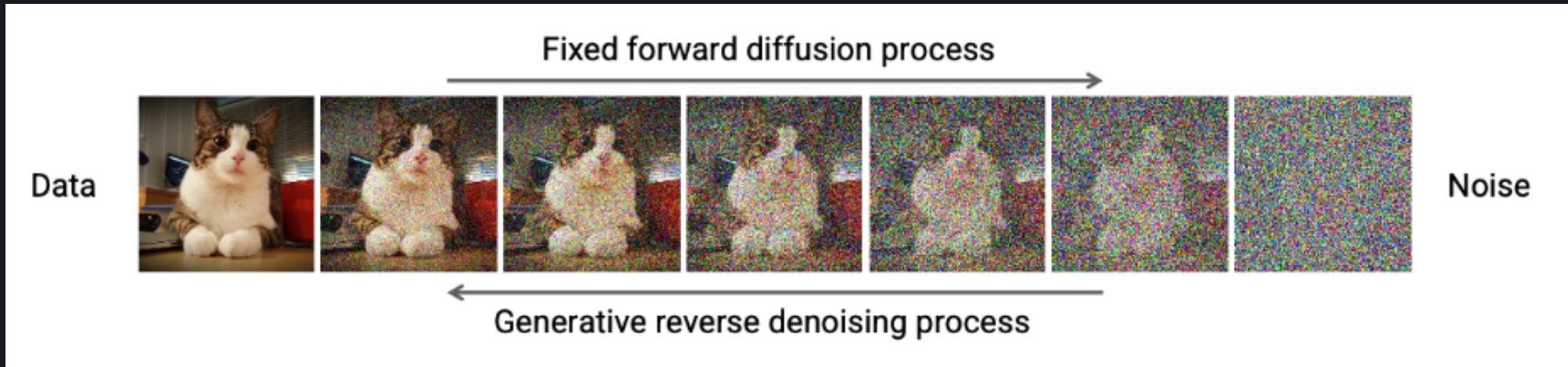


Boosting Generative Image Modeling via Joint Image Feature Synthesis

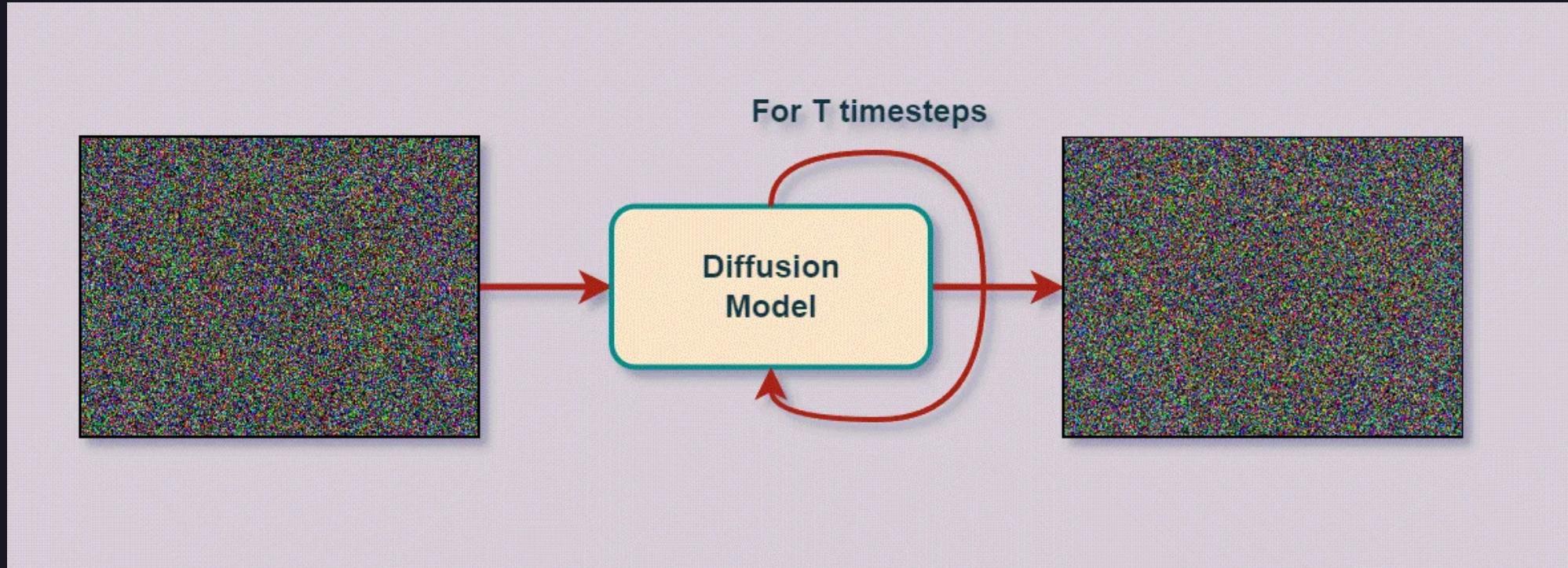
Theodoros Kouzelis, Eystathis Karypidis, Ioannis Kakogeorgiou,
Spyros Gidaris, Nikos Komodakis

Diffusion / Flow Models (in a nutshell)

