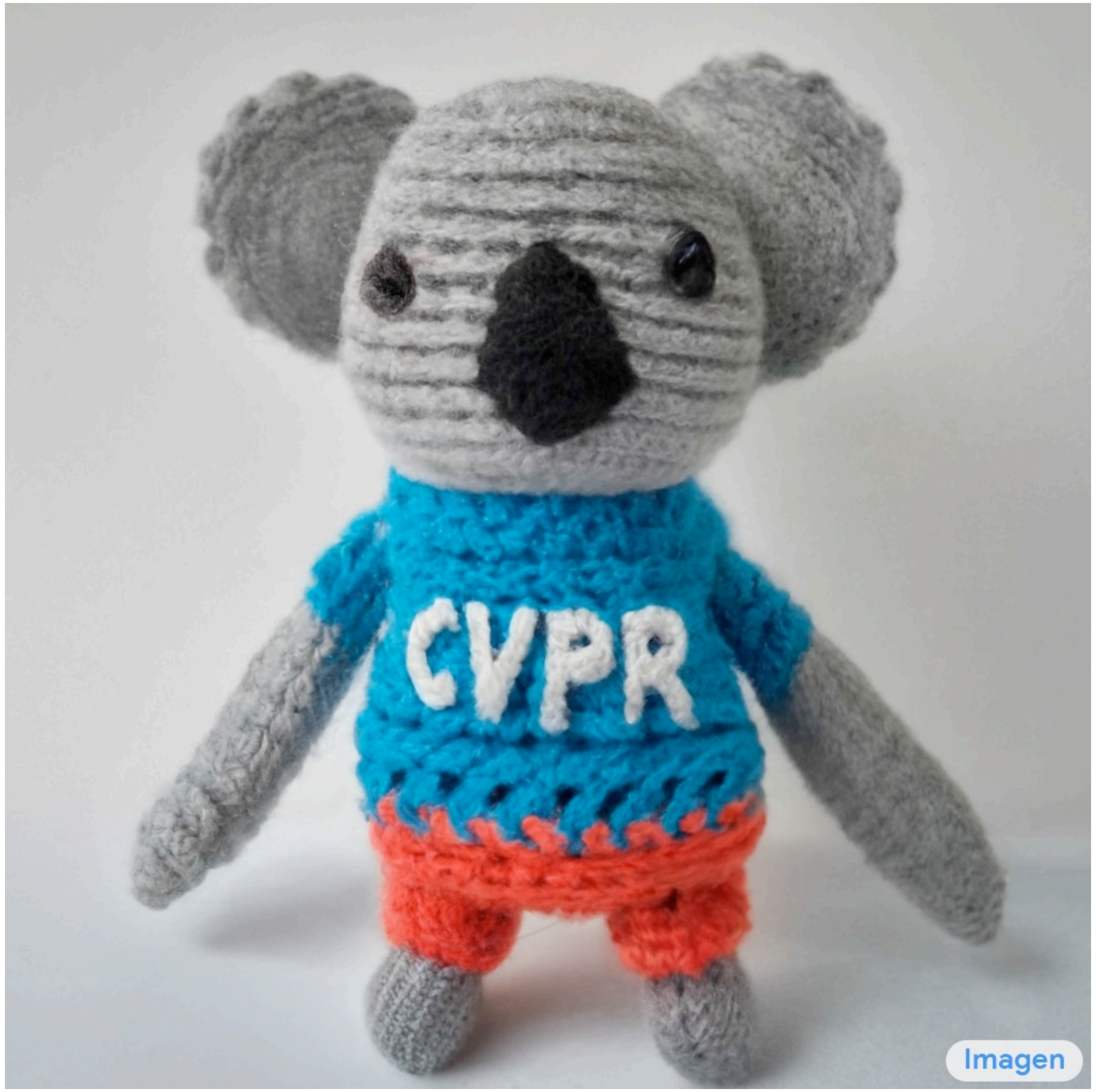


A dragon fruit wearing karate belt in the snow.

Imagen: Google Research



A relaxed garlic with a blindfold reading a newspaper while floating in a pool of tomato soup.



“A cute hand-knitted koala wearing a sweater with “CVPR” written on it.”

Imagen Key Components

- Key modeling components:
  - Cascaded diffusion models.
  - Classifier-free guidance and dynamic thresholding.
  - Frozen large pretrained language models as text encoders (T5-XXL).

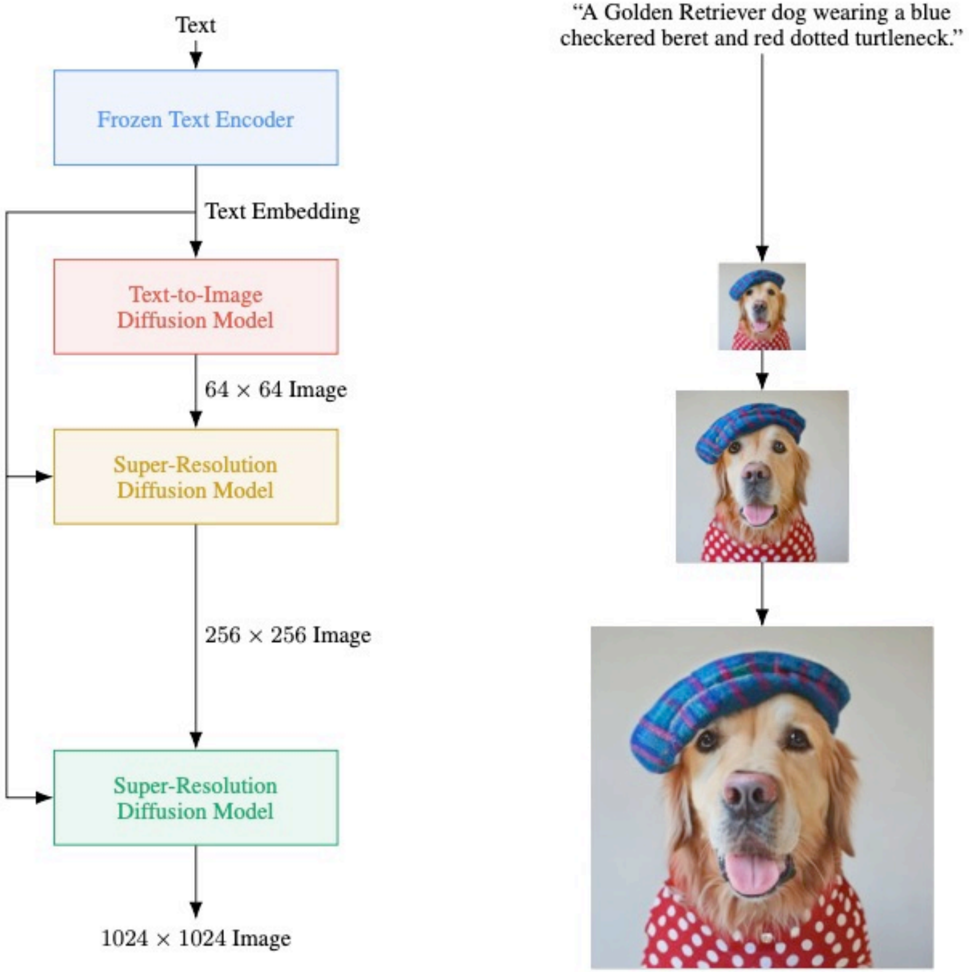

Text
Frozen Text Encoder
Text Embedding
Text-to-Image Diffusion Model
64 × 64 Image
Super-Resolution Diffusion Model
256 × 256 Image
Super-Resolution Diffusion Model
1024 × 1024 Image
"A Golden Retriever dog wearing a blue checkered beret and red dotted turtleneck."

