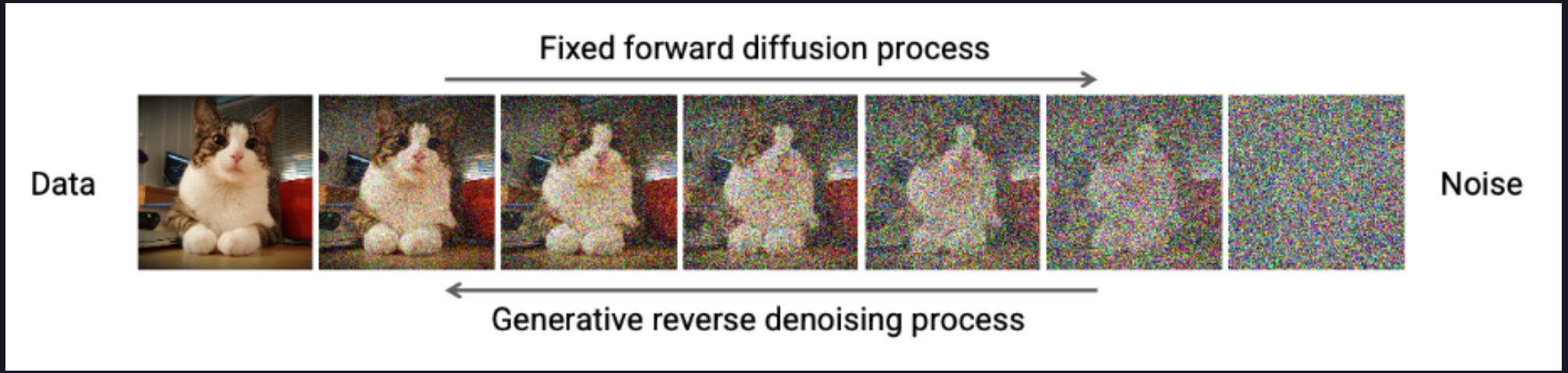
Boosting Generative Image Modeling via Joint Image Feature Syntheis

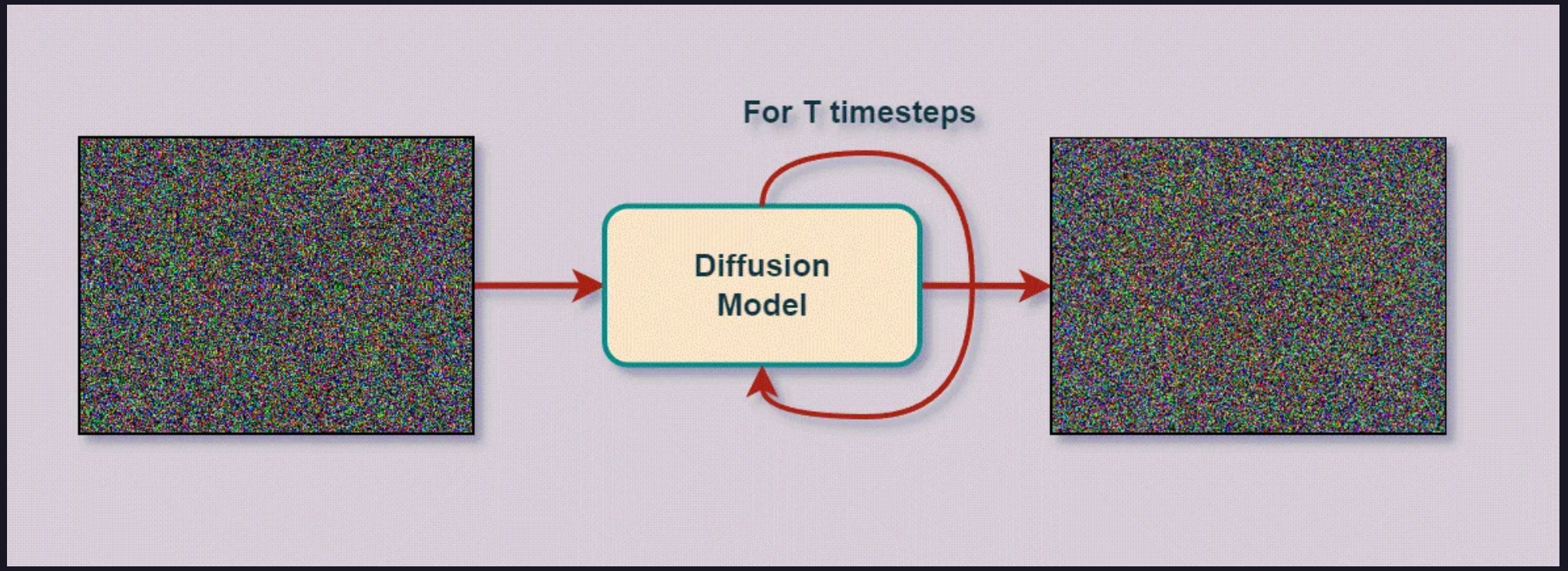
Theodoros Kouzelis, Eystathis Karypidis, Ioannis Kakogeorgiou,
Spyros Gidaris, Nikos Komodakis

Diffusion / Flow Models (in a nutshell)

- Given an Image $\mathbf{x}$ and Gaussina noise $\epsilon \sim \mathcal{N}(0, I)$
- Construct a time dependent noising process:
  - $\mathbf{x}_t = \alpha_t \mathbf{x} + \sigma_t \boldsymbol{\epsilon}$
- Reverse process to generate data from noise:
  - $\mathbf{x}_{t-1} = \tilde{\mathbf{x}}_\mathbf{t} + \boldsymbol{w}_t \cdot \text{Network Output}$



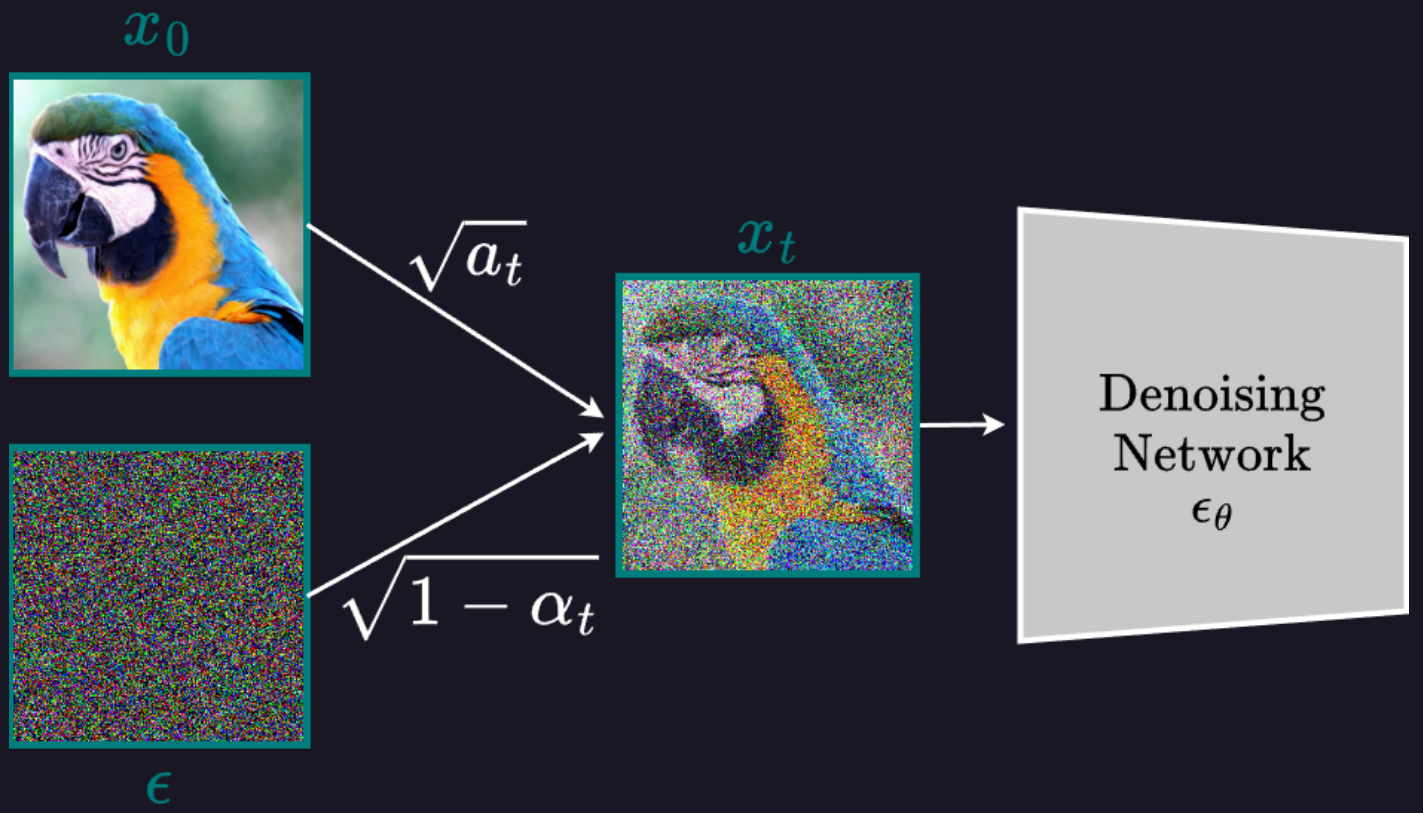
Diffusion / Flow Models



Diffusion Models for Image Generation https://learnopencv.com/image-generation-using-diffusion-models/

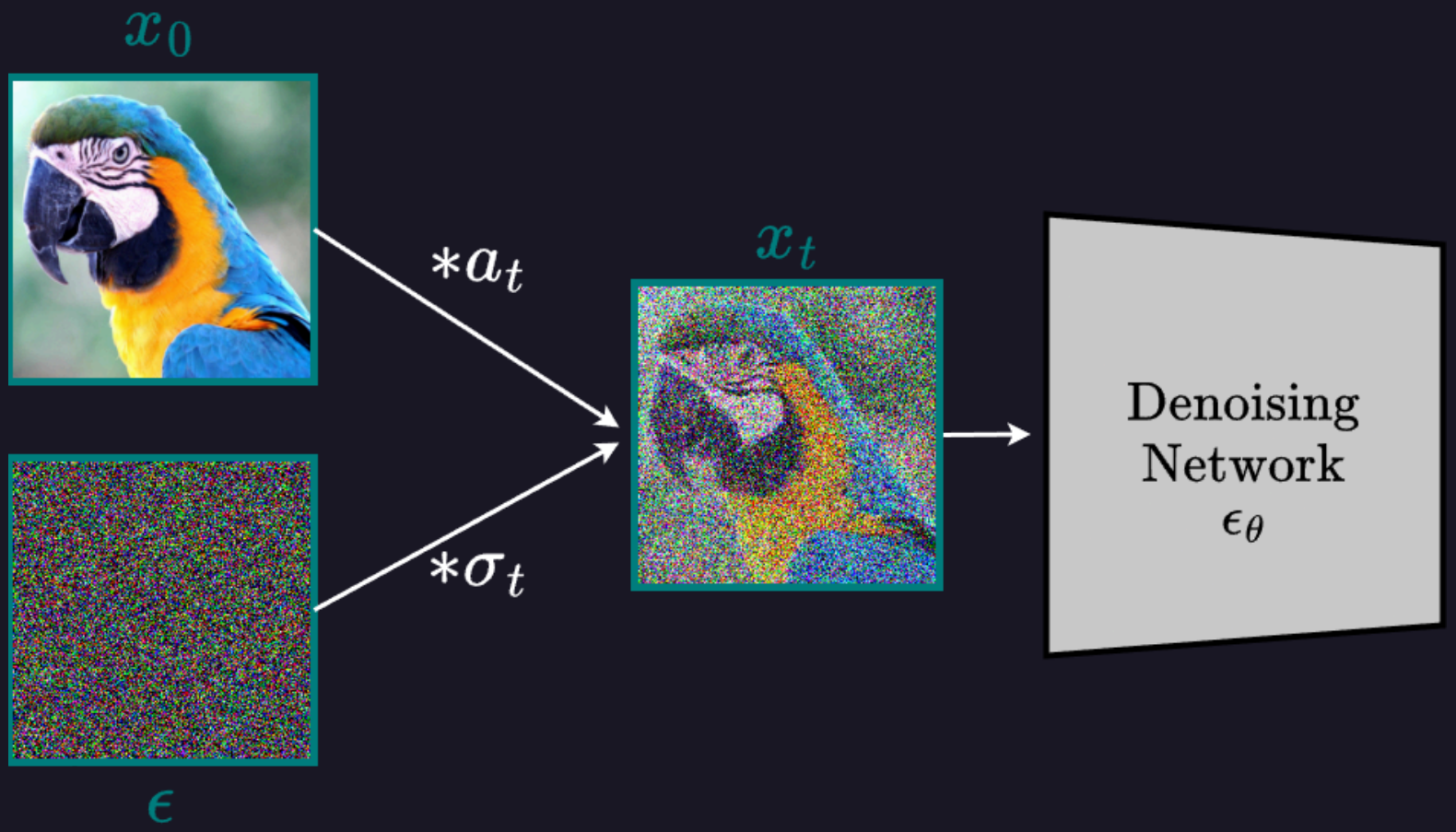
Denoising Objective

- Construct training sample $\mathbf{x}_t = \sqrt{\alpha_t}\mathbf{x} + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}$



