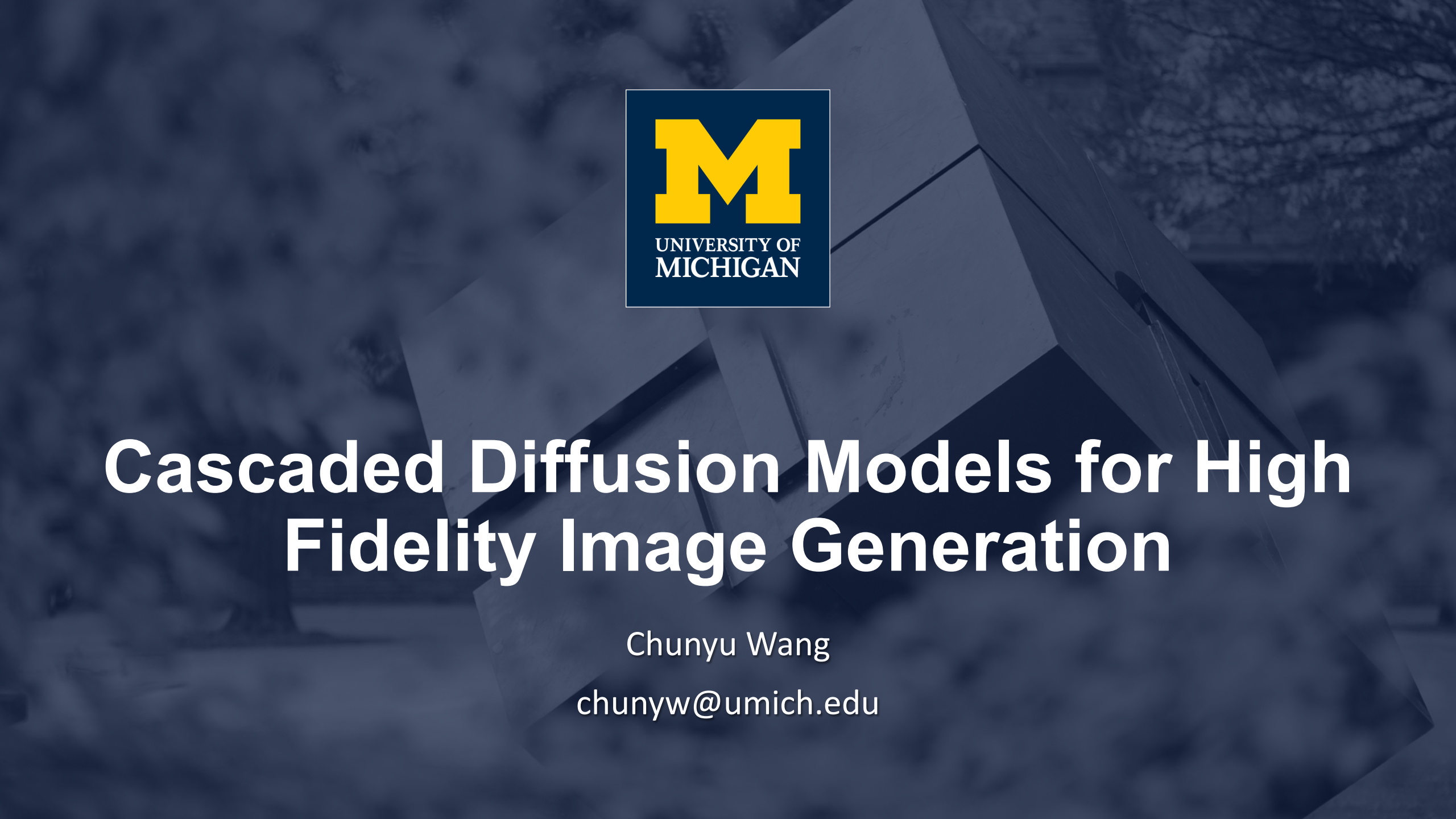


Cascaded Diffusion Models for High Fidelity Image Generation

Chunyu Wang

chunyw@umich.edu

Journal of Machine Learning Research 23 (2022) 1-33 Submitted 6/21; Revised 12/21; Published 1/22

Cascaded Diffusion Models for High Fidelity Image Generation

Jonathan Ho* JONATHANHO@GOOGLE.COM
Chitwan Saharia* SAHARIAC@GOOGLE.COM
William Chan WILLIAMCHAN@GOOGLE.COM
David J. Fleet DAVIDFLEET@GOOGLE.COM
Mohammad Norouzi MNOROUZI@GOOGLE.COM
Tim Salimans SALIMANS@GOOGLE.COM

Outline

- Introduction
- Background
- Methodology
- Experiments & Results
- Related Work
- Conclusion

Outline

- **Introduction**
- Background
- Methodology
- Experiments & Results
- Related Work
- Conclusion

Introduction

Diffusion Models have been shown to be capable of synthesizing high quality images (high-fidelity, sample diversity, stable training)

What is diffusion model?

Diffusion models:
Gradually add Gaussian noise and then reverse

Introduction

Motive?

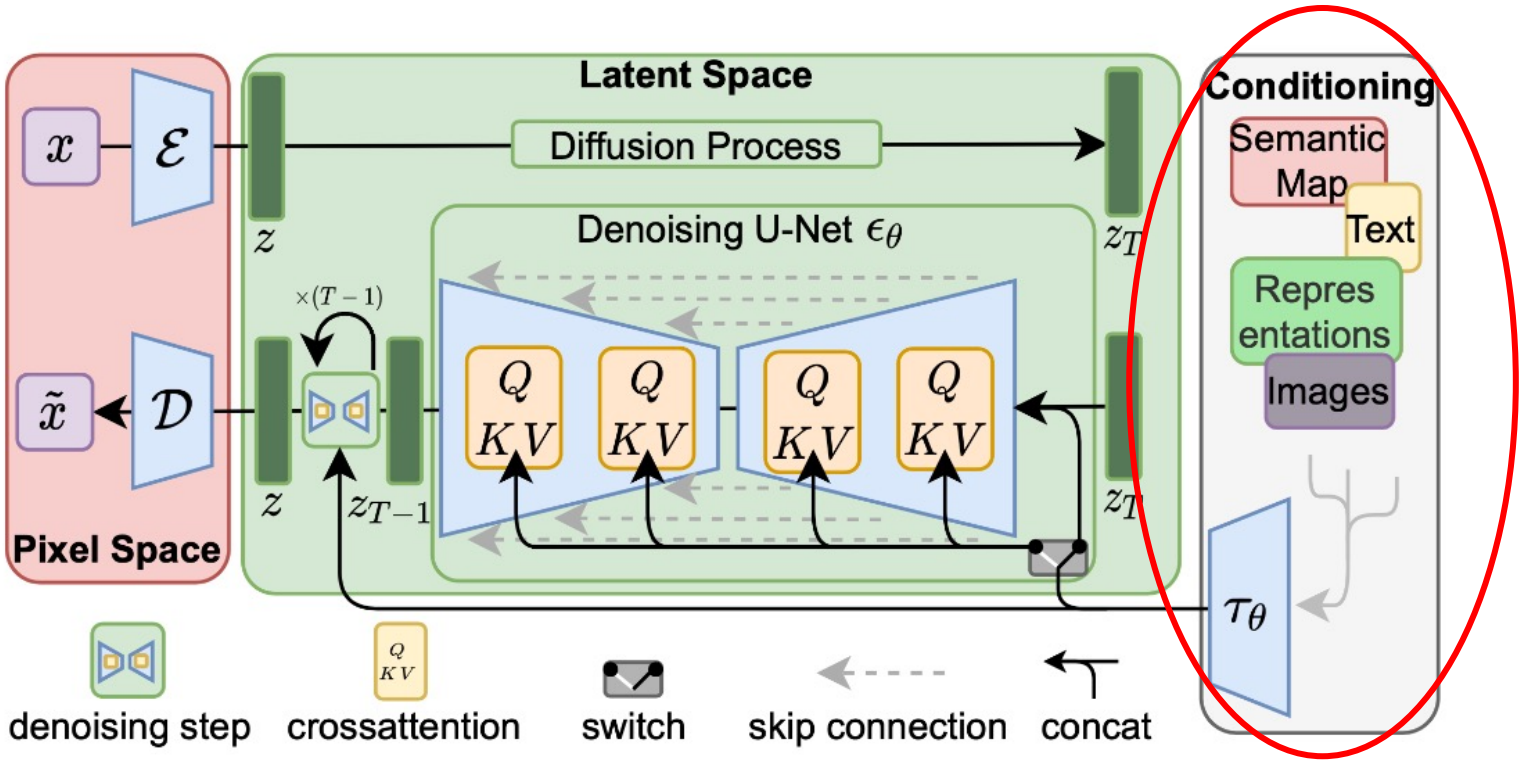
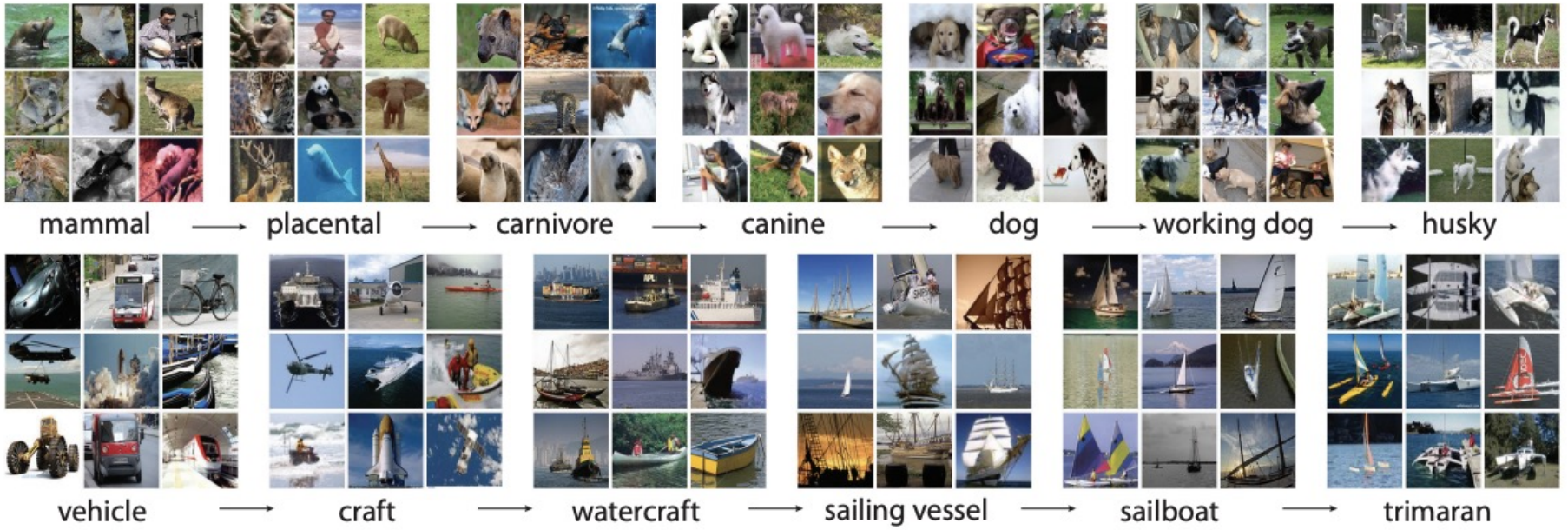


