GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models

Present by: Ahmed, Chandra, Joseph, Muhammad, and Rajat

CAP 6412

Spring 2023

Outline

- Motivation
- Objectives
- Diffusion Model
- GLIDE
- Image Inpainting
- Results
- Conclusion

GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models

Alex Nichol * Prafulla Dhariwal * Aditya Ramesh * Pranav Shyam Pamela Mishkin Bob McGrew
Ilya Sutskever Mark Chen

12.10741v3 [cs.CV] 8 Mar 2022

Abstract

Diffusion models have recently been shown to generate high-quality synthetic images, especially when paired with a guidance technique to trade off diversity for fidelity. We explore diffusion models for the problem of text-conditional image synthesis and compare two different guidance strategies: CLIP guidance and classifier-free guidance. We find that the latter is preferred by human evaluators for both photorealism and caption similarity, and often produces photorealistic samples. Samples from a 3.5 billion parameter text-conditional diffusion model using classifier-free guidance are favored by human evaluators to those from DALL-E, even when the latter uses expensive CLIP reranking. Additionally, we find that our models can be fine-tuned to perform image inpainting, enabling powerful text-driven image editing. We train a smaller model on a filtered dataset and release the code and weights at https://github.com/openai/glide-text2im.

their corresponding text prompts.

On the other hand, unconditional image models can synthesize photorealistic images (Brock et al., 2018; Karras et al., 2019a;b; Razavi et al., 2019), sometimes with enough fidelity that humans can't distinguish them from real images (Zhou et al., 2019). Within this line of research, diffusion models (Sohl-Dickstein et al., 2015; Song & Ermon, 2020b) have emerged as a promising family of generative models, achieving state-of-the-art sample quality on a number of image generation benchmarks (Ho et al., 2020; Dhariwal & Nichol, 2021; Ho et al., 2021).

To achieve photorealism in the class-conditional setting, Dhariwal & Nichol (2021) augmented diffusion models with *classifier guidance*, a technique which allows diffusion models to condition on a classifier's labels. The classifier is first trained on noised images, and during the diffusion sampling process, gradients from the classifier are used to guide the sample towards the label. Ho & Salimans (2021) achieved similar results without a separately trained classifier through the use of *classifier-free guidance*, a form of guidance that interpolates between predictions from a diffusion model with and without labels.

https://github.com/openai/glide-text2im

Motivation

Diffusion models have revolutionized generating photorealistic images from text prompts.

“a hedgehog using a calculator”

Motivation

Diffusion models have revolutionized generating photorealistic images from text prompts.

“a hedgehog using a calculator”

“a painting of a fox in the style of starry night”

Motivation

One of the interesting applications of diffusion models is **Image editing**, which is making realistic edits to an image based on natural language prompts.

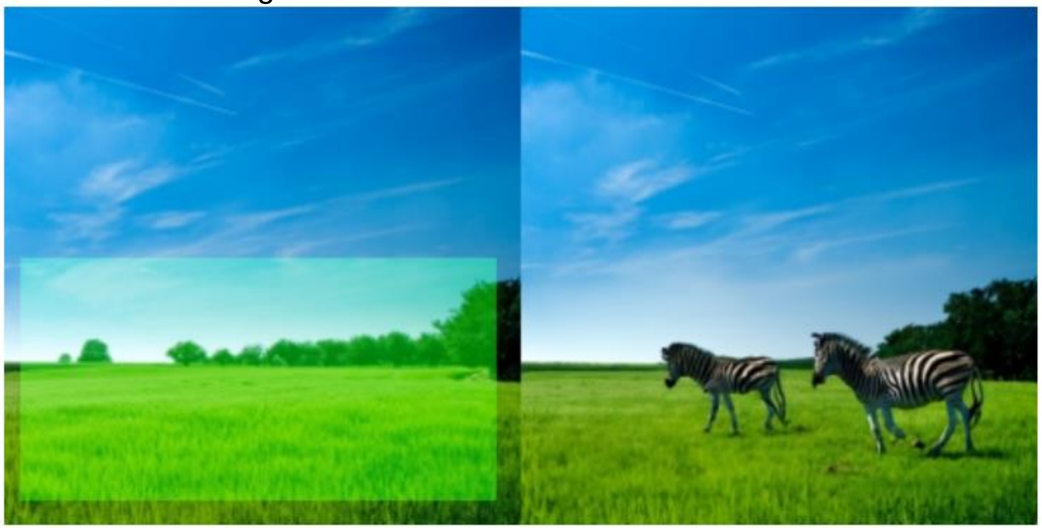
Original

“zebras roaming in the field”

Motivation

One of the interesting applications of diffusion models is **Image editing**, which is making realistic edits to an image based on natural language prompts.

Original Edited



“zebras roaming in the field”

Objectives

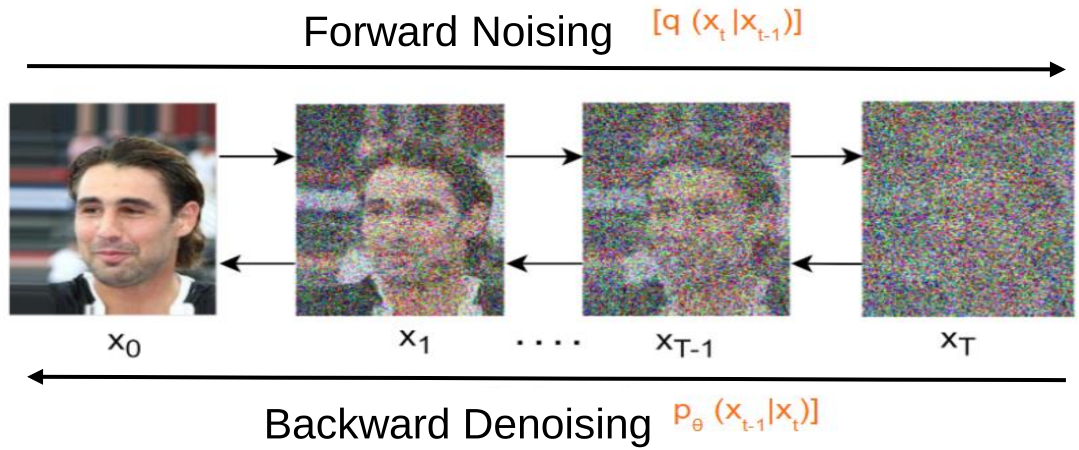
Objectives

- Develop guided diffusion model to generate photorealistic images given text prompts using,
  - CLIP guidance
  - Classifier-free guidance

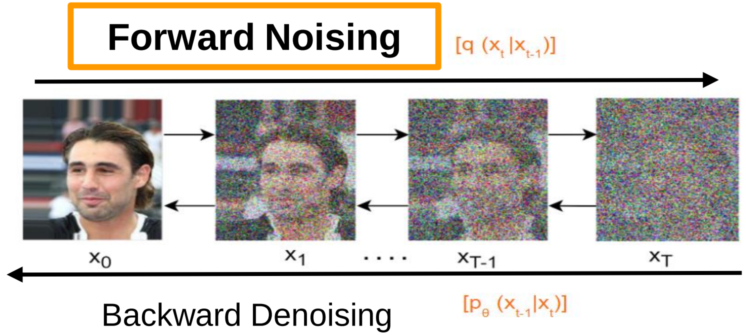
- Perform image inpainting

Diffusion Model

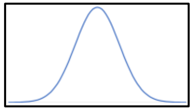
Diffusion Model



- Noise is added iteratively to generate sample noised images.
- A model is learned to take noised image and iteratively generate denoised samples.



Forward process

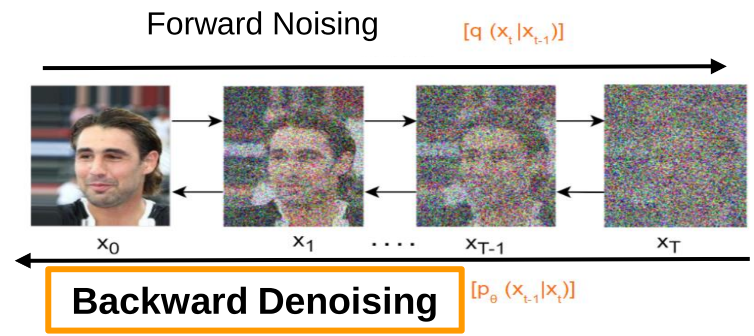
Noise Adding Function 

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

Where:

- N is the Gaussian distribution
- a_t is a hyperparameter variance scheduler
- I is the identity matrix

Backward Process



Inference

$$p_\theta(x_{t-1}|x_t) := \mathcal{N}(\mu_\theta(x_t), \Sigma_\theta(x_t))$$

Where:

- N is the Gaussian distribution
- $\mu_\theta(x_t)$ is the learned mean vector
- $\Sigma_\theta(x_t)$ is the learned covariance vector

Text-Guided Diffusion Model

Text-Conditioned Diffusion?

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(\mu_\theta(x_t), \Sigma_\theta(x_t))$$

Text-Conditioned Diffusion?

You already understand the Diffusion.

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(\mu_\theta(x_t), \Sigma_\theta(x_t))$$

Text-Conditioned Diffusion?

You already understand the Diffusion.

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(\mu_\theta(x_t), \Sigma_\theta(x_t))$$

But you need something More Controlled.

Text-Conditioned Diffusion?

You already understand the Diffusion.

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(\mu_\theta(x_t), \Sigma_\theta(x_t))$$

But you need something More Controlled.

Label = “Goldfinch”

Text-Conditioned Diffusion?

You already understand the Diffusion.

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(\mu_\theta(x_t), \Sigma_\theta(x_t))$$

But you need something More Controlled.

Label = "Goldfinch" 

Text-Conditioned Diffusion?

You already understand the Diffusion.

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(\mu_\theta(x_t), \Sigma_\theta(x_t))$$

But you need something More Controlled.

Label = “Goldfinch” 

Label = “robots meditating in a vipassana retreat”

Text-Conditioned Diffusion?

You already understand the Diffusion.

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(\mu_\theta(x_t), \Sigma_\theta(x_t))$$

But you need something More Controlled.

Label = “Goldfinch”

Label = “robots meditating in a vipassana retreat”

Text-Conditioned Diffusion?

You already understand the Diffusion.

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(\mu_\theta(x_t), \Sigma_\theta(x_t))$$

But you need something More Controlled.

Label = "Goldfinch"

Label = "robots meditating in a vipassana retreat"

$$p_\theta(x_{t-1}|x_t, y)$$

Text-Conditioned Diffusion?

You already understand the Diffusion.

$$p_\theta(x_{t-1}|\boxed{x_t}) = \mathcal{N}(\mu_\theta(x_t), \Sigma_\theta(x_t))$$

But you need something More Controlled.

Label = “Goldfinch”

Label = “robots meditating in a vipassana retreat”

$$p_\theta(x_{t-1}|\boxed{x_t, y})$$

Text-Conditioned Diffusion?

You already understand the Diffusion.

$$p_\theta(x_{t-1}|\boxed{x_t}) = \mathcal{N}(\mu_\theta(x_t), \Sigma_\theta(x_t))$$

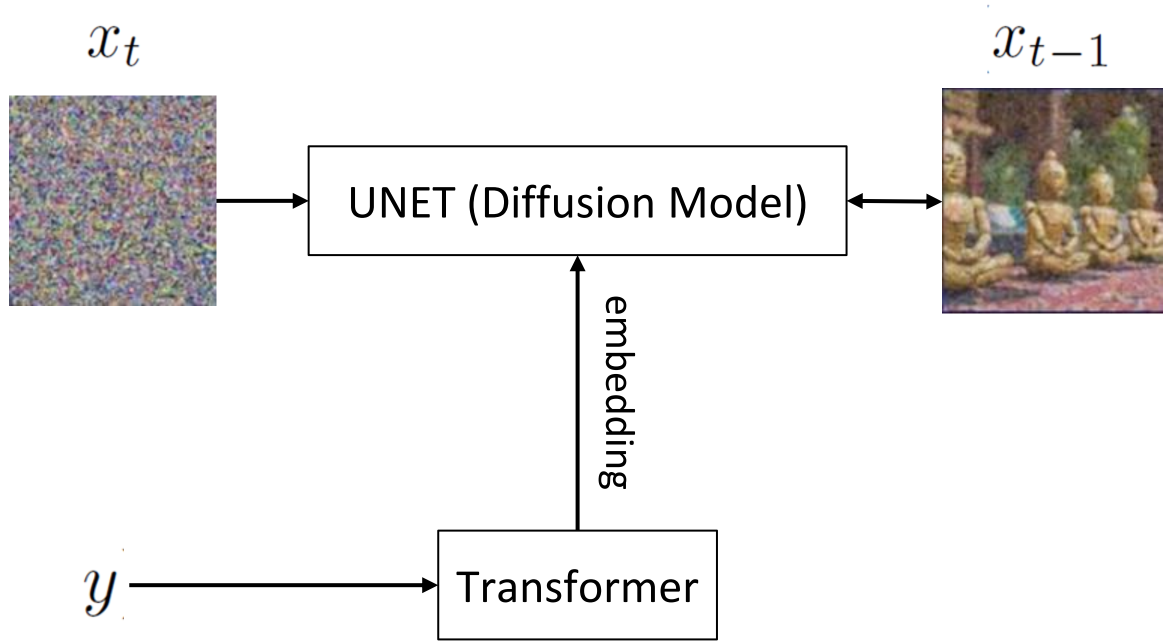
But you need something More Controlled.

Label = "Goldfinch"

Label = "robots meditating in a vipassana retreat"

$$p_\theta(x_{t-1}|\boxed{x_t, y}) \rightarrow$$ Text-Conditioned Diffusion

Text-Conditioned Diffusion



Convert text to discrete tokens & attend to them in UNET

But. Naïve Text Conditional Models = Incoherent Samples

But. Naïve Text Conditional Models = Incoherent Samples

Solution: Guidance

Guidance |

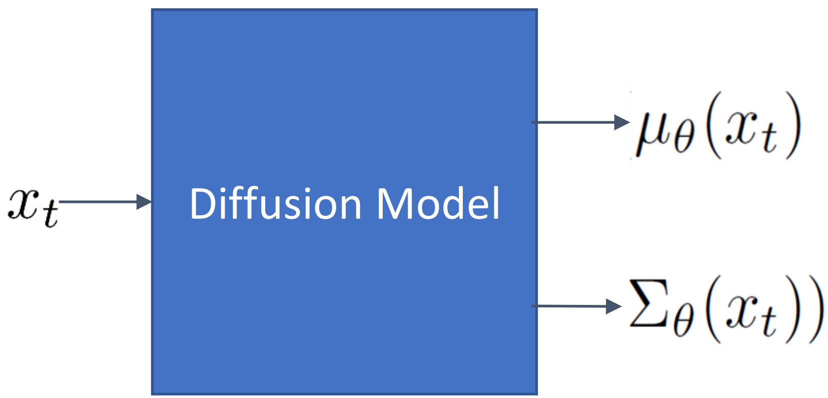
Guidance | ?

Guidance | ?

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(\mu_\theta(x_t), \Sigma_\theta(x_t))$$

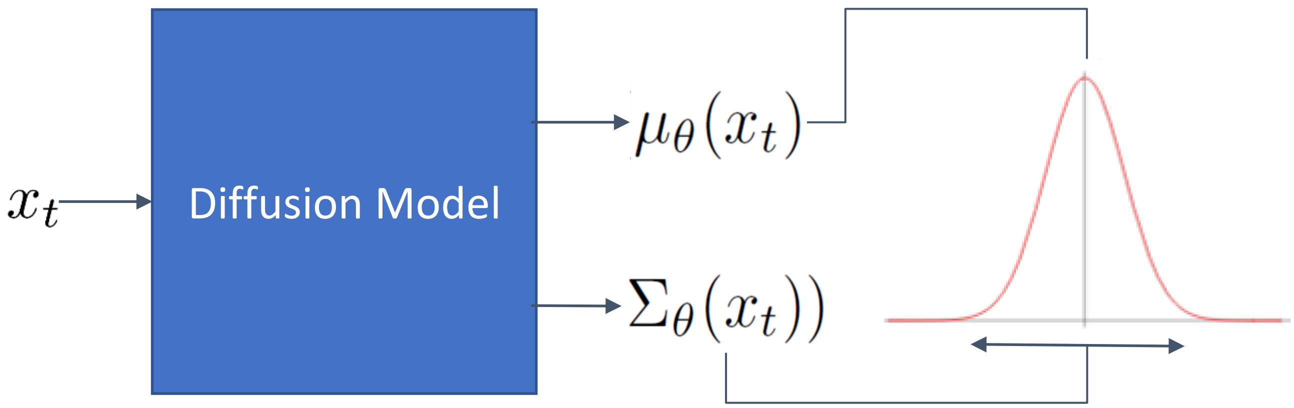
Guidance | ?