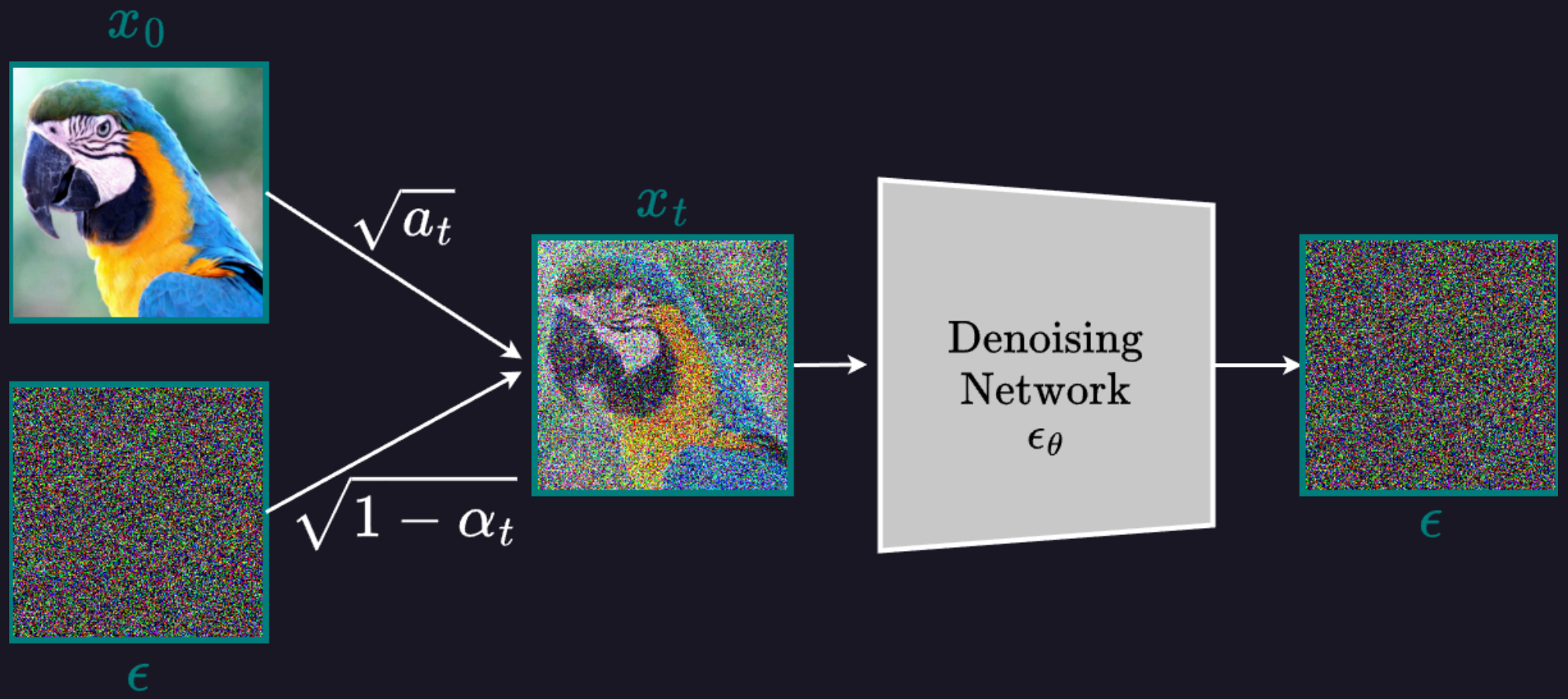
Denoising Objective

- Give $\mathbf{x_t}$ as input to a denoising network ϵ_θ



Denoising Objective

- Loss funtion $\mathcal{L}_{\text{simple}} = \mathbb{E}_x \|\epsilon_\theta(\mathbf{x_t}) - \epsilon\|_2^2$

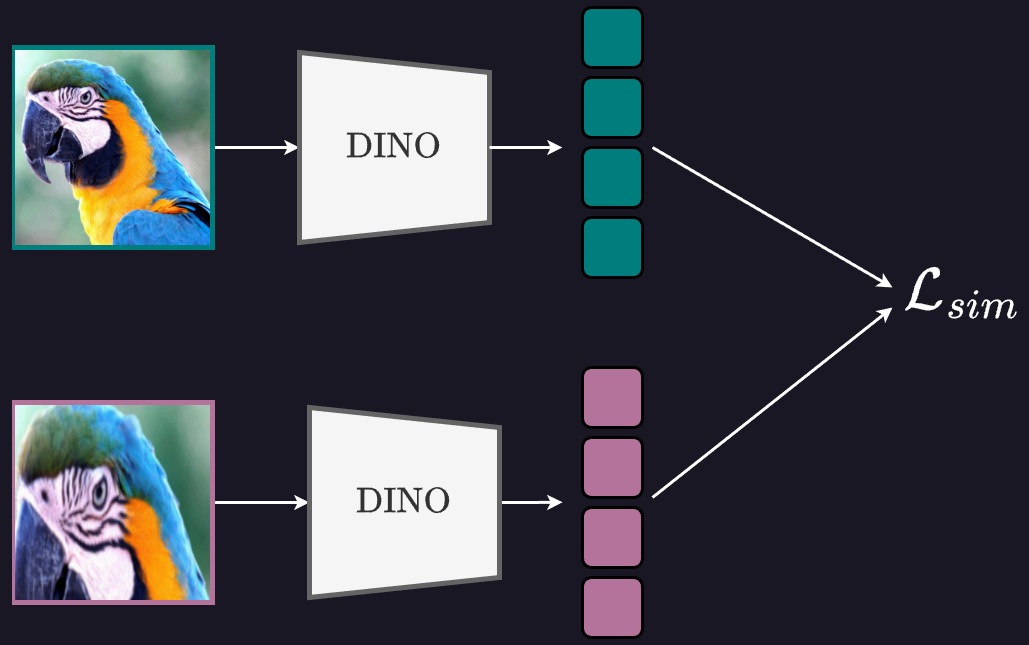


The problem with the Denoising Objective

- Loss funtion $\left\|\epsilon_\theta\left(\mathbf{x_t}\right) - \epsilon\right\|_2^2$
- Not capable of eliminating unnecessary details in $\mathbf{x}$
- Does not result in good representations

DINO (in a nutshell)

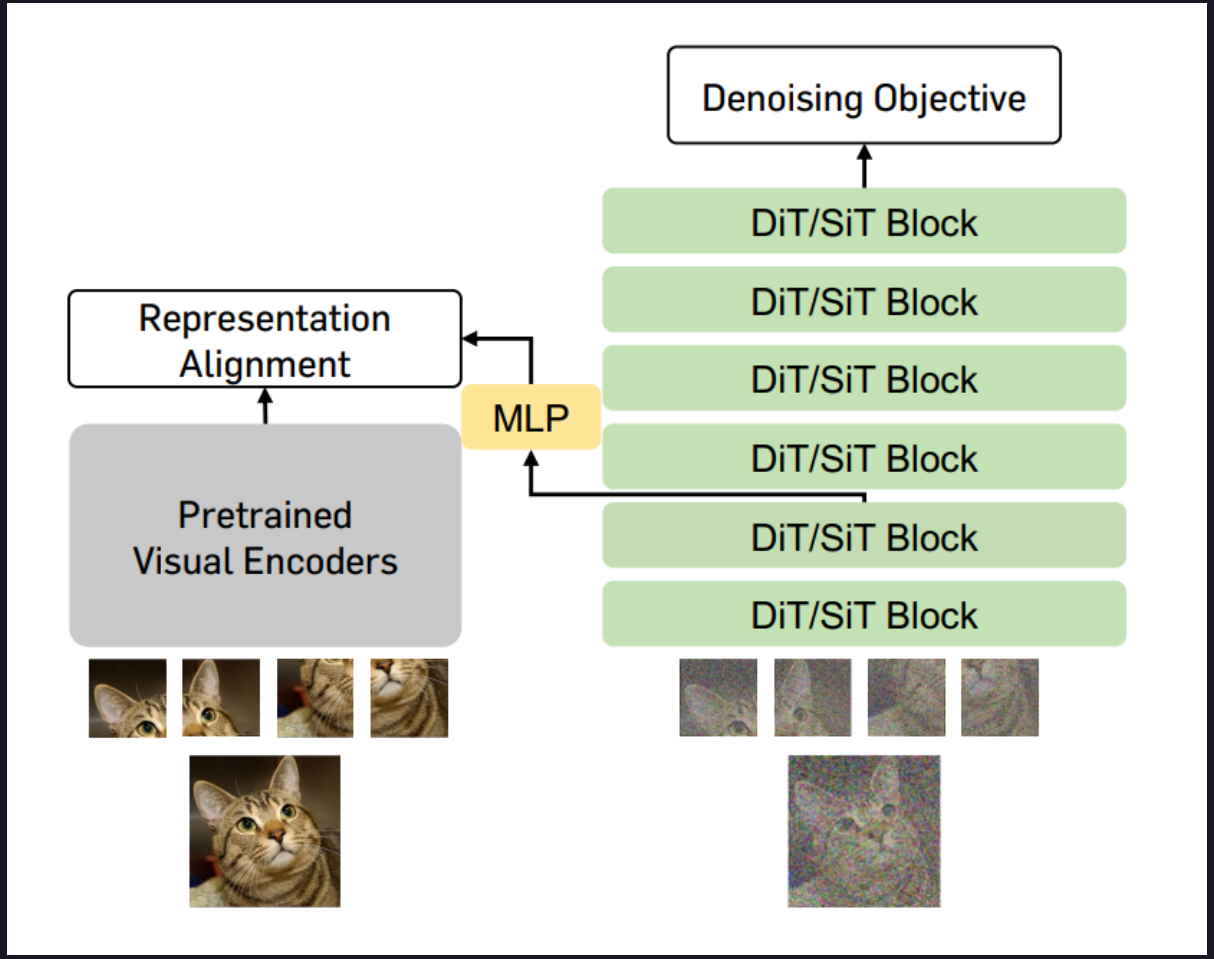
- Take different crops from an image
- Force representations to be similar
- Very strong representations