Figure A.4: Visualization of Imagen. Imagen uses a frozen text encoder to encode the input text into text embeddings. A conditional diffusion model maps the text embedding into a 64 × 64 image. Imagen further utilizes text-conditional super-resolution diffusion models to upsample the image, first 64 × 64 → 256 × 256, and then 256 × 256 → 1024 × 1024.

Imagen Key Observations

- Key Observations:
  - Beneficial to use text conditioning for all super-res models.
  - Noise conditioning augmentation weakens information from low-res models, thus needs text conditioning as extra information input.
  - Scaling text encoder is extremely efficient.
  - More important than scaling diffusion model size.
  - Human raters prefer T5-XXL as the text encoder over CLIP encoder on DrawBench.

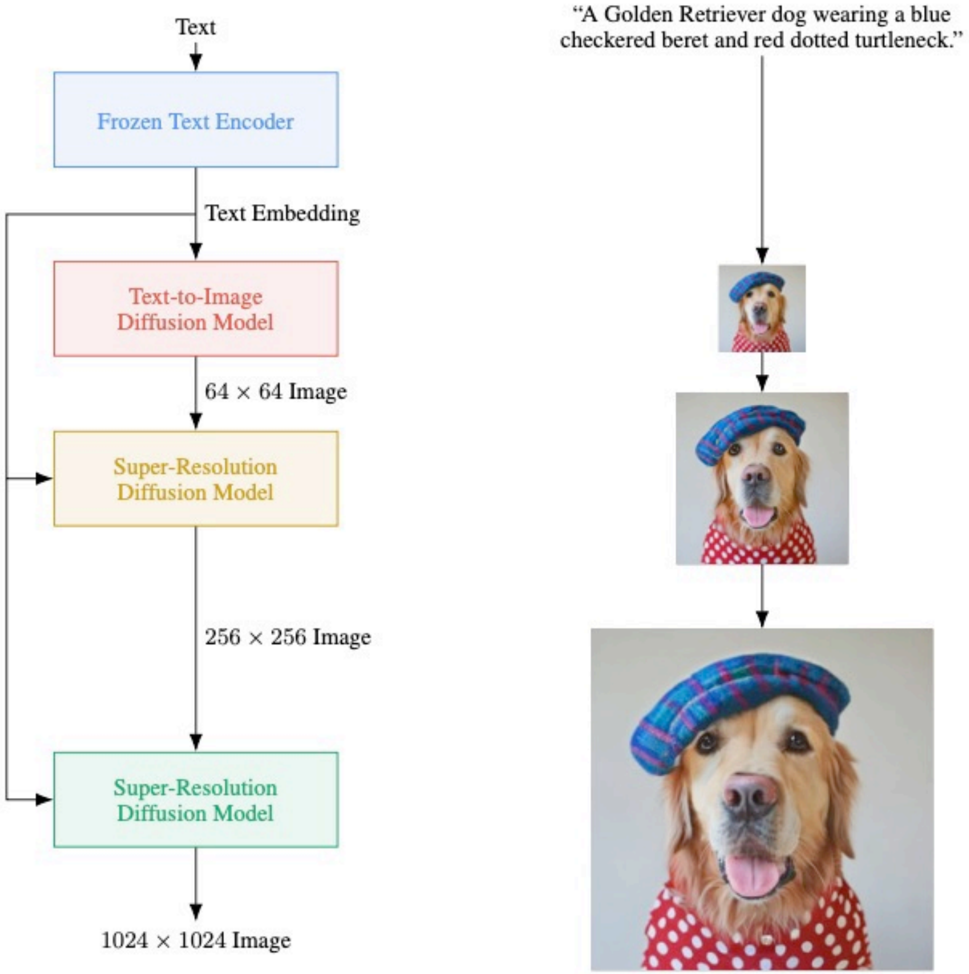
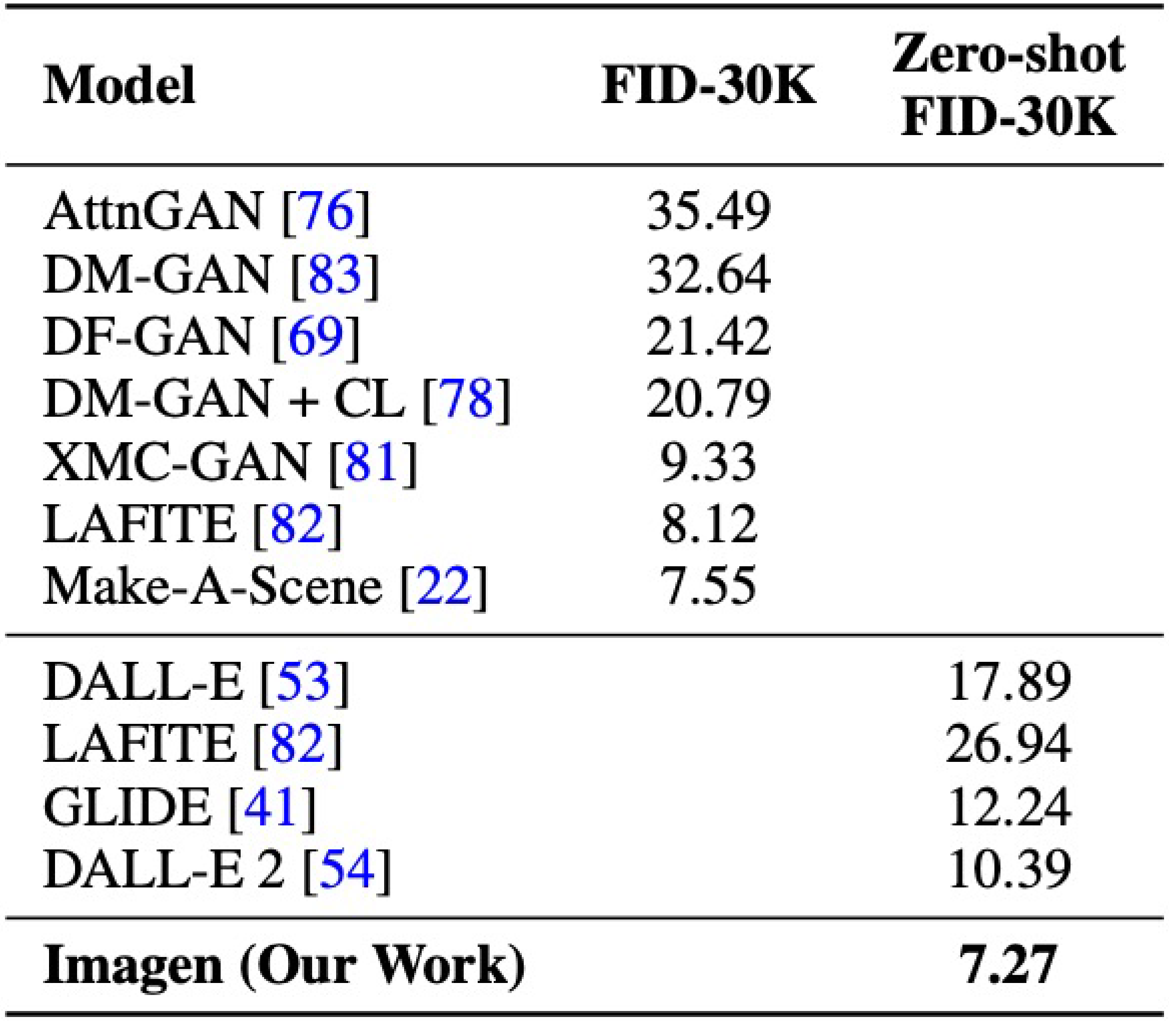