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(\mu_\theta(x_t), \Sigma_\theta(x_t))$$



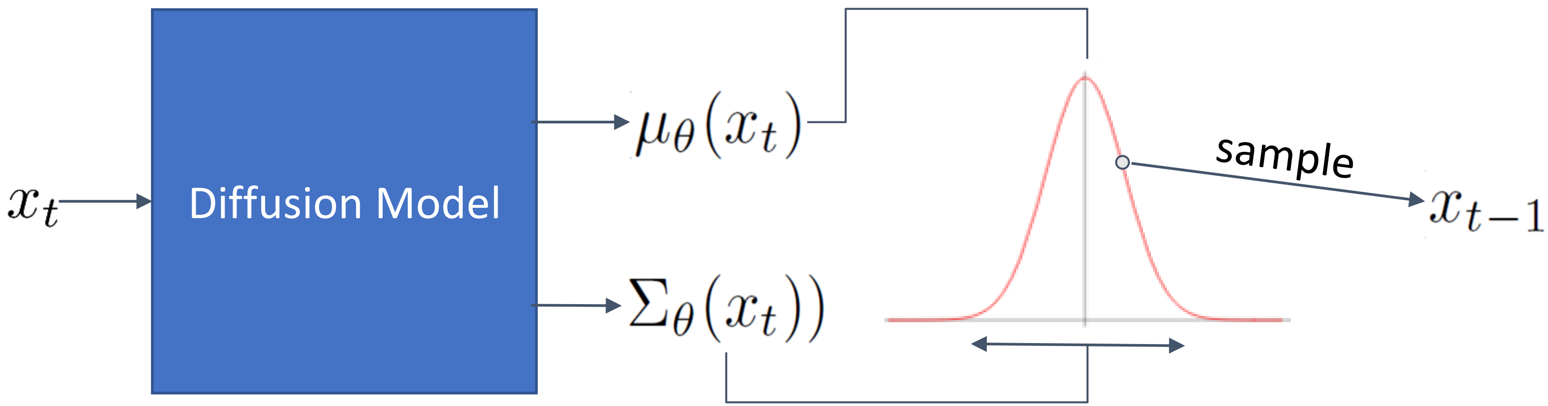
Guidance | ?

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(\mu_\theta(x_t), \Sigma_\theta(x_t))$$



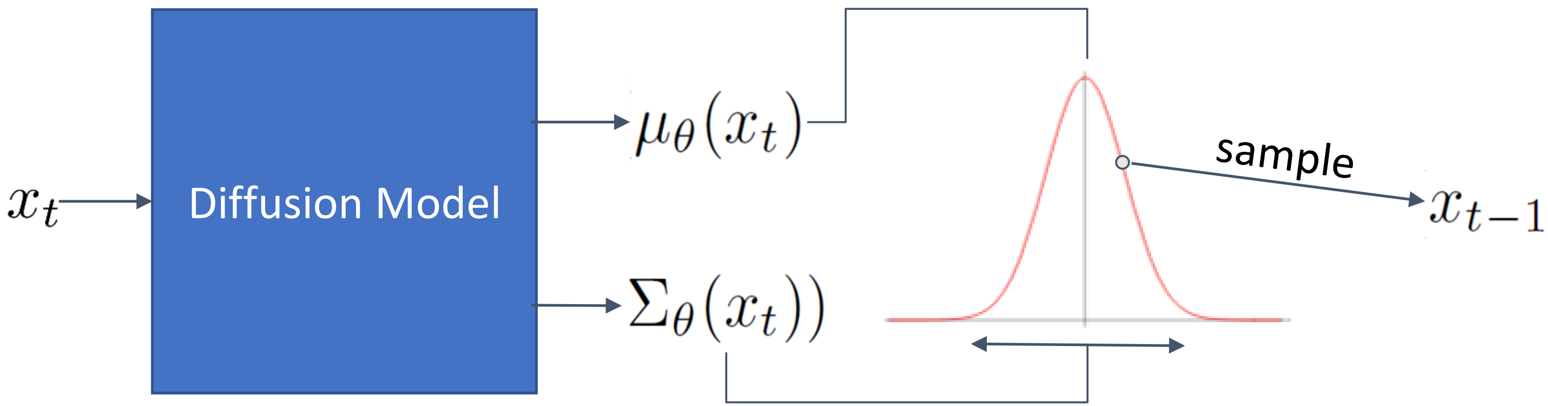
Guidance | ?

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(\mu_\theta(x_t), \Sigma_\theta(x_t))$$



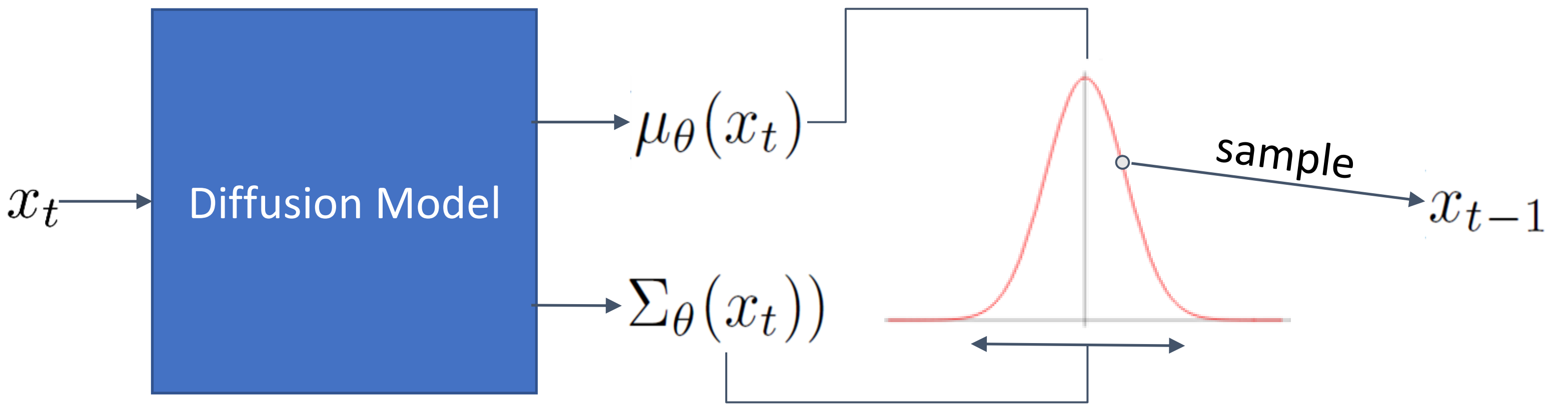
Guidance | ?

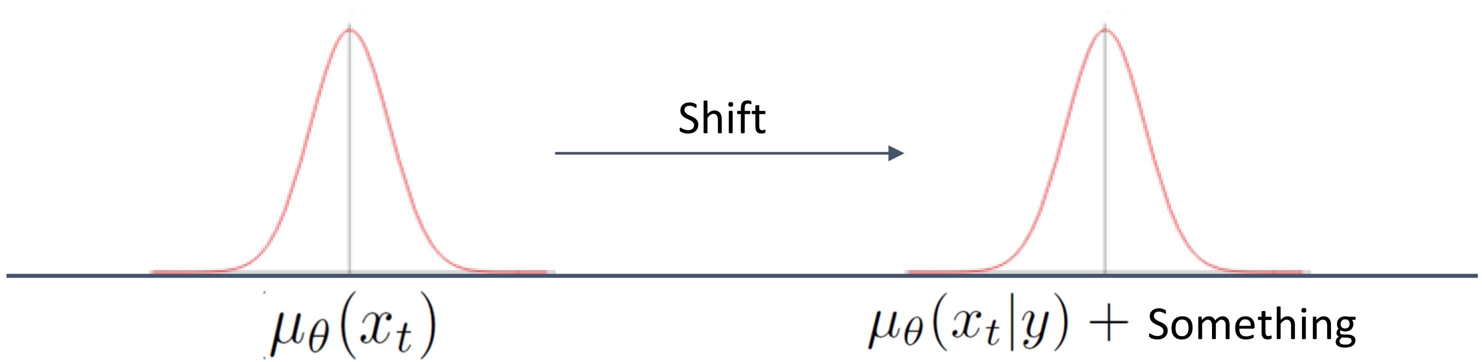
$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(\mu_\theta(x_t), \Sigma_\theta(x_t))$$



Guidance | ?

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(\mu_\theta(x_t), \Sigma_\theta(x_t))$$





Guidance |

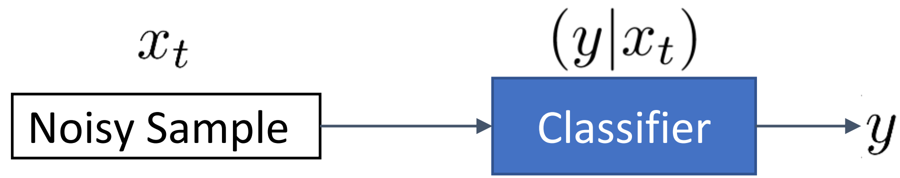
Guidance | Simple Classifier-Based Guidance | Label = “Goldfinch”

Guidance | Simple Classifier-Based Guidance | Label = “Goldfinch”

First: Train a Classifier.

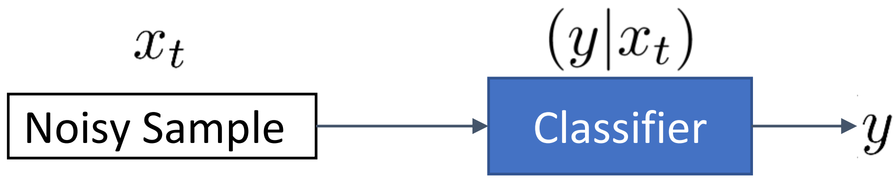
Guidance | Simple Classifier-Based Guidance | Label = “Goldfinch”

First: Train a Classifier.



Guidance | Simple Classifier-Based Guidance | Label = “Goldfinch”

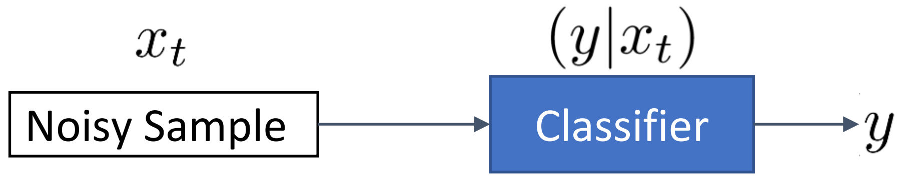
First: Train a Classifier.



Then in Diffusion:

Guidance | Simple Classifier-Based Guidance | Label = "Goldfinch"

First: Train a Classifier.



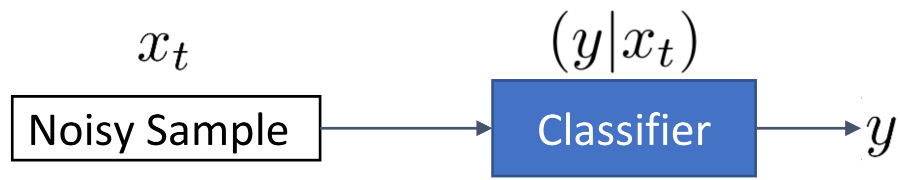
Then in Diffusion:

Pass x_t through classifier; get y. Compute gradient of log-probability of y by x_t

.

Guidance | Simple Classifier-Based Guidance | Label = "Goldfinch"

First: Train a Classifier.

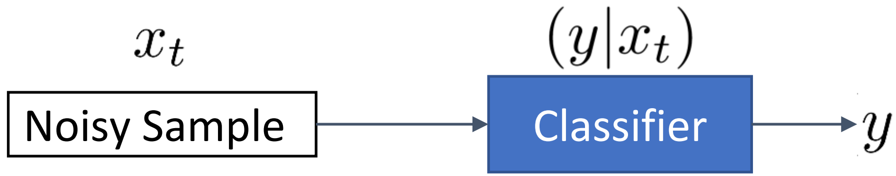


Then in Diffusion:

Pass x_t through classifier; get y. Compute $\nabla_{x_t} \log p_\phi(y|x_t)$

Guidance | Simple Classifier-Based Guidance | Label = “Goldfinch”

First: Train a Classifier.



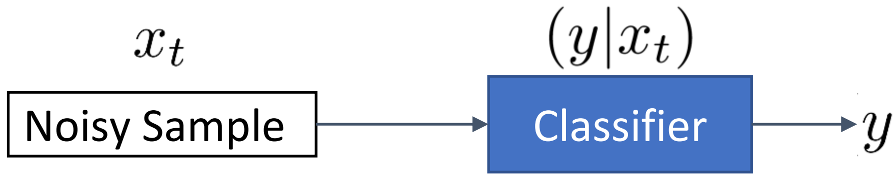
Then in Diffusion:

Pass x_t through classifier; get y. Compute $\nabla_{x_t} \log p_\phi(y|x_t)$

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(\mu_\theta(x_t), \Sigma_\theta(x_t))$$

Guidance | Simple Classifier-Based Guidance | Label = "Goldfinch"

First: Train a Classifier.



Then in Diffusion:

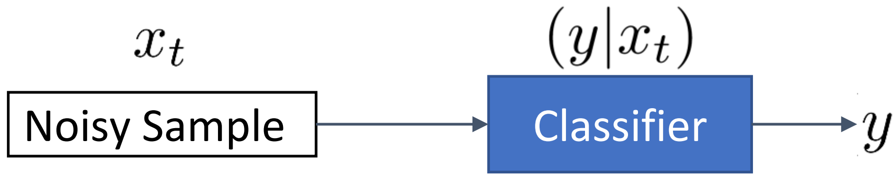
Pass x_t through classifier; get y. Compute $\nabla_{x_t} \log p_\phi(y|x_t)$

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(\boxed{\mu_\theta(x_t)}, \Sigma_\theta(x_t))$$

$$\boxed{\hat{\mu}_\theta(x_t|y) = \mu_\theta(x_t|y) + s \cdot \Sigma_\theta(x_t|y) \nabla_{x_t} \log p_\phi(y|x_t)}$$

Guidance | Simple Classifier-Based Guidance | Label = “Goldfinch”

First: Train a Classifier.



Then in Diffusion:

Pass x_t through classifier; get y . Compute $\boxed{\nabla_{x_t} \log p_\phi(y|x_t)}$

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(\boxed{\mu_\theta(x_t)}, \Sigma_\theta(x_t))$$

$$\boxed{\hat{\mu}_\theta(x_t|y) = \mu_\theta(x_t|y) + s \cdot \Sigma_\theta(x_t|y) \boxed{\nabla_{x_t} \log p_\phi(y|x_t)}}$$

Guidance | Simple Classifier-Based Guidance | Label = “Goldfinch”

First: Train a Classifier.

$$x_t \quad \boxed{\text{Noisy Sample}} \longrightarrow \underset{}{\overset{(y|x_t)}{\boxed{\text{Classifier}}}} \longrightarrow y$$

Then in Diffusion:

Pass x_t through classifier; get y. Compute $\boxed{\nabla_{x_t} \log p_\phi(y|x_t)}$

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(\boxed{\mu_\theta(x_t)}, \Sigma_\theta(x_t))$$

Influence

$$\boxed{\hat{\mu}_\theta(x_t|y) = \mu_\theta(x_t|y) + \boxed{s} \cdot \Sigma_\theta(x_t|y) \boxed{\nabla_{x_t} \log p_\phi(y|x_t)}}$$

Guidance | Simple Classifier-Based Guidance | Label = “Goldfinch”

First: Train a Classifier.

x_t $(y|x_t)$

Noisy Sample → Classifier → y

Then in Diffusion:

Pass x_t through classifier; get y. Compute $\boxed{\nabla_{x_t} \log p_\phi(y|x_t)}$

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(\boxed{\mu_\theta(x_t)}, \Sigma_\theta(x_t))$$

Influence

$$\boxed{\hat{\mu}_\theta(x_t|y) = \mu_\theta(x_t|y) + \boxed{s} \cdot \Sigma_\theta(x_t|y) \boxed{\nabla_{x_t} \log p_\phi(y|x_t)}}$$

But. You need even more Control. Label = “robots meditating in a vipassana retreat”

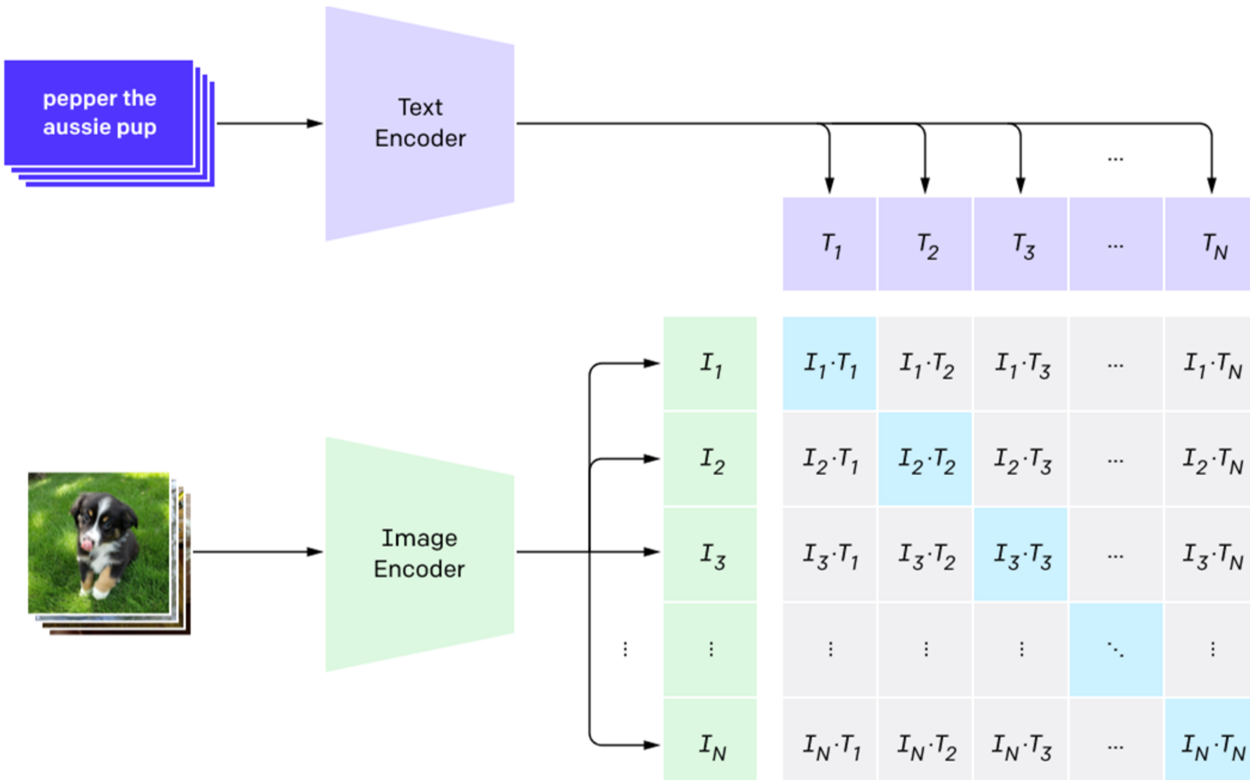
Guidance |

Guidance | CLIP-Based Guidance | Label = “robots meditating in a vipassana retreat”