Caron et al., 2021, Emerging Properties in Self-Supervised Vision Transformers

Representation Alignment

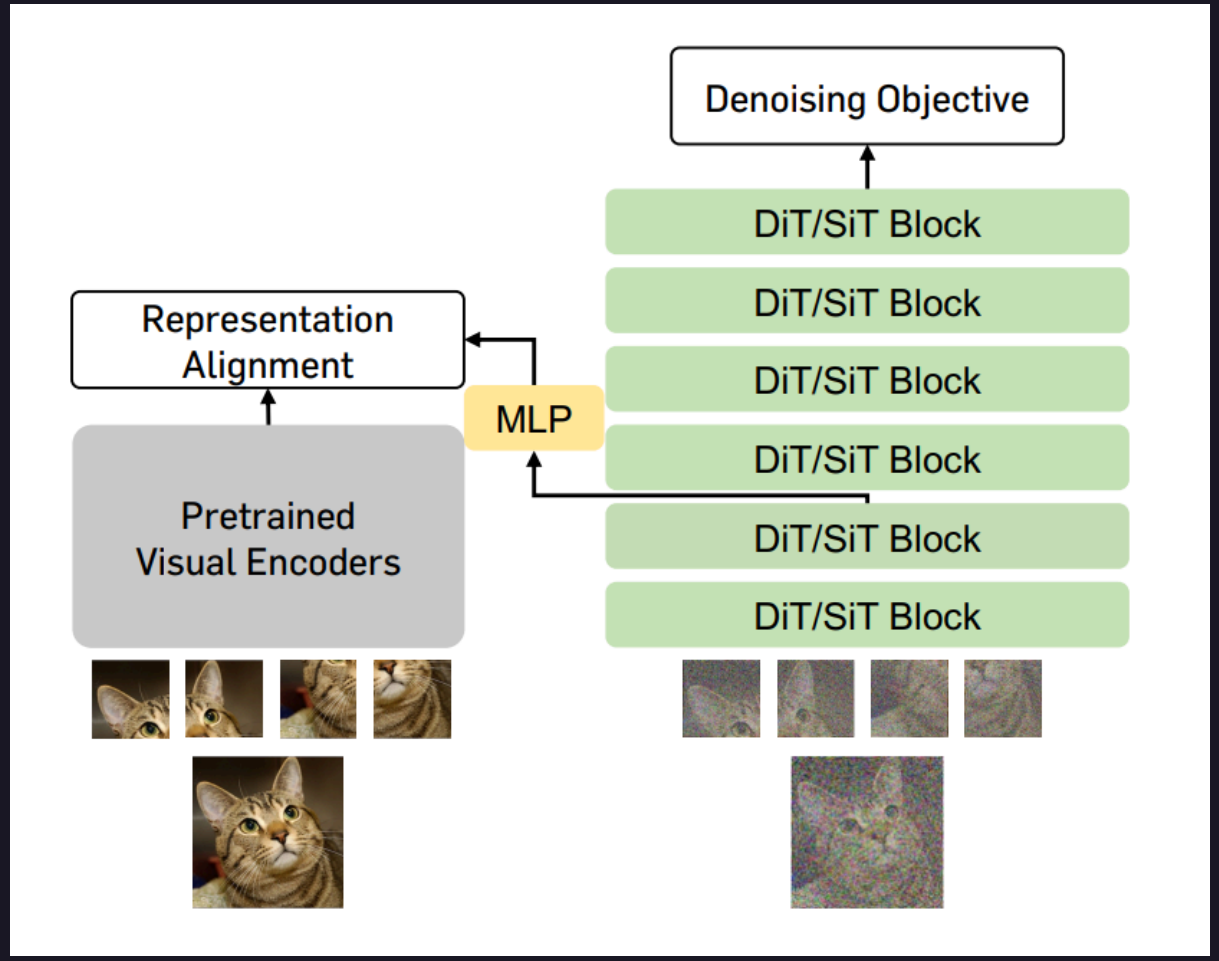
- Align features with the representations of powerful visual encoders.



Yu et al., 2025, Representation Alignment for Generation: Training Diffusion Transformers Is Easier Than You Think

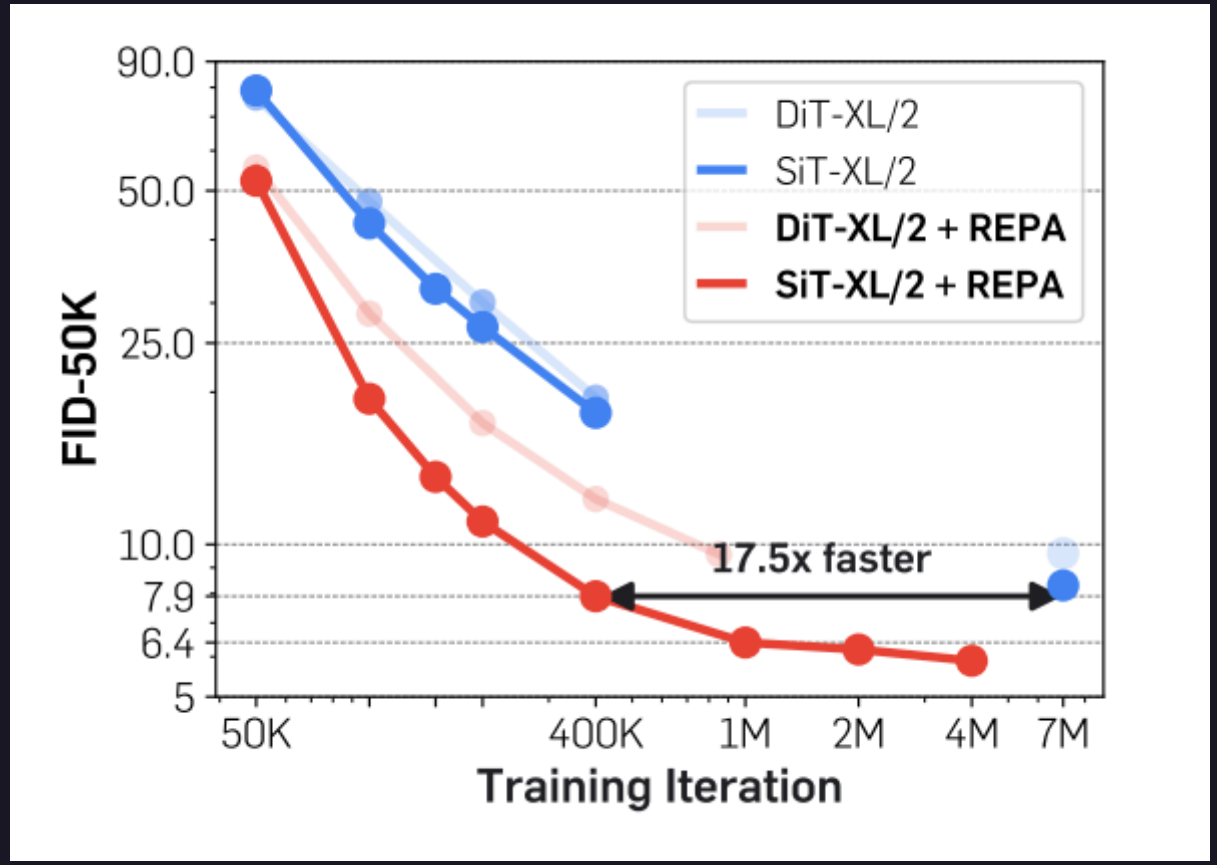
Representation Alignment

- Let h_t be the an intermediate feature
- $y_* = f(x)$ (e.g. DINOv2)
- and h_ϕ a simple projection layer
- REPA loss:
  - $-\mathbb{E}[\frac{1}{N}\sum_i \text{sim}((y_*^i), h_\phi(h_t^i))]$



Representation Alignment

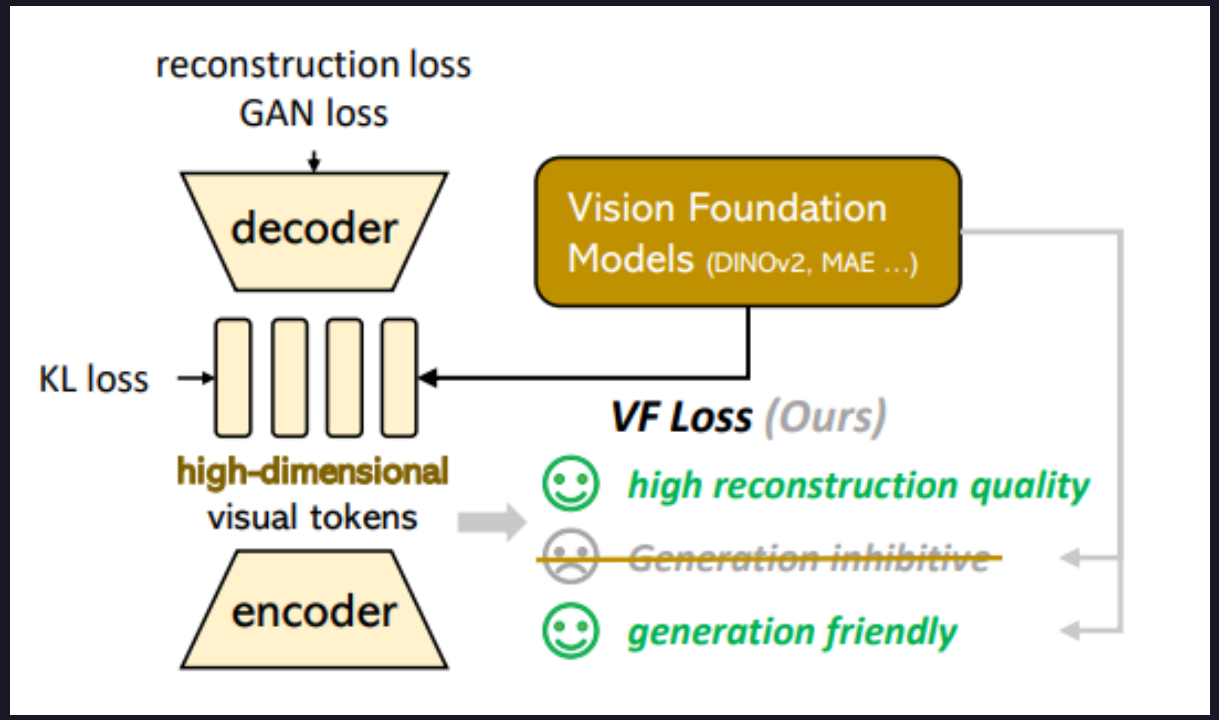
- $\times 17$ faster convergence
- Better generative performance



Yu et al., 2025, Representation Alignment for Generation: Training Diffusion Transformers Is Easier Than You Think

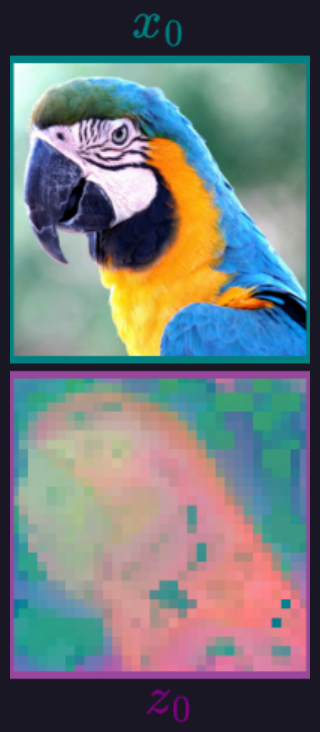
Representation Alignment for VAE Latents

- Aligning the **VAE latents** with DINOv2 also results in better generative performance