Text
Frozen Text Encoder
Text Embedding
Text-to-Image Diffusion Model
64 × 64 Image
Super-Resolution Diffusion Model
256 × 256 Image
Super-Resolution Diffusion Model
1024 × 1024 Image
"A Golden Retriever dog wearing a blue checkered beret and red dotted turtleneck."

Figure A.4: Visualization of Imagen. Imagen uses a frozen text encoder to encode the input text into text embeddings. A conditional diffusion model maps the text embedding into a 64 × 64 image. Imagen further utilizes text-conditional super-resolution diffusion models to upsample the image, first 64 × 64 → 256 × 256, and then 256 × 256 → 1024 × 1024.

Imagen Evaluations

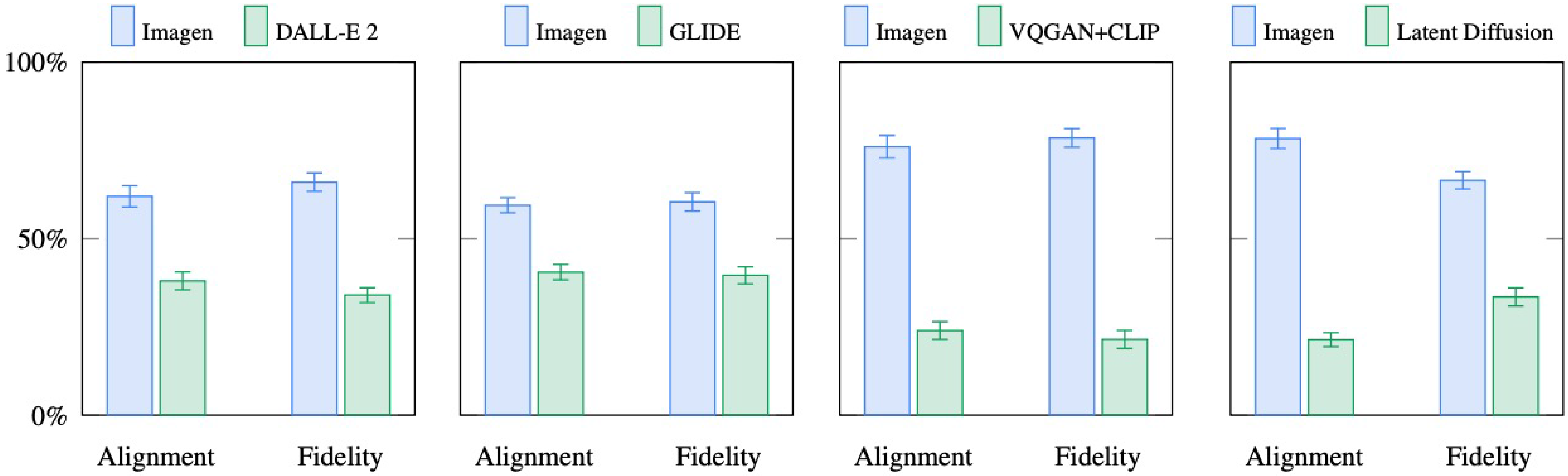
- Imagen got SOTA automatic evaluation scores on COCO dataset:



Model	FID-30K	Zero-shot FID-30K
AttnGAN [76]	35.49	
DM-GAN [83]	32.64	
DF-GAN [69]	21.42	
DM-GAN + CL [78]	20.79	
XMC-GAN [81]	9.33	
LAFITE [82]	8.12	
Make-A-Scene [22]	7.55	
DALL-E [53]		17.89
LAFITE [82]		26.94
GLIDE [41]		12.24
DALL-E 2 [54]		10.39
Imagen (Our Work)		**7.27**

Imagen Evaluations

- Imagen is preferred over recent work by human raters in sample quality & image-text alignment on DrawBench:



Applications

- There are many successful applications of diffusion models (in constantly growing numbers):
  - Image generation, text-to-image generation,controllable generation.
  - Image editing, **image-to-image translation**, **super-resolution**, **segmentation**, **adversarial robustness**.
  - Discrete models, 3D generation, medical imaging, video synthesis.
- Key enabler by diffusion models: Perform high-resolution conditional generation!