Fig. 9. The architecture of latent diffusion model. (Image source: Rombach & Blattmann, et al. 2022)

Introduction

Goal:

improve the sample quality of diffusion models on large high-fidelity data sets with no strong conditioning information



Introduction

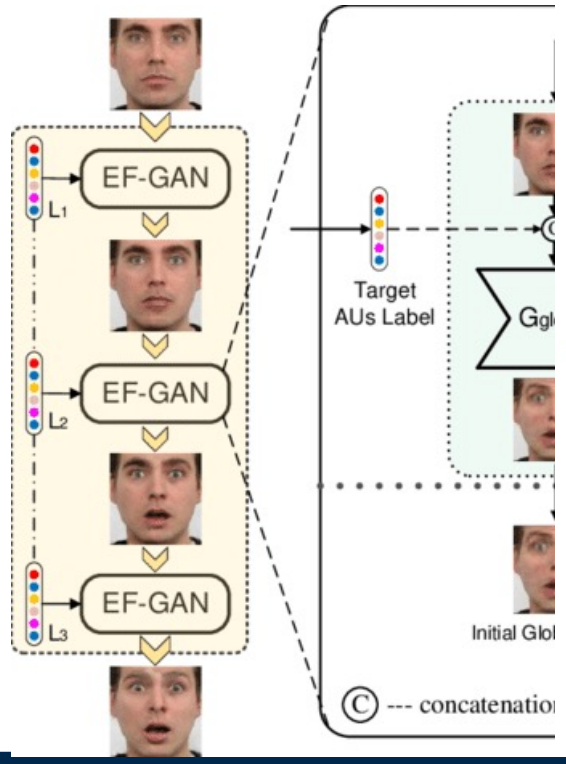
Key method:

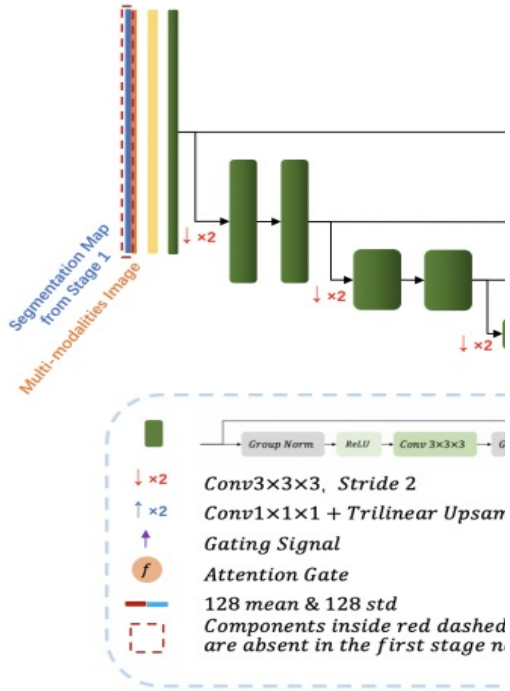
- Cascade DMs (CDM) -- to improve the sample quality of diffusion models on class-conditional ImageNet
  - Base model generates low resolution samples (16x16)
  - Super-resolution models upsample into high resolution samples (16x16→64x64, 128x128, 256x256)

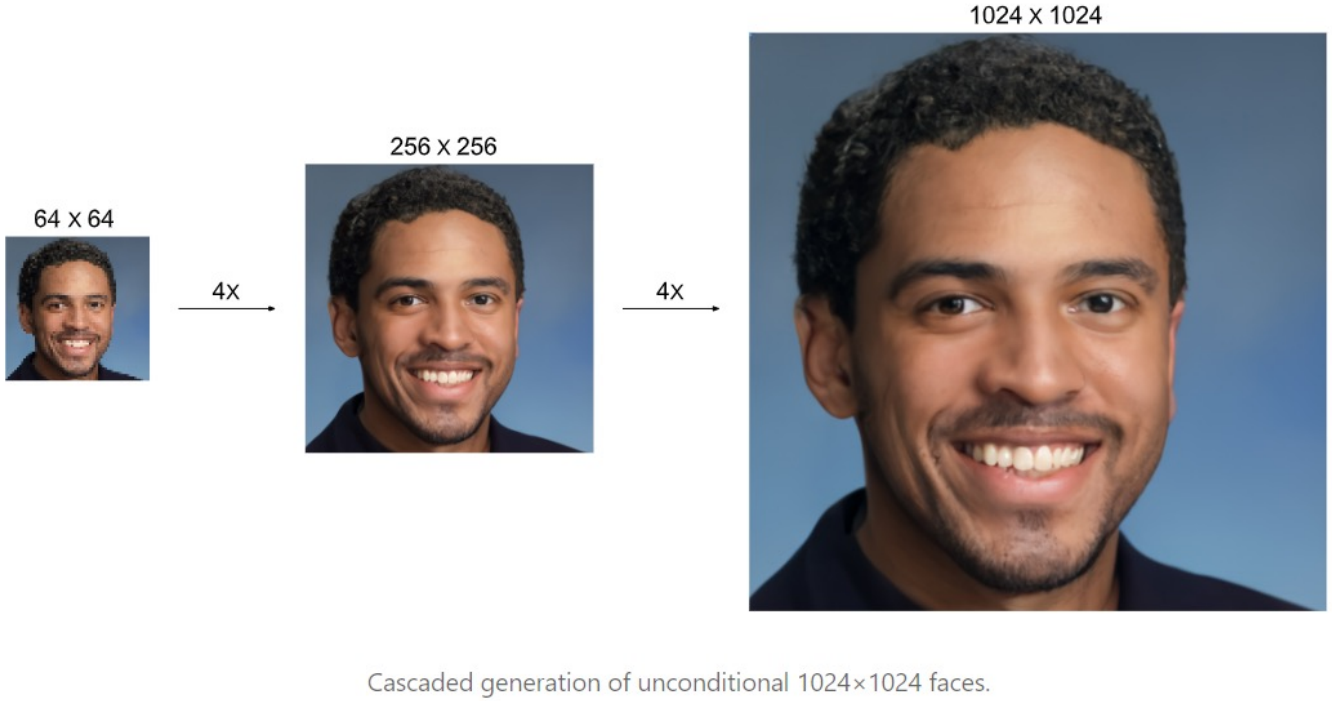
Introduction

Cascading is a technique used in many generative models previously:

Cascade GANs, cascade VAEs, cascade DMs







Cascaded generation of unconditional 1024×1024 faces.