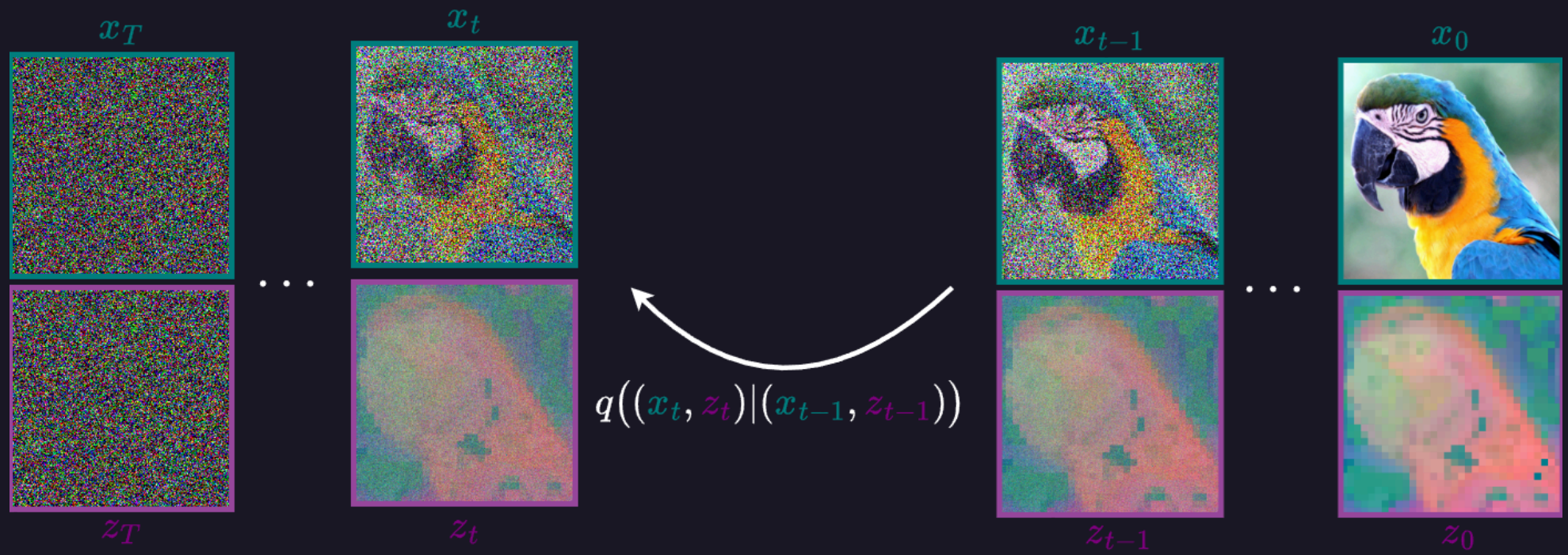
Yao et al., 2025, Reconstruction vs. Generation: Taming Optimization Dilemma in Latent Diffusion Models

