Harnessing Low-Dimensionality in Diffusion Models

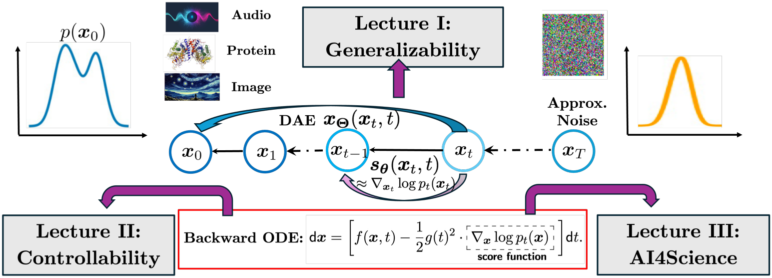
Lecture II: Controllability & Training with Synthetic Data

Qing Qu

September 22, 2025

EECS, University of Michigan

Lecture Schedule



We focus on the **mathematical foundations** of diffusion models through **low-dim structures** and their scientific applications:

- Introduction of Diffusion Models
- Lecture I: **Generalization** of Learning Diffusion Models
- Lecture II: **Controllability** of Diffusion Models
- Lecture III: From Theory to **Scientific Applications**

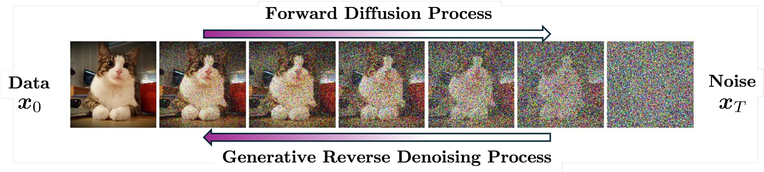
Major References

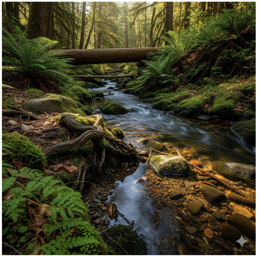
1. Lianghe Shi, Meng Wu, Huijie Zhang, Zekai Zhang, Molei Tao, Qing Qu. A Closer Look at Model Collapse: From a Generalization-to-Memorization Perspective. *Neural Information Processing Systems (NeurIPS'25)*, 2025. (**spotlight, top 3.2%**)
2. Siyi Chen*, Huijie Zhang*, Minzhe Guo, Yifu Lu, Peng Wang, Qing Qu. Exploring Low-Dimensional Subspaces in Diffusion Models for Controllable Image Editing. *Neural Information Processing Systems (NeurIPS'24)*, 2024.
3. Wenda Li, Huijie Zhang, Qing Qu. Shallow Diffuse: Robust and Invisible Watermarking through Low-Dimensional Subspaces in Diffusion Models. *NeurIPS*, 2025 (**spotlight, top 3.2 %**).
4. Xiang Li, Rongrong Wang, Qing Qu. Towards Understanding the Mechanisms of Classifier-Free Guidance. *Neural Information Processing Systems (NeurIPS'25)*, 2025. (**spotlight, top 3.2%**)

Outline

Training with Synthetic Data & Model Collapse

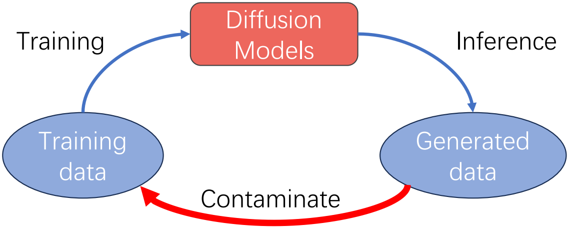
Modern Generative AI - Diffusion Models





Diffusion models can generate high-quality images that are indistinguishable from real ones, even to humans.

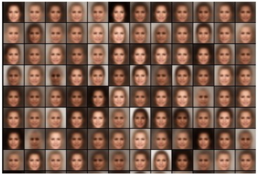
Self-consuming Loop for Training GenAI Models



AI-generated data is mixed into the training dataset for training the next-iteration model.