Guidance | CLIP-Based Guidance | Label = “robots meditating in a vipassana retreat”

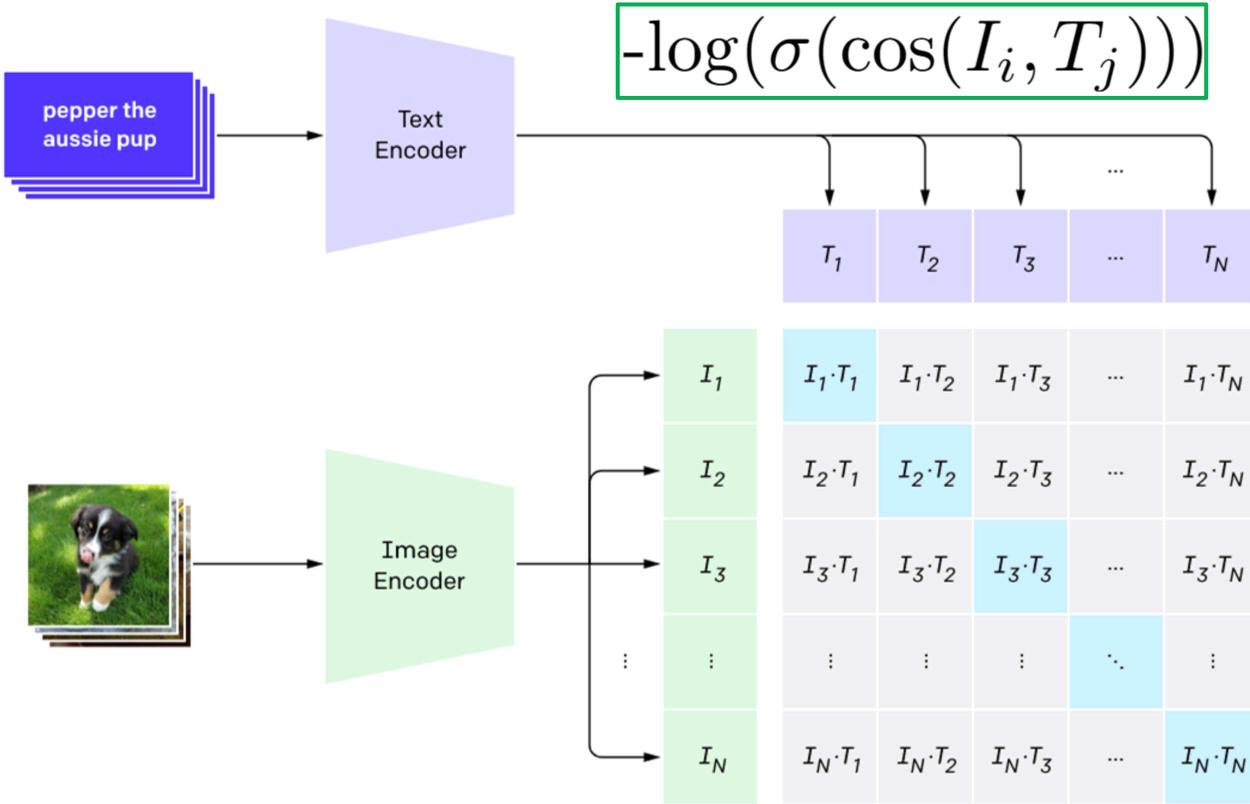
First: Train a CLIP model.

Guidance | CLIP-Based Guidance | Label = "robots meditating in a vipassana retreat"

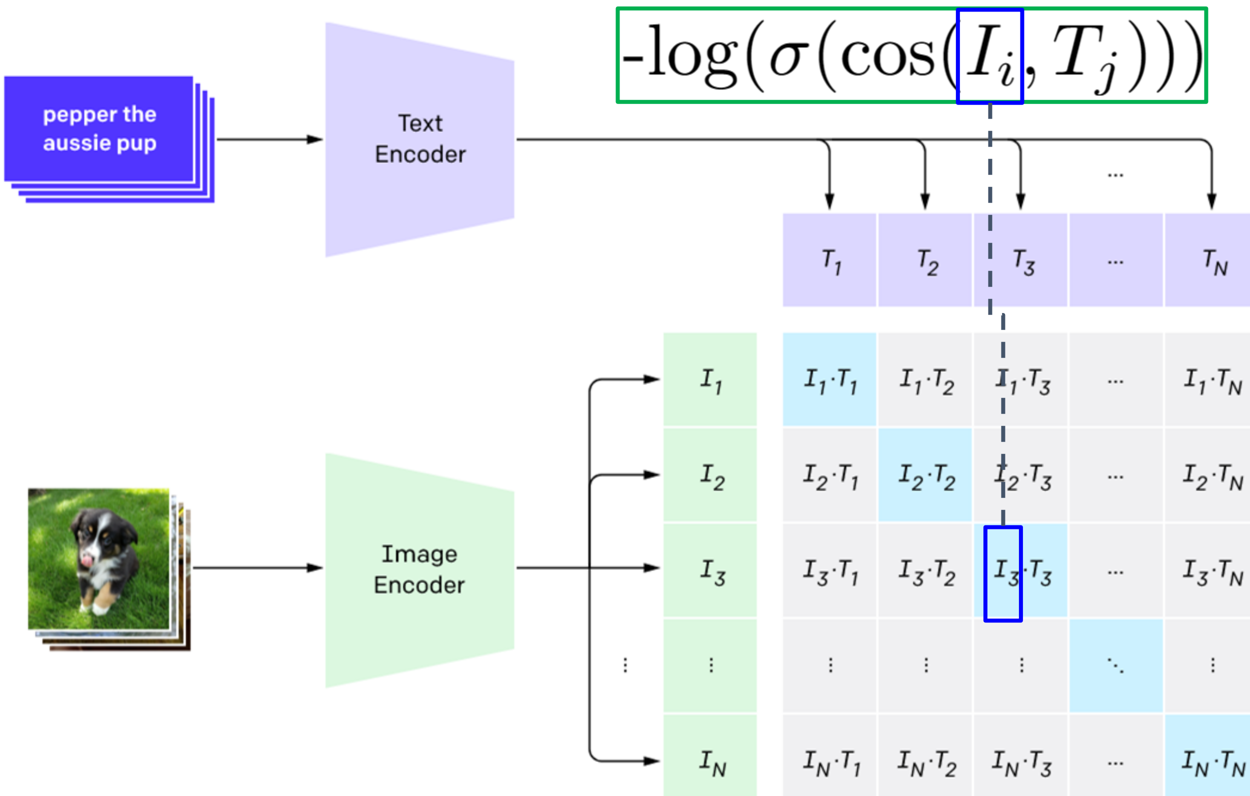


Guidance | CLIP-Based Guidance | Label = "robots meditating in a vipassana retreat"

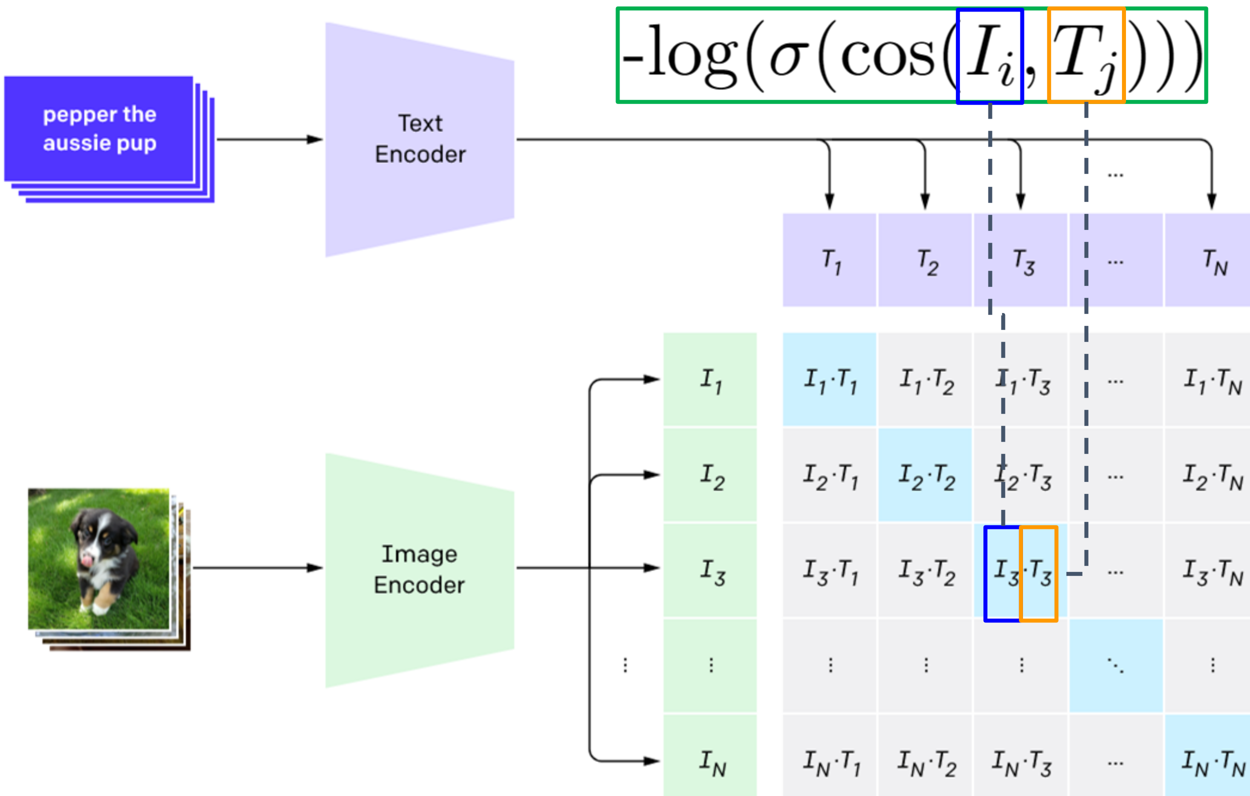
$$\boxed{-\log(\sigma(\cos(I_i, T_j)))}$$



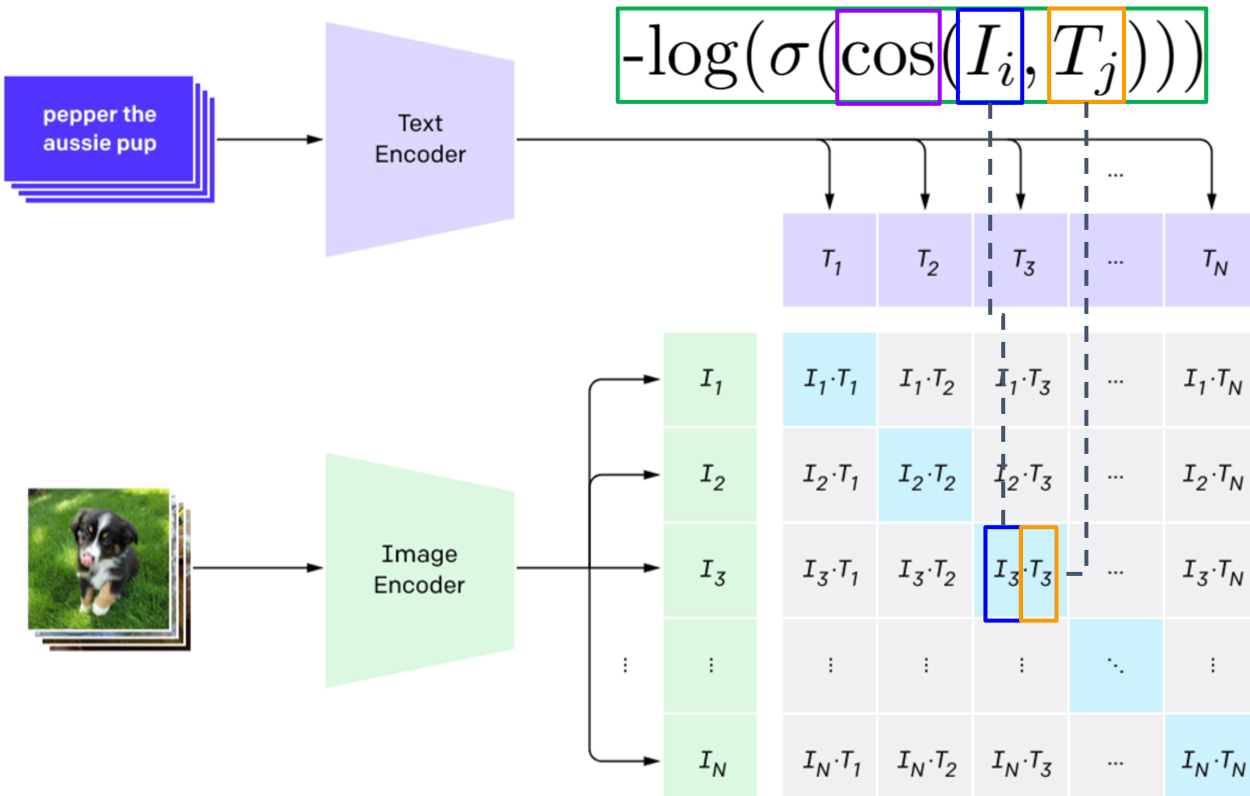
Guidance | CLIP-Based Guidance | Label = “robots meditating in a vipassana retreat”



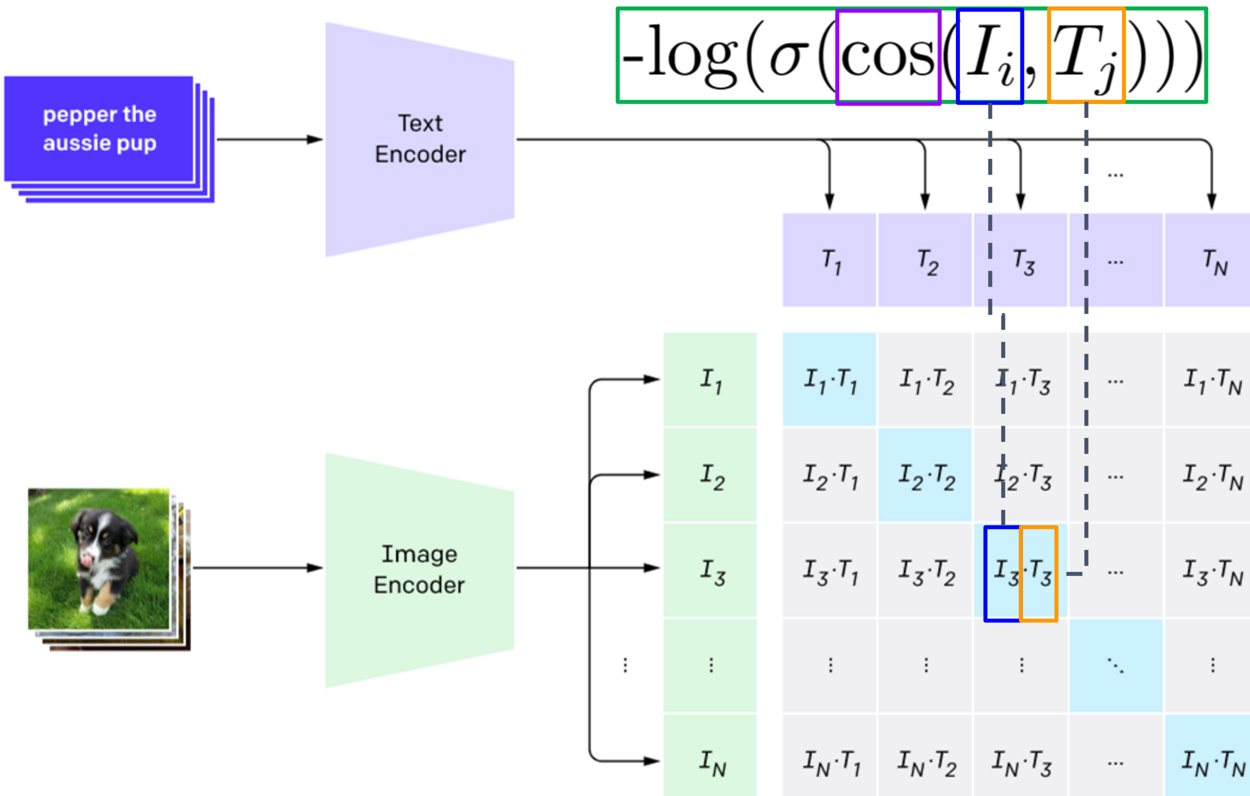
Guidance | CLIP-Based Guidance | Label = “robots meditating in a vipassana retreat”

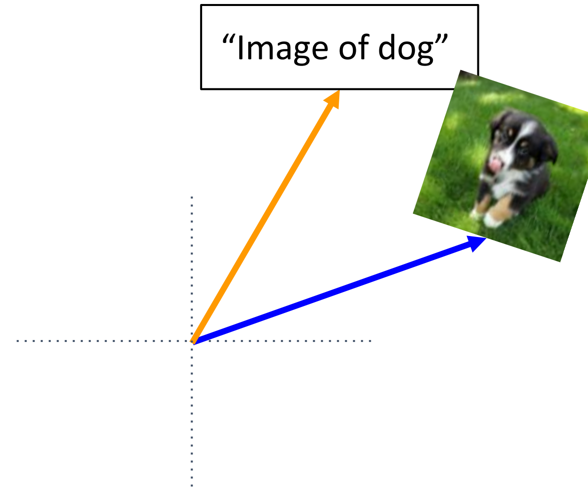


Guidance | CLIP-Based Guidance | Label = “robots meditating in a vipassana retreat”