Introduction

This paper concerns the improvement of diffusion cascading pipelines to attain the best possible sample quality

Introduction

Main contribution

1. CDM yield high fidelity samples superior to BigGAN-deep (Brock et al., 2019) and VQ-VAE-2 (Razavi et al., 2019) in terms of FID score

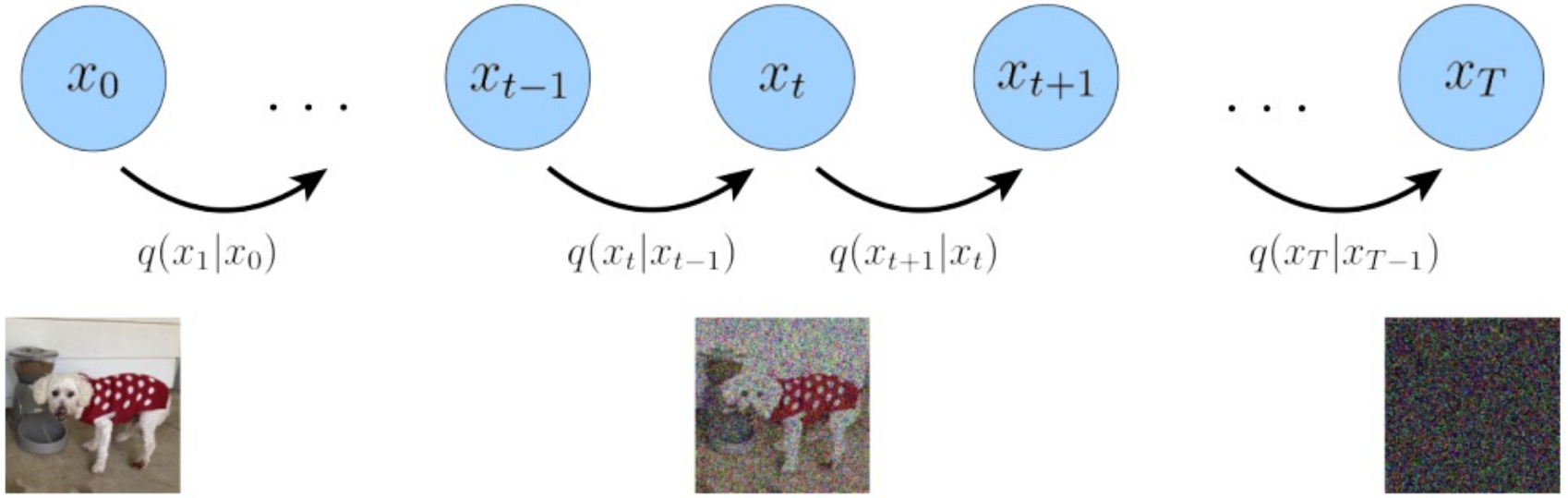
2. Introduces conditioning augmentation as the simplest and the most effective technique to achieve high sample fidelity

Outline

- Introduction
- **Background**
- Methodology
- Experiments & Results
- Related Work
- Conclusion

Background

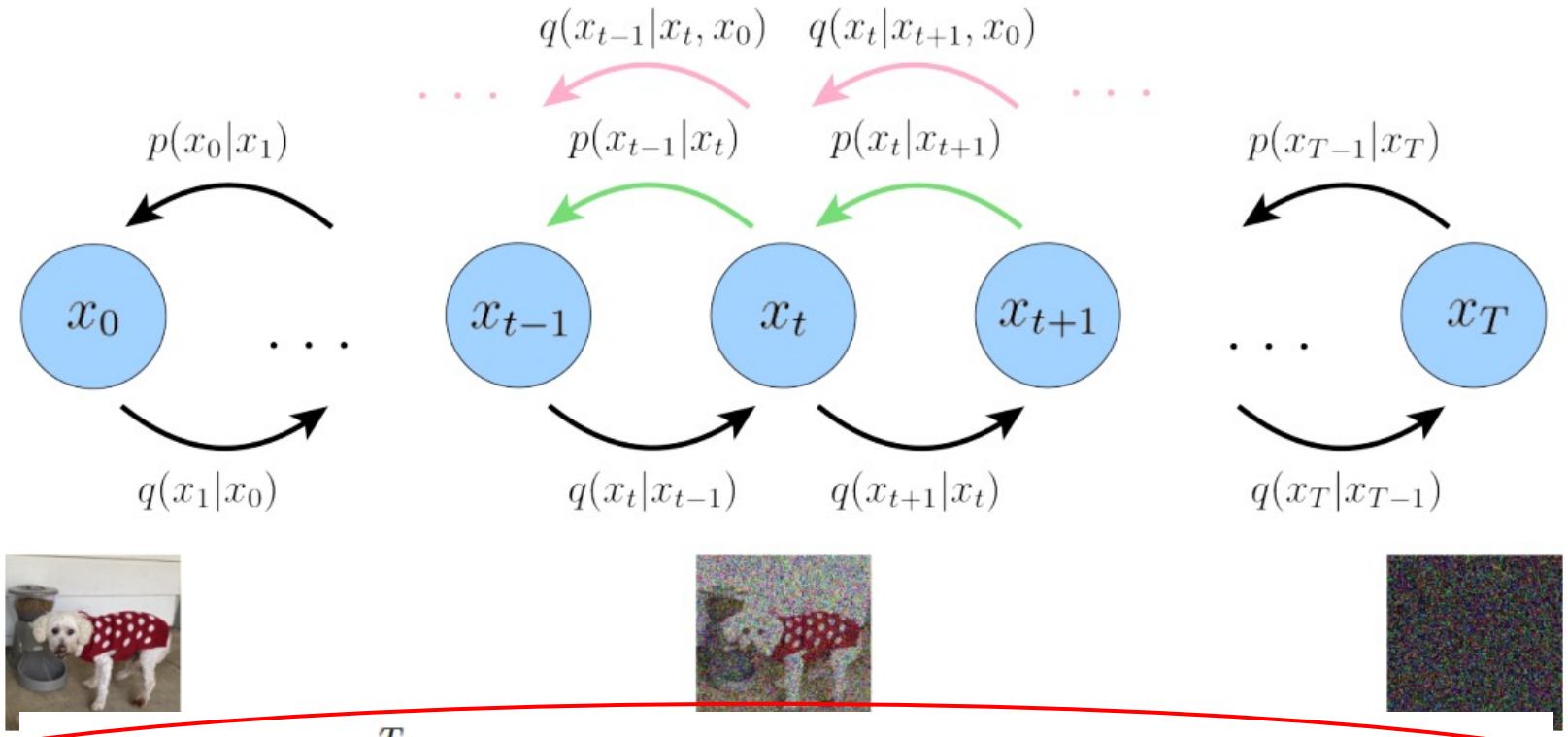
Diffusion Models



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

Background

Diffusion Models



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

Background

Diffusion Models

ELBO $-L_\theta(\mathbf{x}_0) \leq \log p_\theta(\mathbf{x}_0)$

$$L_\theta(\mathbf{x}_0) = \mathbb{E}_q \left[L_T(\mathbf{x}_0) + \sum_{t>1} D_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)) - \log p_\theta(\mathbf{x}_0|\mathbf{x}_1) \right]$$