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

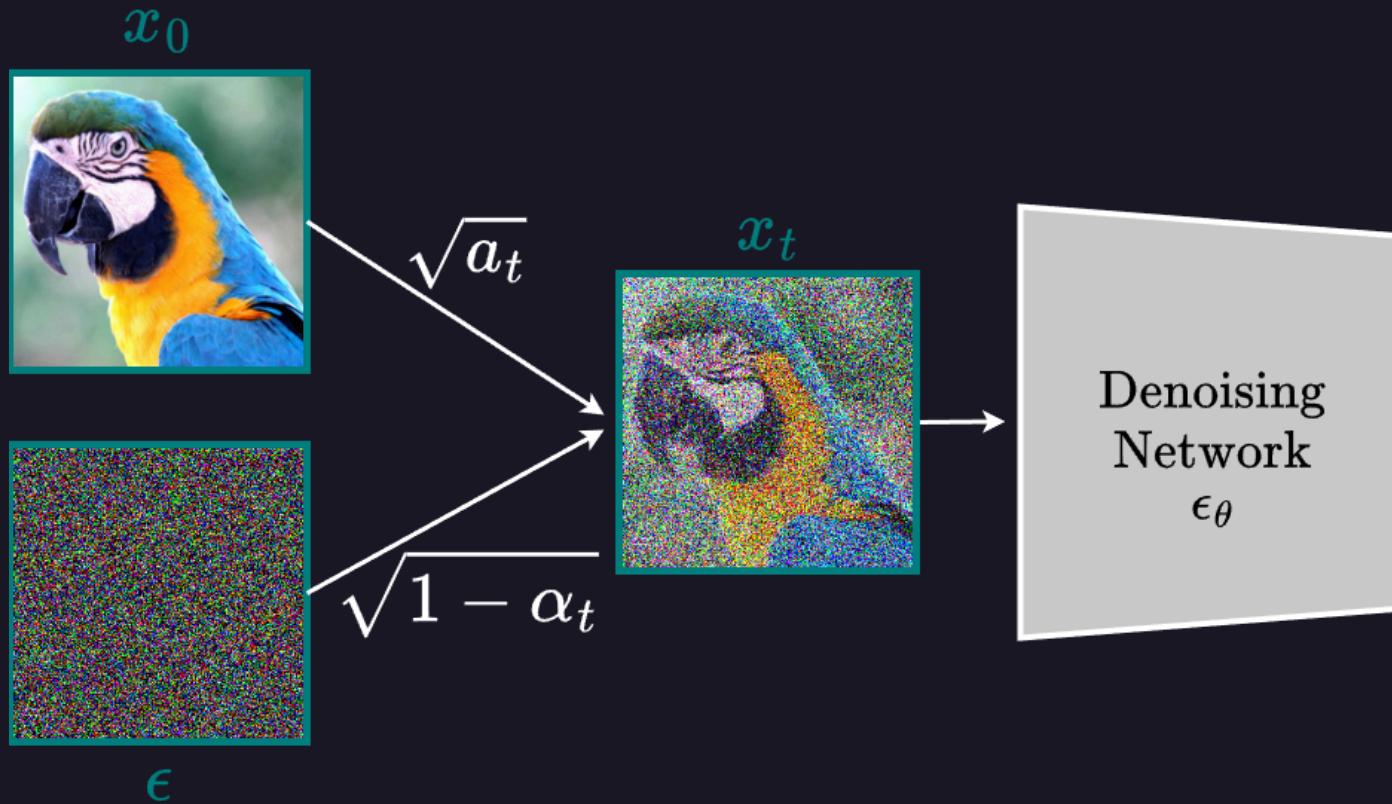


Diffusion / Flow Models



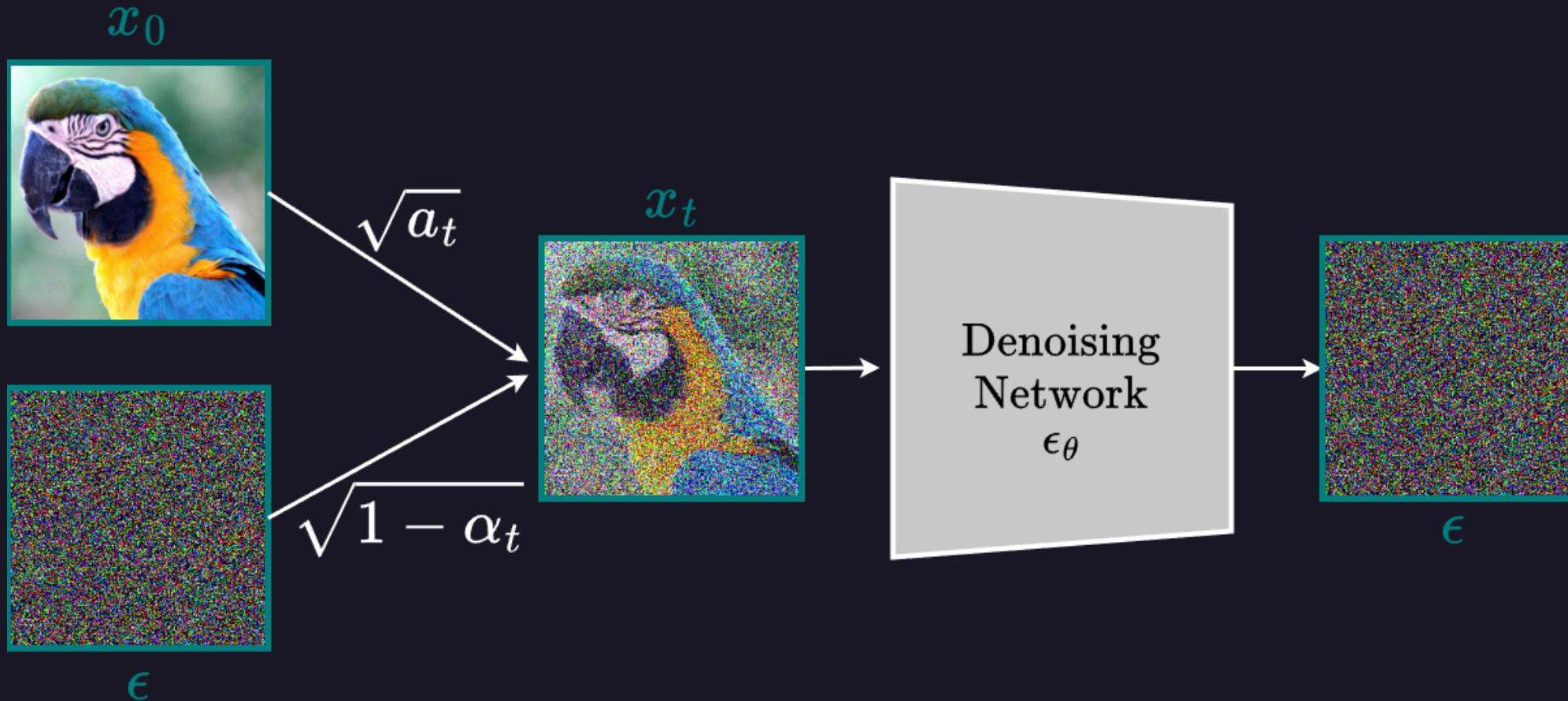
Denoising Objective (DDPM)

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



Denoising Objective (DDPM)

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

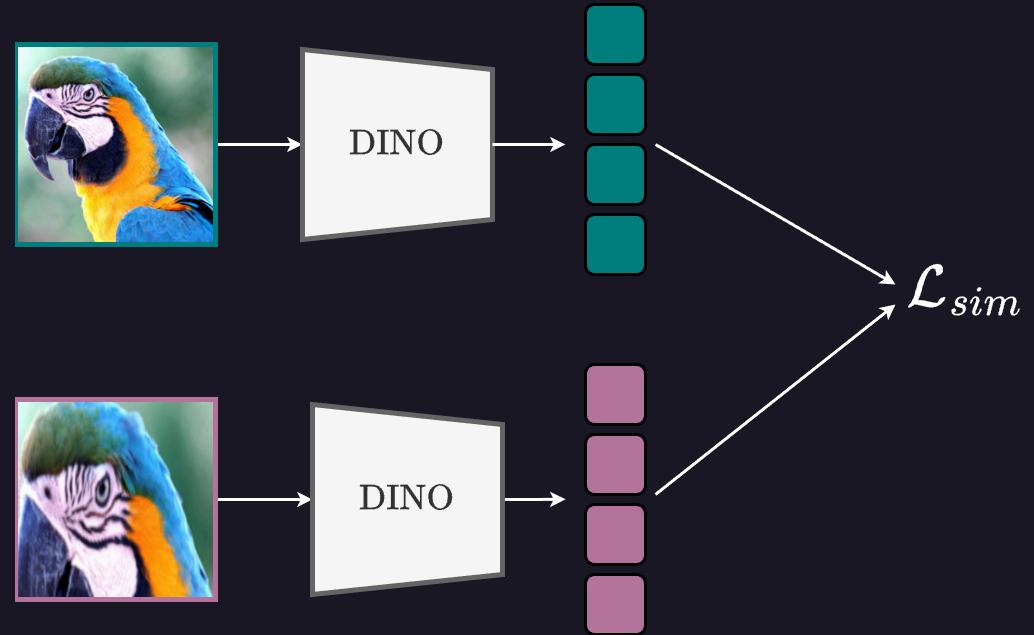


The problem with the Denoising Objective

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

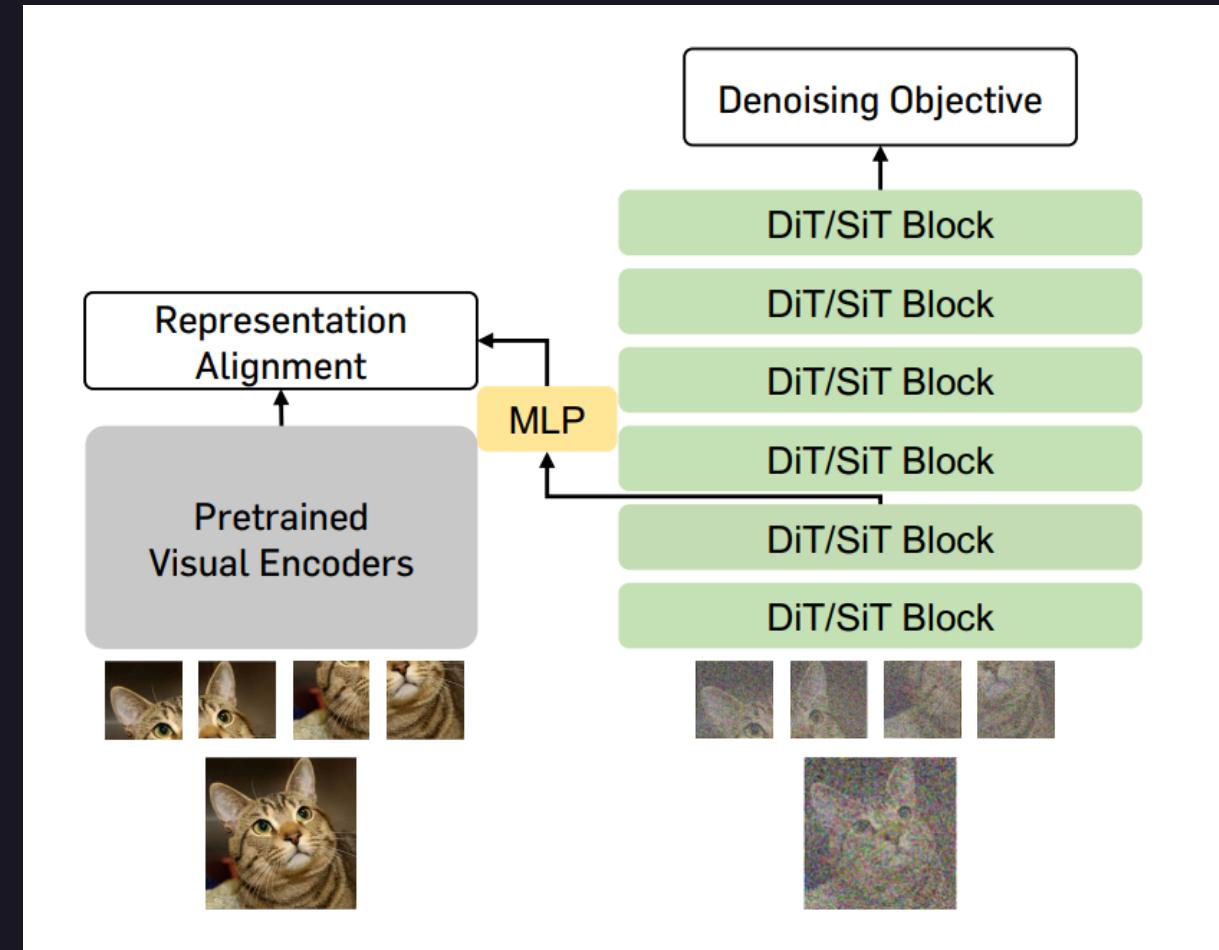
DINO (in a nutshell)