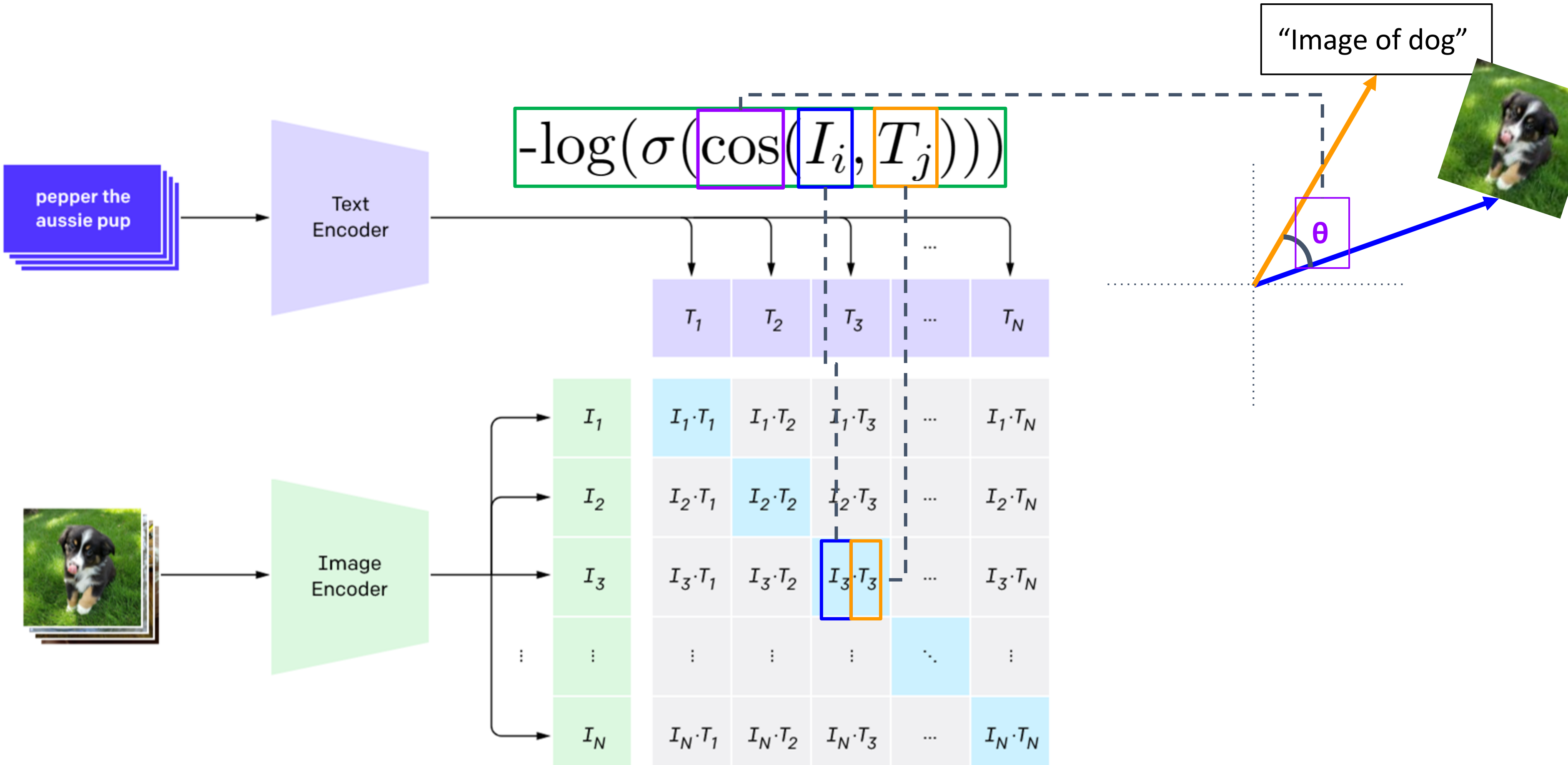


Guidance | CLIP-Based Guidance | Label = “robots meditating in a vipassana retreat”

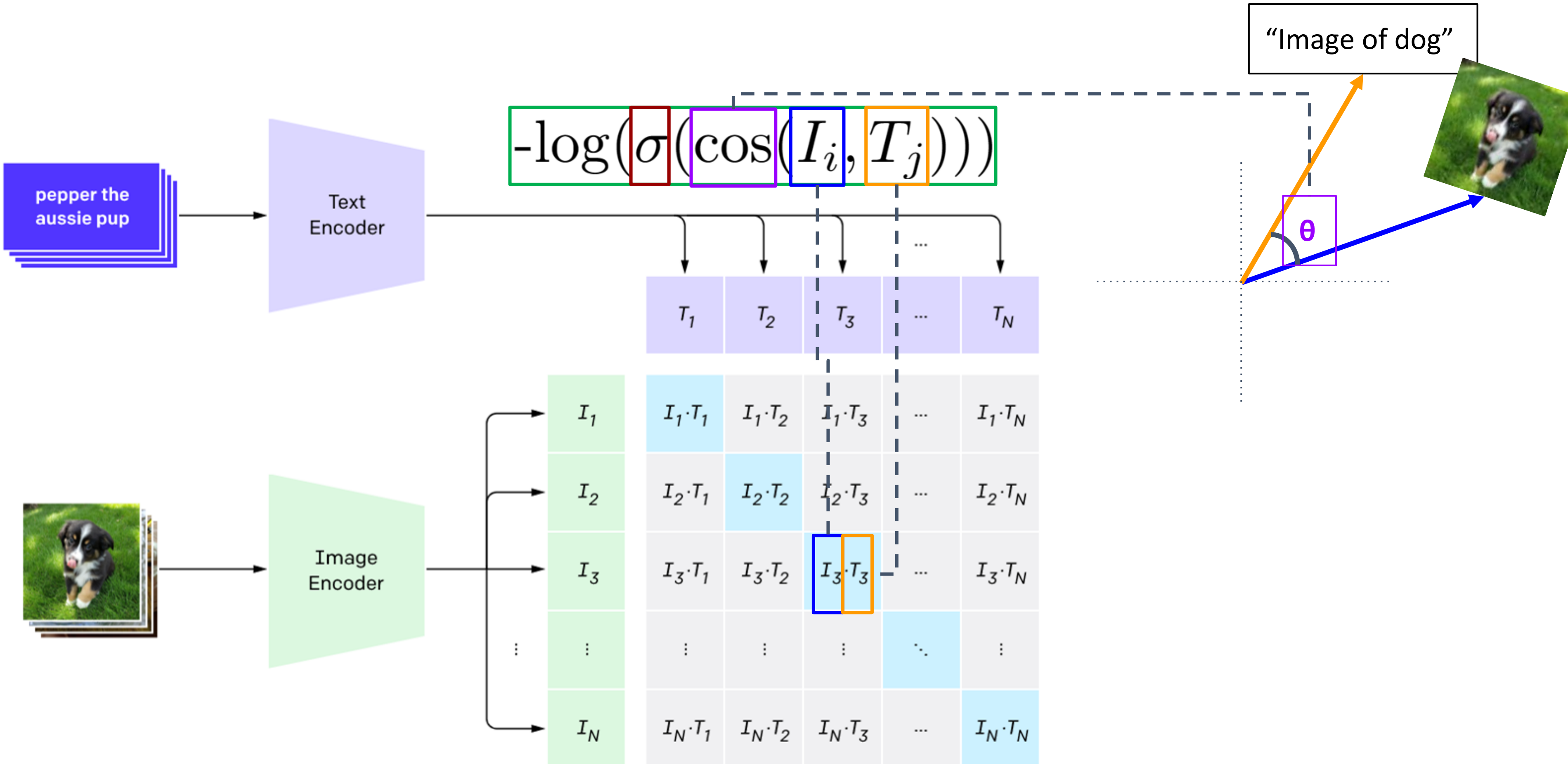




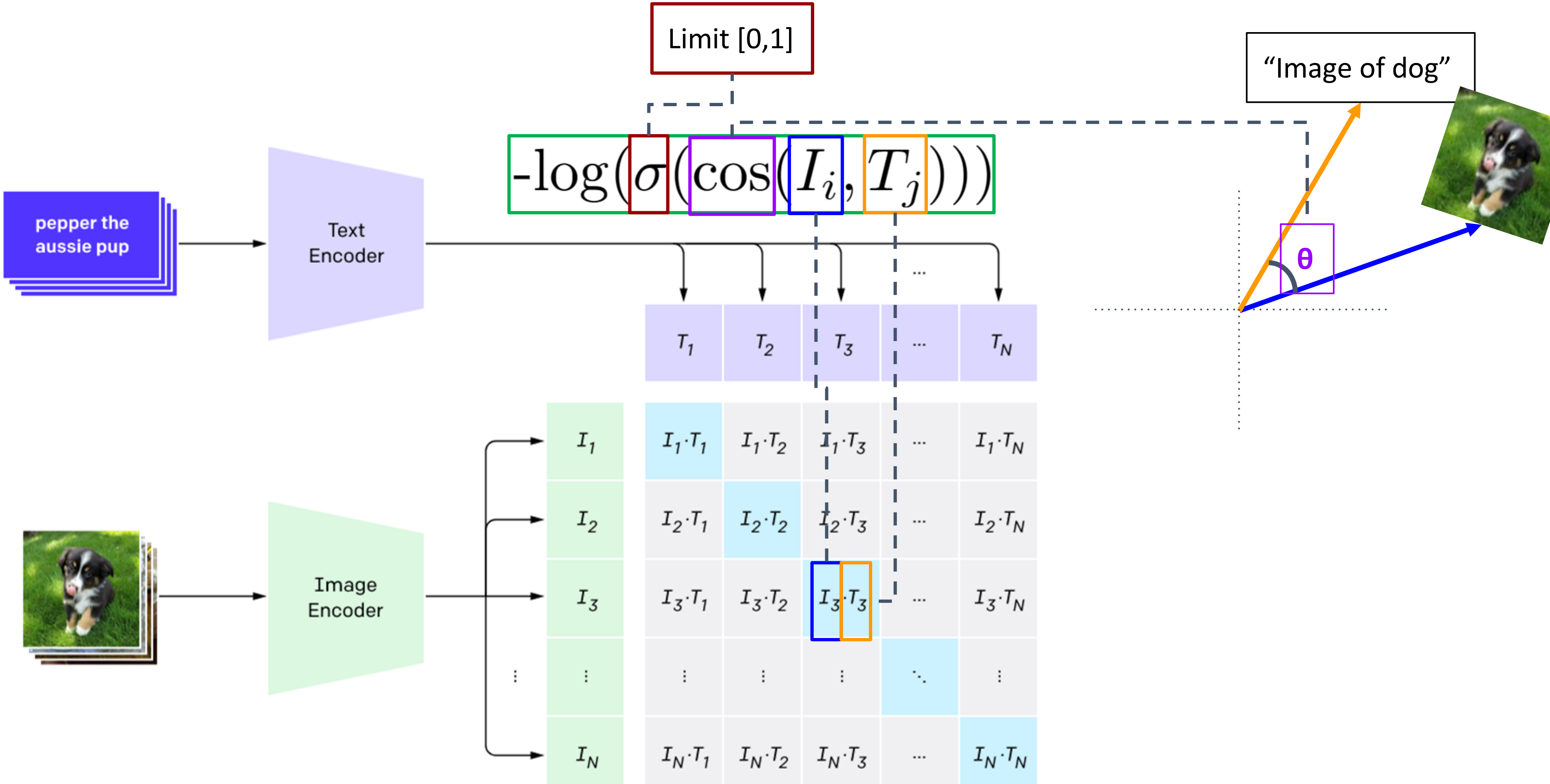
Guidance | CLIP-Based Guidance | Label = “robots meditating in a vipassana retreat”



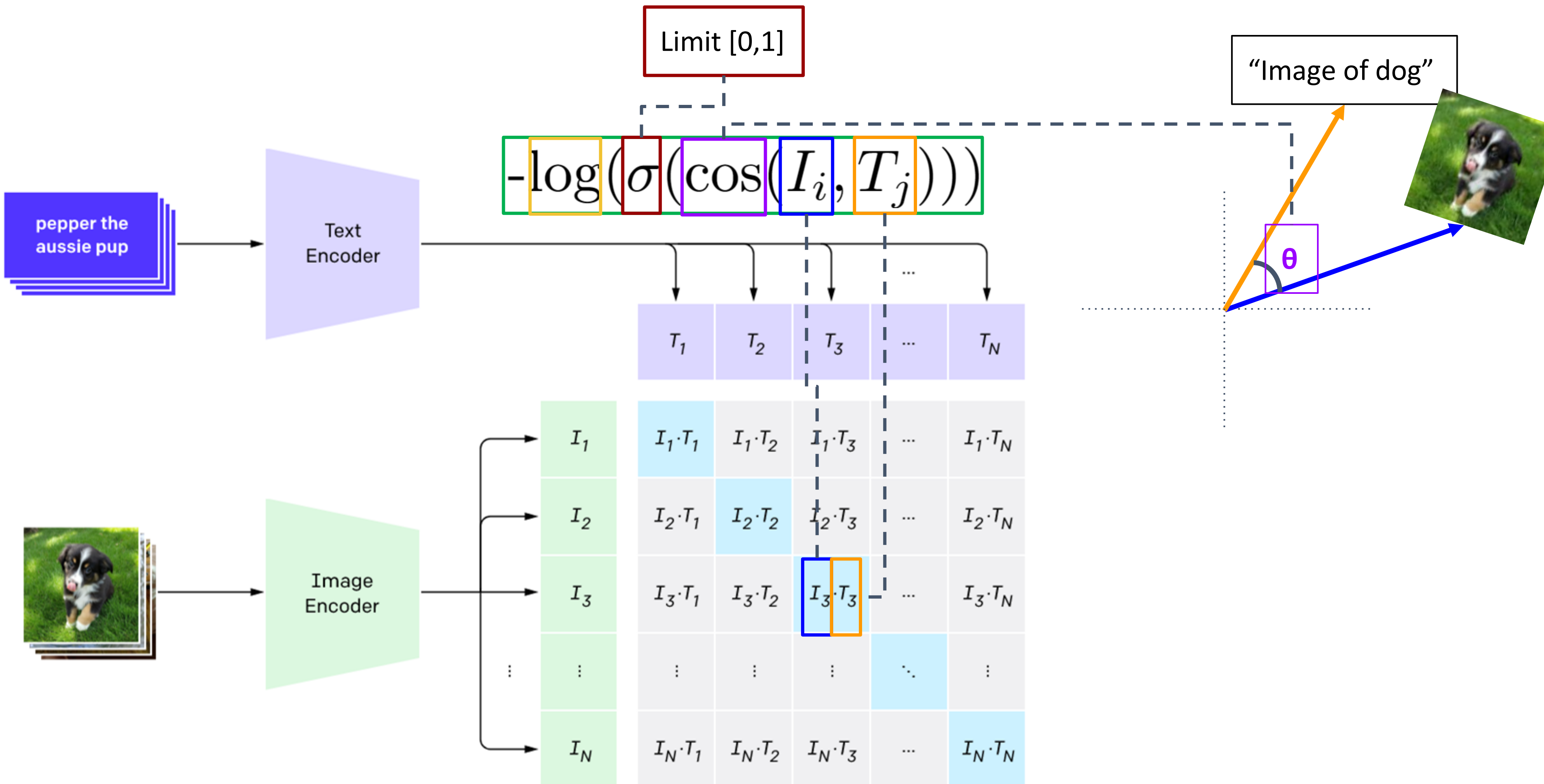
Guidance | CLIP-Based Guidance | Label = “robots meditating in a vipassana retreat”