Background

Diffusion Models

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)).$$

ELBO

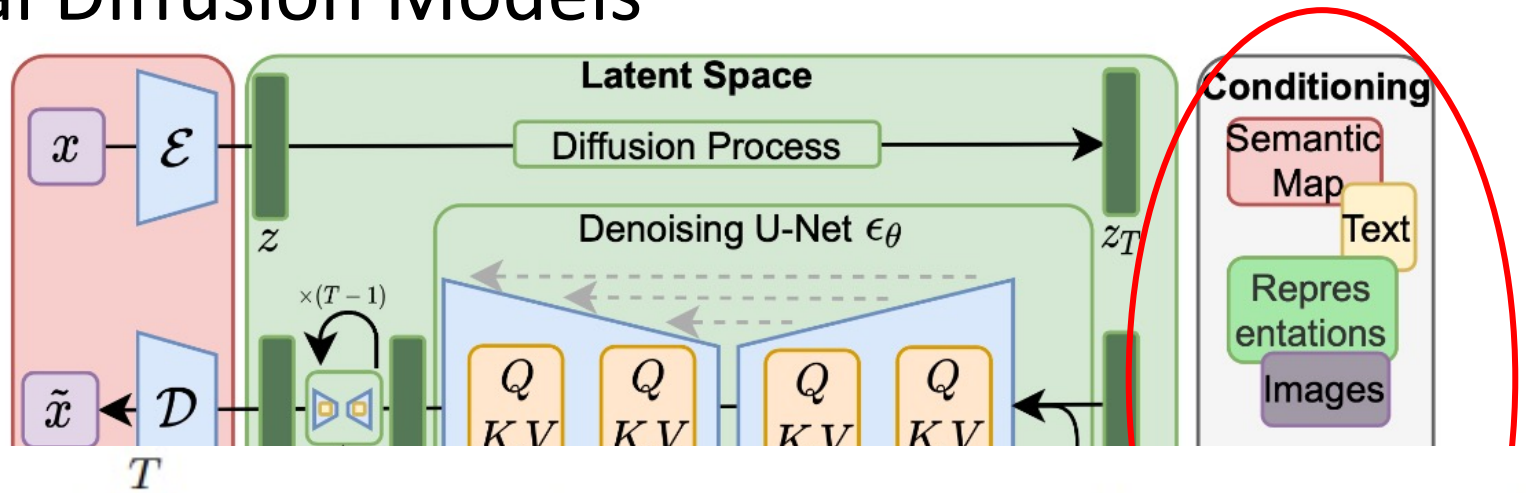
$$L_\theta(\mathbf{x}_0) = \mathbb{E}_q \left[L_T(\mathbf{x}_0) + \sum_{t>1} D_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)) - \log p_\theta(\mathbf{x}_0|\mathbf{x}_1) \right]$$

$$L_{\text{simple}}(\theta) = \mathbb{E}_{\mathbf{x}_0, \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), t \sim \mathcal{U}(\{1, \ldots, T\})} \left[\left\| \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t}\boldsymbol{\epsilon}, t) - \boldsymbol{\epsilon} \right\|^2 \right]$$

$$L_{\text{hybrid}}(\theta) = L_{\text{simple}}(\theta) + \lambda L_{\text{vb}}(\theta) \quad L_{\text{vb}} = \mathbb{E}_{\mathbf{x}_0}[L_\theta(\mathbf{x}_0)]$$

Background

Conditional Diffusion Models



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

$$L_\theta(\mathbf{x}_0|\mathbf{c}) = \mathbb{E}_q\left[L_T(\mathbf{x}_0) + \sum_{t>1} D_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)\parallel p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{c})) - \log p_\theta(\mathbf{x}_0|\mathbf{x}_1,\mathbf{c})\right].$$

Background

Architectures

“The current best architectures for image diffusion models are U-Nets (Ronneberger et al., 2015; Salimans et al., 2017)”

Background

Architectures

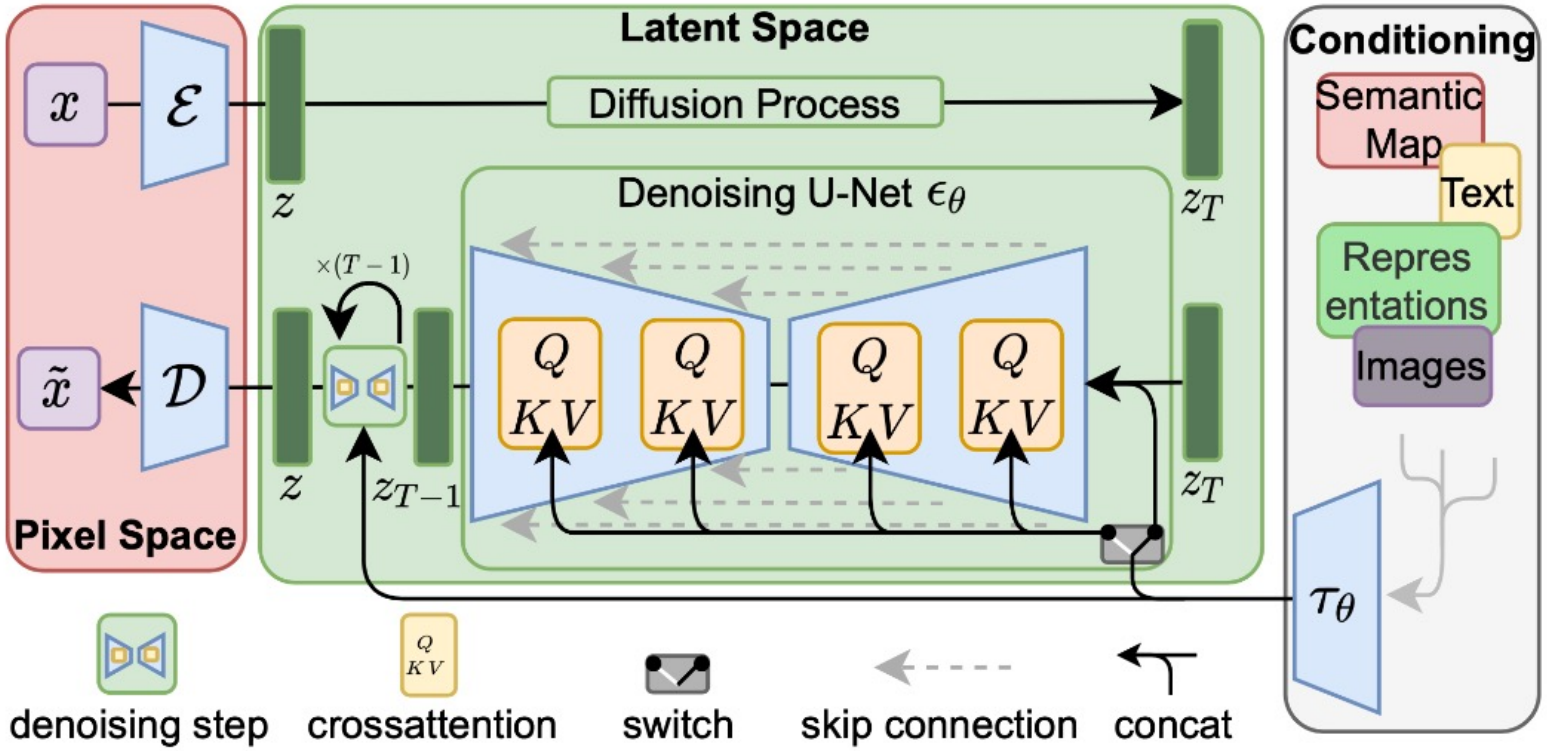
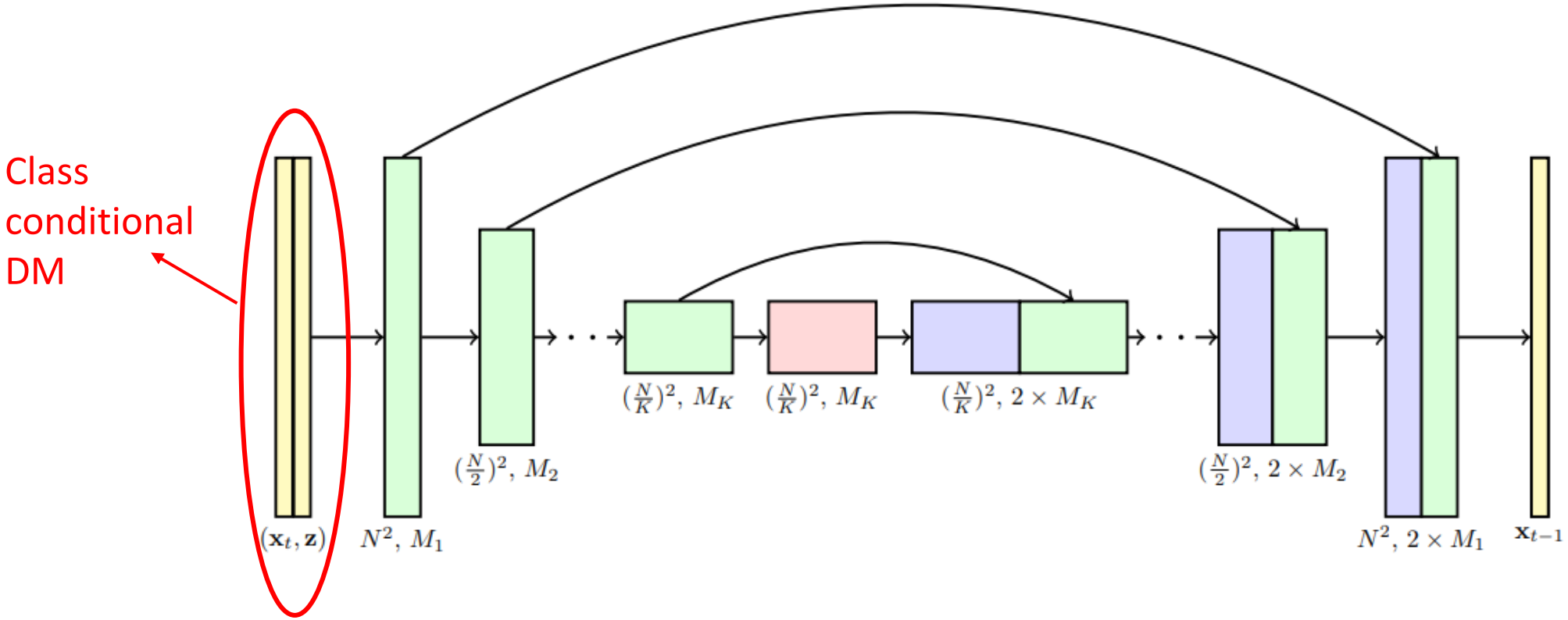