Model Collapse

(Gibney et al.'24, Nature News)



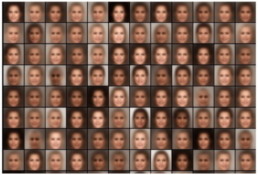
(Gerstgrasser et al.'24, COLM)

- **Model Collapse**: Model performance degrades over iterations[1]. Prior studies have shown that:

[1]An iteration denotes a complete training and sampling cycle, not a single gradient update during training.

Model Collapse


(Gibney et al.'24, Nature News)

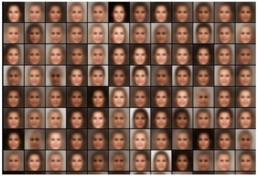

(Gerstgrasser et al.'24, COLM)

- **Model Collapse**: Model performance degrades over iterations[1]. Prior studies have shown that:
  - The **visual quality** of the generated images deteriorates. (FID $\uparrow$)

[1] An iteration denotes a complete training and sampling cycle, not a single gradient update during training.

Model Collapse

(Gibney et al.'24, Nature News)



(Gerstgrasser et al.'24, COLM)

- **Model Collapse**: Model performance degrades over iterations[1]. Prior studies have shown that:
  - The **visual quality** of the generated images deteriorates. (FID $\uparrow$)
  - The **test loss** increases. ($loss \uparrow$)

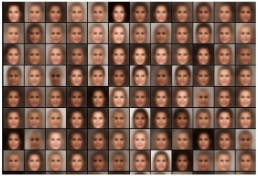
Theorem 2. *For an n-fold synthetic data generation process with $T \geq d+2$ samples per iteration and isotropic features ($\Sigma \stackrel{def}{=} I_d$), the test error for the ridgeless linear predictor $\hat{w}_n$ learned on the accumulated data up to iteration n is given by:*

$$E_{test}^{Accum}(\hat{w}_n) = \frac{\sigma^2 d}{T-d-1}\left(\sum_{i=1}^{n}\frac{1}{i^2}\right) \leq \frac{\sigma^2 d}{T-d-1} \times \frac{\pi^2}{6} \quad (3)$$

[1] An iteration denotes a complete training and sampling cycle, not a single gradient update during training.

Model Collapse

(Gibney et al.'24, Nature News)



(Gerstgrasser et al.'24, COLM)

- **Model Collapse**: Model performance degrades over iterations[1]. Prior studies have shown that:
  - The **visual quality** of the generated images deteriorates. (FID $\uparrow$)
  - The **test loss** increases. ($loss \uparrow$)
  - The **variance** of the generated images decreases. ($\sigma \to 0$)

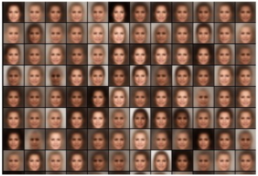
Under the above data-model feedback loop, Shumailov et al. (2024) prove that

$$\hat{\Sigma}^{(t+1)}_{\text{Replace}} \overset{a.s.}{\to} 0 \quad ; \quad \mathbb{E}[\mathbb{W}_2^2(\mathcal{N}(\hat{\mu}^{(t+1)}_{\text{Replace}}, \hat{\Sigma}^{(t+1)}_{\text{Replace}}), \mathcal{N}(\mu^{(0)}, \Sigma^{(0)}))] \to \infty \text{ as } t \to \infty, \tag{4}$$

[1] An iteration denotes a complete training and sampling cycle, not a single gradient update during training.

Model Collapse

(Gibney et al.'24, Nature News)



(Gerstgrasser et al.'24, COLM)

- **Model Collapse**: Model performance degrades over iterations[1]. Prior studies have shown that:
  - The **visual quality** of the generated images deteriorates. (FID $\uparrow$)
  - The **test loss** increases. ($loss \uparrow$)
  - The **variance** of the generated images decreases. ($\sigma \to 0$)

We reveal a **generalization-to-memorization transition** in model collapse, inspiring new mitigation strategies.