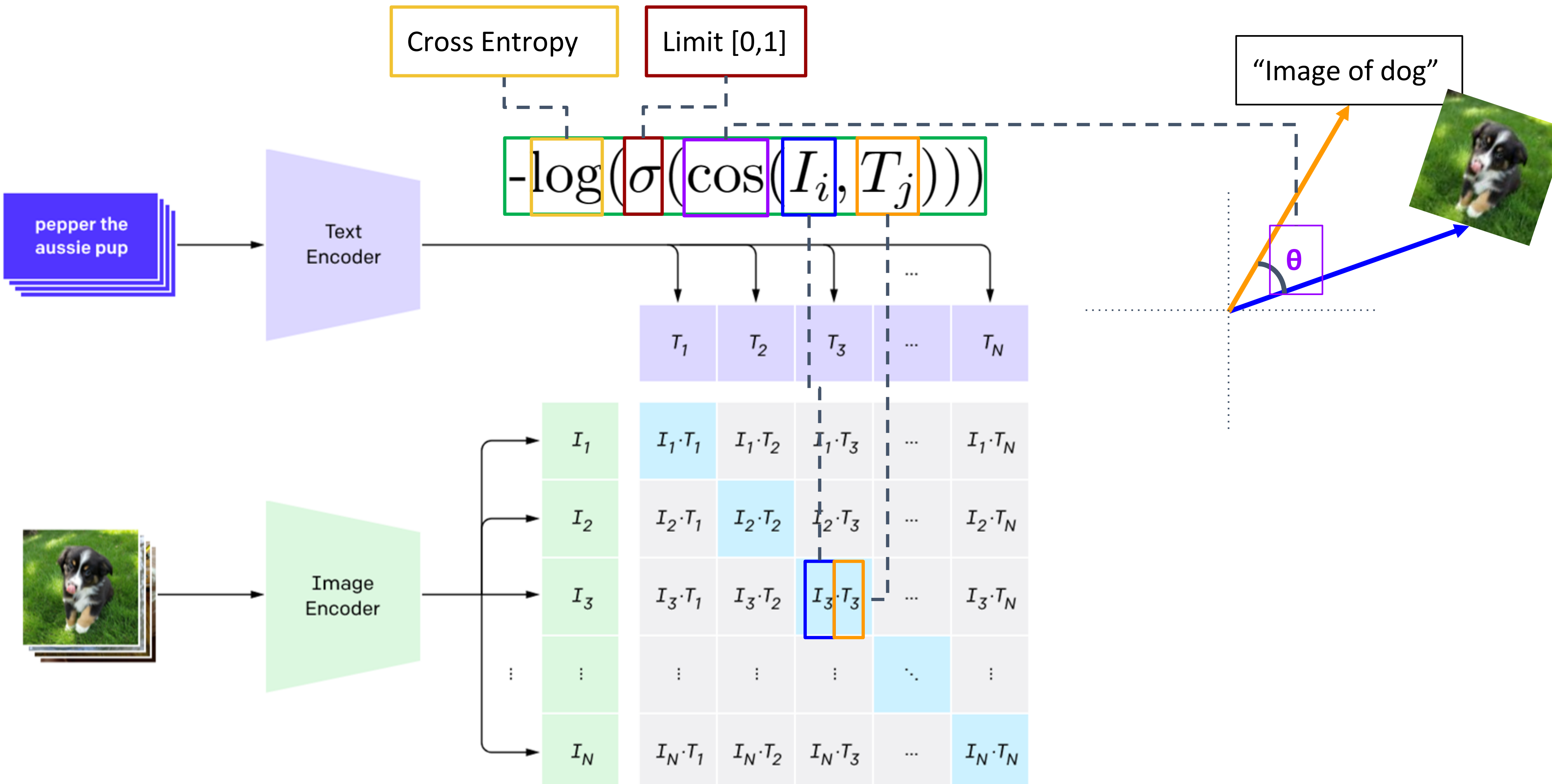
Guidance | CLIP-Based Guidance | Label = “robots meditating in a vipassana retreat”



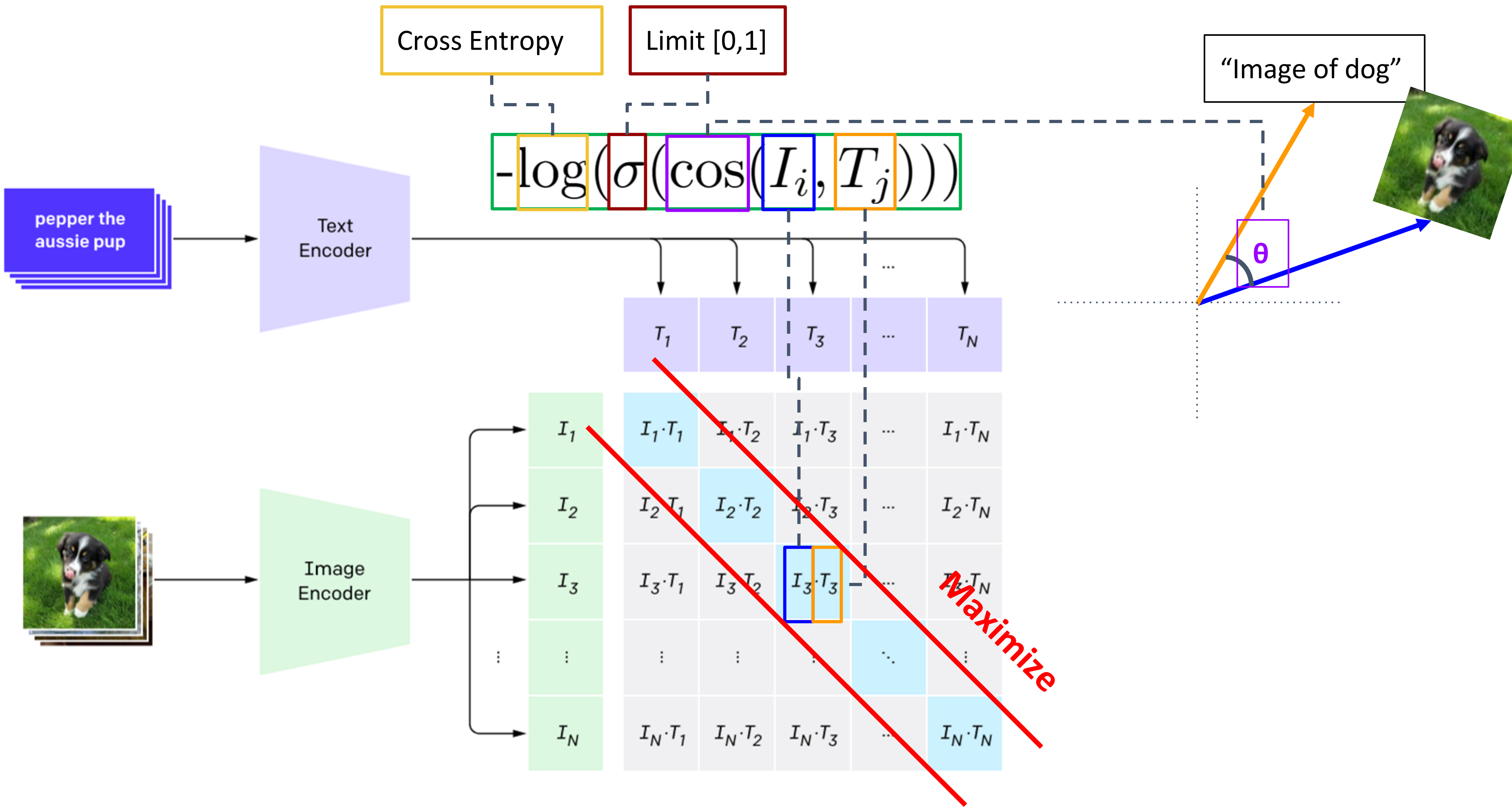
Guidance | CLIP-Based Guidance | Label = “robots meditating in a vipassana retreat”



Guidance | CLIP-Based Guidance | Label = “robots meditating in a vipassana retreat”



Guidance | CLIP-Based Guidance | Label = “robots meditating in a vipassana retreat”



Guidance | CLIP-Based Guidance | Label = “robots meditating in a vipassana retreat”

First: Train a CLIP model.

Guidance | CLIP-Based Guidance | Label = “robots meditating in a vipassana retreat”

First: Train a CLIP model.

Then in Diffusion:

Guidance | CLIP-Based Guidance | Label = “robots meditating in a vipassana retreat”

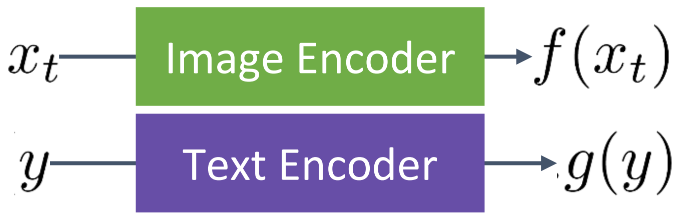
First: Train a CLIP model.

Then in Diffusion:

Guidance | CLIP-Based Guidance | Label = “robots meditating in a vipassana retreat”

First: Train a CLIP model.

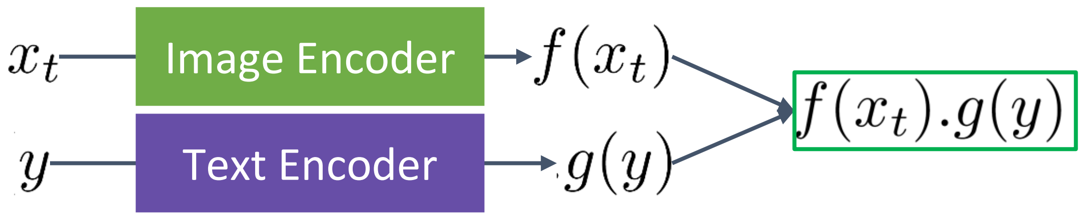
Then in Diffusion:



Guidance | CLIP-Based Guidance | Label = “robots meditating in a vipassana retreat”

First: Train a CLIP model.

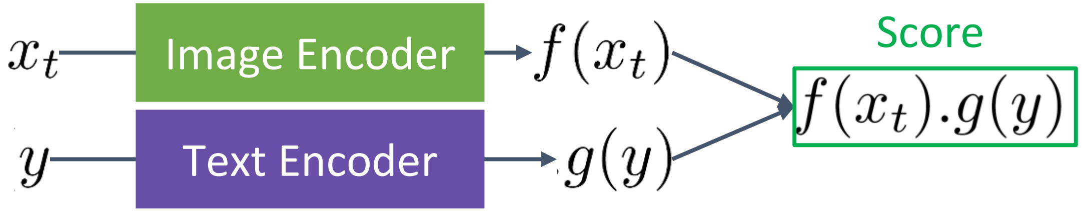
Then in Diffusion:



Guidance | CLIP-Based Guidance | Label = “robots meditating in a vipassana retreat”

First: Train a CLIP model.

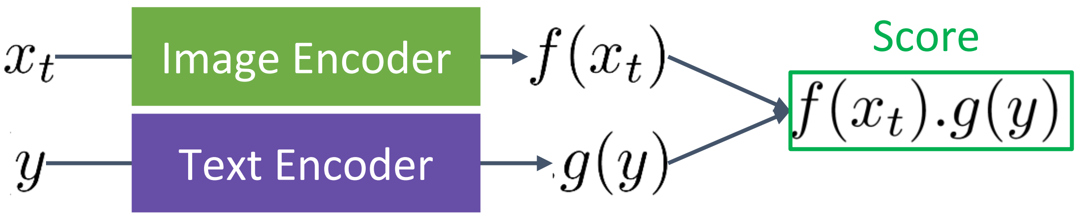
Then in Diffusion:



Guidance | CLIP-Based Guidance | Label = “robots meditating in a vipassana retreat”

First: Train a CLIP model.

Then in Diffusion:

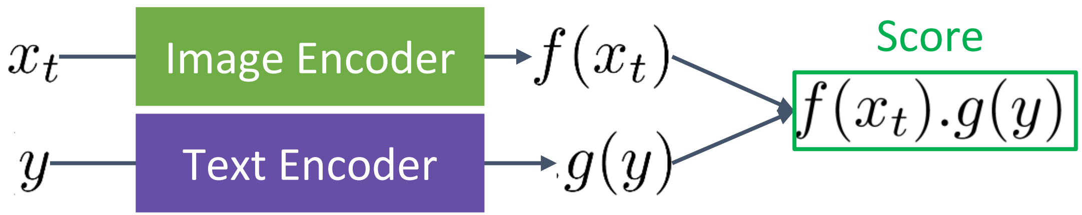


Compute gradient of score by x_t.

Guidance | CLIP-Based Guidance | Label = “robots meditating in a vipassana retreat”

First: Train a CLIP model.

Then in Diffusion:

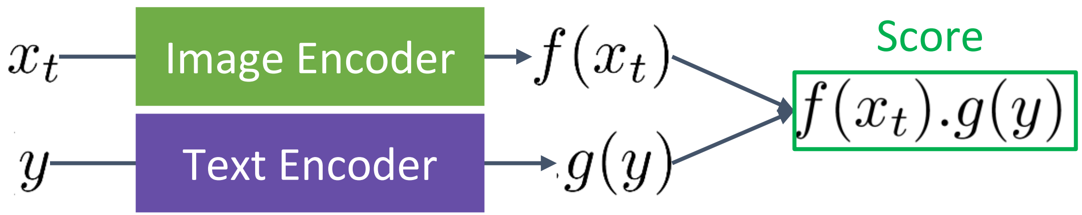


Compute $\nabla_{x_t}(f(x_t).g(y))$

Guidance | CLIP-Based Guidance | Label = “robots meditating in a vipassana retreat”

First: Train a CLIP model.

Then in Diffusion:



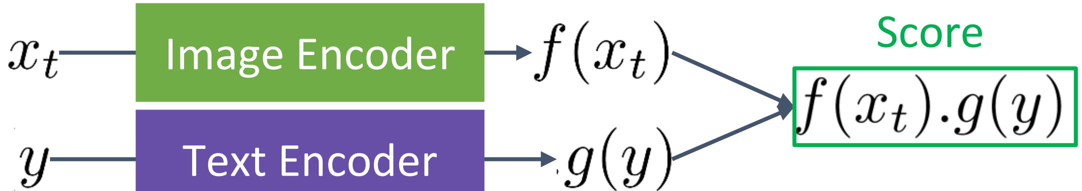
Compute $\nabla_{x_t}(f(x_t).g(y))$

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(\mu_\theta(x_t), \Sigma_\theta(x_t))$$

Guidance | CLIP-Based Guidance | Label = “robots meditating in a vipassana retreat”

First: Train a CLIP model.

Then in Diffusion:



Compute $\nabla_{x_t}(f(x_t).g(y))$

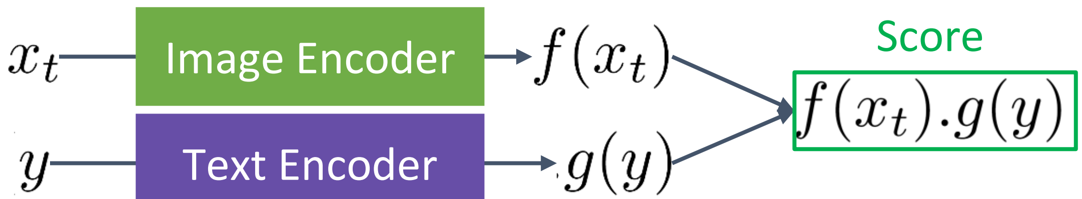
$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(\boxed{\mu_\theta(x_t)}, \Sigma_\theta(x_t))$$

$$\boxed{\hat{\mu}_\theta(x_t|y) = \mu_\theta(x_t|y) + s \cdot \Sigma_\theta(x_t|y)\nabla_{x_t}(f(x_t) \cdot g(y))}$$

Guidance | CLIP-Based Guidance | Label = “robots meditating in a vipassana retreat”

First: Train a CLIP model.

Then in Diffusion:



Compute $\boxed{\nabla_{x_t}(f(x_t).g(y))}$

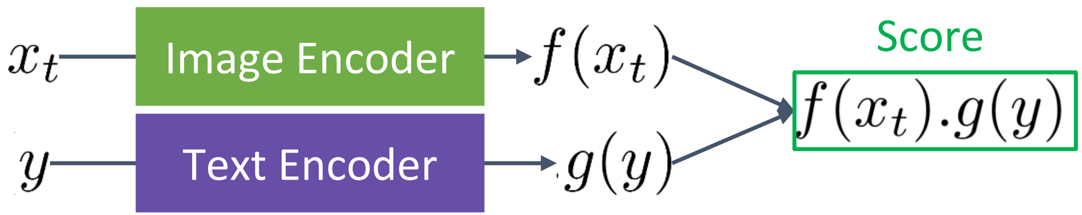
$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(\boxed{\mu_\theta(x_t)}, \Sigma_\theta(x_t))$$

$$\boxed{\hat{\mu}_\theta(x_t|y) = \mu_\theta(x_t|y) + s \cdot \Sigma_\theta(x_t|y)\boxed{\nabla_{x_t}(f(x_t) \cdot g(y))}}$$

Guidance | CLIP-Based Guidance | Label = "robots meditating in a vipassana retreat"

First: Train a CLIP model.

Then in Diffusion:



Compute $\boxed{\nabla_{x_t}(f(x_t).g(y))}$

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(\boxed{\mu_\theta(x_t)}, \Sigma_\theta(x_t))$$

Influence

$$\boxed{\hat{\mu}_\theta(x_t|y) = \mu_\theta(x_t|y) + \boxed{s} \cdot \Sigma_\theta(x_t|y)\boxed{\nabla_{x_t}(f(x_t) \cdot g(y))}}$$

But. The results rely on Pre-Trained (often smaller) Models.

But. The results rely on Pre-Trained (often smaller) Models.

Solution: Classifier-Free Guidance

Guidance |

Guidance | Classifier-Free Guidance | Label = “robots meditating in a vipassana retreat”

Guidance | Classifier-Free Guidance | Label = “robots meditating in a vipassana retreat”

Previously: Train a Guidance Model.

Guidance | Classifier-Free Guidance | Label = “robots meditating in a vipassana retreat”

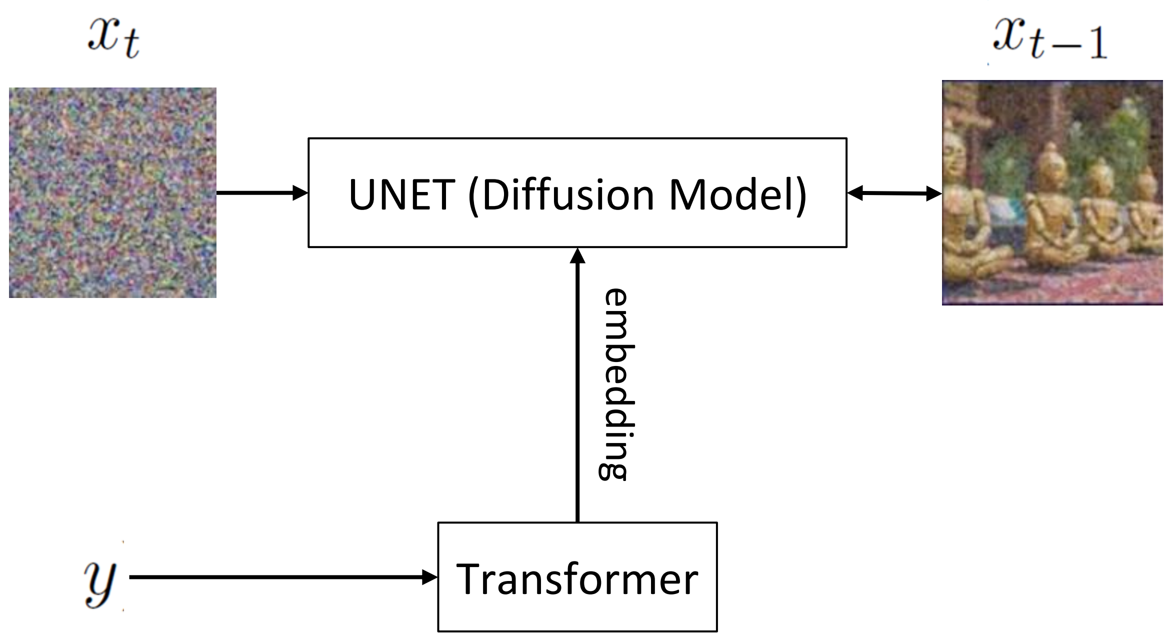
No Separate Guidance Model Needed

Guidance | Classifier-Free Guidance | Label = “robots meditating in a vipassana retreat”

No Separate Guidance Model Needed

Train a Naïve Text Conditional Model.