Fig. 9. The architecture of latent diffusion model. (Image source: Rombach & Blattmann, et al. 2022)

Background

Architectures



Outline

- Introduction
- Background
- **Methodology**
- Experiments & Results
- Related Work
- Conclusion

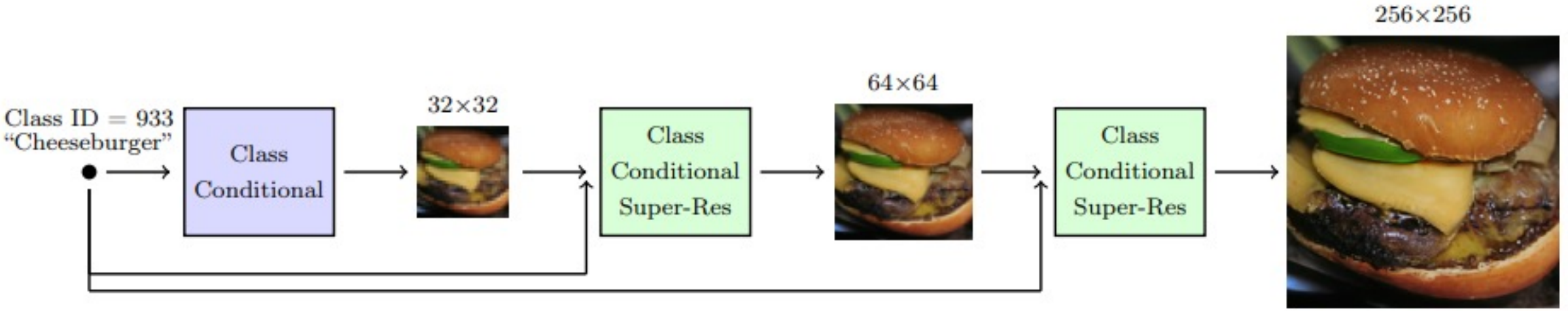
Methodology

Strength of Cascading:

- most of the modeling capacity can be dedicated to low resolution
- train individual models independently
- architectures are tuned at each specific resolution

Methodology

Detailed CDM pipeline for generation of class conditional 256×256 images:



Methodology

conditioning augmentation : The most effective technique is to train each super-resolution model using data augmentation on its low resolution input.

Most effective augmentation:

- On low resolution: add Gaussian noise
- On high resolution: randomly applying Gaussian blur to z

How does *conditioning augmentation* work

(1) Blurring Augmentation

(2) Truncated Conditioning Augmentation

(3) Non-truncated Conditioning Augmentation

Methodology

Suppose

x_0 : high resolution data

z_0 : low resolution counterpart

$p_\theta(z_0)$: low resolution diffusion model

$p_\theta(x_0|\, z_0)$: super-resolution diffusion model

Then cascading pipeline model:

$$p_\theta(x_0) = \int p_\theta(z_0) \cdot p_\theta(x_0|\, z_0)\, dz_0$$

Conditioning Augmentation in CDM

Blurring Augmentation:

Blur z with Gaussian filter

During training, apply this blurring augmentation to 50% of the examples

During inference, no augmentation is applied

Conditioning Augmentation in CDM

Truncated Conditioning Augmentation :

- generating a high resolution sample:

$$p_\theta(\mathbf{x}_0) = \int p_\theta(\mathbf{x}_0|\mathbf{z}_0)p_\theta(\mathbf{z}_0)\,d\mathbf{z}_0 = \int p_\theta(\mathbf{x}_0|\mathbf{z}_0)p_\theta(\mathbf{z}_{0:T})\,d\mathbf{z}_{0:T}.$$

- Truncated Conditioning Augmentation

$$p^s_\theta(\mathbf{x}_0) = \int p_\theta(\mathbf{x}_0|\mathbf{z}_s)p_\theta(\mathbf{z}_s)\,d\mathbf{z}_s = \int p_\theta(\mathbf{x}_0|\mathbf{z}_s)p_\theta(\mathbf{z}_{s:T})\,d\mathbf{z}_{s:T}.$$

Conditioning Augmentation in CDM

Truncated Conditioning Augmentation :

$$p_\theta^s(\mathbf{x}_0) = \int p_\theta(\mathbf{x}_0|\mathbf{z}_s) p_\theta(\mathbf{z}_s) d\mathbf{z}_s = \int p_\theta(\mathbf{x}_0|\mathbf{z}_s) p_\theta(\mathbf{z}_{s:T})\, d\mathbf{z}_{s:T}.$$

- the new base model $p_\theta(\mathbf{z}_s) = \int p_\theta(\mathbf{z}_{s:T}) d\mathbf{z}_{s+1:T}$
- the new super-resolution model

$$p_\theta(\mathbf{x}_0|\mathbf{z}_s) = \int p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{z}_s) d\mathbf{x}_{1:T}$$

Conditioning Augmentation in CDM

Truncated Conditioning Augmentation :

- the new super-resolution model

$$p_\theta(\mathbf{x}_0|\mathbf{z}_s) = \int p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{z}_s) d\mathbf{x}_{1:T}$$

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{z}_s) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t, \mathbf{z}_s, s), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t, \mathbf{z}_s, s)).$$

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t))$$

Conditioning Augmentation in CDM

Truncated Conditioning Augmentation :

Why truncating the low resolution reverse process is a form of data augmentation?

Conditioning Augmentation in CDM

Truncated Conditioning Augmentation :

The training procedure for pθ(x0|zs) involves conditioning on noisy zs ~ q(zs|z0), which, up to scaling, is z0 augmented with Gaussian noise.