[1] An iteration denotes a complete training and sampling cycle, not a single gradient update during training.

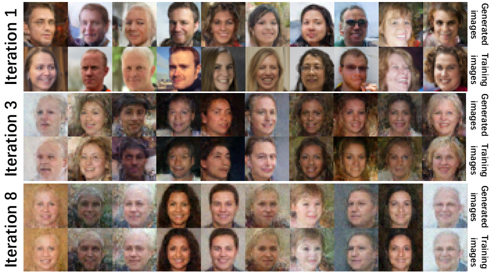
Generalization to Memorization Transition

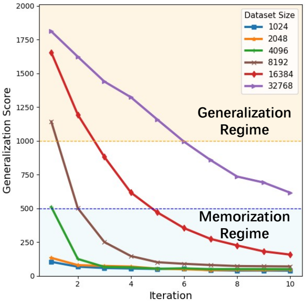
Generalization Score: the average distance between each generated image x in $\mathcal{G}_n$ and its nearest image z in the training dataset $\mathcal{D}_n$:

$$\mathsf{GS}(n) \triangleq \mathsf{Dist}(\mathcal{D}_n, \mathcal{G}_n) = \frac{1}{|\mathcal{G}_n|} \sum_{\boldsymbol{x} \in \mathcal{G}_n} \min_{\boldsymbol{z} \in \mathcal{D}_n} \kappa(\boldsymbol{x}, \boldsymbol{z}),$$

where $\kappa(\cdot, \cdot) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ denotes a distance metric.

Generalization to Memorization Transition





Generalization Score: the average distance between each generated image x in $\mathcal{G}_n$ and its nearest image z in the training dataset $\mathcal{D}_n$:

$$\mathsf{GS}(n) \triangleq \mathsf{Dist}(\mathcal{D}_n, \mathcal{G}_n) = \frac{1}{|\mathcal{G}_n|} \sum_{\boldsymbol{x} \in \mathcal{G}_n} \min_{\boldsymbol{z} \in \mathcal{D}_n} \kappa(\boldsymbol{x}, \boldsymbol{z}),$$

where $\kappa(\cdot,\cdot) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ denotes a distance metric.

Why does the Transition Occur?

Our Hypothesis

With a fixed sample size, information (measured by **entropy**) of the dataset falls over training loops, leading to memorization.

[2] Leonenko Kozachenko. Sample estimate of the entropy of a random vector. Problems of Information Transmission.

Why does the Transition Occur?

Our Hypothesis

With a fixed sample size, information (measured by **entropy**) of the dataset falls over training loops, leading to memorization.

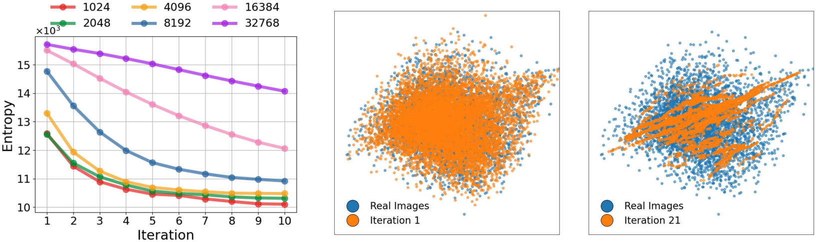
We adopt the Kozachenko-Leonenko (KL) estimator[2] to empirically estimate the entropy of a training dataset $\mathcal{D}$ as

$$\hat{H}_\gamma(\mathcal{D}) = \psi(|\mathcal{D}|) - \psi(\gamma) + \log c_d + \frac{d}{|\mathcal{D}|} \sum_{\boldsymbol{x} \in \mathcal{D}} \log \varepsilon_\gamma(\boldsymbol{x}),$$

where $\psi : \mathbb{N} \to \mathbb{R}$ is the digamma function; c_d denotes the volume of the unit ball in the d-dimensional space; and $\varepsilon_\gamma(\boldsymbol{x}) = \kappa(\boldsymbol{x}, \boldsymbol{x}_\gamma)$ represents the γ-nearest neighbor distance.