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



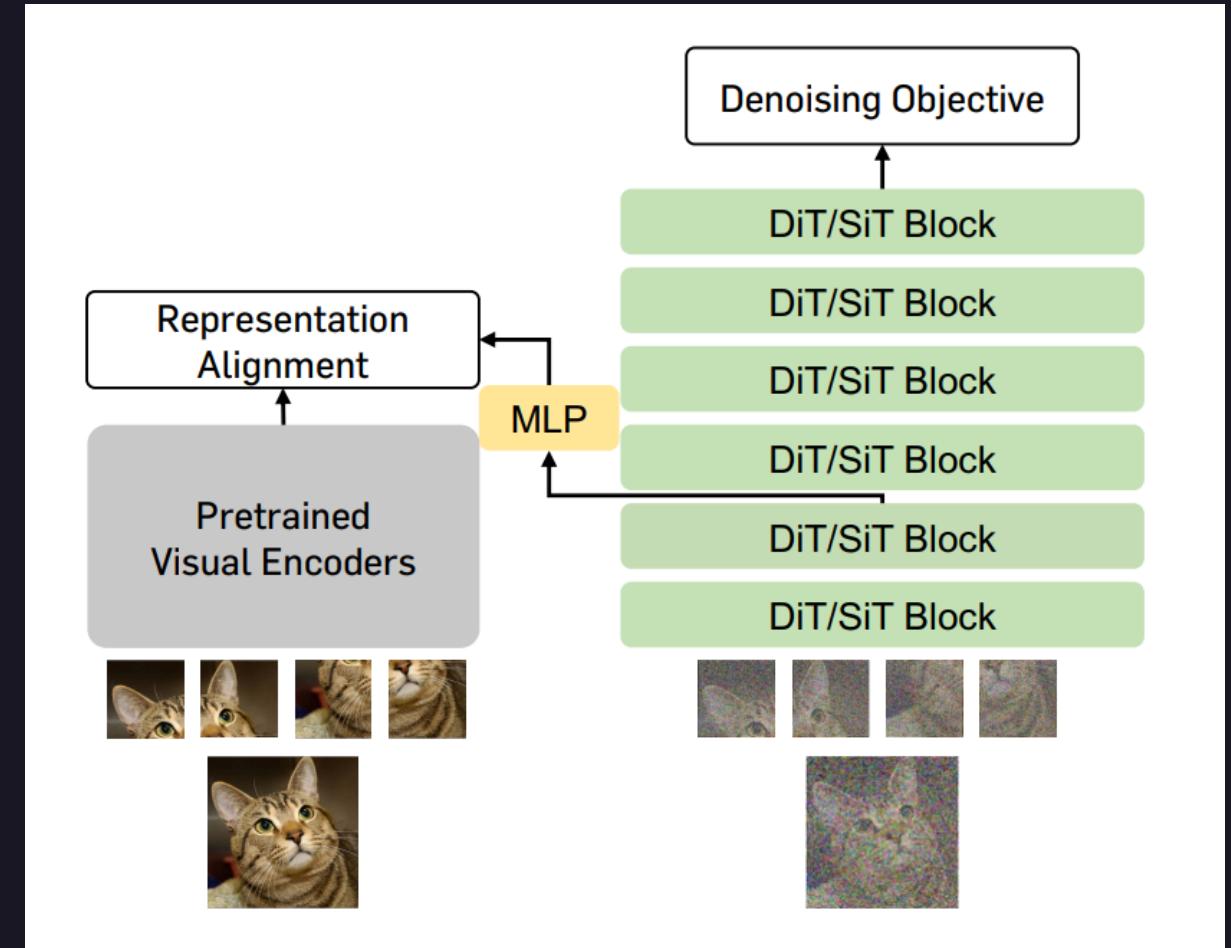
Representation Alignment

- Align features with the representations of powerful visual encoders.