ReDi: Joint image-feature Synthesis



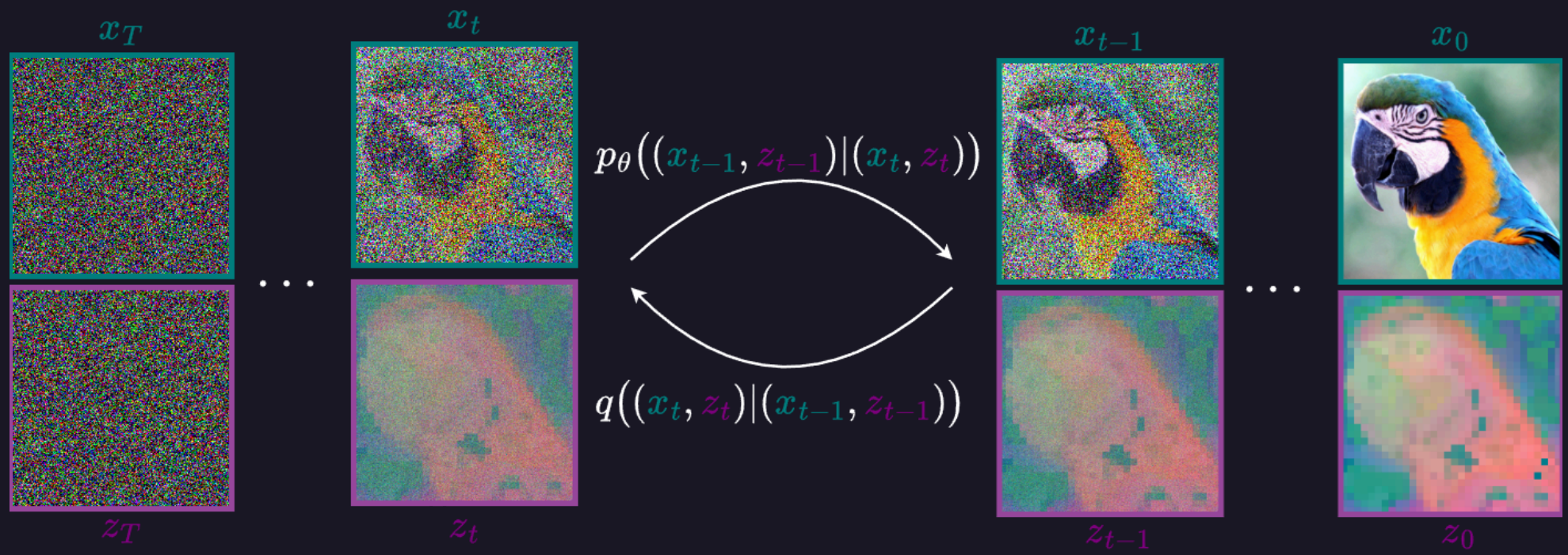
ReDi: Joint image-feature Synthesis

Joint Forward Process

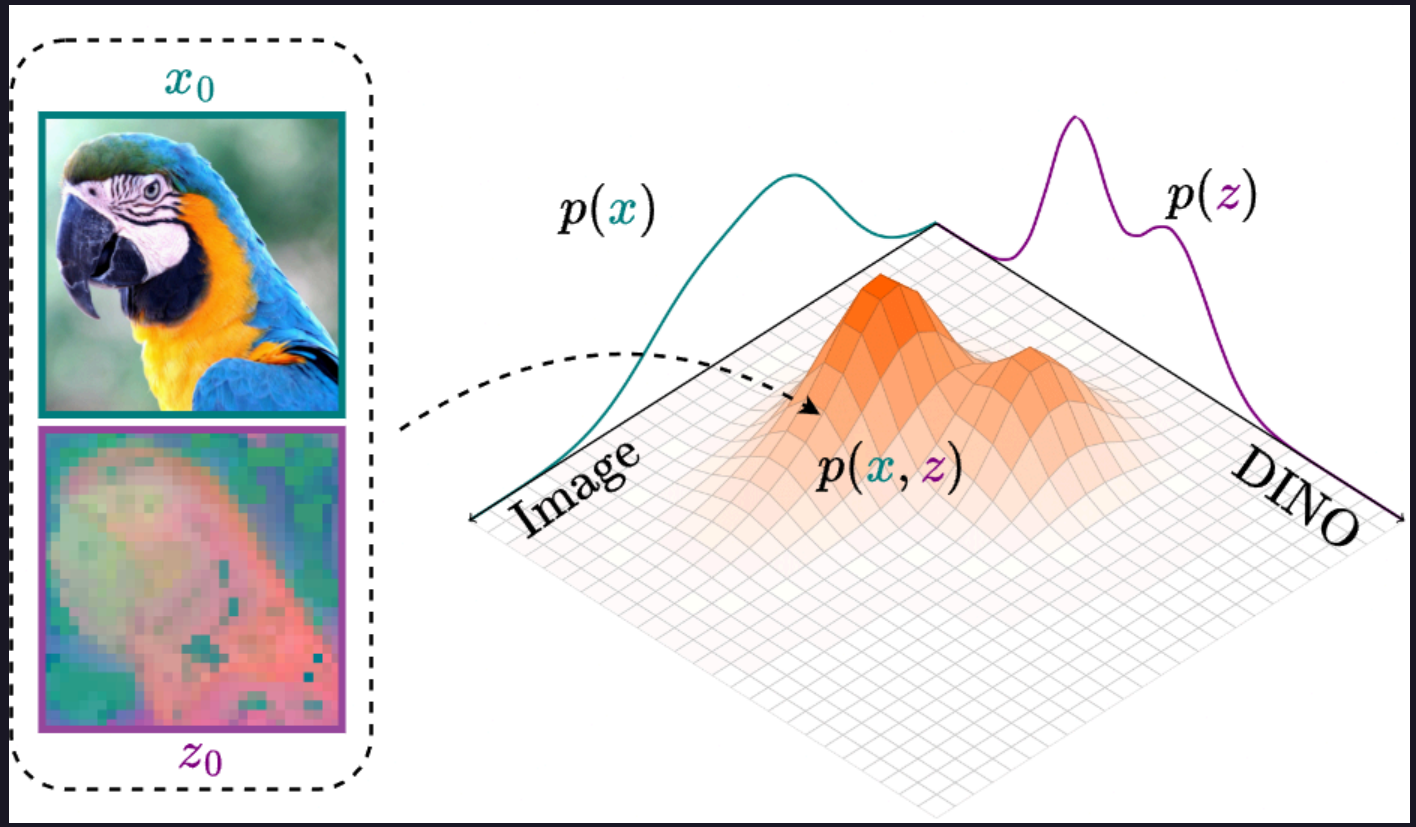


ReDi: Joint image-feature Synthesis

Joint Reverse Process

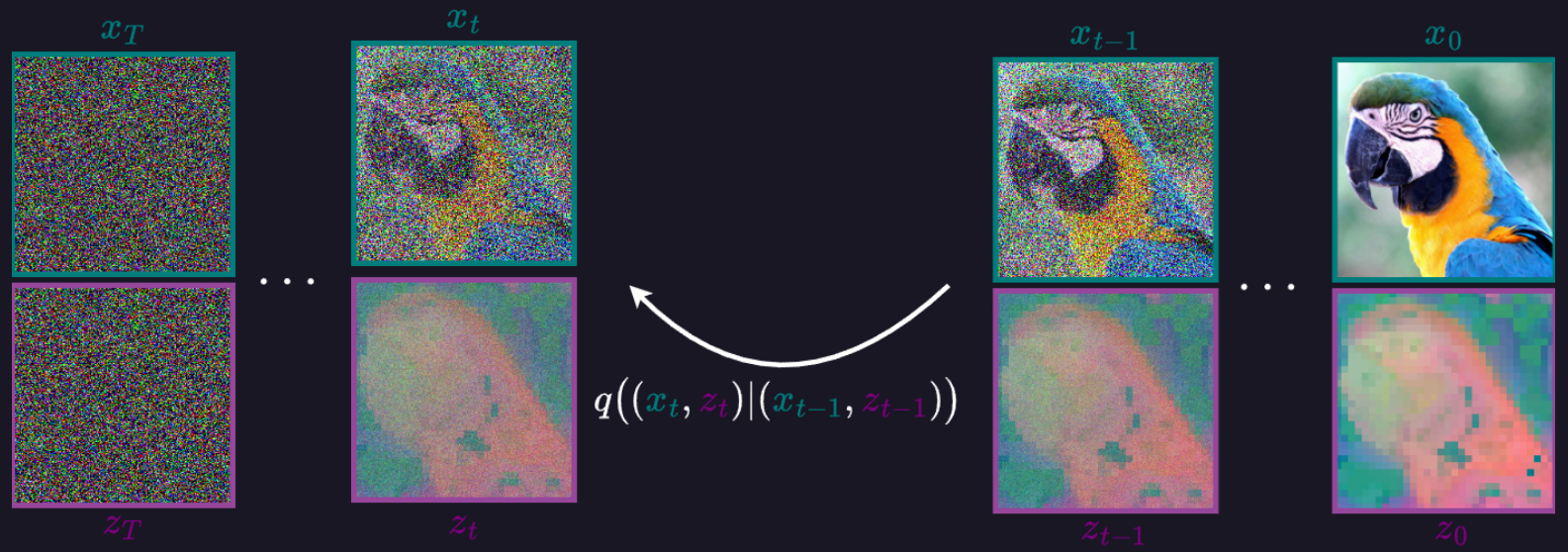


ReDi: Joint image-feature Synthesis



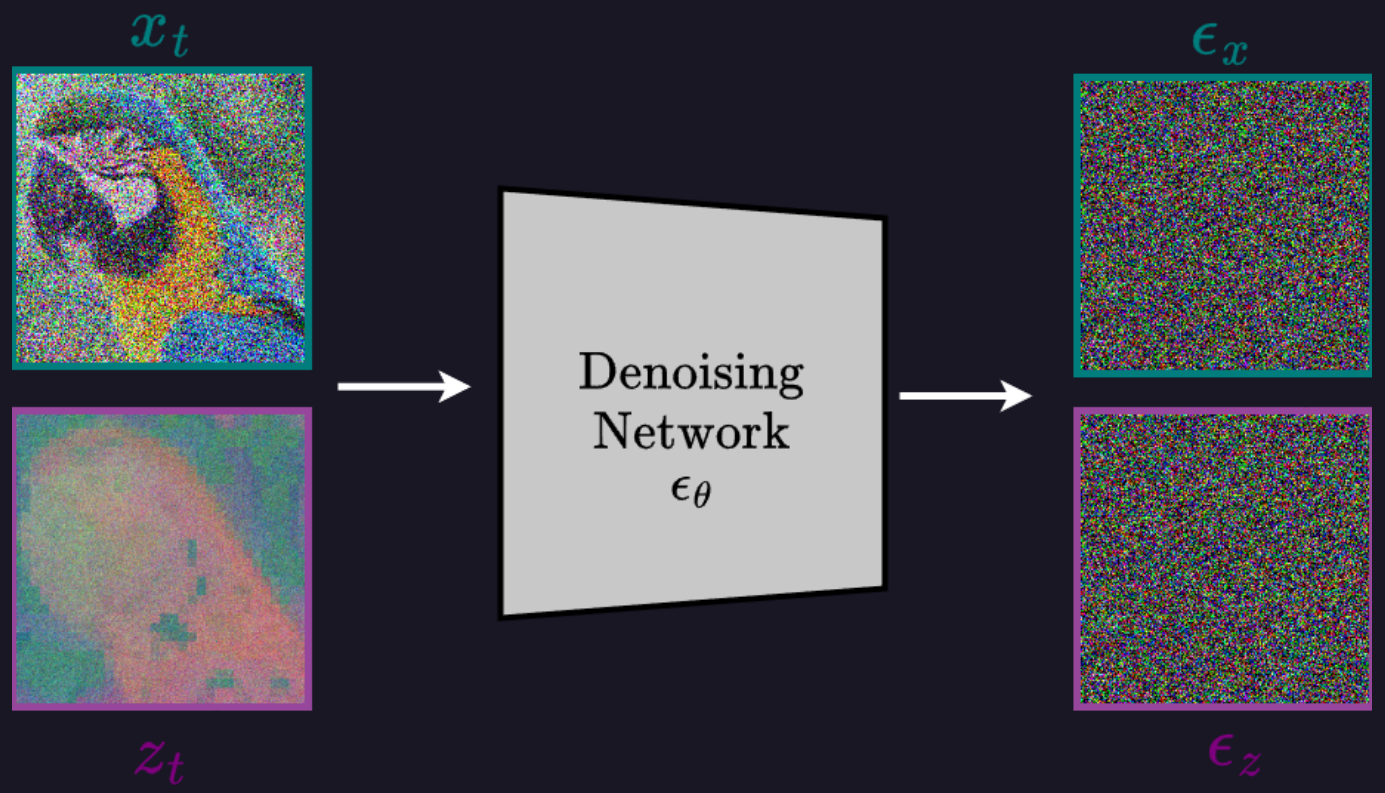
Joint Forward Process

- VAE Latents: $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_x$
- DINOv2: $\mathbf{z}_t = \sqrt{\bar{\alpha}_t}\mathbf{z}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_z$



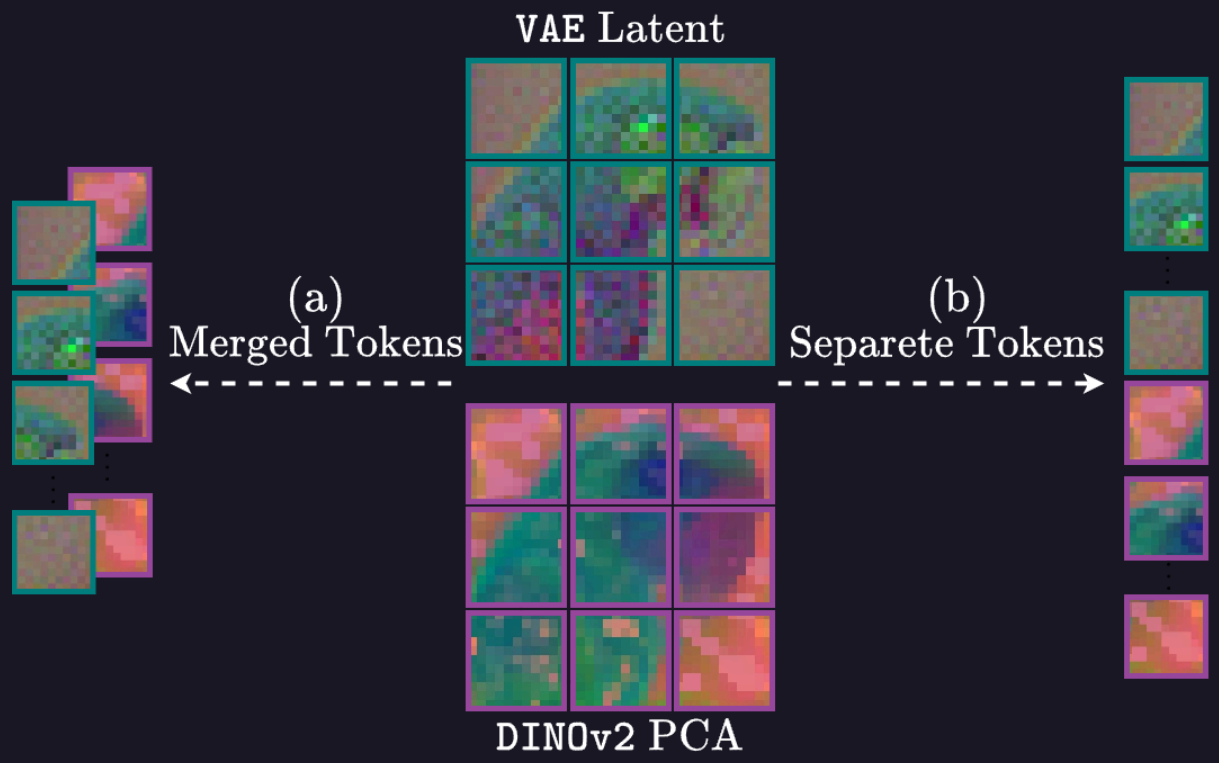
Joint Denoising Objective

- $\mathcal{L}_{joint} = \mathop{\mathbb{E}}_{\mathbf{x}_0, \mathbf{z}_0} \left[\| \boldsymbol{\epsilon}_\theta^x(\mathbf{x}_t, \mathbf{z}_t, t) - \boldsymbol{\epsilon}_x \|_2^2 + \| \boldsymbol{\epsilon}_\theta^z(\mathbf{x}_t, \mathbf{z}_t, t) - \boldsymbol{\epsilon}_z \|_2^2 \right]$



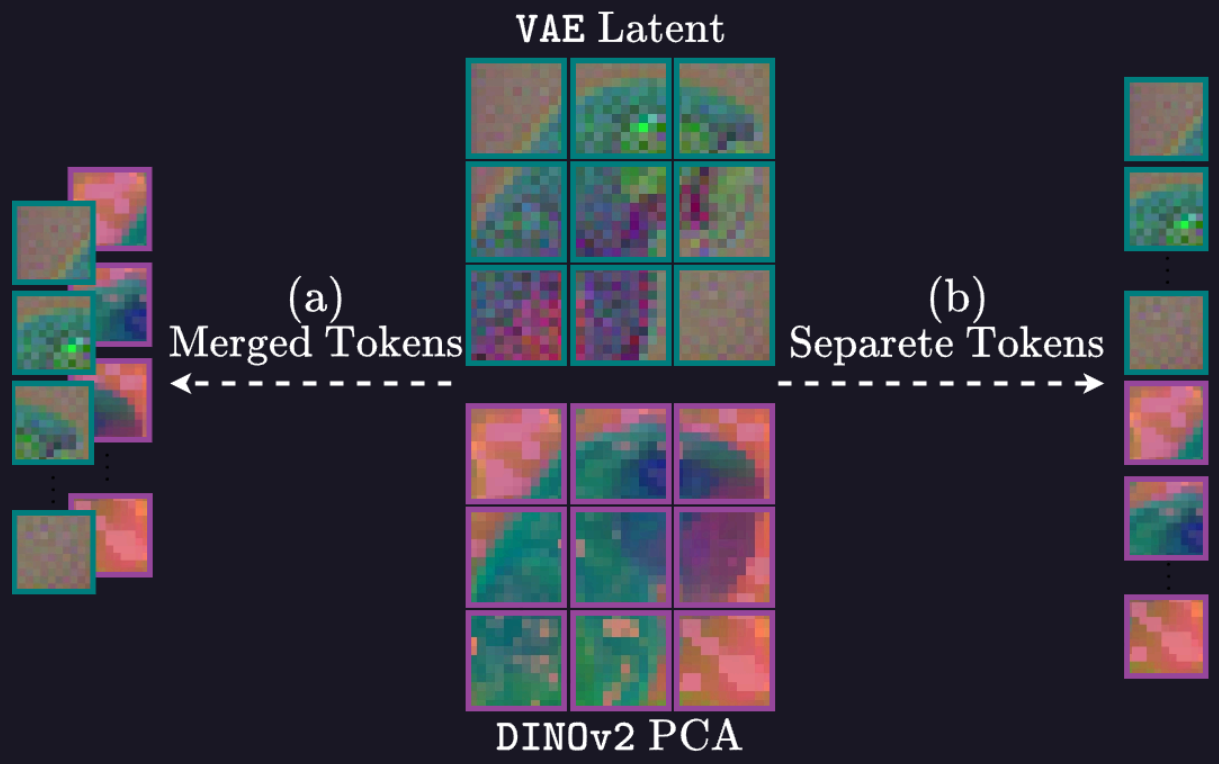
Fusing the two modelities in a single input sequence

- Merged Tokens
- Separete Tokens



Fusing the two modelities in a single input sequence

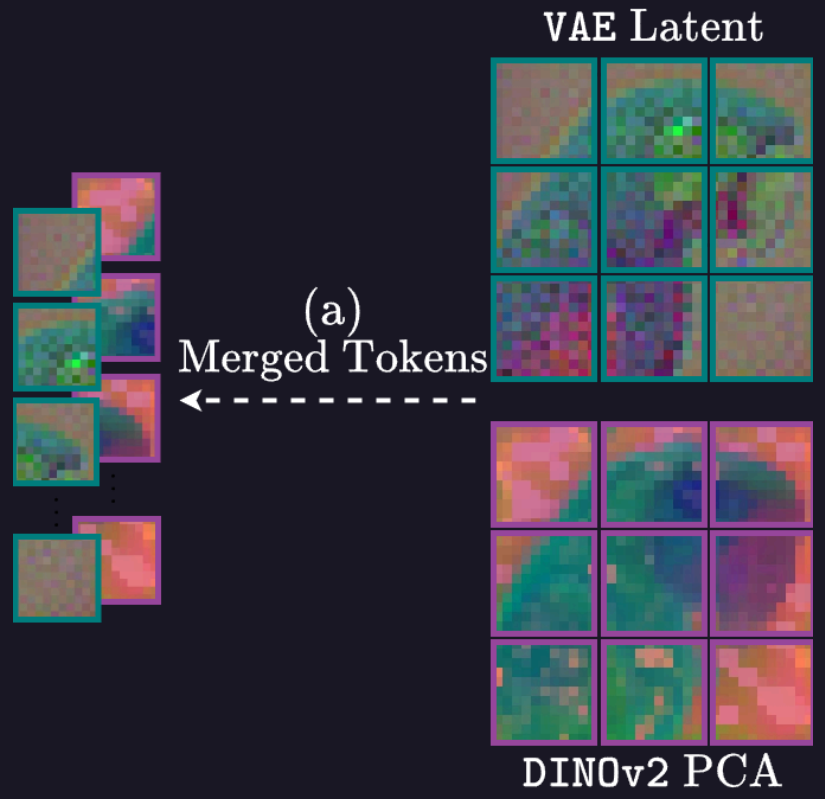
- Modality-specific **embedding** matrices:
  - $\mathbf{W}^{x}_{\text{emp}} \in \mathbb{R}^{C_x \times C_d}$
  - $\mathbf{W}^{z}_{\text{emp}} \in \mathbb{R}^{C_z \times C_d}$
- Modality-specific **projection** matrices:
  - $\mathbf{W}^{x}_{\text{dec}} \in \mathbb{R}^{C_d \times C_x}$
  - $\mathbf{W}^{z}_{\text{dec}} \in \mathbb{R}^{C_d \times C_x}$