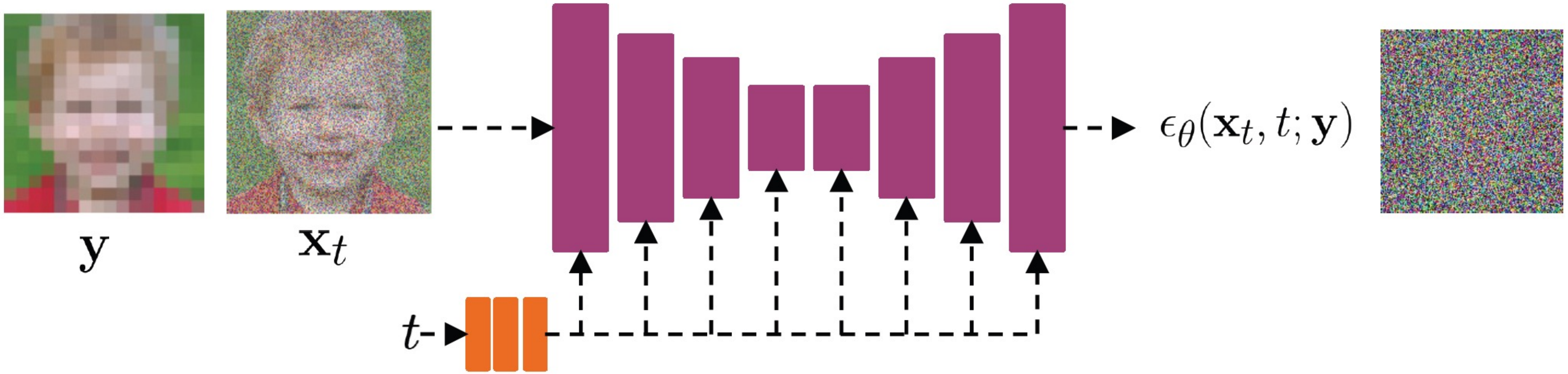
Super-Resolution via Repeated Refinement

- Image super-resolution can be considered as training $p(x|y)$ where y is a low-resolution image and x is the corresponding high-resolution image.

- Train a score model for x conditioned on y using:

$$\mathbb{E}_{x,y}\mathbb{E}_{\epsilon\sim\mathcal{N}(0,I)}\mathbb{E}_t||\epsilon_\theta(x_t, t; y) - \epsilon||_p^p.$$

- The conditional score is simply a U-Net with x_t and y (resolution image) concatenated:



Super-Resolution via Repeated Refinement

Natural Image Super-Resolution 64×64 → 256×256

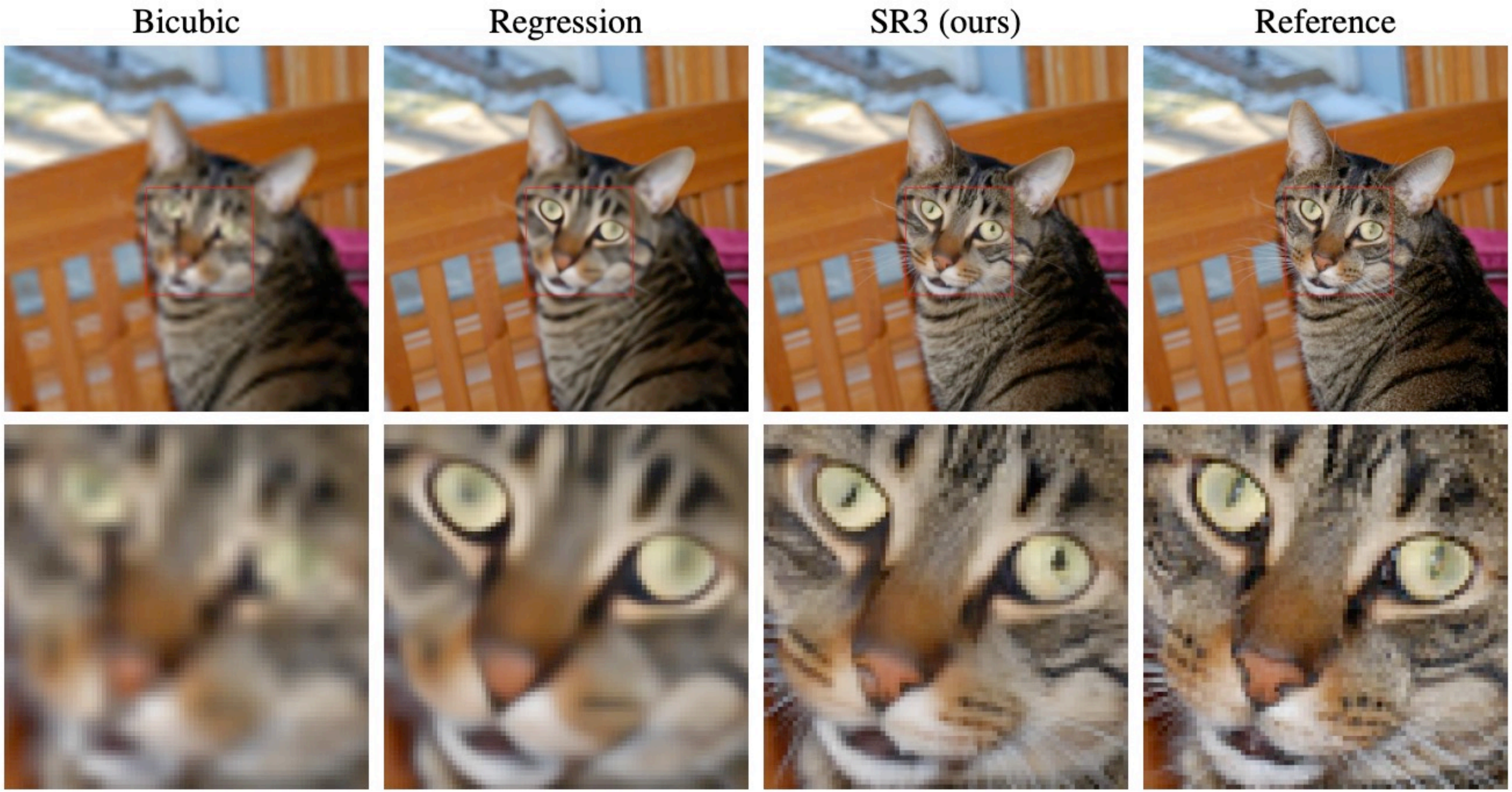


Image-to-Image Translation: Palette: Image-to-Image Diffusion Models

- Many image-to-image translation applications can be considered as training $p(x|y)$ where y is the input image.

- For example, for colorization, x is a colored image and y is a gray-level image.

- Train a score model for x conditioned on y using: $\mathbb{E}_{x,y}\mathbb{E}_{\epsilon\sim\mathcal{N}(0,I)}\mathbb{E}_t||\epsilon_\theta(x_t,t;y)-\epsilon||_p^p$.