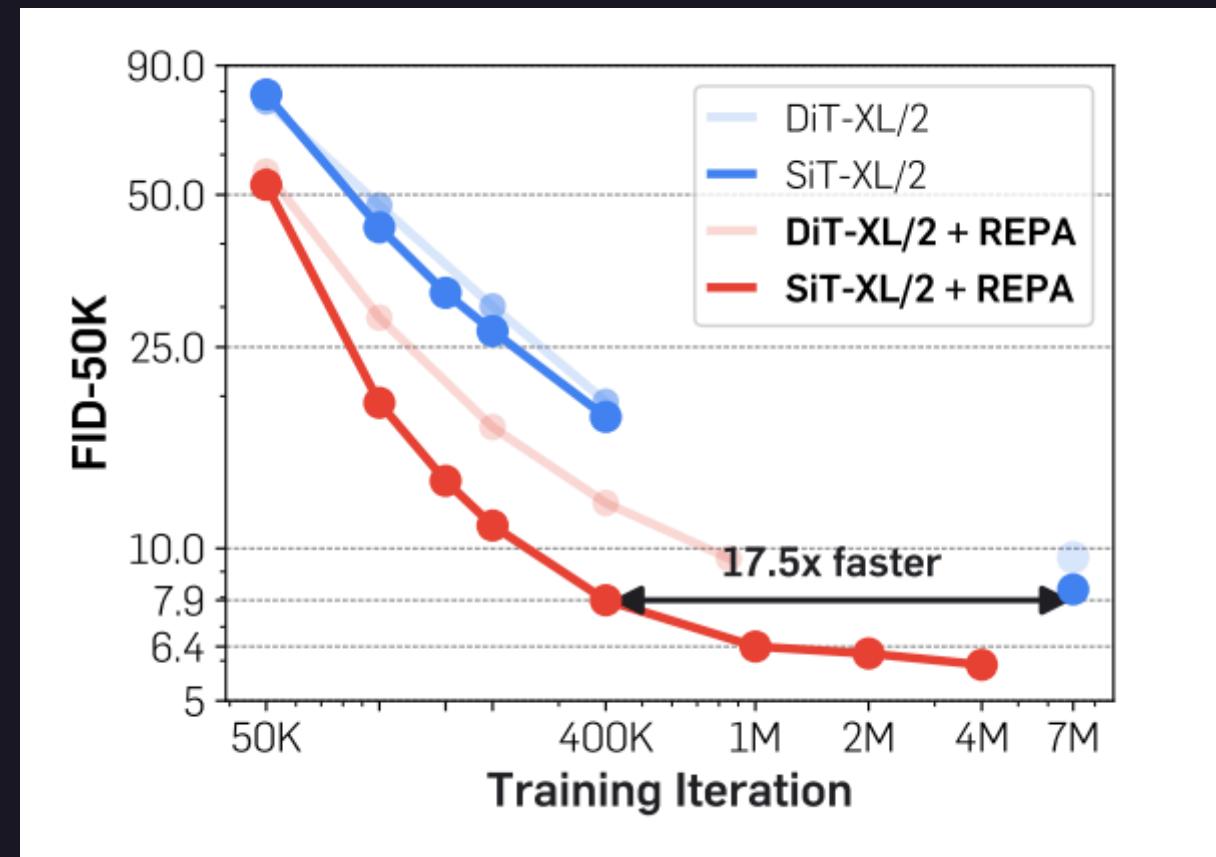
Representation Alignment

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



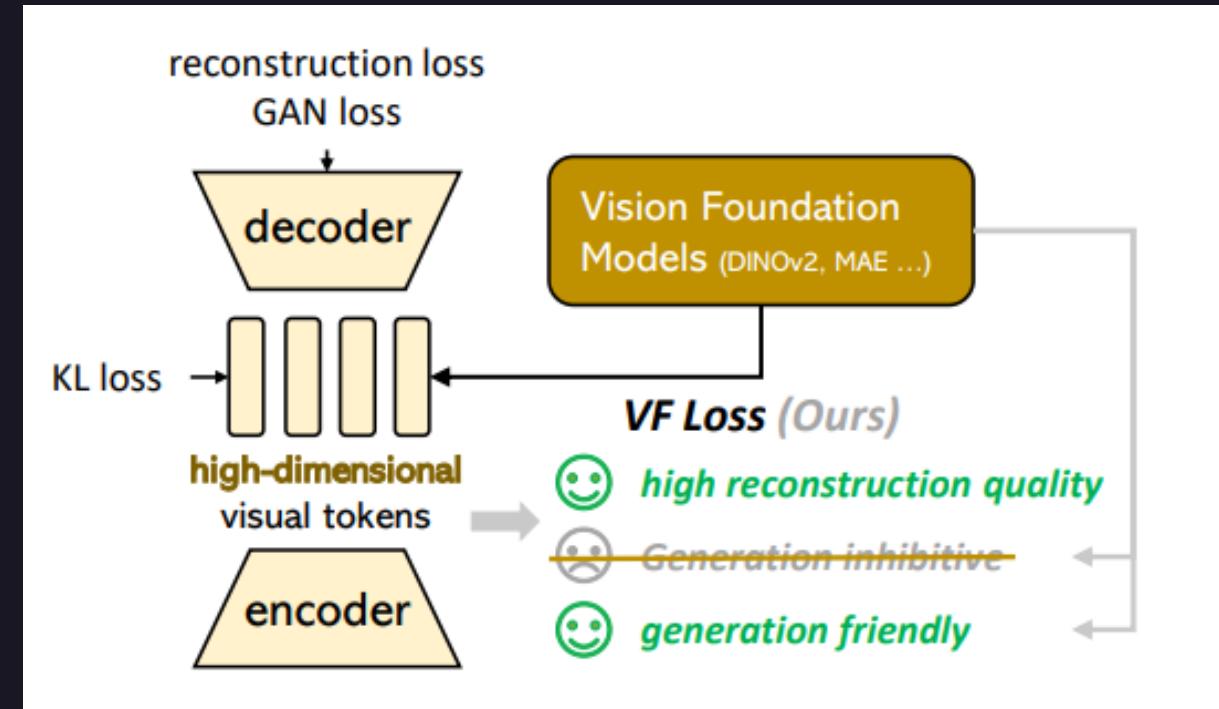
Representation Alignment

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



Representation Alignment for VAE Latents

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

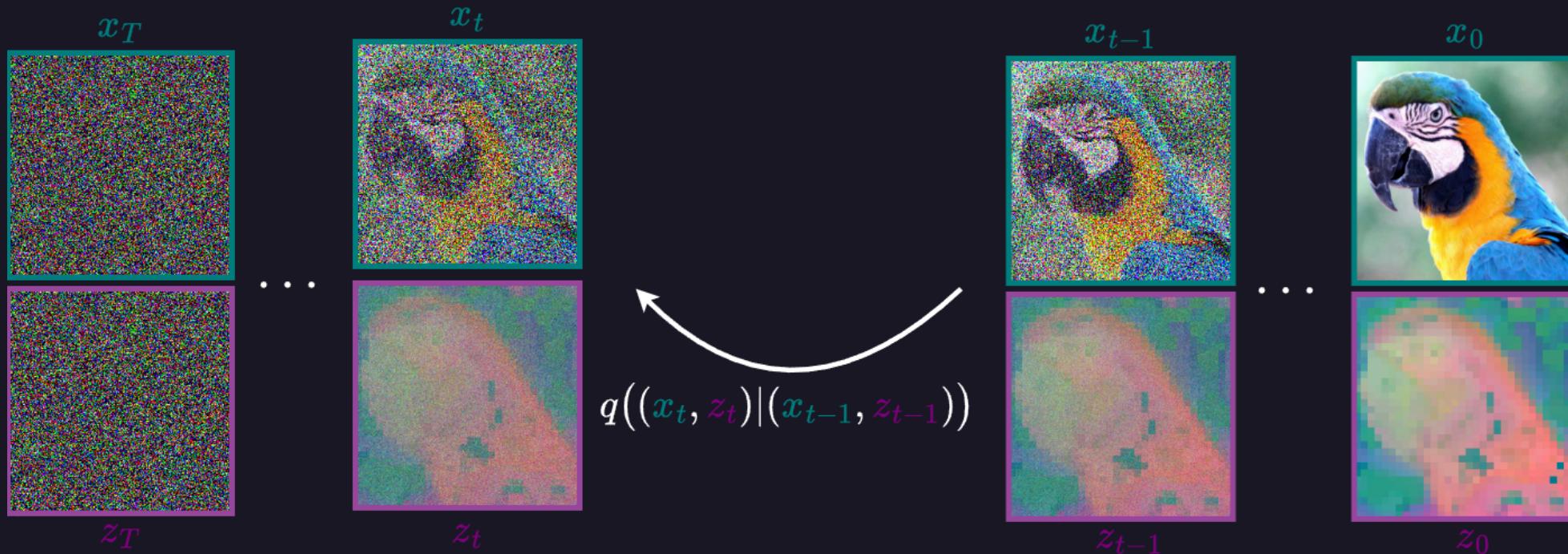


ReDi: Joint image-feature Synthesis



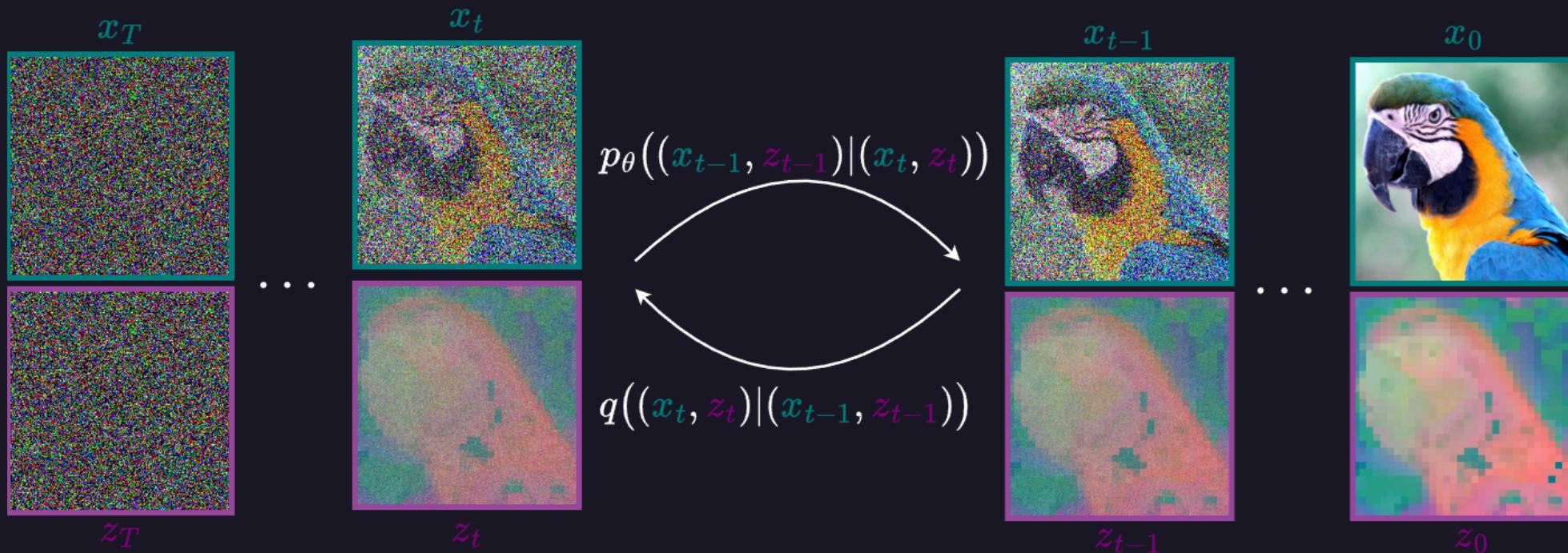
ReDi: Joint image-feature Synthesis

Joint Forward Process

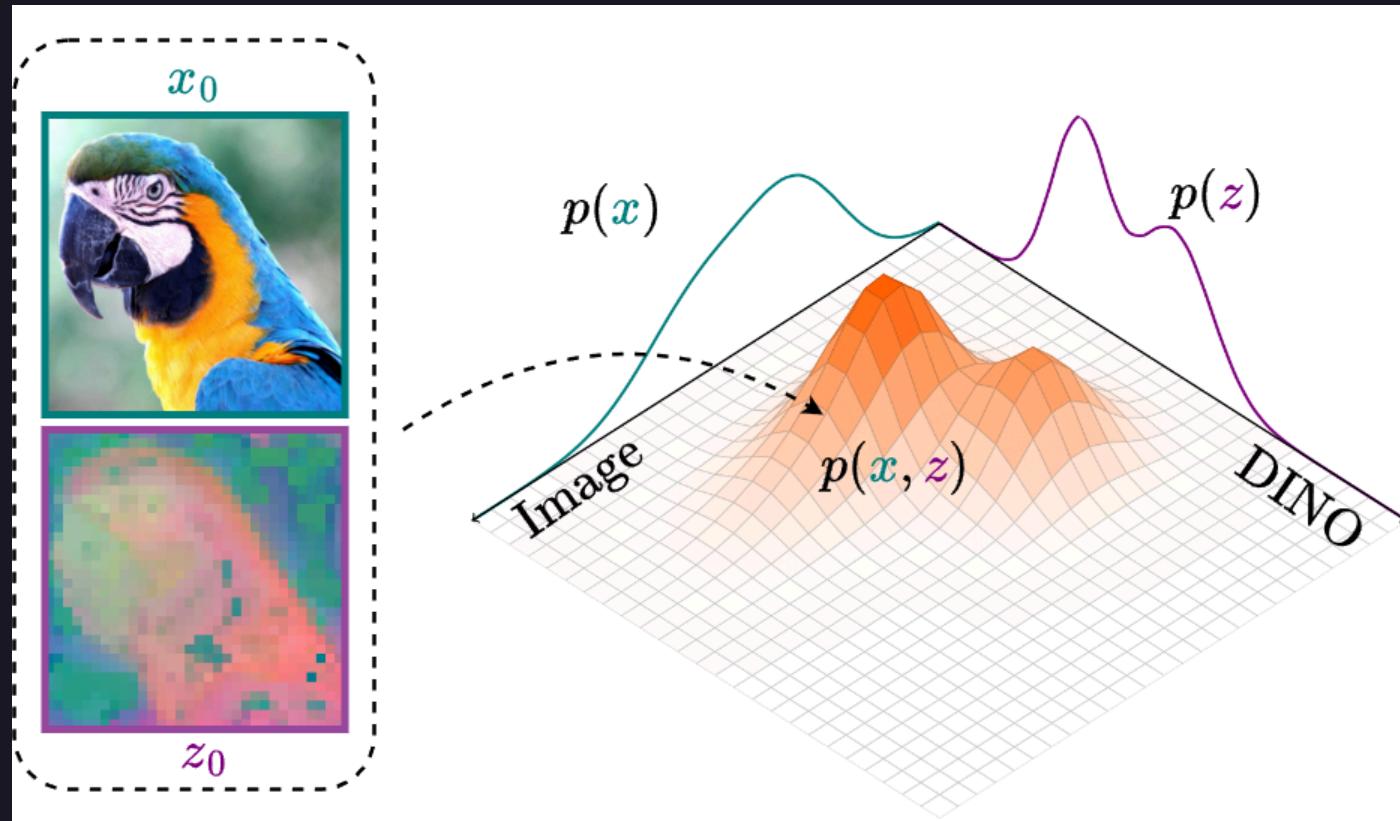


ReDi: Joint image-feature Synthesis