New ELBO:

Low resolution model

Super resolution model

$$-\log p_\theta^s(\mathbf{x}_0) \leq \mathbb{E}_q \left[L_T(\mathbf{z}_0) + \sum_{t>s} D_{\mathrm{KL}}(q(\mathbf{z}_{t-1}|\mathbf{z}_t, \mathbf{z}_0) \parallel p_\theta(\mathbf{z}_{t-1}|\mathbf{z}_t)) + L_\theta(\mathbf{x}_0|\mathbf{z}_s) \right]$$

Conditioning Augmentation in CDM

Algorithm 1 Training a two-stage CDM with Gaussian conditioning augmentation

repeat ▷ Train base model
  $(\mathbf{z}_0, \mathbf{c}) \sim p(\mathbf{z}, \mathbf{c})$ ▷ Sample low-resolution image and label
  $t \sim \mathcal{U}(\{1, \dots, T\})$
  $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  $\mathbf{z}_t = \sqrt{\bar{\alpha}_t}\mathbf{z}_0 + \sqrt{1 - \bar{\alpha}_t}\boldsymbol{\epsilon}$
  $\theta \leftarrow \theta - \eta\nabla_\theta \left\|\boldsymbol{\epsilon}_\theta(\mathbf{z}_t, t, \mathbf{c}) - \boldsymbol{\epsilon}\right\|^2$ ▷ Simple loss (can be replaced with a hybrid loss)
until converged
repeat ▷ Train super-resolution model (in parallel with the base model)
  $(\mathbf{x}_0, \mathbf{z}_0, \mathbf{c}) \sim p(\mathbf{x}, \mathbf{z}, \mathbf{c})$ ▷ Sample low- and high-resolution images and label
  $s, t \sim \mathcal{U}(\{1, \dots, T\})$
  $\boldsymbol{\epsilon}_\mathbf{z}, \boldsymbol{\epsilon}_\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ ▷ Note: $\boldsymbol{\epsilon}_\mathbf{z}, \boldsymbol{\epsilon}_\mathbf{x}$ should have the same shapes as $\mathbf{z}_0, \mathbf{x}_0$, respectively
  $\mathbf{z}_t = \sqrt{\bar{\alpha}_s}\mathbf{z}_0 + \sqrt{1 - \bar{\alpha}_s}\boldsymbol{\epsilon}_\mathbf{z}$ ▷ Apply Gaussian conditioning augmentation
  $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t}\boldsymbol{\epsilon}_\mathbf{x}$
  $\theta \leftarrow \theta - \eta\nabla_\theta \left\|\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, \mathbf{z}_s, s, \mathbf{c}) - \boldsymbol{\epsilon}_\mathbf{x}\right\|^2$
until converged

Conditioning Augmentation in CDM

Non-truncated Conditioning Augmentation :

Non-truncated CA uses the same model modifications and training procedure as truncated conditioning augmentation

Only difference:

At sampling time, we always sample z_0 using the full, non-truncated low resolution reverse process

$$\mathbf{z}'_s \sim q(\mathbf{z}_s|\mathbf{z}_0)$$

Conditioning Augmentation in CDM

Algorithm 2 Sampling from a two-stage CDM with Gaussian conditioning augmentation

Require: $\mathbf{c}$: class label
Require: s: conditioning augmentation truncation time
$\mathbf{z}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
if using truncated conditioning augmentation **then**
  for $t = T, \ldots, s+1$ **do**
    $\mathbf{z}_{t-1} \sim p_\theta(\mathbf{z}_{t-1}|\mathbf{z}_t, \mathbf{c})$
  end for
else
  for $t = T, \ldots, 1$ **do**
    $\mathbf{z}_{t-1} \sim p_\theta(\mathbf{z}_{t-1}|\mathbf{z}_t, \mathbf{c})$
  end for
  $\mathbf{z}_s \sim q(\mathbf{z}_s|\mathbf{z}_0)$ ▷ Overwrite previously sampled value of $\mathbf{z}_s$
end if
$\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $t = T, \ldots, 1$ **do**
  $\mathbf{x}_{t-1} \sim p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{z}_s, \mathbf{c})$
end for
return $\mathbf{x}_0$

Conditioning Augmentation in CDM

Non-truncated Conditioning Augmentation :

Advantage over truncated Conditioning Augmentation :

- During the search phase over s, we need to store the low resolution samples just once

Conditioning Augmentation in CDM

Truncated and non-truncated conditioning augmentation should perform similarly, because z_s and z_s' should have similar marginal distributions if the low resolution model is trained well enough

Outline

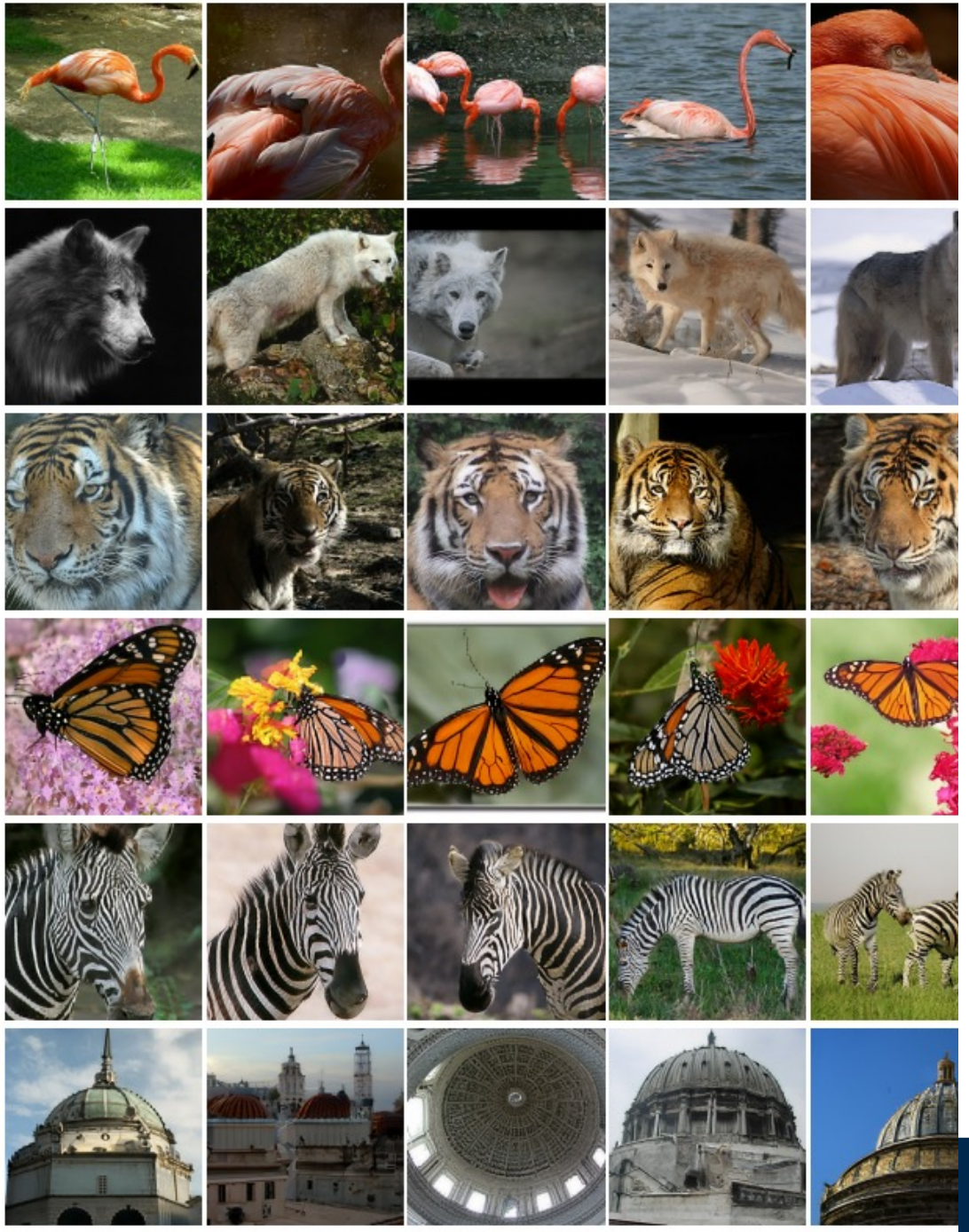
- Introduction
- Background
- Methodology
- **Experiments & Results**
- Related Work
- Conclusion