- The conditional score is simply a U-Net with x_t and y concatenated:

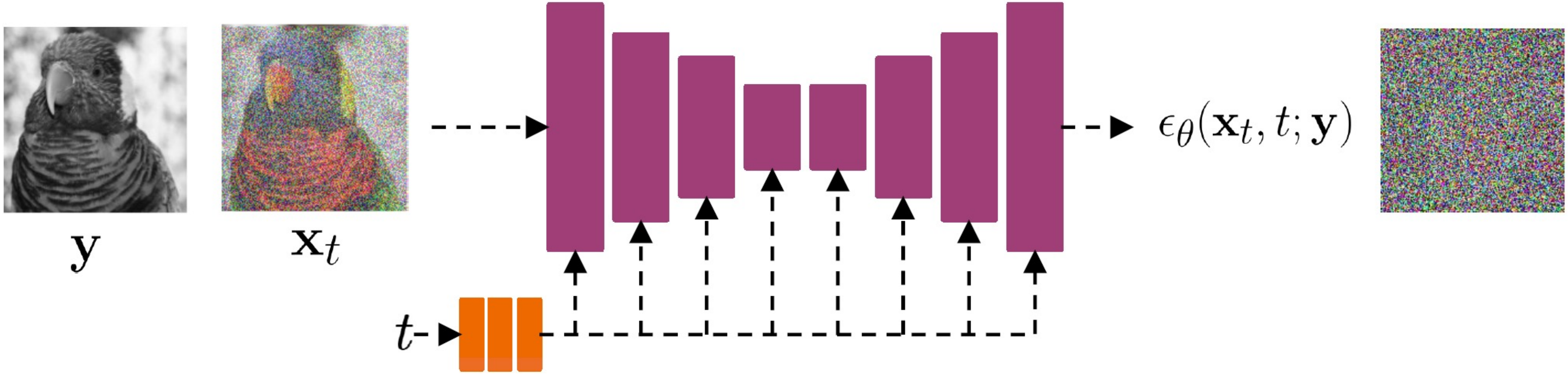
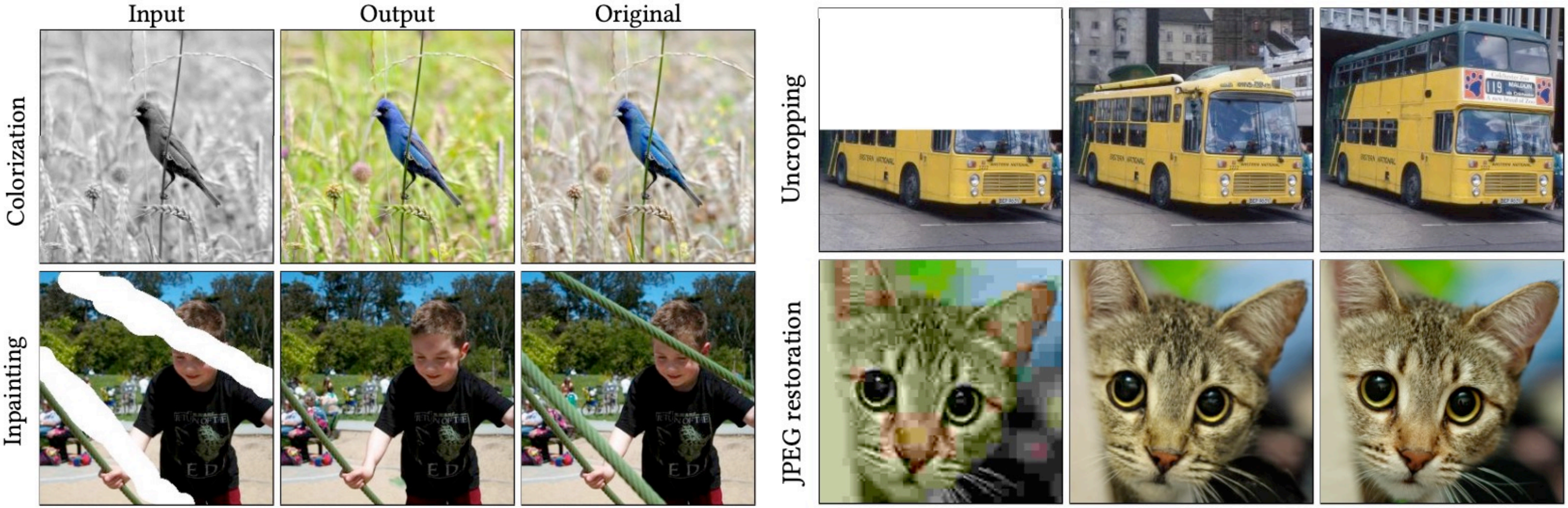
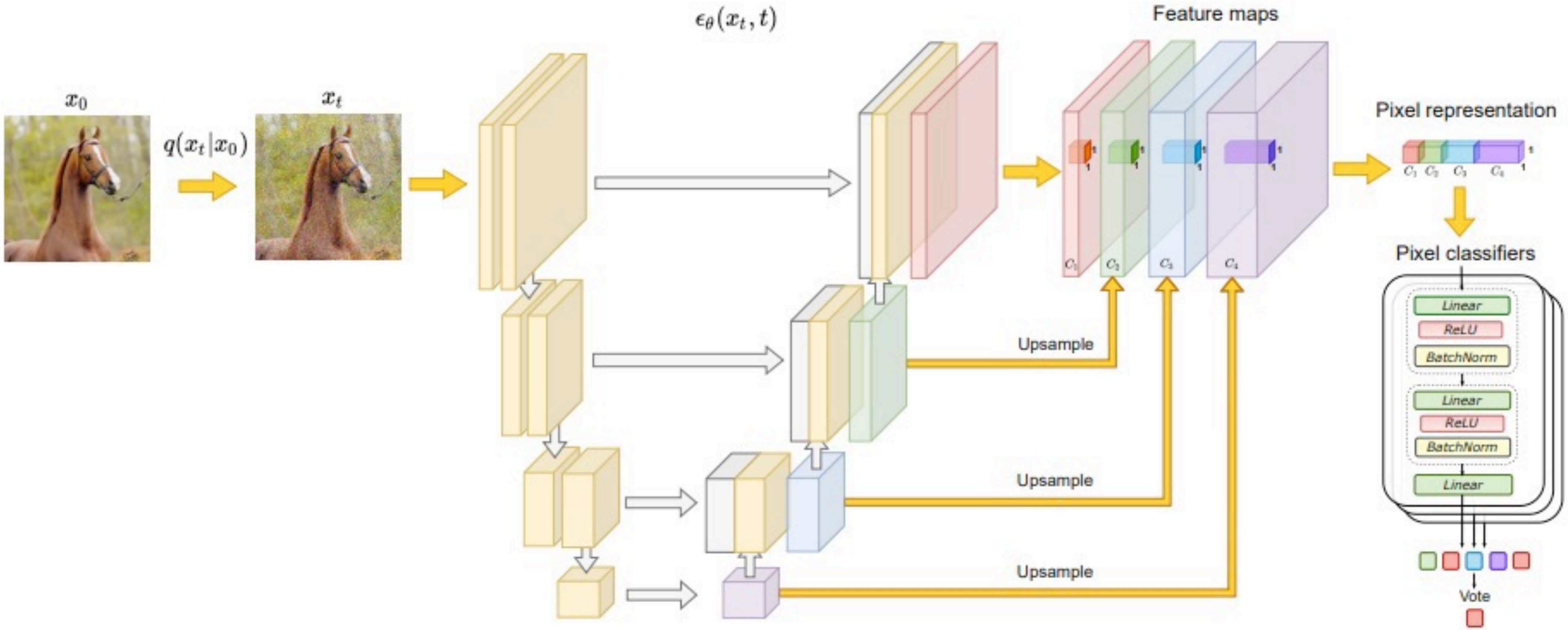