Merged Tokens

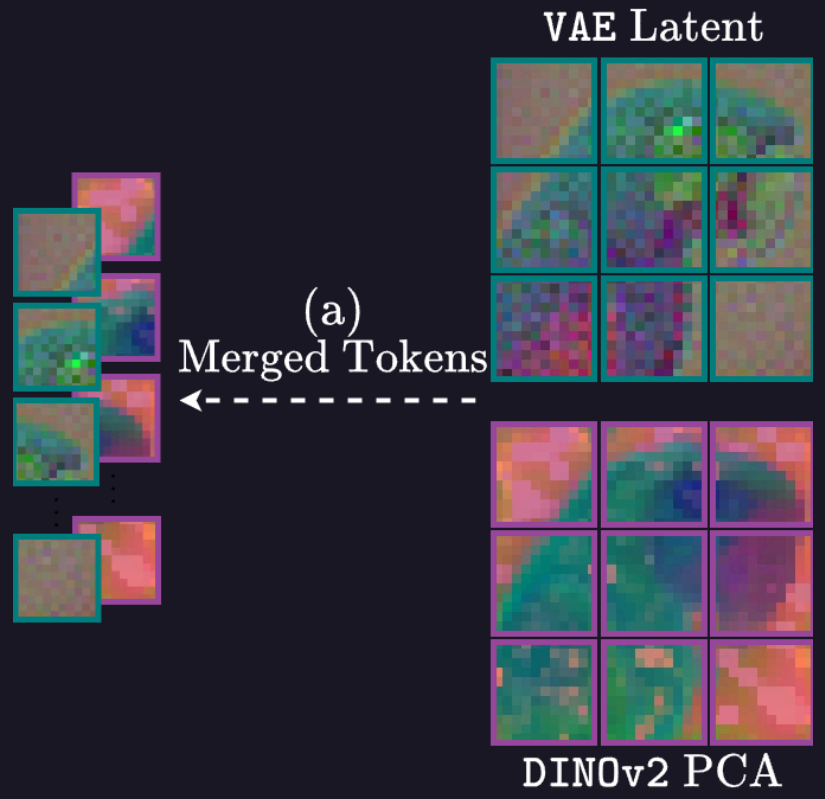
The tokens are summed channel-wise:

- $\mathbf{h}_t = \mathbf{x}_t \mathbf{W}^x_{\text{emb}} + \mathbf{z}_t \mathbf{W}^z_{\text{emb}} \in \mathbb{R}^{L \times C_d}$
- The transformer processes h_t to produce o_t
- $\boldsymbol{\epsilon}^x_\theta = \mathbf{o}_t \mathbf{W}^x_{\text{dec}}, \quad \boldsymbol{\epsilon}^z_\theta = \mathbf{o}_t \mathbf{W}^z_{\text{dec}}.$



Merged Tokens

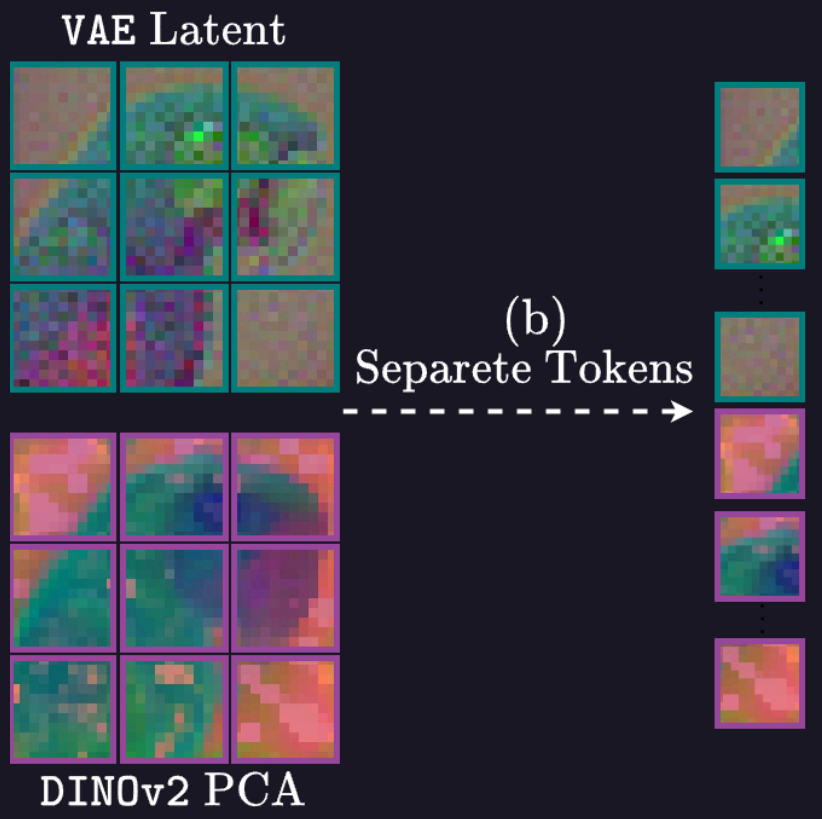
- **Early fusion.**
- **Maintains computational efficiency, as the token count remains unchanged.**



Separete Tokens

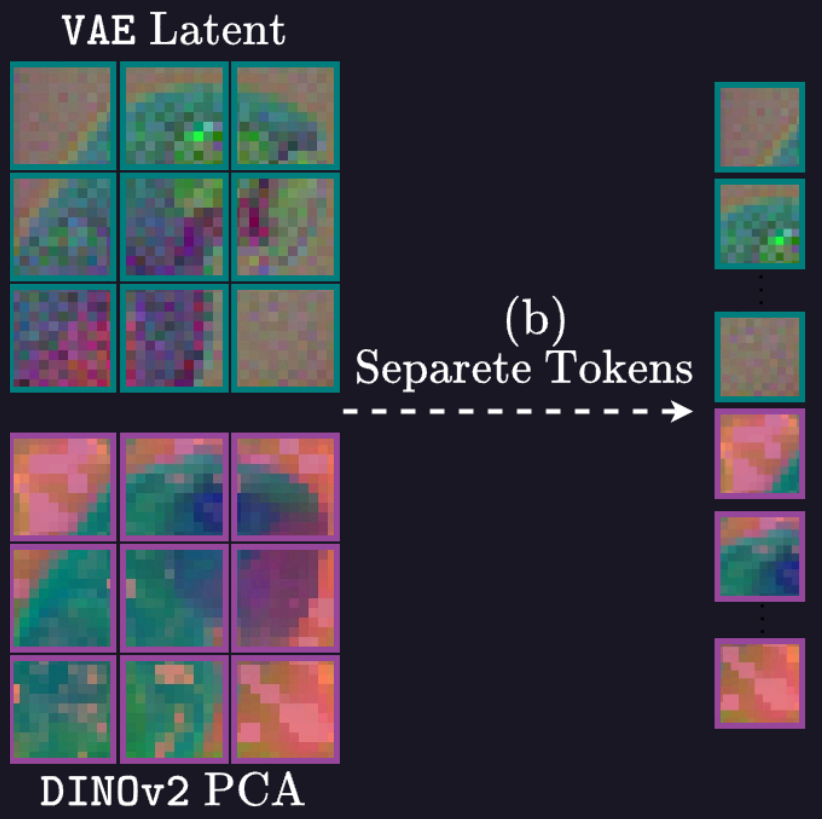
Tokens are concatenated along the sequence dimension:

- $\mathbf{h}_t = [\mathbf{x}_t\mathbf{W}^x_{\text{emb}}\,,\,\mathbf{z}_t\mathbf{W}^z_{\text{emb}}] \in \mathbb{R}^{2L\times C_d},$
- The transformer outputs separate representations $\mathbf{o}_t = [\mathbf{o}^x_t\,,\,\mathbf{o}^z_t]$
- $\boldsymbol{\epsilon}^x_\theta = \mathbf{o}^x_t\mathbf{W}^x_{\text{dec}}, \quad \boldsymbol{\epsilon}^z_\theta = \mathbf{o}^z_t\mathbf{W}^z_{\text{dec}}.$



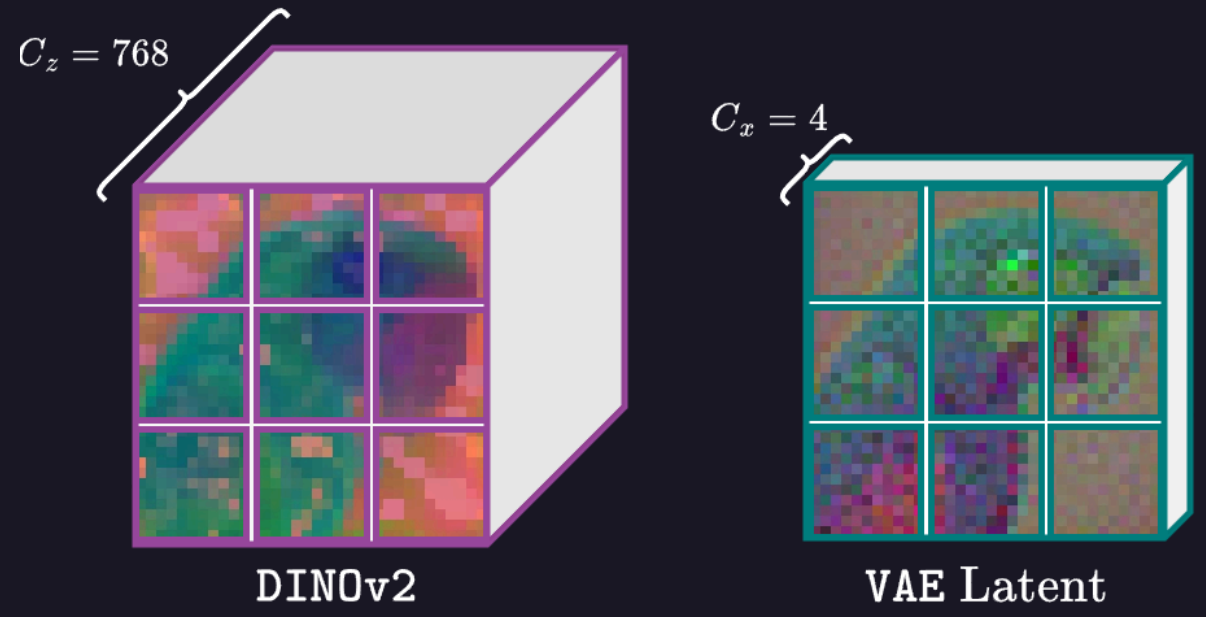
Separete Tokens

- Preserves modality-specific information.
- **Increased computation** due to increased token count ($\times 2$).