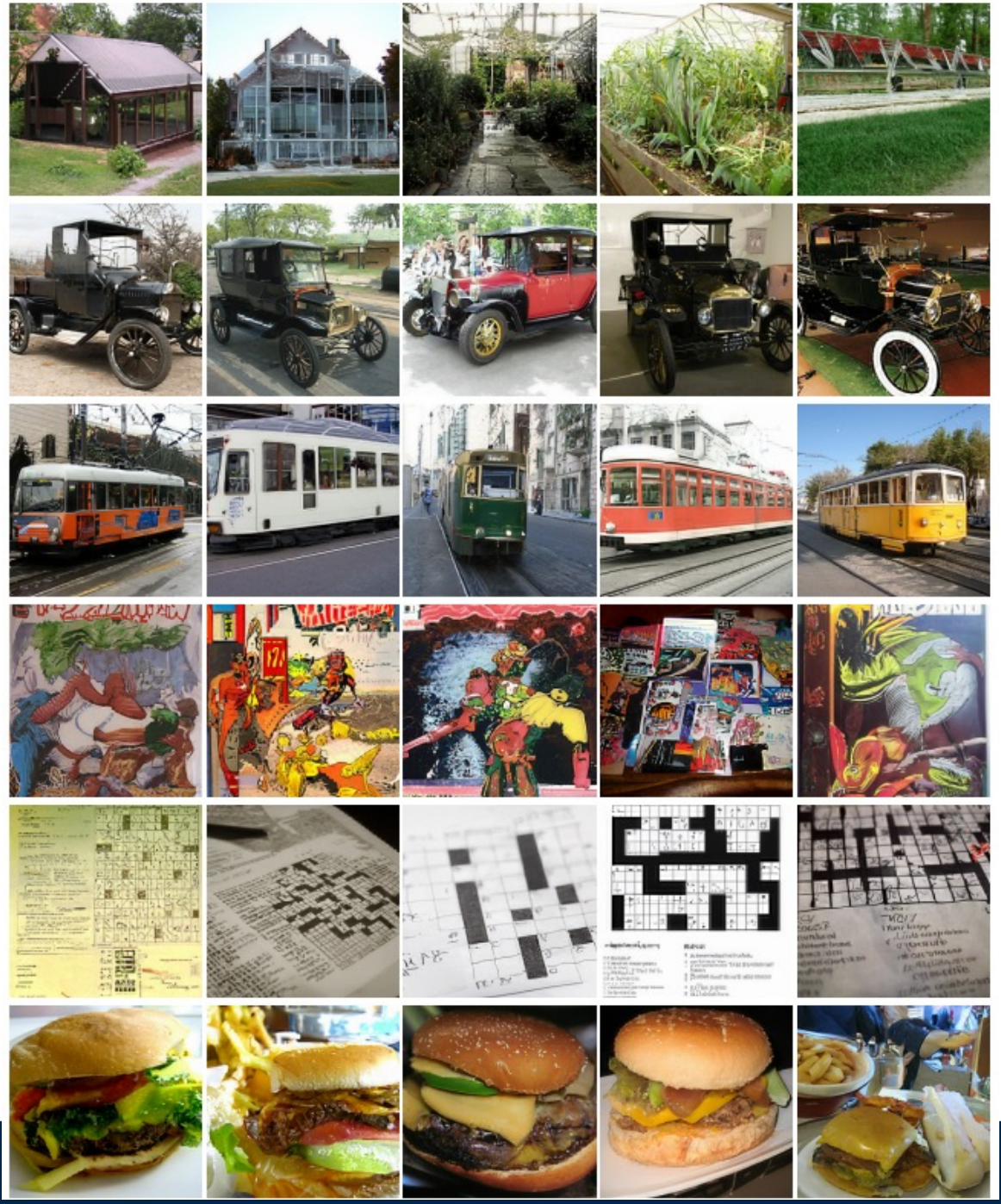
Experiments

Model: a 32×32 base model, a 32×32→64×64 super-resolution model, followed by 64×64→128×128 or 64×64→256×256 super-resolution models.

Models at 32×32 and 64×64 resolutions use 4000 diffusion timesteps. Models at 128×128 and 256×256 resolutions use 100 sampling steps.

Dataset: ImageNet





Compared to other generatice models

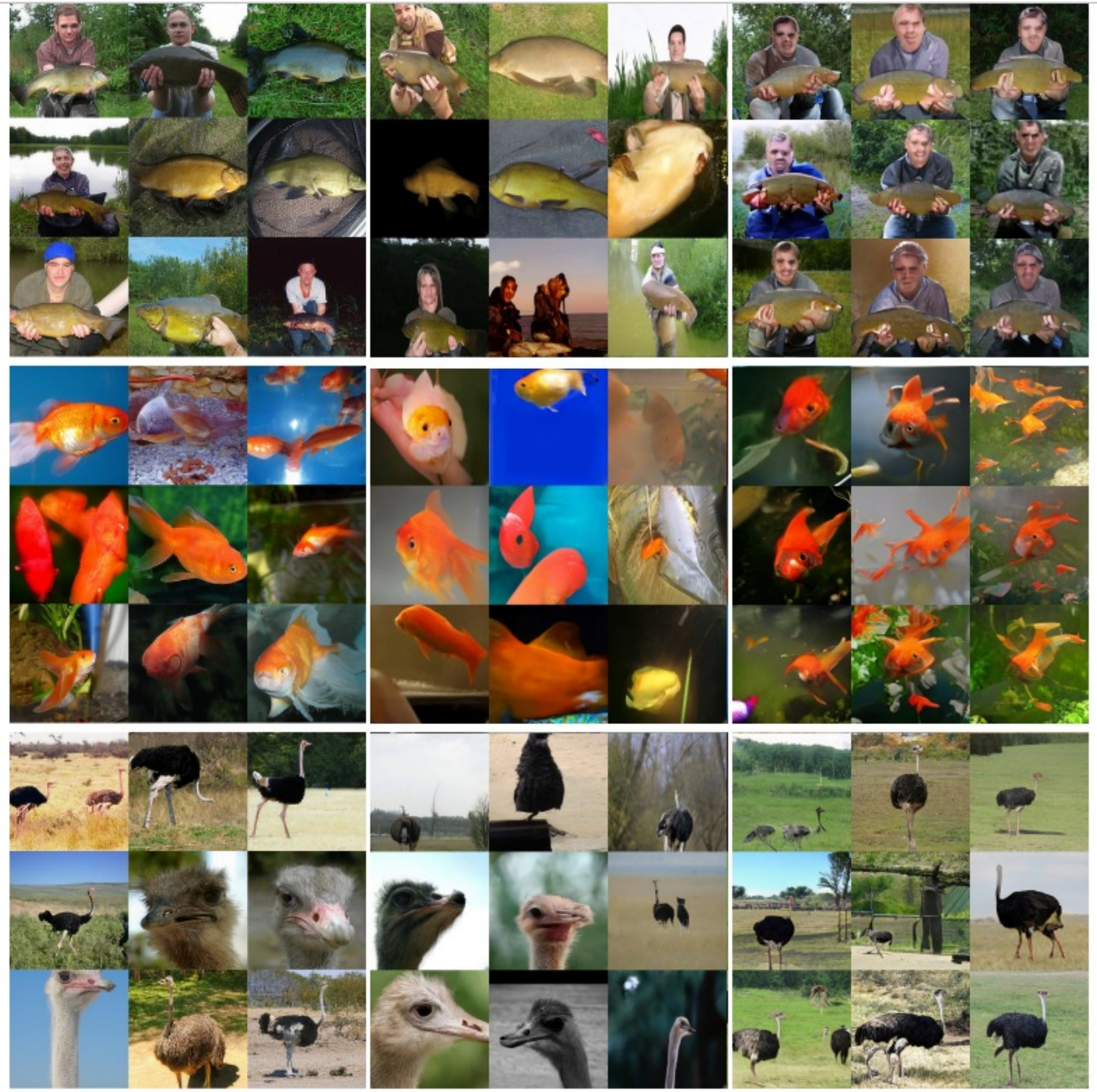
Model	FID vs train	FID vs validation	IS
32×32 resolution			
CDM (ours)	1.11	1.99	26.01 ± 0.59
64×64 resolution			
BigGAN-deep, by (Dhariwal and Nichol, 2021)	4.06		
Improved DDPM (Nichol and Dhariwal, 2021)	2.92		
ADM (Dhariwal and Nichol, 2021)	2.07		
CDM (ours)	**1.48**	2.48	67.95 ± 1.97
128×128 resolution			
BigGAN-deep (Brock et al., 2019)	5.7		124.5
BigGAN-deep, max IS (Brock et al., 2019)	25		**253**
LOGAN (Wu et al., 2019)	**3.36**		148.2
ADM (Dhariwal and Nichol, 2021)	5.91		
CDM (ours)	3.52	3.76	128.80 ± 2.51
256×256 resolution			
BigGAN-deep (Brock et al., 2019)	6.9		171.4
BigGAN-deep, max IS (Brock et al., 2019)	27		**317**
VQ-VAE-2 (Razavi et al., 2019)	31.11		
Improved DDPM (Nichol and Dhariwal, 2021)	12.26		
SR3 (Saharia et al., 2021)	11.30		
ADM (Dhariwal and Nichol, 2021)	10.94		100.98
ADM+upsampling (Dhariwal and Nichol, 2021)	7.49		127.49
CDM (ours)	**4.88**	4.63	158.71 ± 2.26