Image-to-Image Translation:
Palette: Image-to-Image Diffusion Models



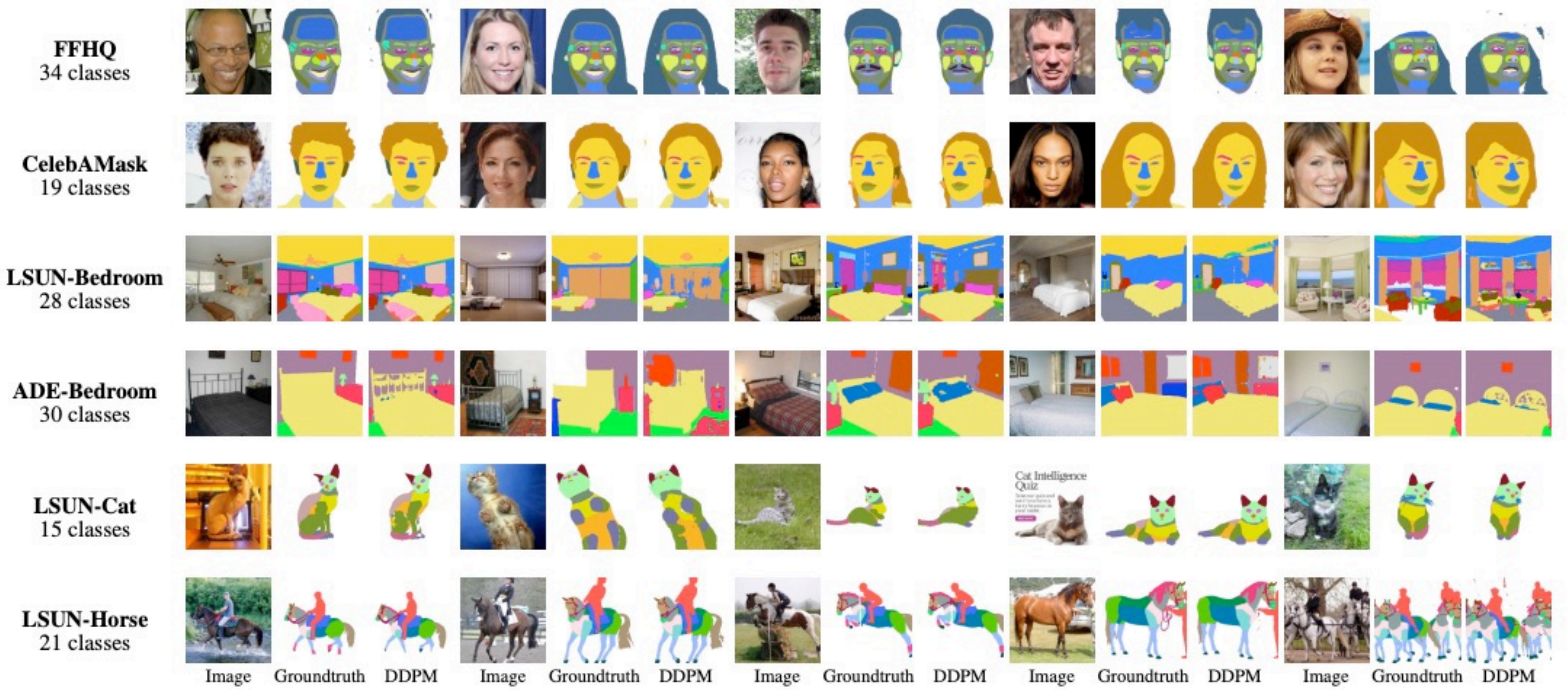
Semantic Segmentation: Label-Efficient Semantic Segmentation with Diffusion Models

- Can we use representation learned from diffusion models for downstream applications such as semantic segmentation?



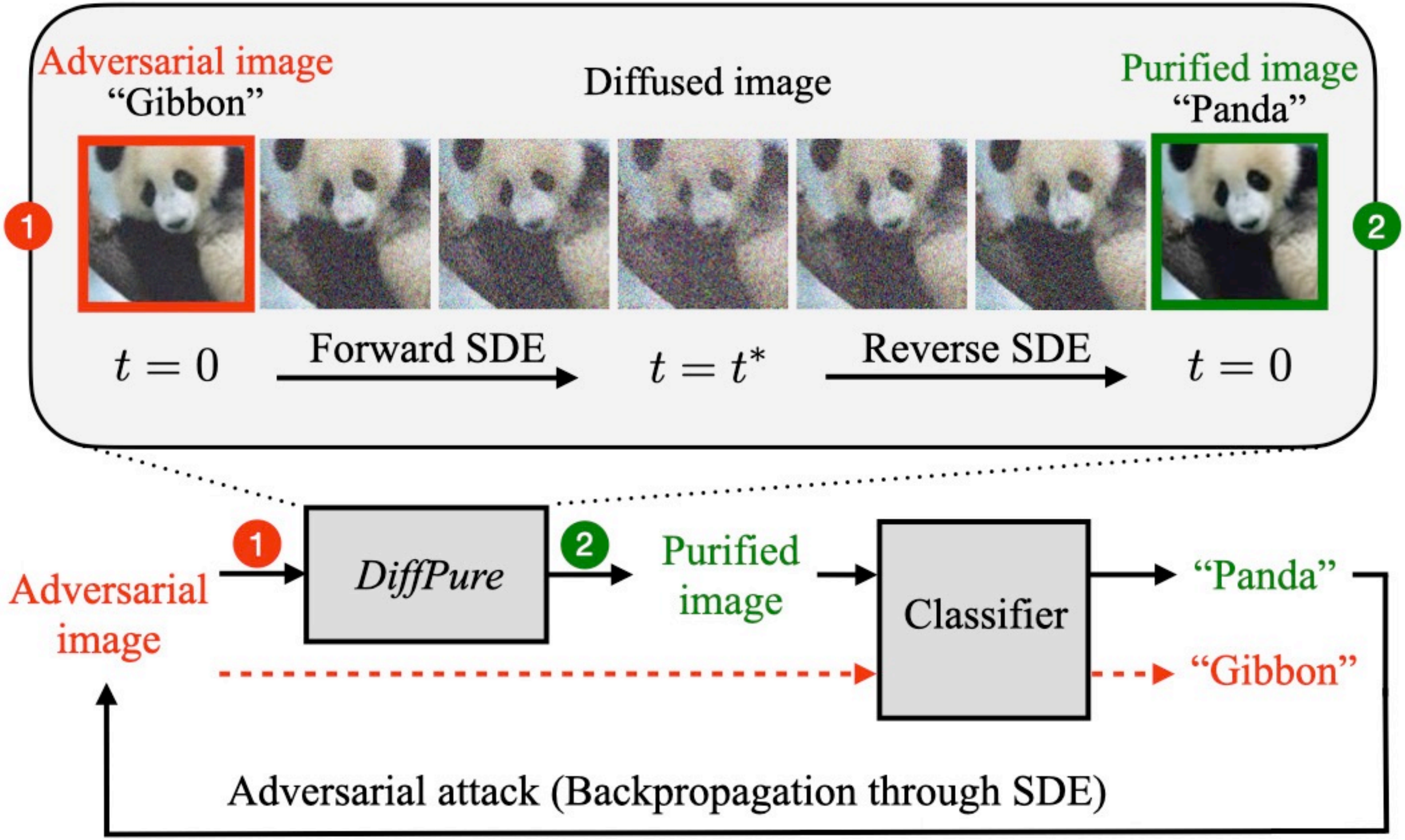
Semantic Segmentation: Label-Efficient Semantic Segmentation with Diffusion Models