Joint Reverse Process

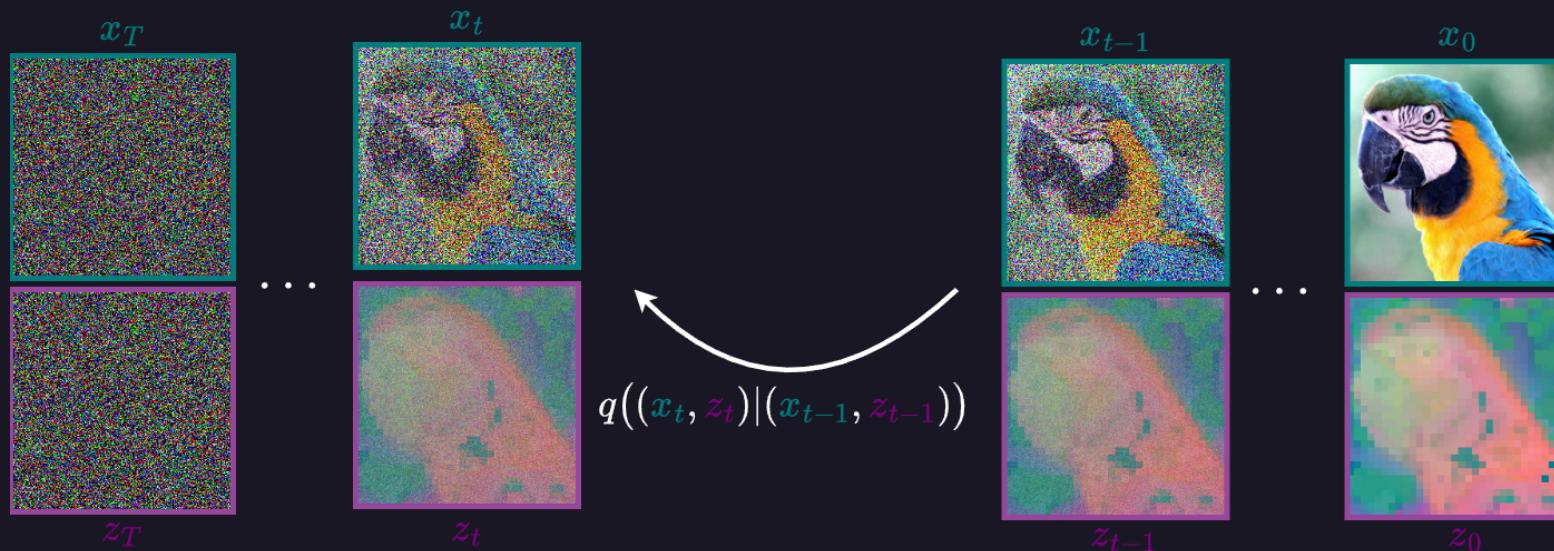


ReDi: Joint image-feature Synthesis



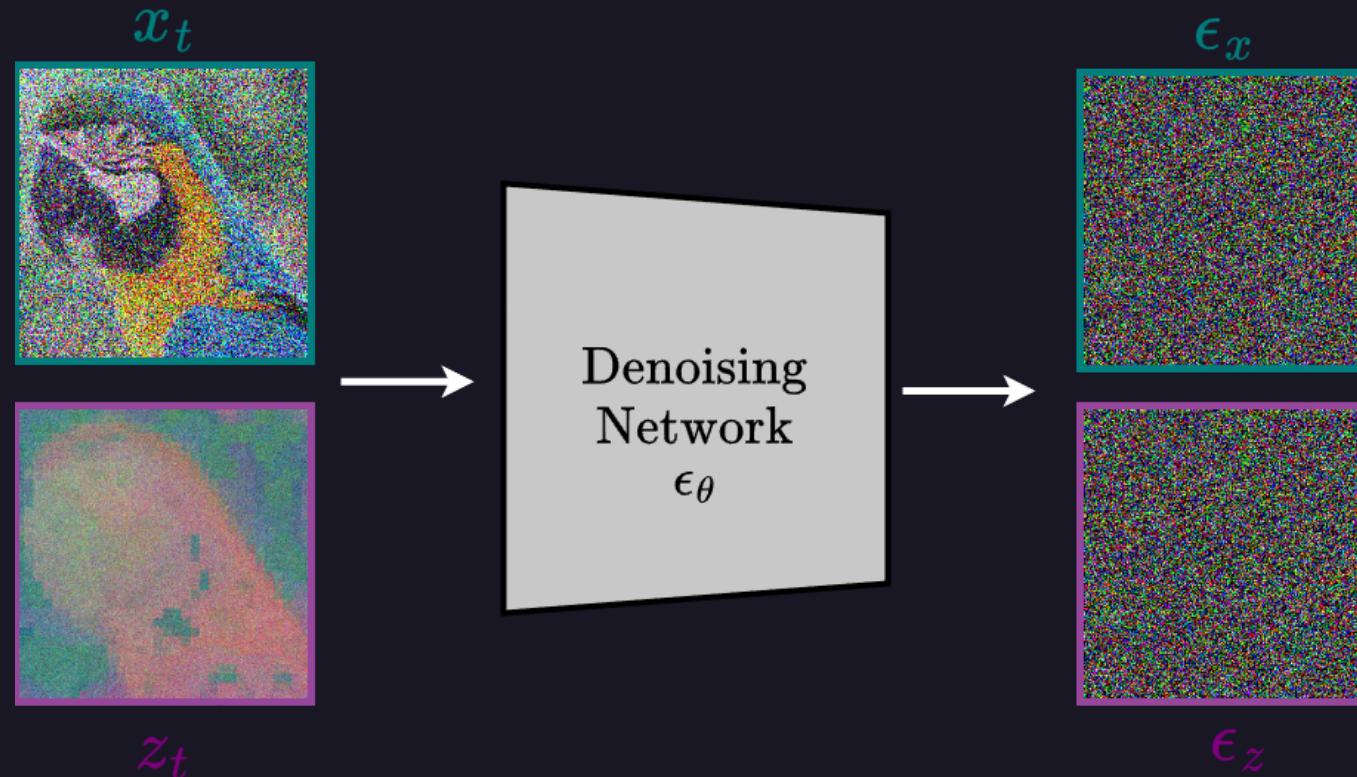
Joint Forward Process

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



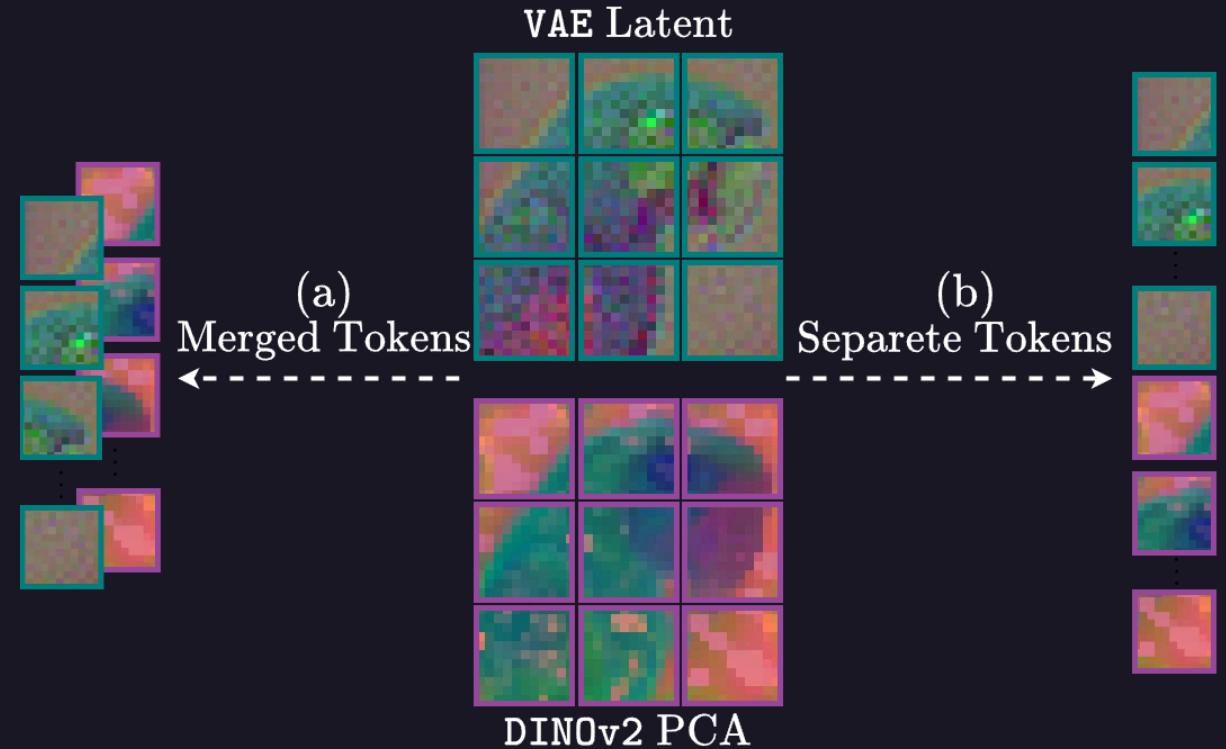
Joint Denoising Objective

- $\mathcal{L}_{joint} = \mathbb{E}_{\mathbf{x}_0, \mathbf{z}_0} \left[\|\epsilon_{\theta}^x(\mathbf{x}_t, \mathbf{z}_t, t) - \epsilon_x\|_2^2 + \|\epsilon_{\theta}^z(\mathbf{x}_t, \mathbf{z}_t, t) - \epsilon_z\|_2^2 \right]$



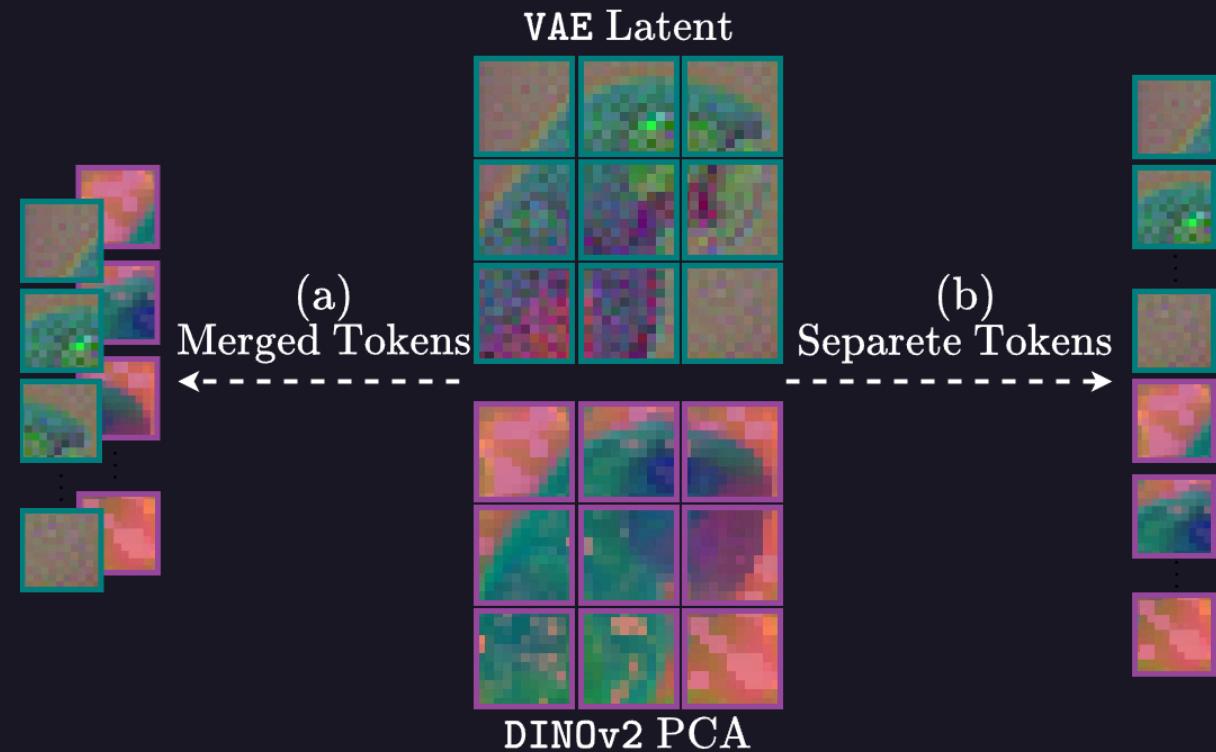
Fusing the two modelities in a single input sequence