Text-Conditioned Diffusion



Convert text to discrete tokens & attend to them in UNET

Guidance | Classifier-Free Guidance | Label = “robots meditating in a vipassana retreat”

No Separate Guidance Model Needed

Train a Naïve Text Conditional Model.

Guidance | Classifier-Free Guidance | Label = “robots meditating in a vipassana retreat”

No Separate Guidance Model Needed

Train a Naïve Text Conditional Model.

Use noisy x_t and y to train it.

Guidance | Classifier-Free Guidance | Label = “robots meditating in a vipassana retreat”

No Separate Guidance Model Needed

Train a Naïve Text Conditional Model.

Use noisy x_t and y to train it.

Sometimes don’t pass labels. $y = \emptyset$

Guidance | Classifier-Free Guidance | Label = “robots meditating in a vipassana retreat”

No Separate Guidance Model Needed

Train a Naïve Text Conditional Model.

Use noisy x_t and y to train it.

Sometimes don’t pass labels. $y = \emptyset$

Then at Inference:

Guidance | Classifier-Free Guidance | Label = “robots meditating in a vipassana retreat”

No Separate Guidance Model Needed

Train a Naïve Text Conditional Model.

Use noisy x_t and y to train it.

Sometimes don’t pass labels. $y = \emptyset$

Then at Inference:

Guidance | Classifier-Free Guidance | Label = “robots meditating in a vipassana retreat”

No Separate Guidance Model Needed

Train a Naïve Text Conditional Model.

Use noisy x_t and y to train it.

Sometimes don’t pass labels. $y = \emptyset$

Then at Inference:

Guidance | Classifier-Free Guidance | Label = “robots meditating in a vipassana retreat”

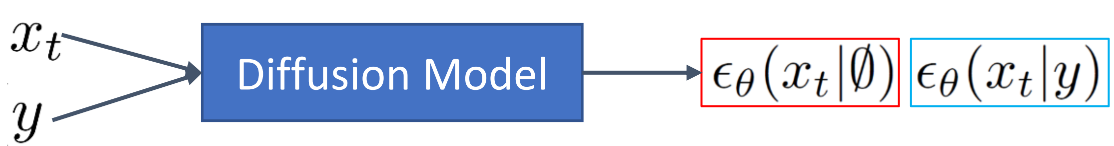
No Separate Guidance Model Needed

Train a Naïve Text Conditional Model.

Use noisy x_t and y to train it.

Sometimes don’t pass labels. $y = \emptyset$

Then at Inference:



Guidance | Classifier-Free Guidance | Label = “robots meditating in a vipassana retreat”

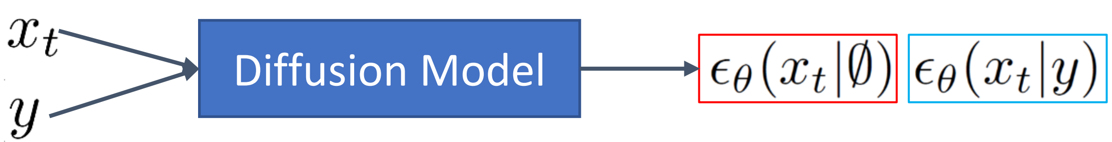
No Separate Guidance Model Needed

Train a Naïve Text Conditional Model.

Use noisy x_t and y to train it.

Sometimes don’t pass labels. $y = \emptyset$

Then at Inference:



Move the prediction from $\boxed{\epsilon_\theta(x_t|\emptyset)}$ to $\boxed{\epsilon_\theta(x_t|y)}$.

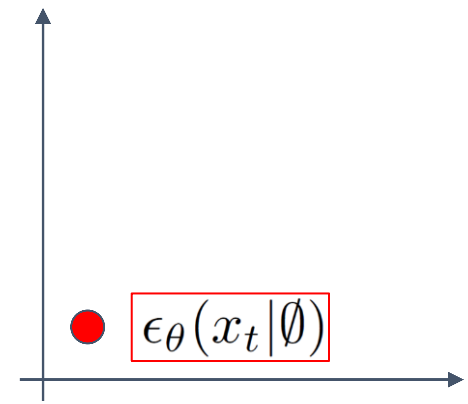
Guidance | Classifier-Free Guidance | Label = “robots meditating in a vipassana retreat”

No Separate Guidance Model Needed

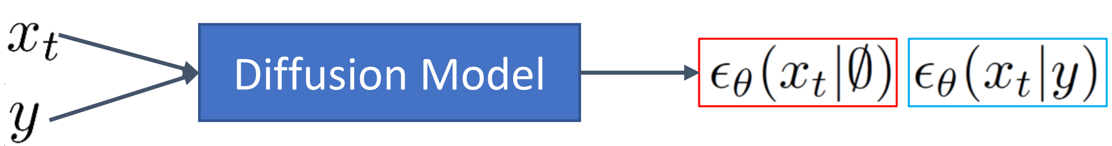
Train a Naïve Text Conditional Model.

Use noisy x_t and y to train it.

Sometimes don’t pass labels. $y = \emptyset$



Then at Inference:



Move the prediction from $\epsilon_\theta(x_t|\emptyset)$ to $\epsilon_\theta(x_t|y)$.

$$\hat{\epsilon}_\theta(x_t|y) = \epsilon_\theta(x_t|\emptyset)$$

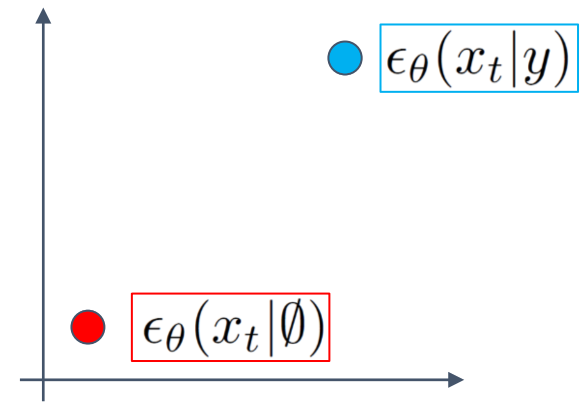
Guidance | Classifier-Free Guidance | Label = "robots meditating in a vipassana retreat"

No Separate Guidance Model Needed

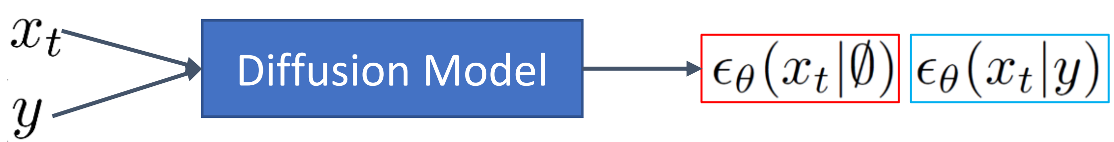
Train a Naïve Text Conditional Model.

Use noisy x_t and y to train it.

Sometimes don't pass labels. $y = \emptyset$



Then at Inference:



Move the prediction from $\boxed{\epsilon_\theta(x_t|\emptyset)}$ to $\boxed{\epsilon_\theta(x_t|y)}$.

$$\hat{\epsilon}_\theta(x_t|y) = \epsilon_\theta(x_t|\emptyset)$$

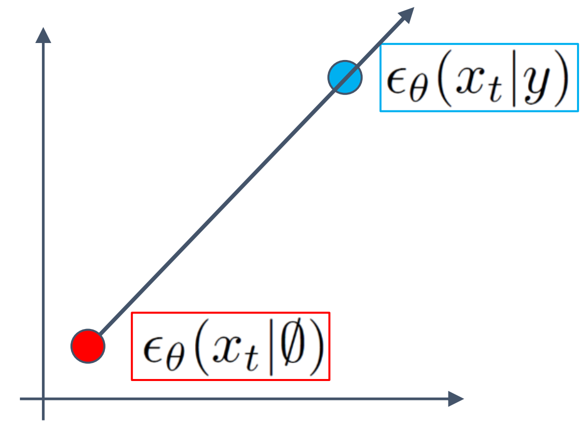
Guidance | Classifier-Free Guidance | Label = “robots meditating in a vipassana retreat”

No Separate Guidance Model Needed

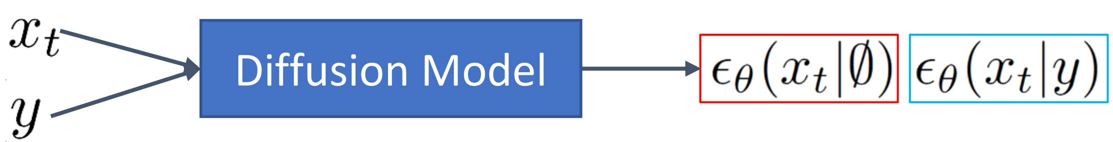
Train a Naïve Text Conditional Model.

Use noisy x_t and y to train it.

Sometimes don’t pass labels. $y = \emptyset$



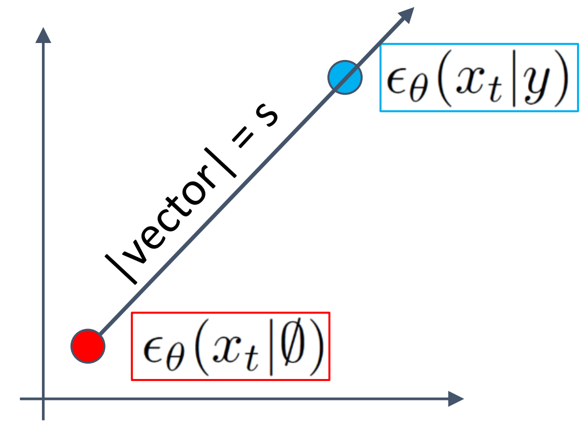
Then at Inference:



Move the prediction from $\boxed{\epsilon_\theta(x_t|\emptyset)}$ to $\boxed{\epsilon_\theta(x_t|y)}$.

$$\hat{\epsilon}_\theta(x_t|y) = \epsilon_\theta(x_t|\emptyset) + \quad (\epsilon_\theta(x_t|y) - \epsilon_\theta(x_t|\emptyset))$$

Guidance | Classifier-Free Guidance | Label = “robots meditating in a vipassana retreat”



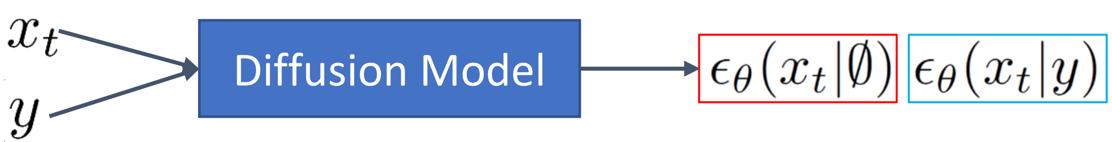
No Separate Guidance Model Needed

Train a Naïve Text Conditional Model.

Use noisy x_t and y to train it.

Sometimes don’t pass labels. $y = \emptyset$

Then at Inference:



Move the prediction from $\epsilon_\theta(x_t|\emptyset)$ to $\epsilon_\theta(x_t|y)$.

$$\hat{\epsilon}_\theta(x_t|y) = \epsilon_\theta(x_t|\emptyset) + s \cdot (\epsilon_\theta(x_t|y) - \epsilon_\theta(x_t|\emptyset))$$

Guidance |

Guidance | Visualizing scale parameter s

Guidance | Visualizing scale parameter s

"a stained glass window of a panda eating bamboo"

Guidance | Visualizing scale parameter s

"a stained glass window of a panda eating bamboo"

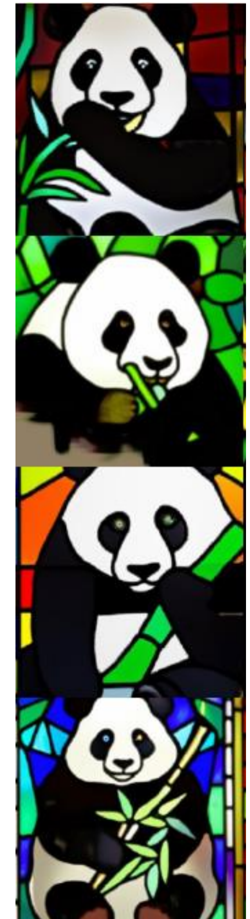
s = 0

Guidance | Visualizing scale parameter s

"a stained glass window of a panda eating bamboo"

$s = 0$

...



$s = 3$

Inception Score

Inception Score

- Named after the Inception classifier model used
- A way to evaluate samples without humans that still correlates well with human evaluation
- To calculate inception score …
    - Inception model is ran on the generated images to get …
        - $p(y|x)$, conditional label distribution (distribution of labels for a given image)
        - $p(y)$, marginal distribution (distribution of labels across all images)
    - Relative entropy is measured between $p(y|x)$ and $p(y)$
- Measures if the images generated are **distinct** and **varied**

Inception Score

- Named after the Inception classifier model used
- A way to evaluate samples without humans that still correlates well with human evaluation
- To calculate inception score …
  - Inception model is ran on the generated images to get …
    - $p(y|x)$, conditional label distribution (distribution of labels for a given image)
    - $p(y)$, marginal distribution (distribution of labels across all images)
  - Relative entropy is measured between $p(y|x)$ and $p(y)$
- Measures if the images generated are **distinct** and **varied**

Inception Score (cont.)

Inception Score Function

$$\exp(\mathbb{E}_{\boldsymbol{x}} \mathrm{KL}(p(y|\boldsymbol{x}) || p(y)))$$

Where:

- E_x is the expected value
- KL is relative entropy
- p(y|x) and (y) are values gotten from the inception model

Frechet Inception Distance / FID Score

- Drawback to IS: does not compare to real world samples in its calculation
- FID was created to address this drawback
- To calculate the FID score…
  - Runs Inception model on real life images and fake images
  - Then difference in the two resulting gaussians is taken, giving us our FID score

FID Score (cont.)

FID Score Function

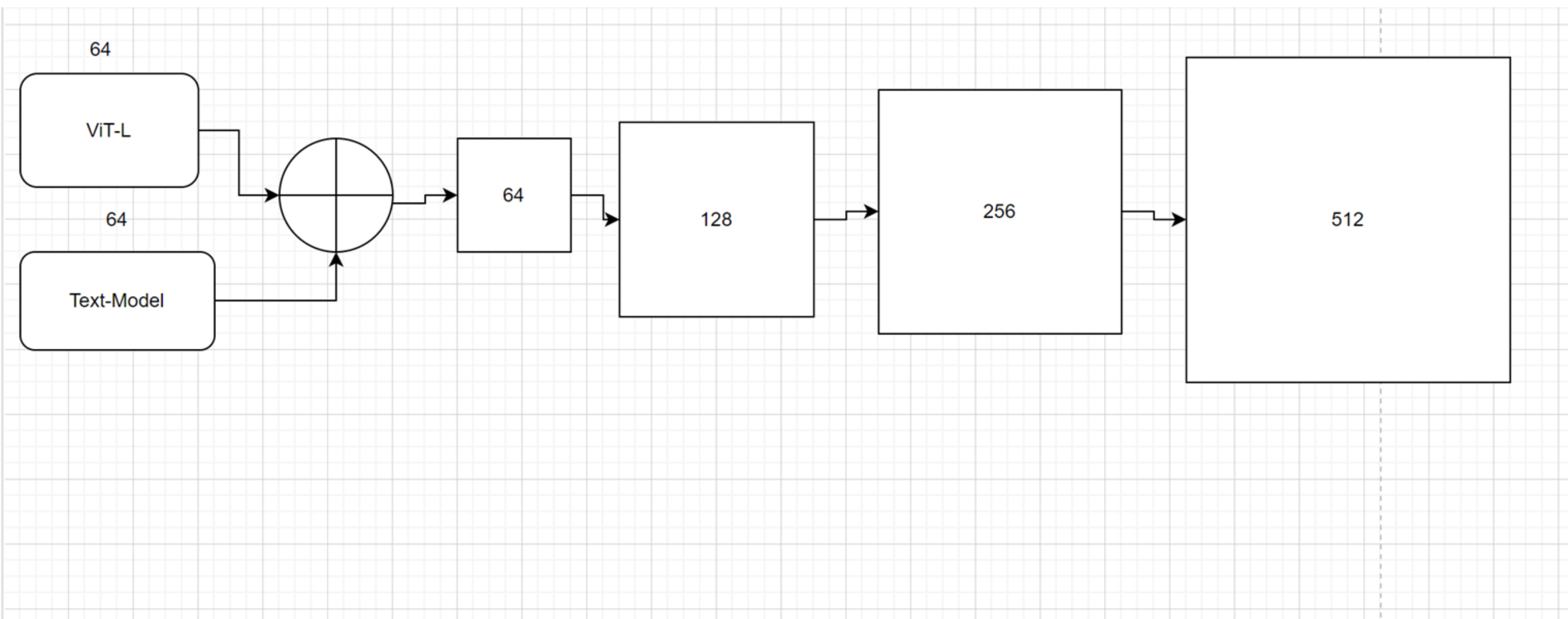
$$d^2((\boldsymbol{m}, \boldsymbol{C}), (\boldsymbol{m}_w, \boldsymbol{C}_w)) = \|\boldsymbol{m} - \boldsymbol{m}_w\|_2^2 + \mathrm{Tr}\big(\boldsymbol{C} + \boldsymbol{C}_w - 2\big(\boldsymbol{C}\boldsymbol{C}_w\big)^{1/2}\big)$$