- The experimental results show that the proposed method outperforms Masked Autoencoders, GAN and VAE-based models.

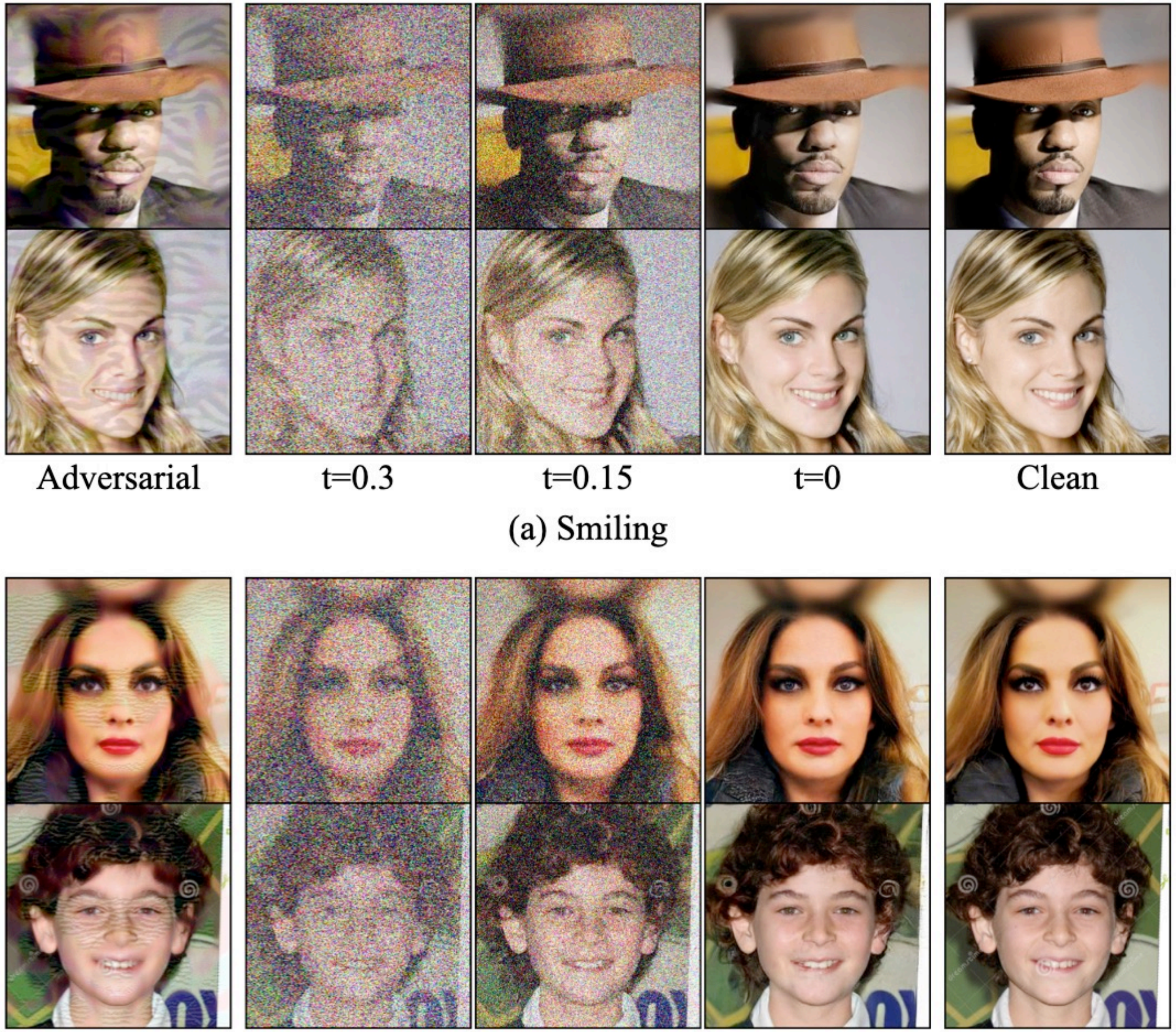


Adversarial Robustness:
Diffusion Models for Adversarial Purification

Lecture #15

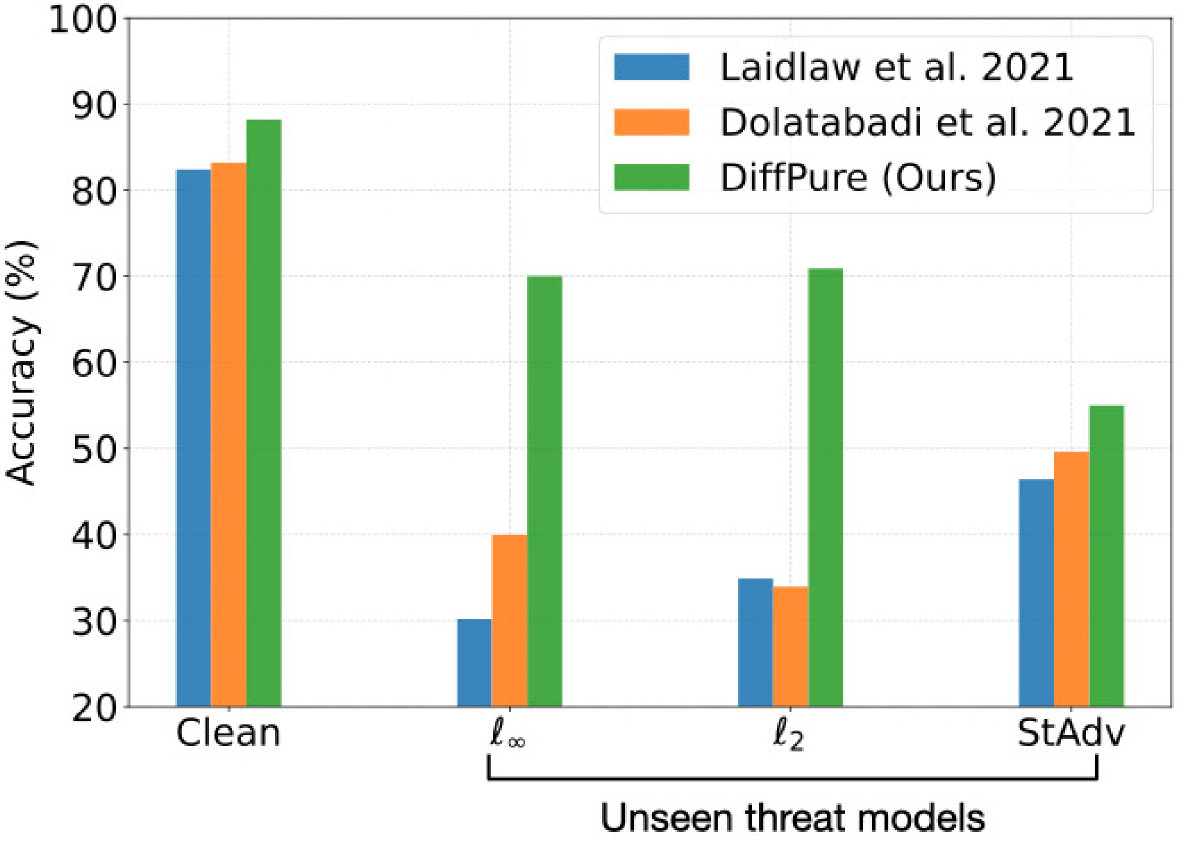


Adversarial Robustness: Diffusion Models for Adversarial Purification



(a) Smiling

(b) Eyeglasses



Comparison with state-of-the-art defense methods against unseen threat models (including AutoAttack ℓ_∞, AutoAttack ℓ_2, and StdAdv) on ResNet-50 for CIFAR-10.

Open Problems

- Diffusion models are a special form of VAEs and continuous normalizing flows:
  - Why do diffusion models perform so much better than these models?
  - How can we improve VAEs and normalizing flows with lessons learned from diffusion models?
- Sampling from diffusion models is still slow especially for interactive applications:
  - The best we could reach is 4-10 steps. How can we have one step samplers?
  - Do we need new diffusion processes?

Open Problems

- Diffusion models can be considered as latent variable models, but their latent space lacks semantics:
  - How can we do latent-space semantic manipulations in diffusion models?
- How can diffusion models help with discriminative applications?
  - Representation learning (high-level vs low-level).
  - Uncertainty estimation.
  - Joint discriminator-generator training.

Open Problems

- What are the best network architectures for diffusion models?
  - Can we go beyond existing U-Nets?
  - How can we feed the time input and other conditioning?
  - How can we improve the sampling efficiency using better network designs?
- How can we apply diffusion models to other data types?
  - 3D data (e.g., distance functions, meshes, voxels, volumetric representations), video, text, graphs, etc.
  - How should we change diffusion models for these modalities?

Open Problems

- Compositional and controllable generation:
  - How can we go beyond images and generate scenes?
  - How can we have more fine-grained control in generation?
- Diffusion models for X:
  - Can we better solve applications that were previously addressed by GANs and other generative models?
  - Which applications will benefit most from diffusion models?

References (1/5)

1. Xiao et al.,“Tackling the Generative Learning Trilemma with Denoising Diffusion GANs”, ICLR 2022.

2. Kingma et al., “Variational Diffusion Models”, NeurIPS 2021.

3. Vahdat & Kautz, “NVAE: A Deep Hierarchical Variational Autoencoder”, NeurIPS 2020.

4. Song et al., “Denoising Diffusion Implicit Models”, ICLR 2021.

5. Karras et al., “Elucidating the Design Space of Diffusion-Based Generative Models”, arXiv 2022.