[2] Leonenko Kozachenko. Sample estimate of the entropy of a random vector. Problems of Information Transmission.

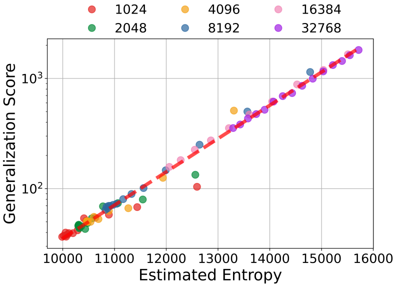
The Entropy of the Training Datasets



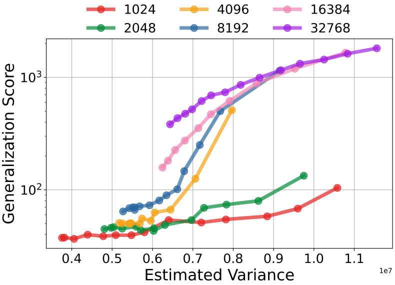
Left: Entropy of training data over self-consuming iterations under different data sizes. (Experiments conducted on Cifar-10 using DDPM)

Middle and Right: PCA visualization of data before and after collapse.

The Relation Between Entropy and Generalization Score



(a) Generalization score vs. estimated entropy.



(b) Generalization score vs. trace of covariance.

- All the points in (a) align well on a single line.
- The Generalization score shows only a weak, size-dependent correlation with variance.
- Entropy is therefore the more robust indicator.

Mitigating Collapse via Entropy-Based Sample Selection

Intuition. Given a candidate pool $\mathcal{S}$, consisting of both real and previously AI-generated images, choose a subset $\mathcal{D} \subset \mathcal{S}$ of size N that **maximizes training-set entropy**:

$$\max_{\mathcal{D}\subset\mathcal{S},\ |\mathcal{D}|=N} \underbrace{\sum_{\boldsymbol{x}\in\mathcal{D}} \log \min_{y\in\mathcal{D}\setminus\{\boldsymbol{x}\}} \kappa(\boldsymbol{x}, \boldsymbol{y})}_{\hat{H}_1(\mathcal{D})} .$$

- Yields a diverse, high-entropy training set for next-generation models.
- Difficult to optimize globally; requires approximation methods.

Mitigating Collapse via Entropy-Based Sample Selection

Algorithm I: Greedy Selection

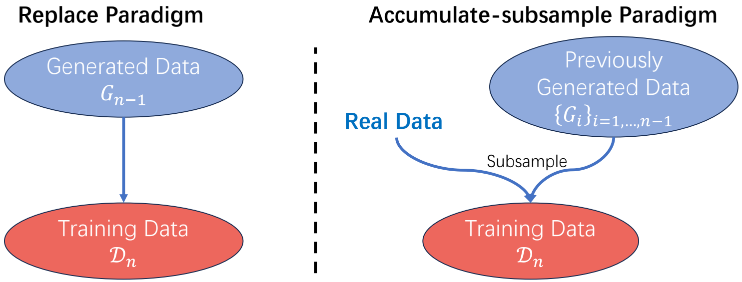
1. **Initialization**: randomly pick $x_0 \in \mathcal{S}$ and set $\mathcal{D} \leftarrow \{x_0\}$.
2. **Iterative step** (Terminate at $|\mathcal{D}| = N$):

$$\boldsymbol{x}_{\text{sel}} = \arg\max_{\boldsymbol{x} \in \mathcal{S} \setminus \mathcal{D}} \left[\min_{y \in \mathcal{D}} \kappa(\boldsymbol{x}, \boldsymbol{y}) \right], \qquad \mathcal{D} \leftarrow \mathcal{D} \cup \{\boldsymbol{x}_{\text{sel}}\}.$$