Where:

- (m, C) is the normal distribution from running Inception on real life images
  - m and C representing it’s mean and Covariance vectors, respectively
- (m_w, C_w) the distribution from Inception on the generated images
- Tr is the trace matrix operation

Setup:

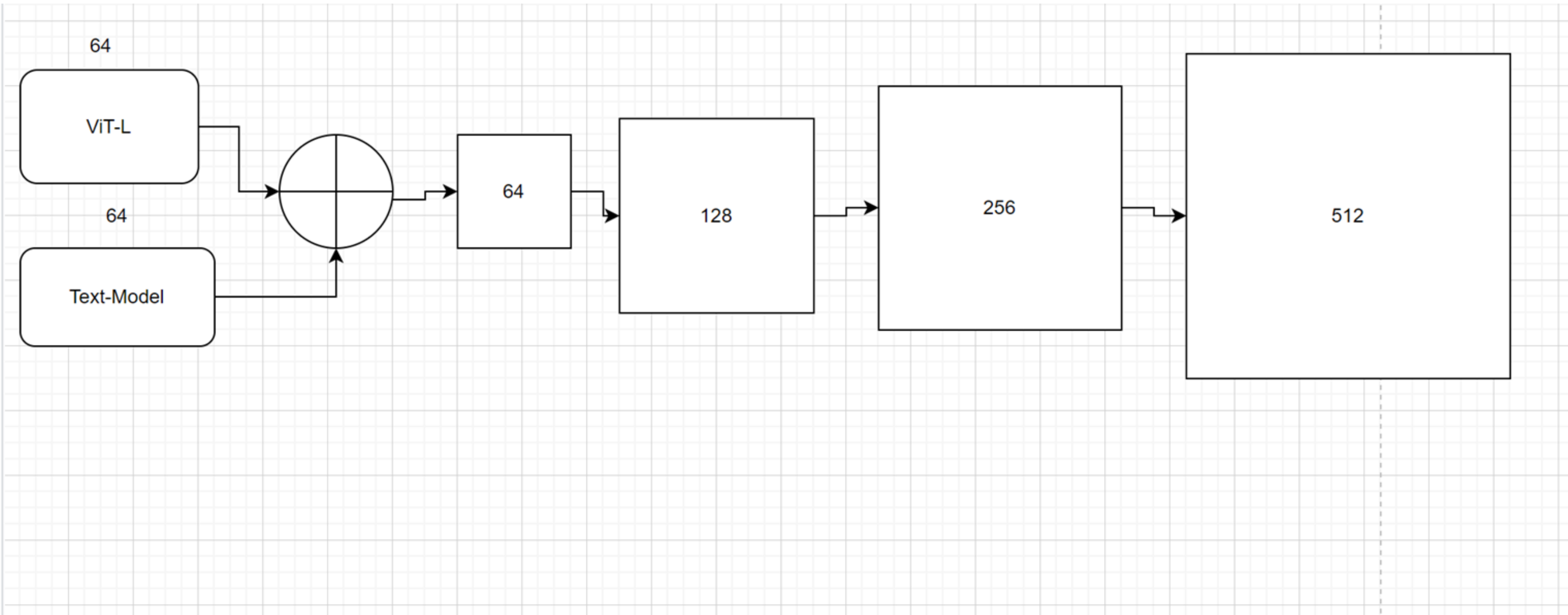
Setup:



Setup:

Dataset: MS COCO images, textual prompts

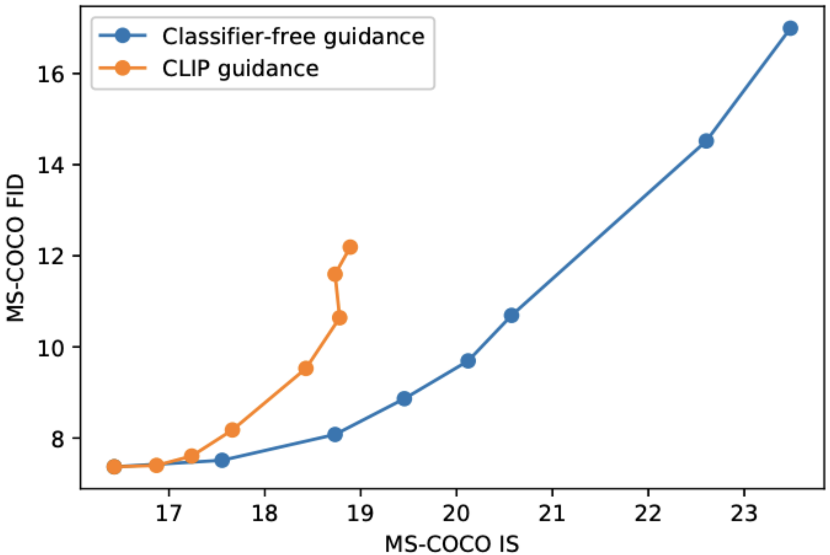
Batch size: VITL/Text model, 2048. Upsampling block: 2048/4 = 512



Evaluation: FID vs Inception score

Zero-Shot FID is calculated from samples observed from classes which were not observed during training

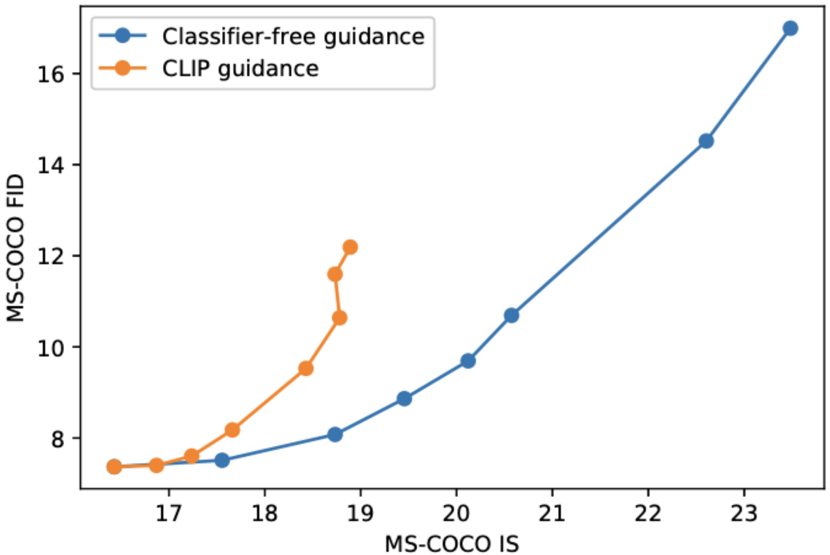
Evaluation: FID vs Inception score



(b) IS/FID

Zero-Shot FID is calculated from samples observed from classes which were not observed during training

Evaluation: FID vs Inception score

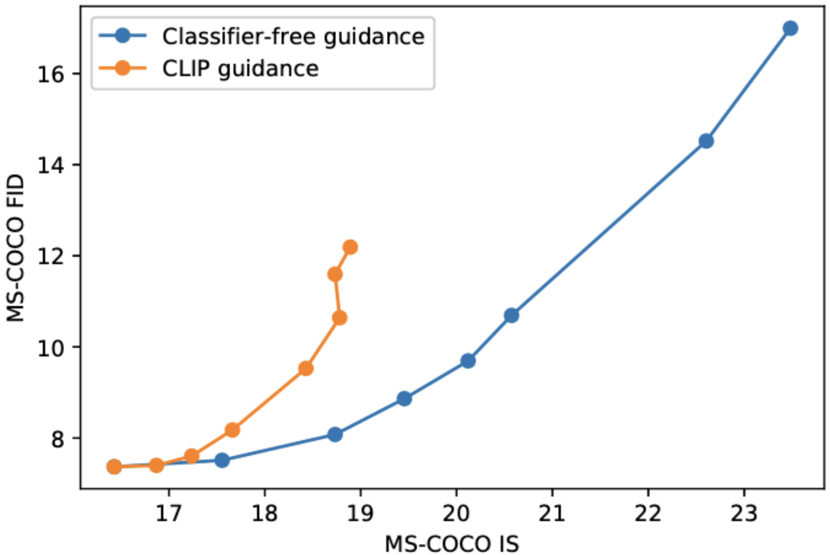


(b) IS/FID

- Best method: FID: low, IS: High

Zero-Shot FID is calculated from samples observed from classes which were not observed during training

Evaluation: FID vs Inception score



(b) IS/FID

Model	FID	Zero-shot FID
AttnGAN (Xu et al., 2017)	35.49	
DM-GAN (Zhu et al., 2019)	32.64	
DF-GAN (Tao et al., 2020)	21.42	
DM-GAN + CL (Ye et al., 2021)	20.79	
XMC-GAN (Zhang et al., 2021)	9.33	
LAFITE (Zhou et al., 2021)	**8.12**	
DALL-E (Ramesh et al., 2021)		~ 28
LAFITE (Zhou et al., 2021)		26.94
GLIDE		**12.24**
GLIDE (Validation filtered)		**12.89**

- Best method: FID: low, IS: High

Zero-Shot FID is calculated from samples observed from classes which were not observed during training

ELO SCORES.

$$L_{\text{elo}} := -\sum_{i,j} A_{ij} \cdot \log\left(\frac{1}{1 + 10^{(\sigma_i - \sigma_j)/400}}\right)$$

ELO SCORES.

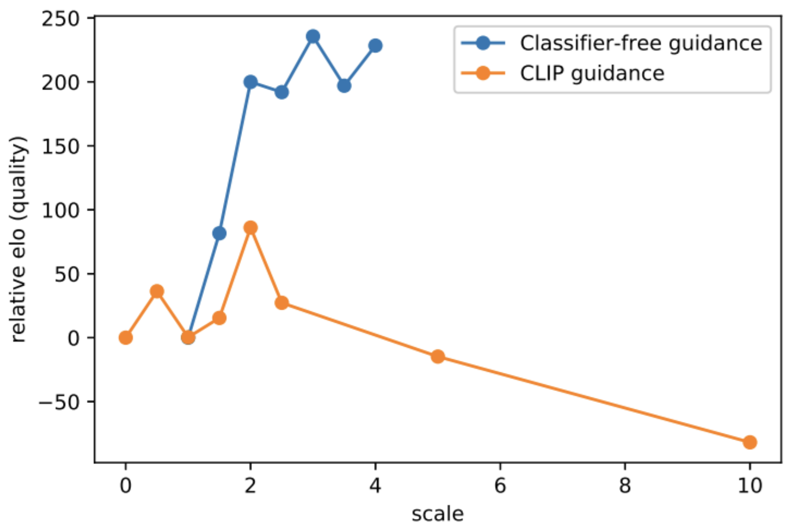
Elo scores are computed by minimizing the objective:

$$L_{\text{elo}} := -\sum_{i,j} A_{ij} \cdot \log\left(\frac{1}{1 + 10^{(\sigma_i - \sigma_j)/400}}\right)$$

Elo scores vs guidance scale.

Elo Score is a metric that measures the relative performance in zero shot learning

Elo scores vs guidance scale.

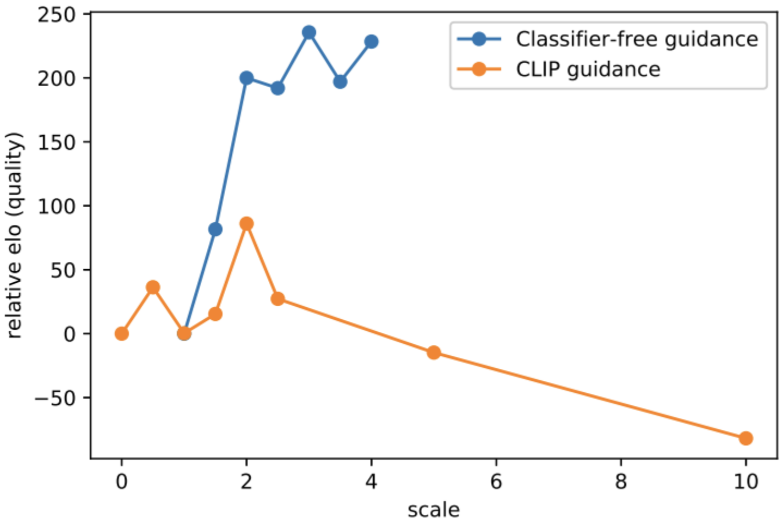


(a) Photorealism

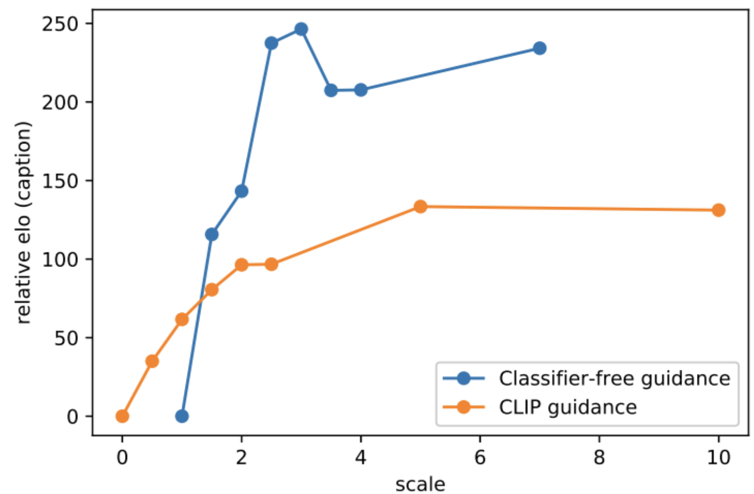
Elo Score is a metric that measures the relative performance in zero shot learning

Elo scores vs guidance scale.

(a) Photorealism



(a) Photorealism



(b) Caption Similarity

Elo Score is a metric that measures the relative performance in zero shot learning

Elo scores vs guidance scale.

Elo scores vs guidance scale.

Guidance	Photorealism	Caption
Unguided	-88.6	-106.2
CLIP guidance	-73.2	29.3
Classifier-free guidance	**82.7**	**110.9**

Table 3. Human evaluation results comparing GLIDE to DALL-E. We report win probabilities of our model for both photorealism and caption similarity. In the final row, we apply the dVAE used by DALL-E to the outputs of GLIDE.

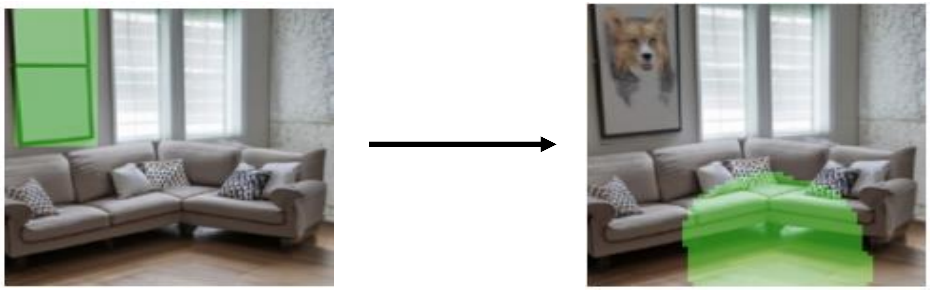
	DALL-E Temp.	Photo-realism	Caption Similarity
No reranking	1.0	91%	83%
	0.85	84%	80%
DALL-E reranked	1.0	89%	71%
	0.85	87%	69%
DALL-E reranked + GLIDE blurred	1.0	72%	63%
	0.85	66%	61%

Image Inpainting

Image Inpainting

a painting of a dog
on the wall

Image Inpainting



a painting of a dog
on the wall

Image Inpainting

a painting of a dog
on the wall

“a round coffee table
in front of a couch”

Image Inpainting

a painting of a dog
on the wall

“a round coffee table
in front of a couch”

Image Inpainting

a painting of a dog
on the wall

“a round coffee table
in front of a couch”

“a vase of flowers on
a coffee table”

Image Inpainting

a painting of a dog
on the wall

“a round coffee table
in front of a couch”

“a vase of flowers on
a coffee table”

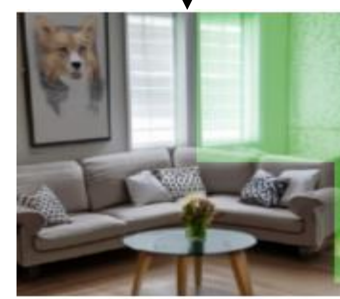
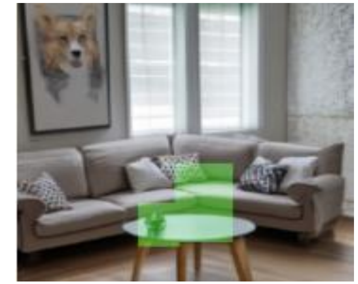


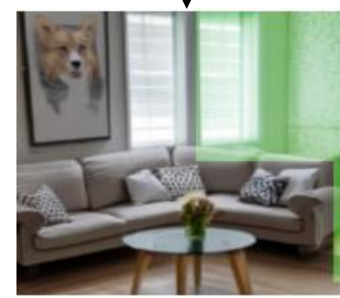
Image Inpainting

a painting of a dog on the wall

“a round coffee table in front of a couch”



“a vase of flowers on a coffee table”



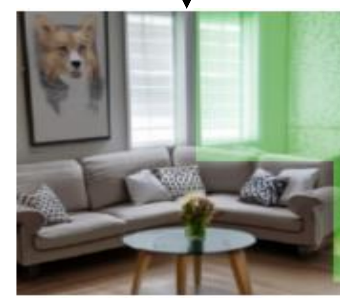
“a couch in the corner of a room”

Image Inpainting

a painting of a dog
on the wall

“a round coffee table
in front of a couch”

“a vase of flowers on
a coffee table”



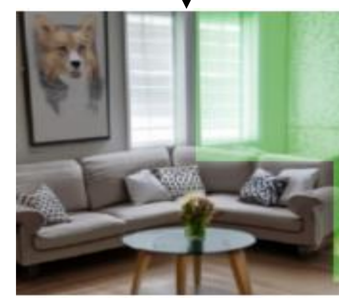
“a couch in the
corner of a room”

Image Inpainting

a painting of a dog
on the wall

“a round coffee table
in front of a couch”

“a vase of flowers on
a coffee table”



“a couch in the
corner of a room”

- Input: Image + Mask + Guiding Text
- Output: New Image
- Process repeated at each time step, by <u>progressively adding</u> new elements to the scene.