6. Salimans & Ho, “Progressive Distillation for Fast Sampling of Diffusion Models”, ICLR 2022.

7. Dockhorn et al., “Score-Based Generative Modeling with Critically-Damped Langevin Diffusion”, ICLR 2022.

References (2/5)

8. Gao et al., “Learning energy-based models by diffusion recovery likelihood”, ICLR 2021.
9. Vahdat et al., “Score-Based Generative Modeling in Latent Space”, NeurIPS 2021.
10. Rombach et al., “High-Resolution Image Synthesis with Latent Diffusion Models”, CVPR 2022.
11. Ramesh et al., “Hierarchical Text-Conditional Image Generation with CLIP Latents”, arXiv 2022.
12. Saharia et al., “Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding”, arXiv 2022.
13. Saharia et al., “Palette: Image-to-Image Diffusion Models”, arXiv 2021.
14. Dhariwal and Nichol, “Diffusion models beat GANs on image synthesis”, NeurIPS 2021.

References (3/5)

15. Ho & Salimans, "Classifier-Free Diffusion Guidance", NeurIPS 2021.

16. Ho et al., "Cascaded Diffusion Models for High Fidelity Image Generation", JMLR 2022.

17. Nichol et al., "GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models", ICML 2021.

18. Radford et al., "Learning Transferable Visual Models From Natural Language Supervision", 2021.

19. Ramesh et al., "Hierarchical Text-Conditional Image Generation with CLIP Latents", arXiv 2022.

20. Saharia et al., "Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding", arXiv 2022.

References (4/5)

21. Preechakul et al., “Diffusion Autoencoders: Toward a Meaningful and Decodable Representation”, CVPR 2022.

22. Saharia et al., “Image Super-Resolution via Iterative Refinement”, 2021.

23. Saharia et al., “Palette: Image-to-Image Diffusion Models”, 2022.

24. Choi et al., ILVR: “Conditioning Method for Denoising Diffusion Probabilistic Models”, ICCV 2021.

25. Baranchuk et al., “Label-Efficient Semantic Segmentation with Diffusion Models”, ICLR 2022.

26. Meng et al., SDEdit: “Guided Image Synthesis and Editing with Stochastic Differential Equations”, ICLR 2022.

27. Nie et al., “Diffusion Models for Adversarial Purification”, ICML 2022.

References (5/5)

Based on:

- Kreis, Gao & Vahdat, Tutorial on Denoising Diffusion-based Generative Modeling: Foundations and Applications, CVPR 2022. `https://cvpr2022-tutorial-diffusion-models.github.io/`

Newer tutorial:

- Song, Meng & Vahdat, Denoising Diffusion Models: A Generative Learning Big Bang, CVPR 2023. `https://cvpr2023-tutorial-diffusion-models.github.io/`

Introduction to Deep Generative Modeling

Lecture #15

HY-673 – Computer Science Dep., University of Crete

Professors: Yannis Pantazis, Yannis Stylianou

Teaching Assistant: Michail Raptakis