- Merged Tokens
- Separate Tokens



Fusing the two modelities in a single input sequence

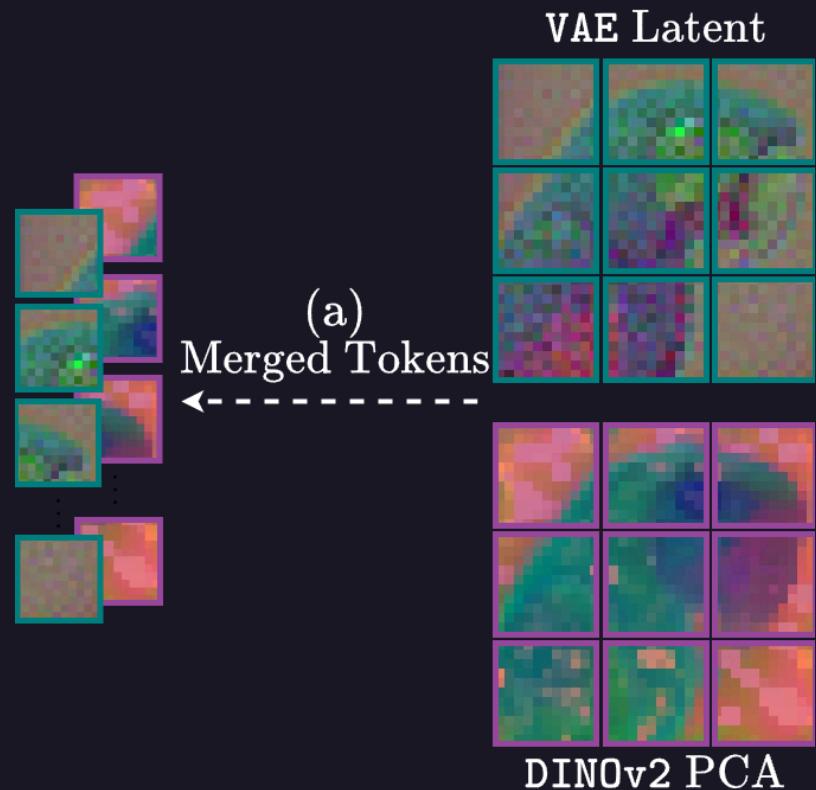
- Modality-specific **embedding** matrices:
 - $\mathbf{W}_{\text{emp}}^x \in \mathbb{R}^{C_x \times C_d}$
 - $\mathbf{W}_{\text{emp}}^z \in \mathbb{R}^{C_z \times C_d}$
- Modality-specific **projection** matrices:
 - $\mathbf{W}_{\text{dec}}^x \in \mathbb{R}^{C_d \times C_x}$
 - $\mathbf{W}_{\text{dec}}^z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



Merged Tokens

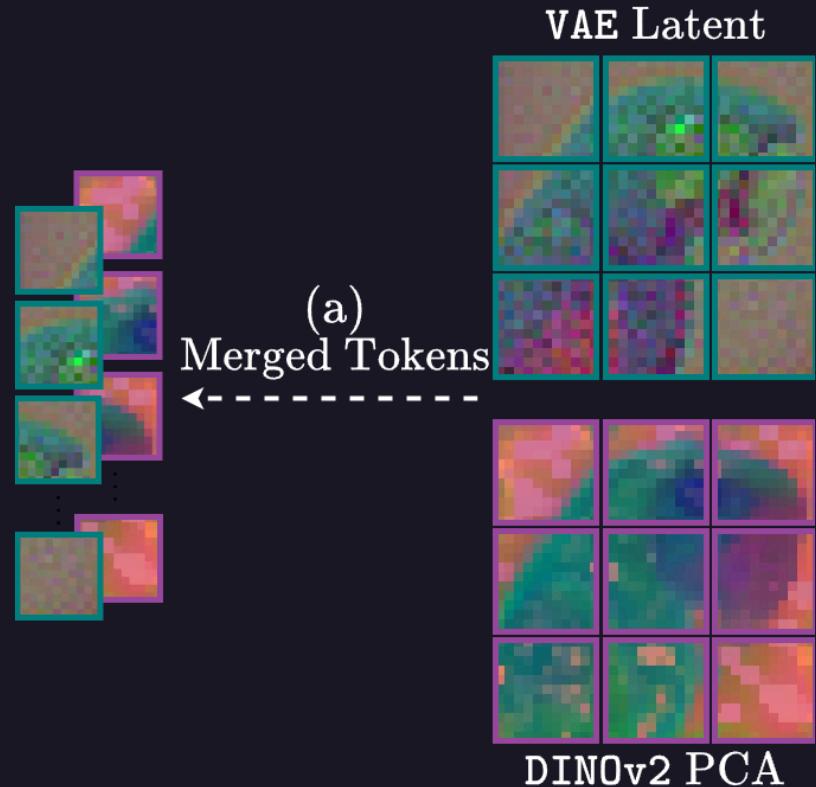
The tokens are summed channel-wise:

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



Merged Tokens

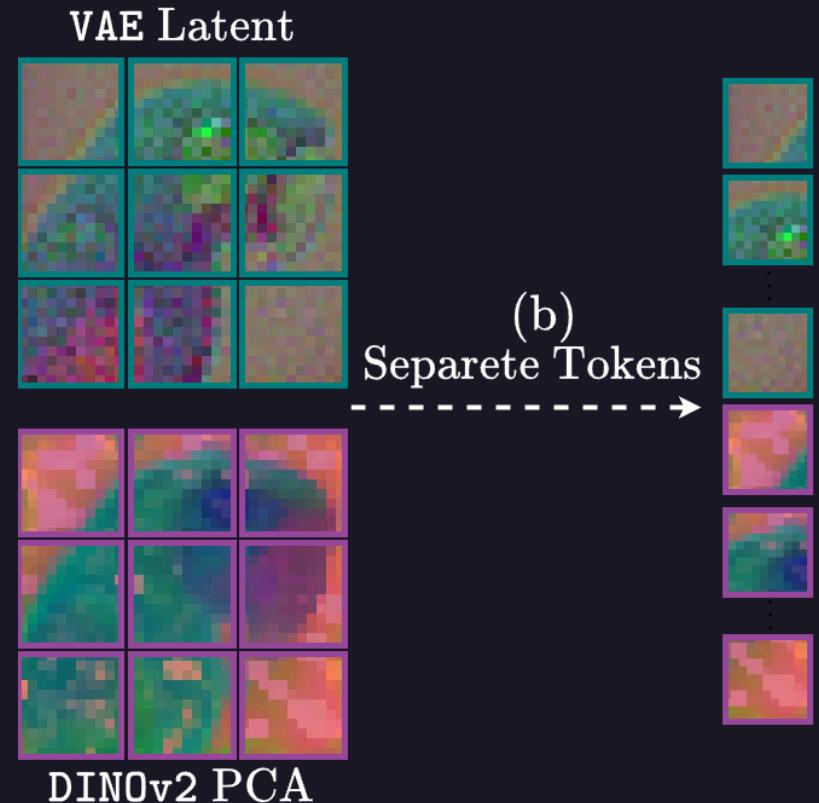
- Early fusion.
- Maintains computational efficiency, as the token count remains unchanged.