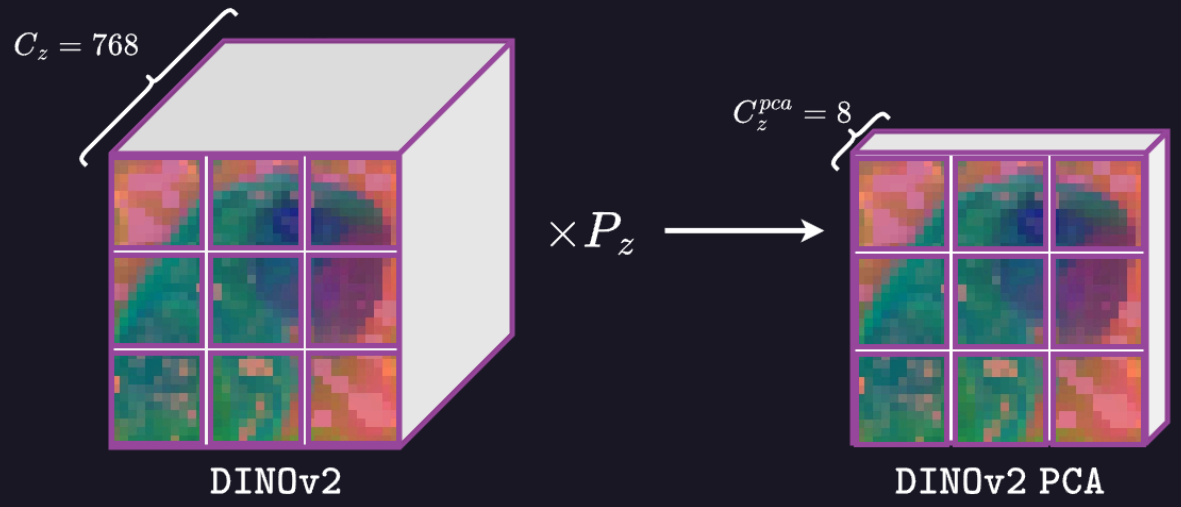
Dimensionality-Reduced Visual Representation

- The channels of `DINOv2` significantly exceeds that of `VAE latent`



Dimensionality-Reduced Visual Representation

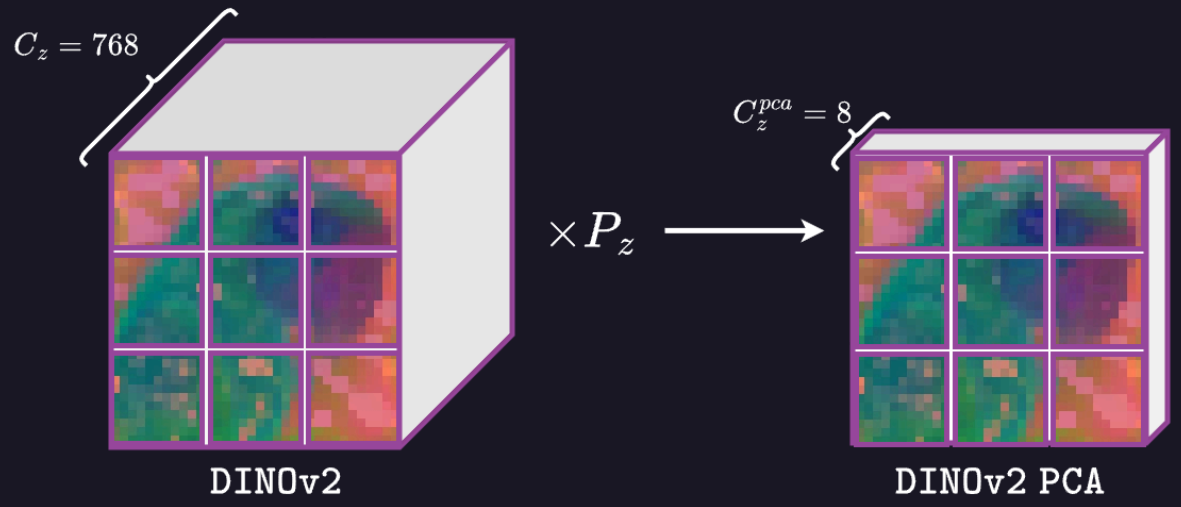
- Sample N features from training set and calculate a PCA projection matrix P_z



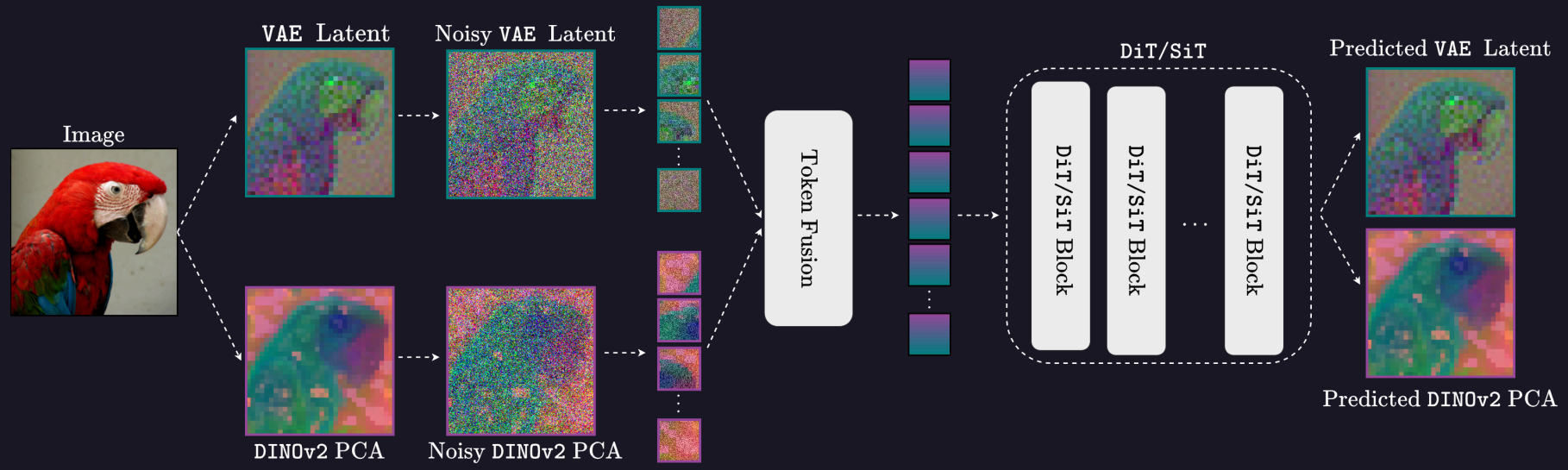
Dimensionality-Reduced Visual Representation

- Project to principal subspace:

$$z_{pca} = z \times P_z$$



Redi Pipeline



Represenation Guidance

- Inspired by Classifier-Free Guidance.
- During inference we modify the posterior distribution to:
  $\hat{p}_\theta(\mathbf{x}_t, \mathbf{z}_t) \propto p_\theta(\mathbf{x}_t) p(\mathbf{z}_t|\mathbf{x}_t)^{w_r}$
- Samples are pushed toward higher likelihood of the conditional distribution
  $p_\theta(\mathbf{z}_t|\mathbf{x}_t)$

Represenation Guidance

- Taking the log derivative yields the guided score function:

$$\nabla_{\mathbf{x}_t} \log \hat{p}_\theta(\mathbf{x}_t, \mathbf{z}_t) = \nabla_{\mathbf{x}_t} \log \left[p_\theta(\mathbf{x}_t) p(\mathbf{z}_t | \mathbf{x}_t)^{w_r} \right]$$

Represenation Guidance

- Taking the log derivative yields the guided score function:

$$\begin{aligned}\nabla_{\mathbf{x}_t} \log \hat{p}_\theta(\mathbf{x}_t, \mathbf{z}_t) &= \nabla_{\mathbf{x}_t} \log \left[p_\theta(\mathbf{x}_t) p(\mathbf{z}_t|\mathbf{x}_t)^{w_r} \right] \\ &= \nabla_{\mathbf{x}_t} \log p_\theta(\mathbf{x}_t) + w_r \big(\nabla_{\mathbf{x}_t} \log p_\theta(\mathbf{z}_t|\mathbf{x}_t) \big)\end{aligned}$$

Represenation Guidance

- Taking the log derivative yields the guided score function:

$$\begin{aligned}
\nabla_{\mathbf{x}_t} \log \hat{p}_\theta(\mathbf{x}_t, \mathbf{z}_t) &= \nabla_{\mathbf{x}_t} \log \left[p_\theta(\mathbf{x}_t) p(\mathbf{z}_t|\mathbf{x}_t)^{w_r} \right] \\
\nabla_{\mathbf{x}_t} \log \hat{p}_\theta(\mathbf{x}_t, \mathbf{z}_t) &= \nabla_{\mathbf{x}_t} \log p_\theta(\mathbf{x}_t) + w_r \big(\nabla_{\mathbf{x}_t} \log p_\theta(\mathbf{z}_t|\mathbf{x}_t) \big) \\
&= \nabla_{\mathbf{x}_t} \log p_\theta(\mathbf{x}_t) + w_r \big(\nabla_{\mathbf{x}_t} \log p_\theta(\mathbf{x}_t, \mathbf{z}_t) - \nabla_{\mathbf{x}_t} \log p_\theta(\mathbf{x}_t) \big).
\end{aligned}$$

Represenation Guidance

$$\nabla_{\mathbf{x}_t} \log \hat{p}_\theta(\mathbf{x}_t, \mathbf{z}_t) = \nabla_{\mathbf{x}_t} \log p_\theta(\mathbf{x}_t) + w_r \big(\nabla_{\mathbf{x}_t} \log p_\theta(\mathbf{x}_t, \mathbf{z}_t) - \nabla_{\mathbf{x}_t} \log p_\theta(\mathbf{x}_t) \big).$$

- From the equivalance between denoisers and scores we have:
  $\hat{\boldsymbol{\epsilon}}_\theta(\mathbf{x}_t, \mathbf{z}_t, t) = \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) + w_r \left(\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, \mathbf{z}_t, t) - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \right).$

Represenation Guidance

$$\hat{\boldsymbol{\epsilon}}_\theta(\mathbf{x}_t, \mathbf{z}_t, t) = \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) + w_r \left(\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, \mathbf{z}_t, t) - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right).$$

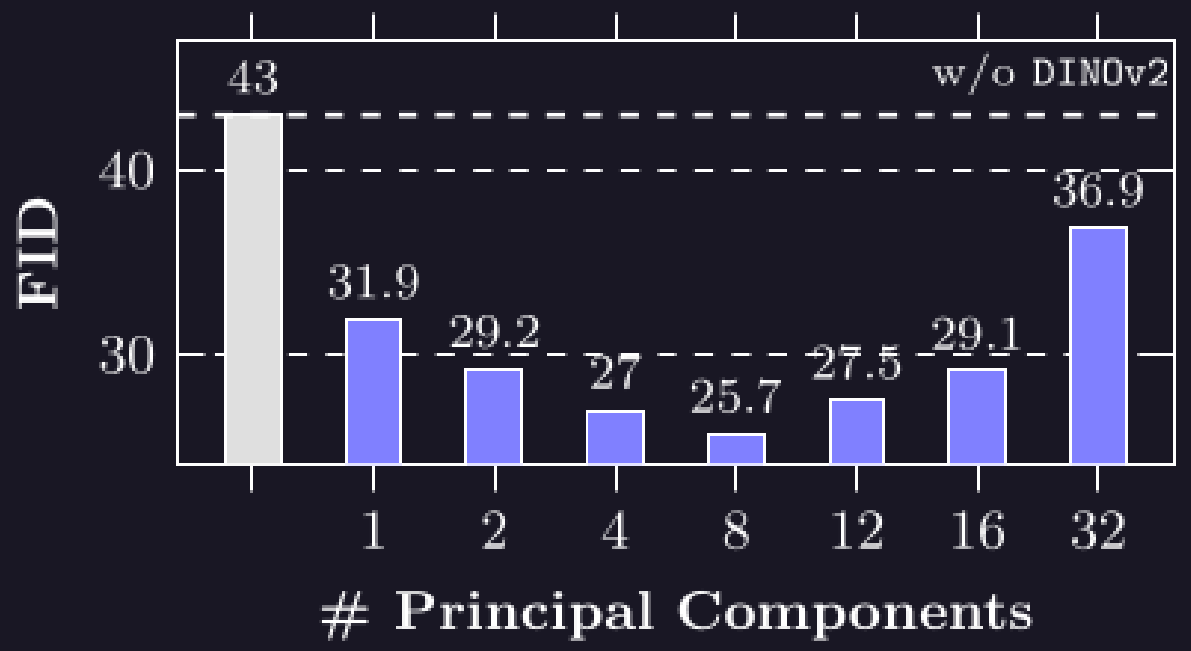
- We train both $\boldsymbol{e_\theta}(\mathbf{x}_t, \mathbf{z}_t, t)$ and $\boldsymbol{e_\theta}(\mathbf{x}_t, t)$ jointly.
- With probability $p_{drop} = 0.2$:
  - Zero out $\mathbf{z}_t$ (setting $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) = \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, \mathbf{0}, t)$)
  - Disable the visual representation denoising loss.