(a) Class-conditional ImageNet sample quality results for classifier guidance-free methods

Compared to other generatice models

Model	Top-1 Accuracy	Top-5 Accuracy
128×128 resolution		
Real	68.82%	88.79%
BigGAN-deep (Brock et al., 2019)	40.64%	64.44%
HAM (De Fauw et al., 2019)	54.05%	77.33%
CDM (ours)	**59.84%**	**81.79%**
256×256 resolution		
Real	73.09%	91.47%
BigGAN-deep (Brock et al., 2019)	42.65%	65.92%
VQ-VAE-2 (Razavi et al., 2019)	54.83%	77.59%
CDM (ours)	**63.02%**	**84.06%**

(b) Classification Accuracy Score (CAS) results



CDM (ours) VQ-VAE-2 BigGAN-deep

Compared to non-cascaded DM

Model	FID vs train	FID vs validation	IS
Improved DDPM (Nichol and Dhariwal, 2021)	2.92		
Our reimplementation	2.44	**2.91**	49.81 ± 0.65
+ more sampling steps	**2.35**	**2.91**	**52.72 ± 1.15**

(a) Improvements to a non-cascaded baseline

Ablation

Conditioning	FID vs train	FID vs validation	IS
No cascading	2.35	2.91	52.72 ± 1.15
$16\times16 \rightarrow 64\times64$ cascading			
$s = 0$	6.02	5.84	35.59 ± 1.19
$s = 101$	3.41	3.67	44.72 ± 1.12
$s = 1001$	**2.13**	**2.79**	$\mathbf{54.47 \pm 1.05}$

Experiments at 64x64

Conditioning	FID vs train	FID vs validation	IS
No conditioning augmentation (baseline)			
$s = 0$	1.71	2.46	61.34 ± 1.58
Truncated conditioning augmentation			
$s = 251$	1.50	**2.44**	66.76 ± 1.76
$s = 501$	**1.48**	2.48	67.95 ± 1.97
$s = 751$	**1.48**	2.51	**68.48 ± 1.77**
$s = 1001$	1.49	2.51	67.95 ± 1.51
$s = 1251$	1.51	2.54	67.20 ± 1.94
$s = 1501$	1.54	2.56	67.09 ± 1.67
Non-truncated conditioning augmentation			
$s = 251$	1.58	2.50	66.21 ± 1.51
$s = 501$	1.53	2.51	**67.59 ± 1.85**
$s = 751$	**1.48**	2.47	67.48 ± 1.31
$s = 1001$	1.49	2.48	66.51 ± 1.59
$s = 1251$	**1.48**	**2.46**	66.28 ± 1.49
$s = 1501$	1.50	2.47	65.59 ± 0.86

Outline

- Introduction
- Background
- Methodology
- Experiments & Results
- **Related Work**
- Conclusion

Related Works

Cascading pipelines have been investigated in work on VQ-VAEs (van den Oord et al., 2016c; Razavi et al., 2019) and autoregressive models (Menick and Kalchbrenner, 2019).

Cascading pipelines have also been investigated for diffusion models, such as SR3 (Saharia et al., 2021), Improved DDPM (Nichol and Dhariwal, 2021), and concurrently in ADM (Dhariwal and Nichol, 2021)

Related Works

- Concurrent work (Dhariwal and Nichol, 2021) used an improved architecture, named ADM, and a classifier guidance technique in which a class-conditional diffusion model sampler is modified to simultaneously take gradient steps to maximize the score of an extra trained image classifier

Outline

- Introduction
- Background
- Methodology
- Experiments & Results
- Related Work
- **Conclusion**

Conclusion

- CDMs are capable of outperforming state-of-the-art generative models on the ImageNet class-conditional generation benchmark when paired with conditioning augmentation

Quiz

1. According to the paper, should we set the truncated augmentation timestep s as large as possible to get better results?

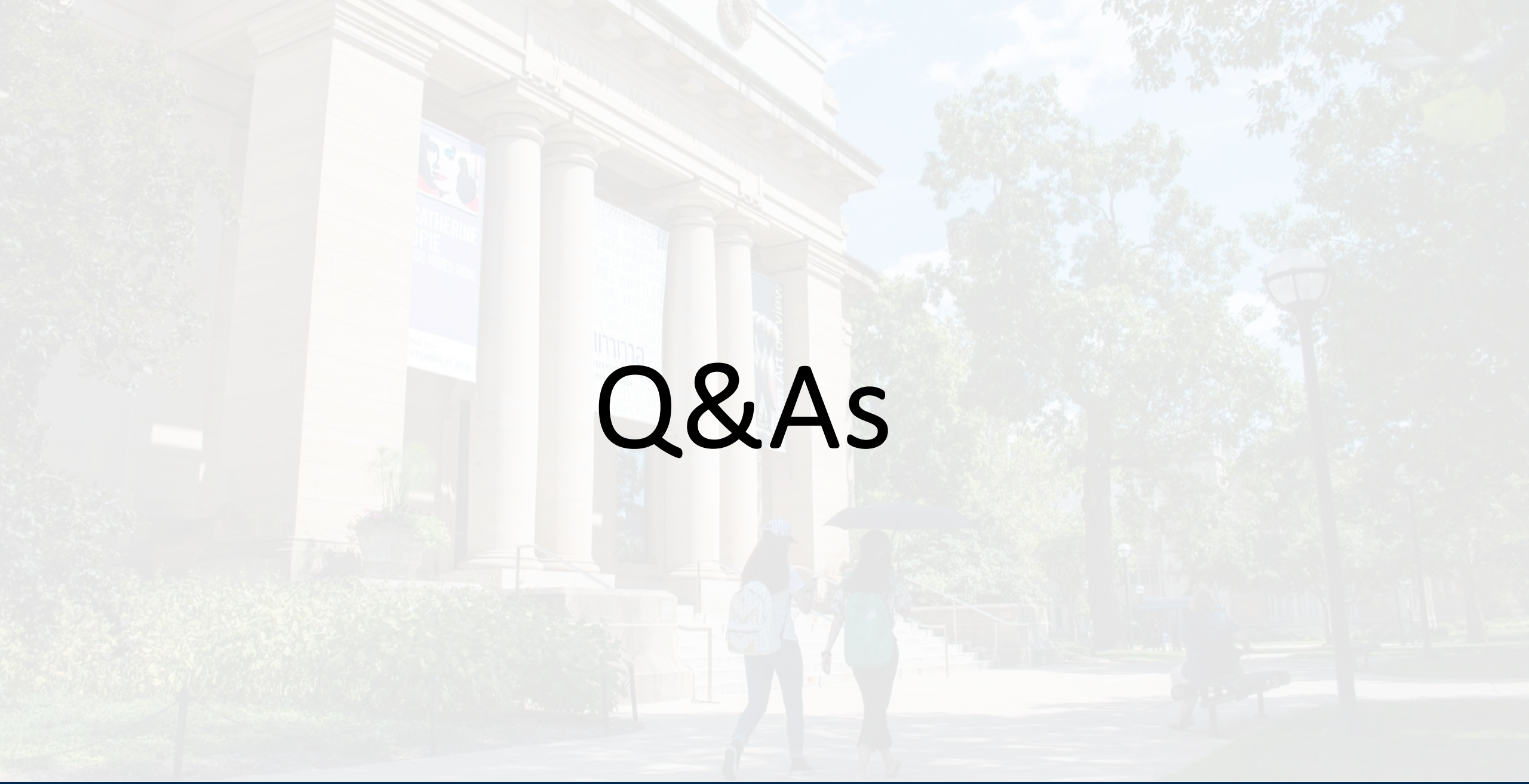
No.

Quiz

2. According to the paper, which Conditioning Augmentation does the author propose to use in practice, truncated or non-truncated? Why?

Non-truncated.

They are approximately equally effective at improving sample quality. But the truncated augmentation requires more memory space.



Q&As

Thank you!