Separate Tokens

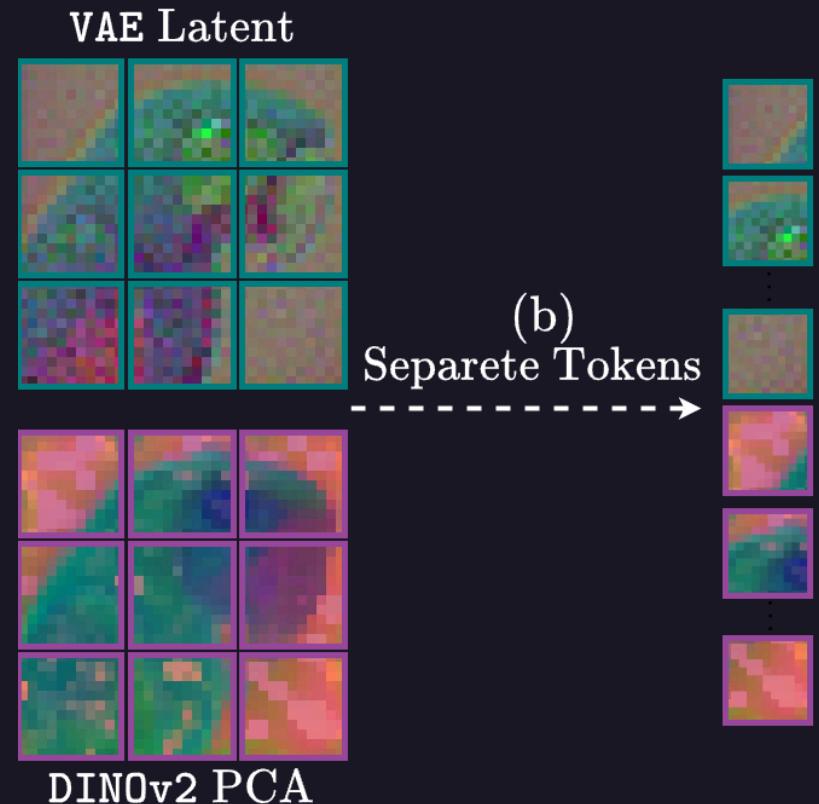
Tokens are concatenated along the sequence dimension:

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



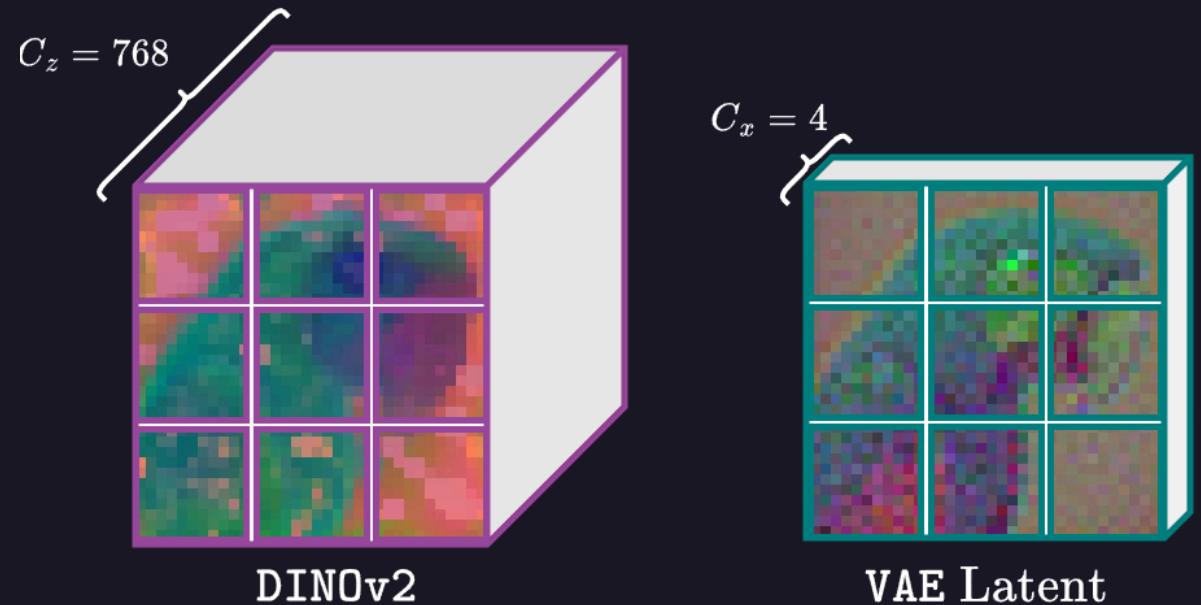
Separate Tokens

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



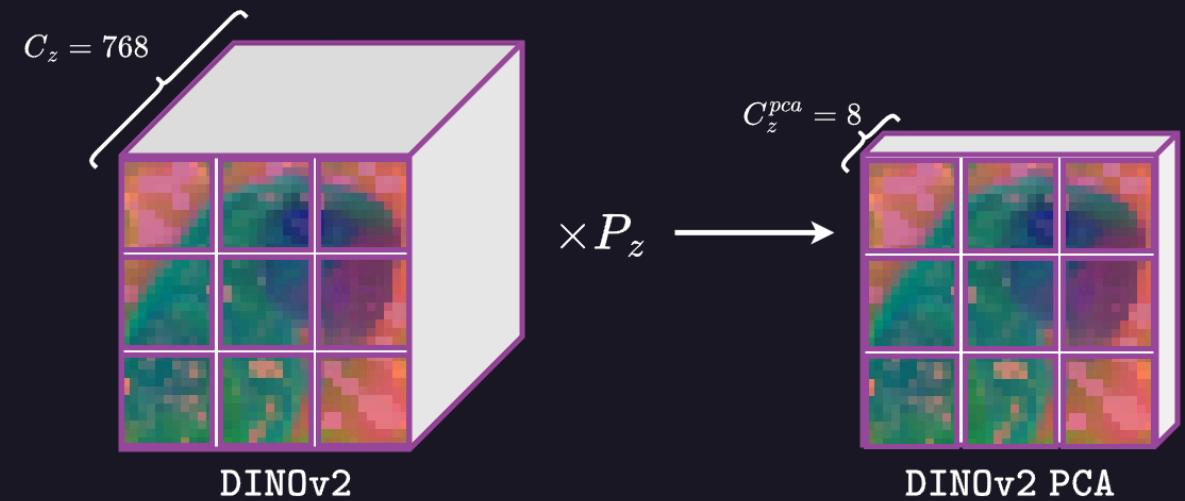
Dimensionality-Reduced Visual Representation

- The channels of DINOv2 significantly exceeds that of VAE latent



Dimensionality-Reduced Visual Representation

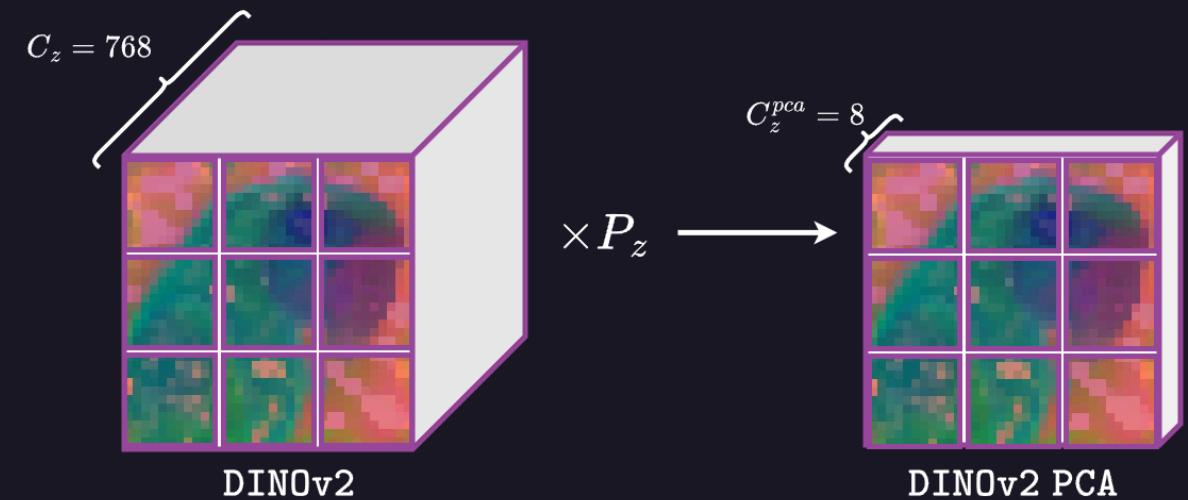
- Sample N features from training set and calculate a PCA projection matrix P_z



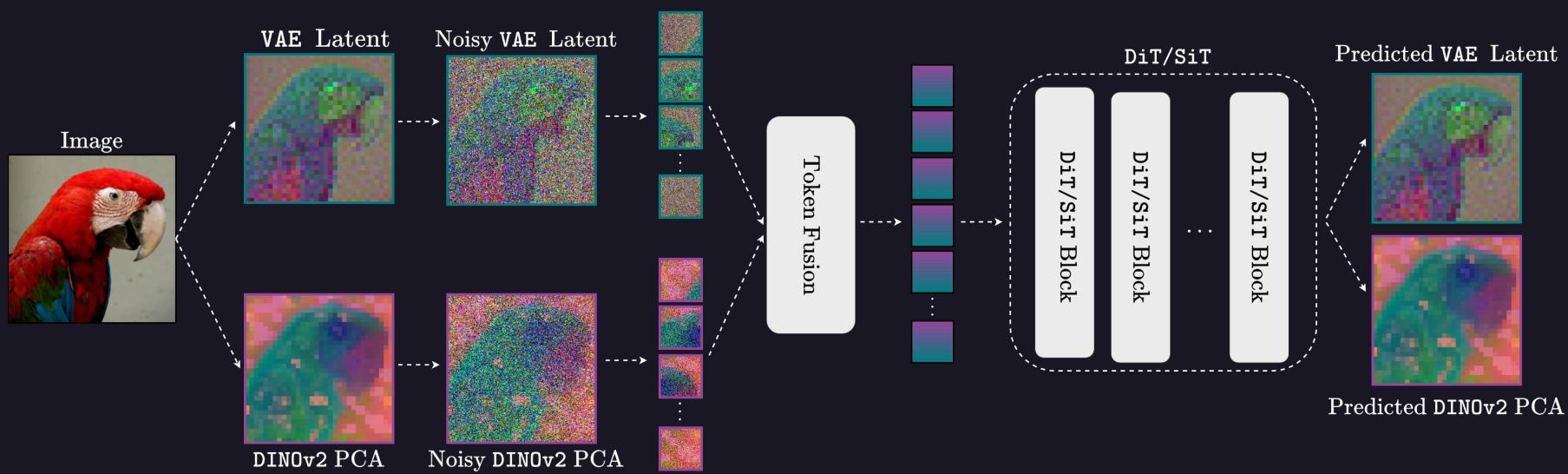
Dimensionality-Reduced Visual Representation

- 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:

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

Representation Guidance

- Taking the log derivative yields the guided score function:

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

Represenation Guidance

- Taking the log derivative yields the guided score function:

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

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

Representation Guidance

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

- From the equivalence between denoisers and scores we have:

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

Represenation Guidance

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

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

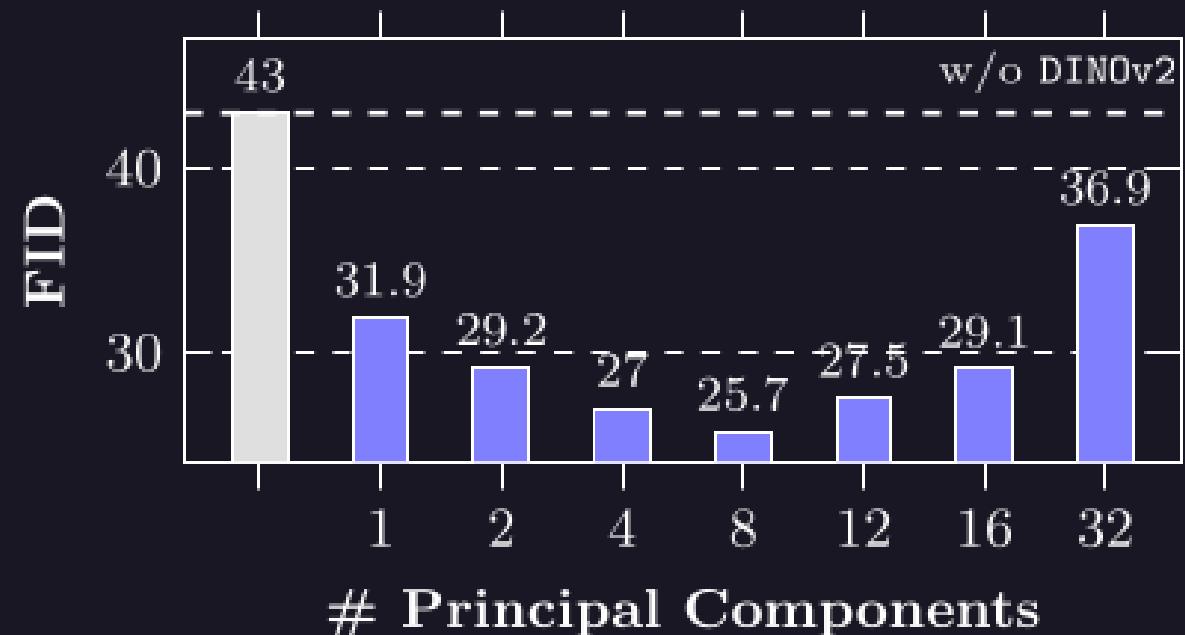
Experimental Results



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 inference 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 inference compute

Separate Tokens

- ~ $\times 2$ training and inference 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

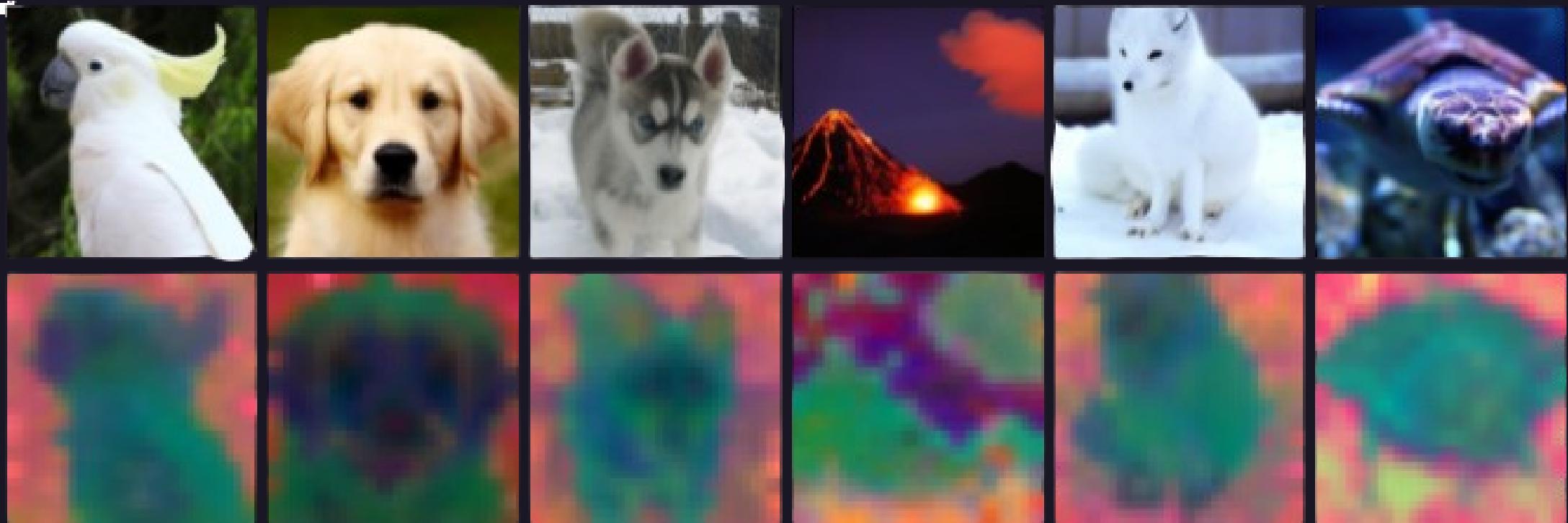
DiT-L/2	458M	400K	23.2
w/ REPA	458M	400K	15.6
w/ ReDi (ours)	458M	400K	10.5
SiT-L/2	458M	400K	18.5
w/ REPA	458M	400K	9.7
w/ ReDi (ours)	458M	400K	9.4
DiT-XL/2	675M	400K	19.5
w/ REPA	675M	400K	12.3
DiT-XL/2	675M	7M	9.6
w/ REPA	675M	850K	9.6
w/ ReDi (ours)	675M	400K	8.7
SiT-XL/2	675M	400K	17.2
w/ REPA	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.↑
<i>Autoregressive Models</i>						
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
<i>Latent Diffusion Models</i>						
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
<i>Leveraging Visual Representations</i>						
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

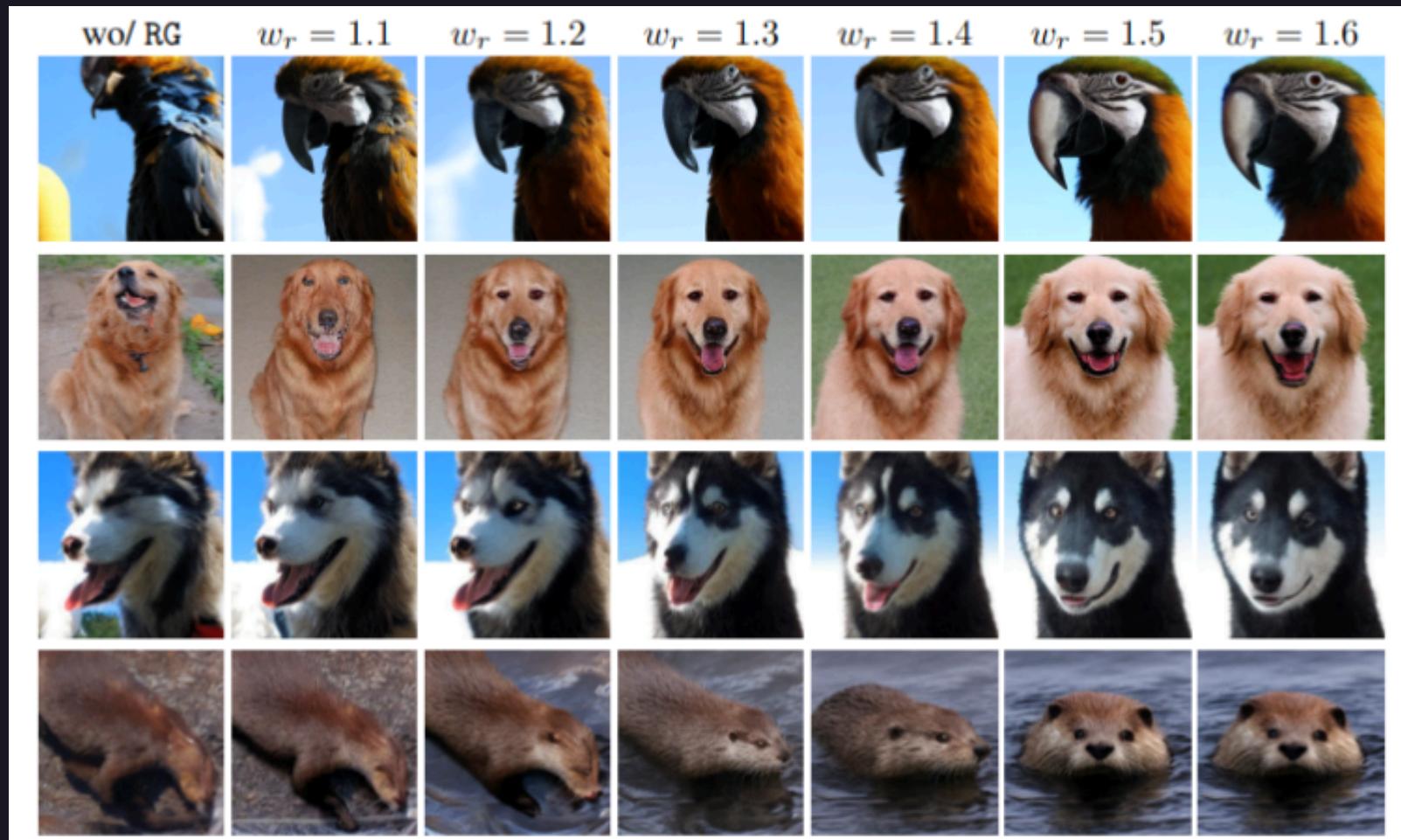


Unconditional Generation

- 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
- Leverage the generated representations

Thank You 😊