Experimental Results 🔬 📊

Ablation

Dimentionality reduction

- Intermediate subspace ($r = 8$)
- Retains sufficient expressivity to guide generation
- Does not dominate model capacity.



Ablation

Merged Tokens vs. Separate Tokens.

Model	#Tokens	Throughput ↑	FID↓
DiT-B/2	256	4.52	43.5
w/ ReDi (MR)	256	4.51	25.7
w/ ReDi (SP)	512	2.26	24.7

Merged Tokens vs. Separate Tokens.

Merged Tokens

- ~Same training and infernce compute

Model	#Tokens	Throughput ↑	FID↓
DiT-B/2	256	4.52	43.5
w/ ReDi (MR)	256	4.51	25.7
w/ ReDi (SP)	512	2.26	24.7

Merged Tokens vs. Separate Tokens.

Merged Tokens

- ~Same training and infernce compute

Separete Tokens

- ~$\times 2$ training and infernce compute
- Slightly better performance

Model	#Tokens	Throughput ↑	FID↓
DiT-B/2	256	4.52	43.5
w/ ReDi (MR)	256	4.51	25.7
w/ ReDi (SP)	512	2.26	24.7

Main benchmark

Conditional Generation

- $\sim \times 6$ faster than REPA
- Converged performance :
  - REPA : 5.9 FID
  - ReDi : 3.3 FID

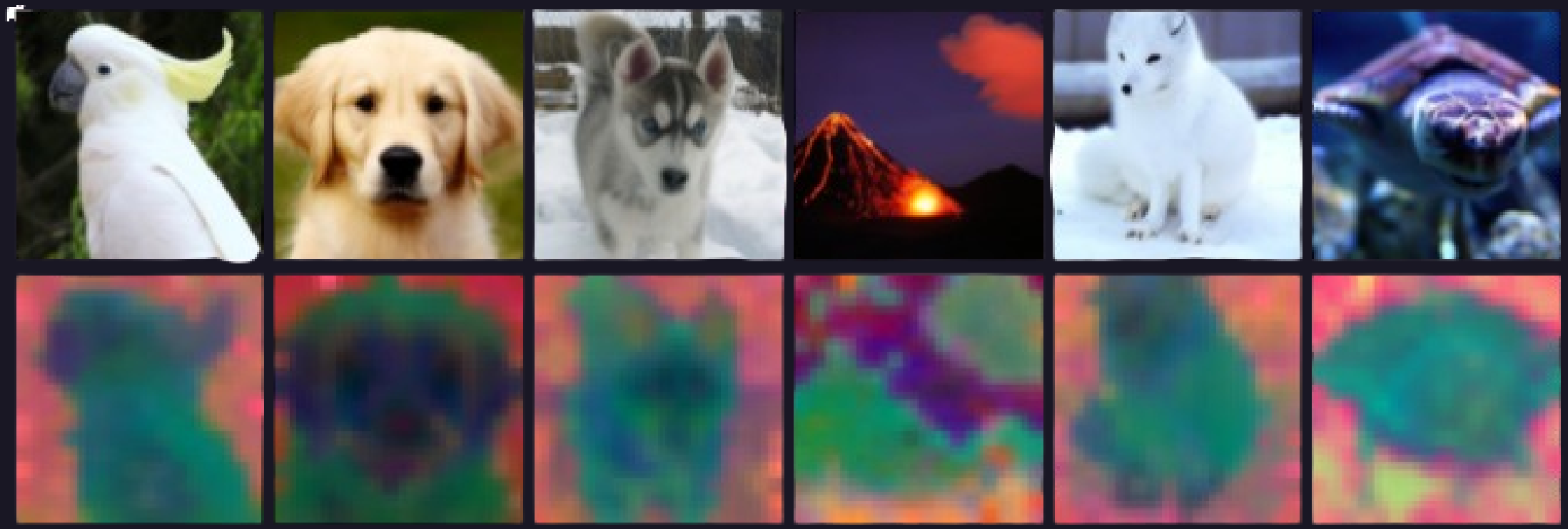
DiT-L/2	458M	400K	23.2
w/ REPA	458M	400K	15.6
w/ ReDi (ours)	458M	400K	10.5
SiT-L/2	458M	400K	18.5
w/ REPA	458M	400K	9.7
w/ ReDi (ours)	458M	400K	9.4
DiT-XL/2	675M	400K	19.5
w/ REPA	675M	400K	12.3
DiT-XL/2	675M	7M	9.6
w/ REPA	675M	850K	9.6
w/ ReDi (ours)	675M	400K	8.7
SiT-XL/2	675M	400K	17.2
w/ REPA	675M	400K	7.9
w/ ReDi (ours)	675M	400K	7.5
SiT-XL/2	675M	7M	8.3
w/ REPA	675M	4M	5.9
w/ ReDi (ours)	675M	700K	5.6
w/ ReDi (ours)	675M	4M	3.3

Main benchmark

Conditional Generation w/ CFG

Model	Epochs	FID↓	sFID↓	IS↑	Pre.↑	Rec.↑
Autoregressive Models						
VAR	350	1.80	-	365.4	0.83	0.57
MagViTv2	1080	1.78	-	319.4	0.83	0.57
MAR	800	1.55	-	303.7	0.81	0.62
Latent Diffusion Models						
LDM	200	3.60	-	247.7	0.87	0.48
U-ViT-H/2	240	2.29	5.68	263.9	0.82	0.57
DiT-XL/2	1400	2.27	4.60	278.2	0.83	0.57
MaskDiT	1600	2.28	5.67	276.6	0.80	0.61
SD-DiT	480	3.23	-	-	-	-
SiT-XL/2	1400	2.06	4.50	270.3	0.82	0.59
FasterDiT	400	2.03	4.63	264.0	0.81	0.60
MDT	1300	1.79	4.57	283.0	0.81	0.61
Leveraging Visual Representations						
REPA	800	1.80	4.50	284.0	0.81	0.61
ReDi (ours)	350	1.72	4.68	278.7	0.77	0.63
ReDi (ours)	800	1.61	4.66	295.1	0.78	0.64

Selected Samples

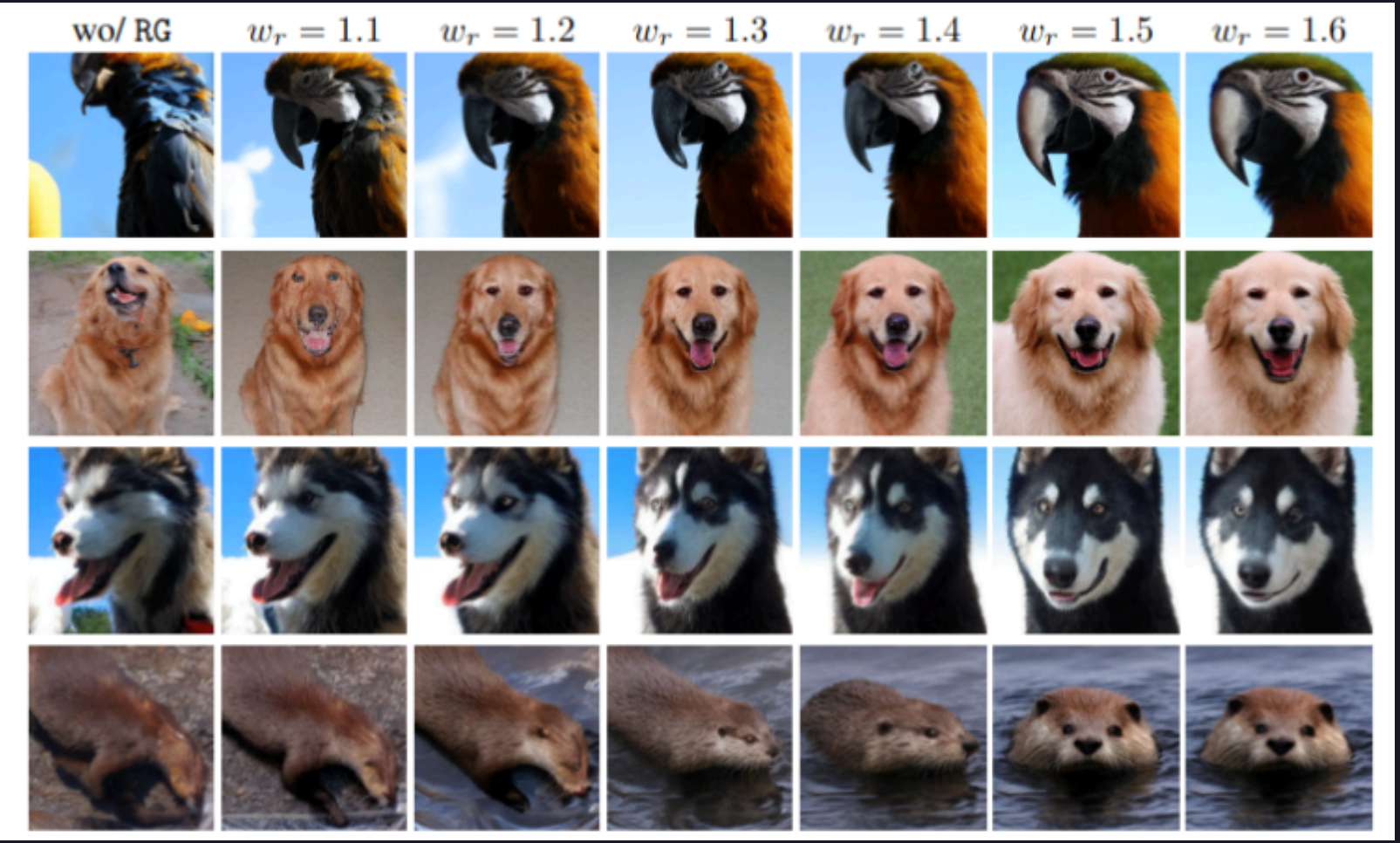


Unconditional Generation

- Representation Guidance (RG) is very usefull for unconditional generation
- ReDi significantly closes the gap between conditional and unconditional generation

Model	#Params	FID↓
DiT-B/2 (conditional)	130M	43.5
DiT-B/2	130M	69.3
w/ ReDi (ours)	130M	51.7
w/ ReDi+RG (ours)	130M	47.3
DiT-XL/2 (conditional)	675M	19.5
DiT-XL/2	675M	44.6
w/ ReDi (ours)	675M	25.1
w/ ReDi+RG (ours)	675M	22.6

Representation Guidance



Possible Future Research Directions

- Multiple Representations (e.g. `DINOv2` and `CLIP`)
- Better compression approach for `DINOv2`
- Leverege the generated representations

Thank You 😀