Earlier: Song et Al , 2020

Earlier: Song et Al , 2020

“a couch in the corner of a room”

Earlier: Song et Al , 2020

“a couch in the corner of a room”

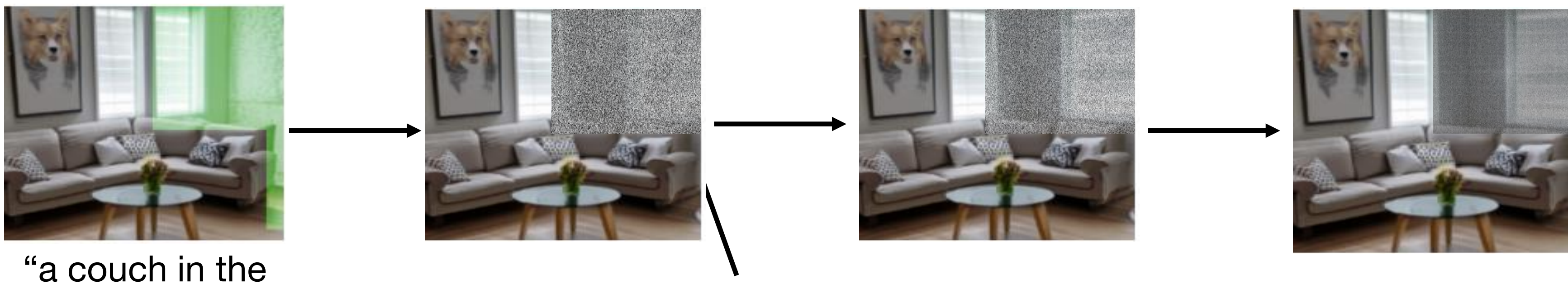
Gaussian Noise

Earlier: Song et Al , 2020

“a couch in the corner of a room”

Gaussian Noise

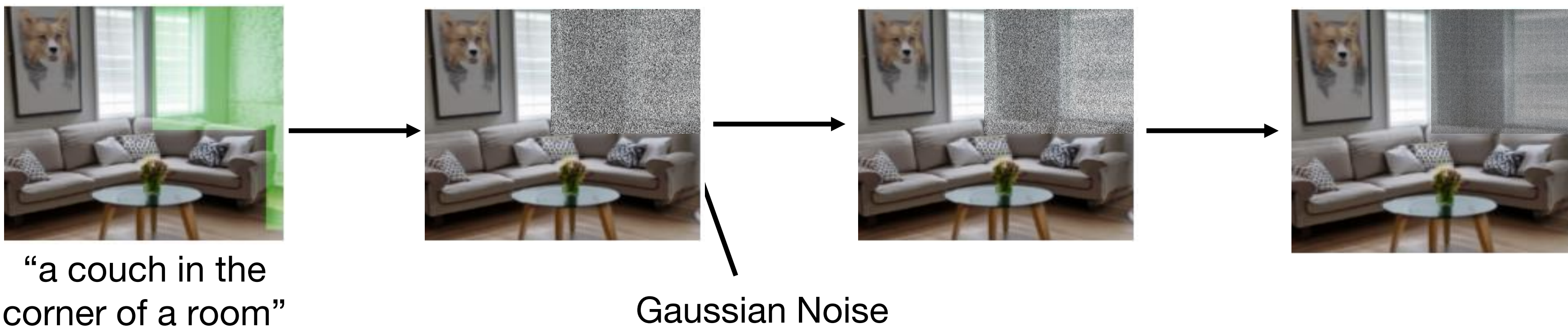
Earlier: Song et Al , 2020



“a couch in the corner of a room”

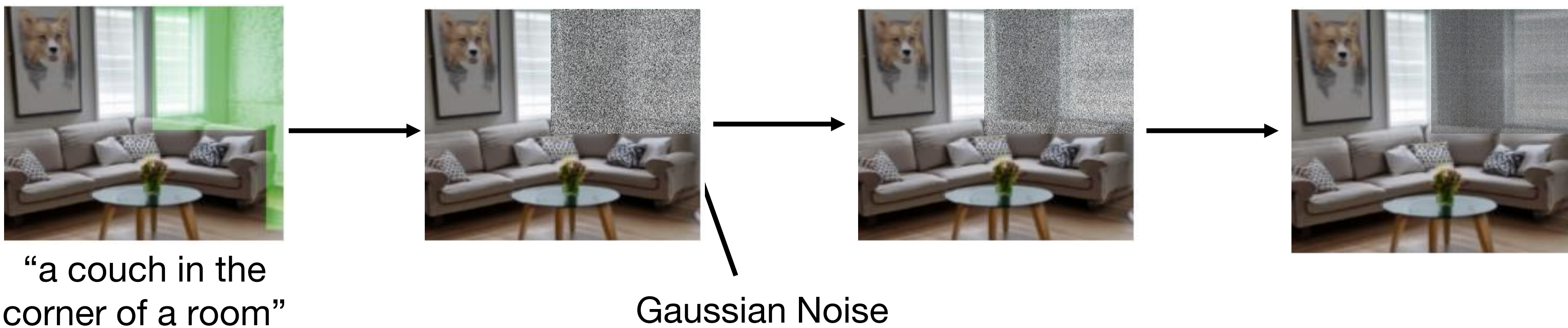
Gaussian Noise

Earlier: Song et Al , 2020



- Adding gaussian noise at mask regions.
- Leads to checkerboard artifacts.
- Network never looks at surrounding context during training

Earlier: Song et Al , 2020

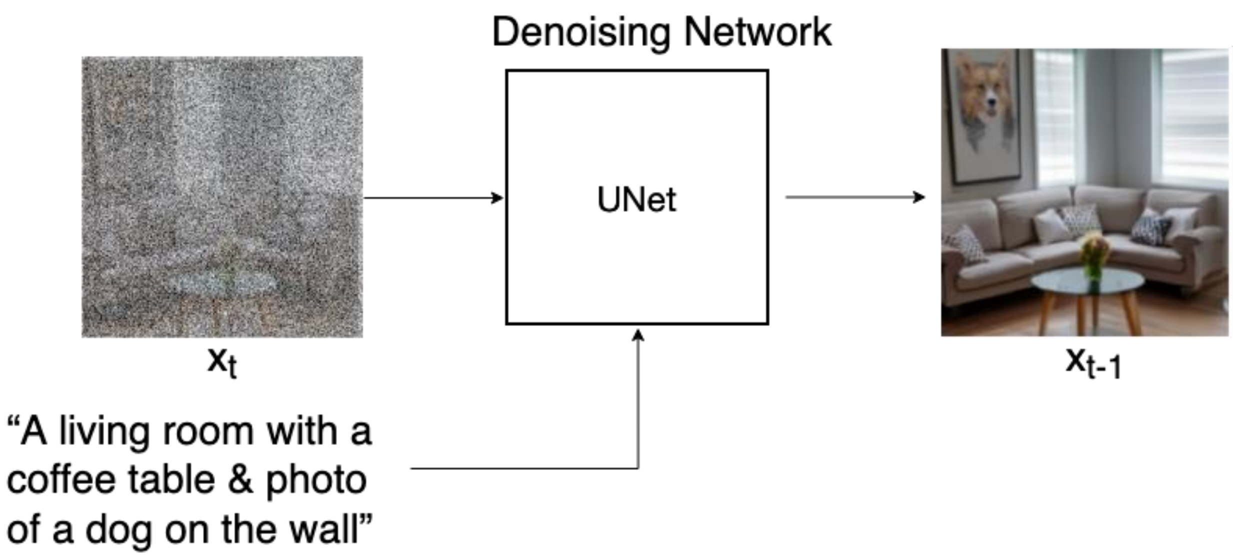


- Adding gaussian noise at mask regions.
- Leads to checkerboard artifacts.
- Network never looks at surrounding context during training

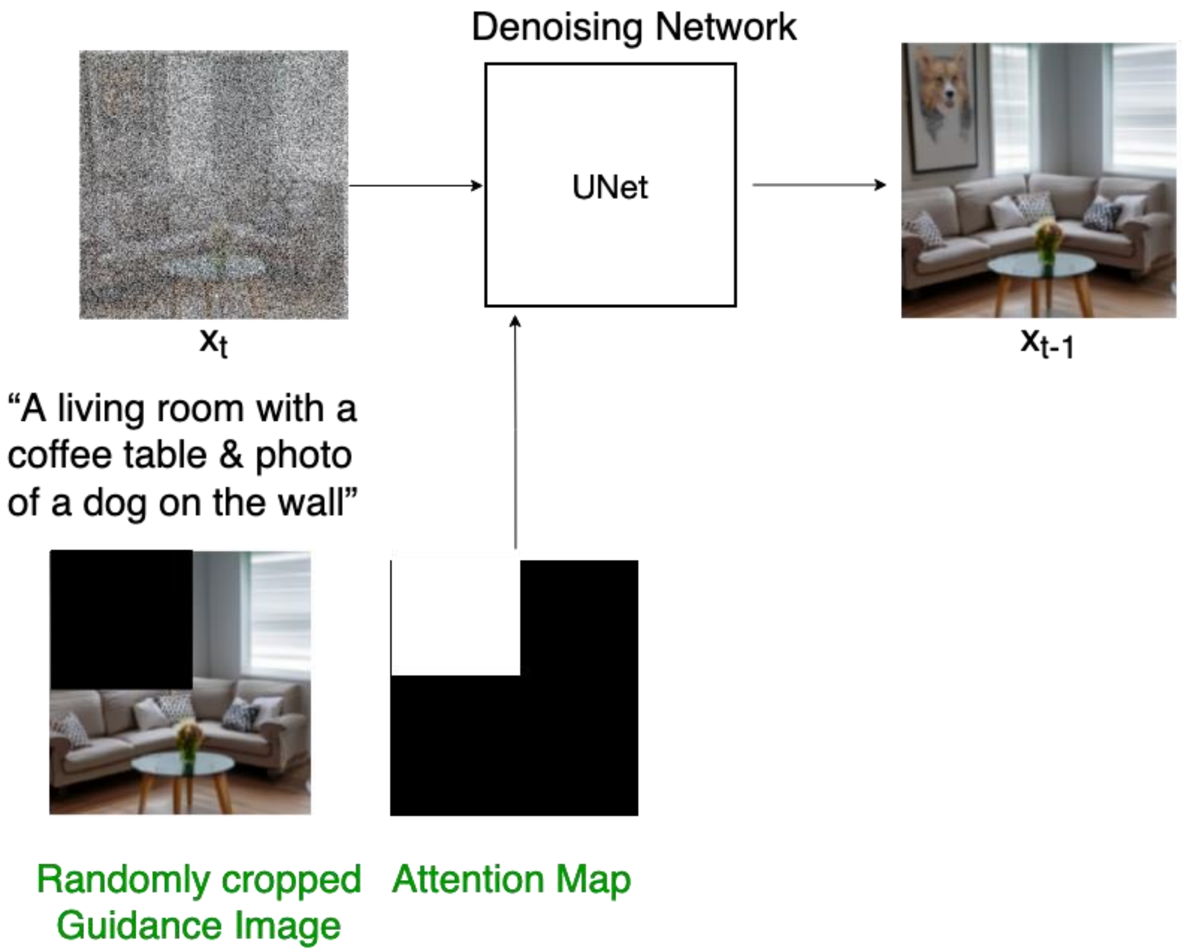
Text guided Image Generation

Naive GLIDE model:

Reverse Diffusion Process

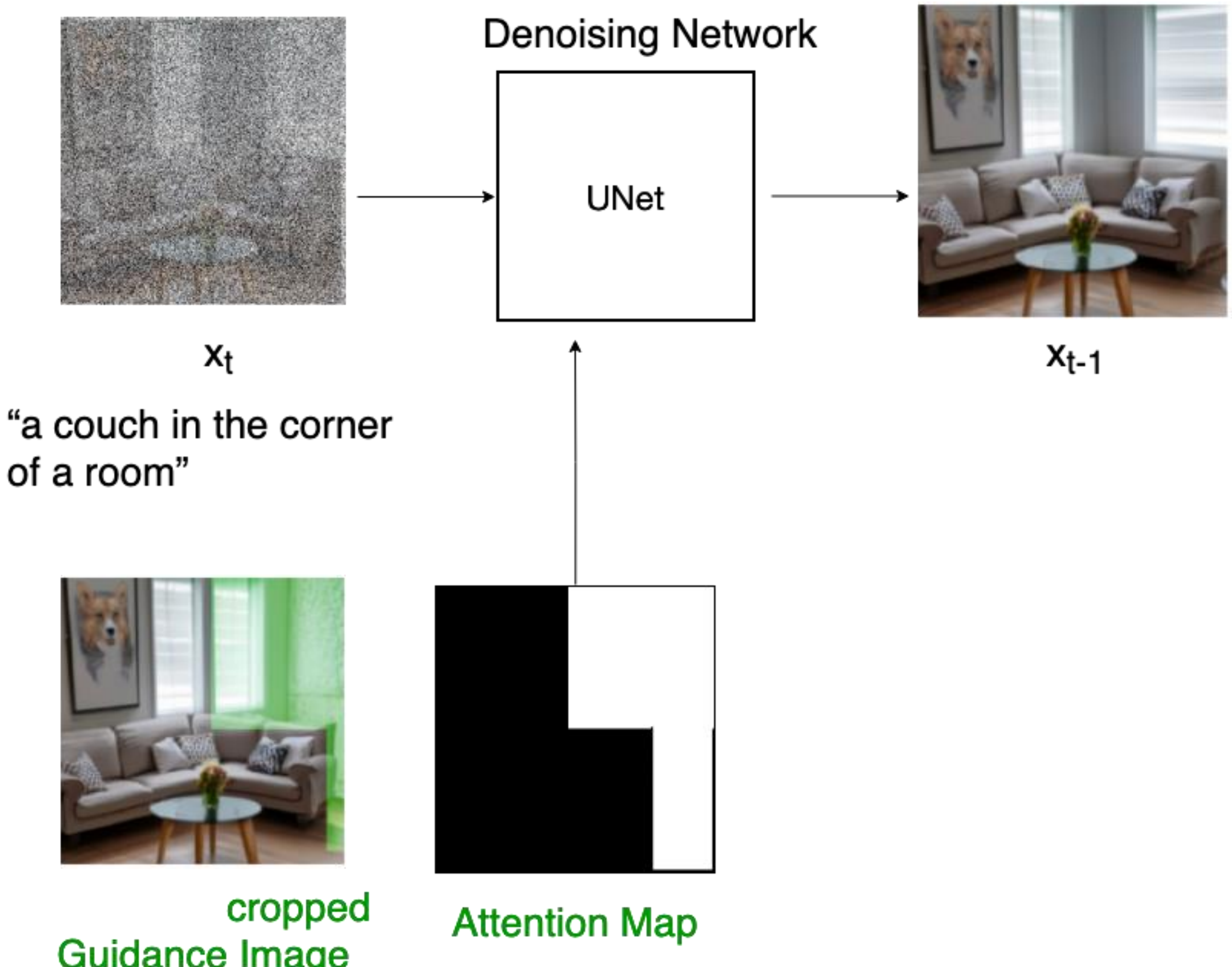


Training: Image Inpainting



- <u>Force the network to learn global context.</u>

Inference: Image Inpainting



Conclusion & Future Work:

Conclusion & Future Work:

- Classifier Free is better than CLIP guidance.
    - Interestingly, even though CLIP trained on <Image, text pairs>.
- Controlling scale adjust tradeoff b/w photorealism and diversity
    - Better than Gans: only photorealism, no diversity.
- Diffusion is iterative:
    - Scene editing requires careful prompts at regular intervals of generation.
    - Specify full scene semantics earlier & “learn” when to apply?

Conclusion & Future Work:

- Classifier Free is better than CLIP guidance.
  - Interestingly, even though CLIP trained on <Image, text pairs>.
- Controlling scale adjust tradeoff b/w photorealism and diversity
  - Better than Gans: only photorealism, no diversity.
- Diffusion is iterative:
  - Scene editing requires careful prompts at regular intervals of generation.
  - Specify full scene semantics earlier & “learn” when to apply?

References

- Nichol, Alex, et al. "Glide: Towards photorealistic image generation and editing with text-guided diffusion models." arXiv preprint arXiv:2112.10741 (2021).
- Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. Advances in Neural Information Processing Systems 30 (NIPS 2017), 2017.
- Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. Improved techniques for training gans. arXiv:1606.03498, 2016.