Algorithm II: Threshold Decay Filter

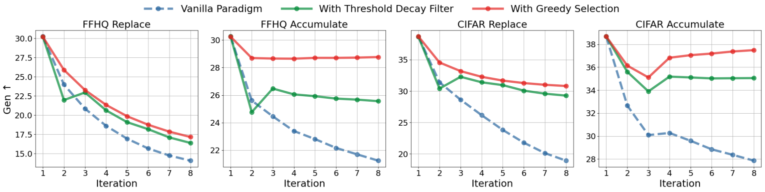
This method extends greedy selection by introducing an additional hyperparameter that controls the degree of greediness.

Two Different Paradigms of Self-consuming Training Loops

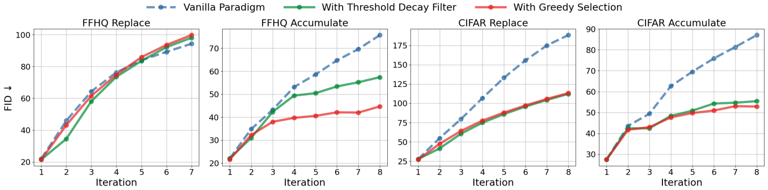


Our experiments are conducted under two distinct paradigms explored in prior studies.

Results: Generalization Score & FID



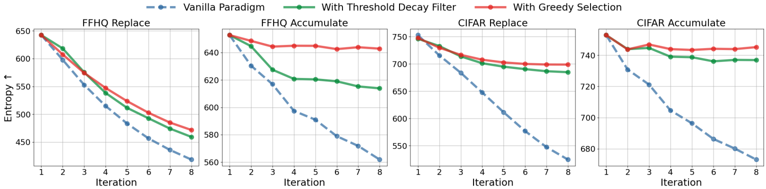
(a) Generalization Score over iterations



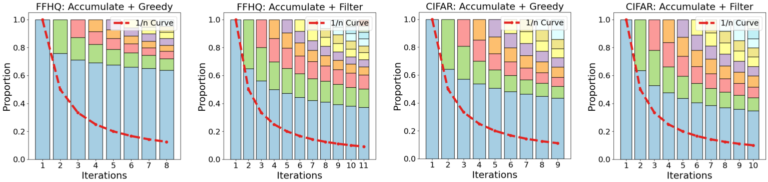
(b) FID over iterations

Entropy-based selection methods help preserve generalization performance and mitigate the rise in **FID**.

Analysis for the Improvement



(a) Estimated entropy over iterations



(b) Data composition over iterations

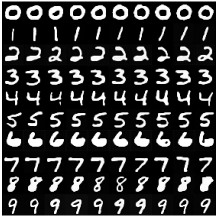
Through **Greedy selection** strategy, we maximize the entropy and observe a preference for selecting real data (blue) over synthetic data (others).

Mitigating Diversity Collapse of Classifier Free Guidance

Comparison of MNIST generations with different methods:

(a) Unconditional

Visually ambiguous!



(b) Classifier-free guidance (CFG)

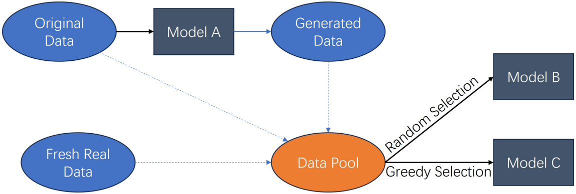
Looks better, but lacks diversity.

(c) CFG with Greedy Selection

Maintain both quality and diversity!

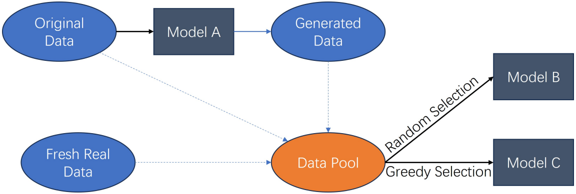
Training Under More Realistic Settings

A more realistic setting where fresh real images are incorporated into each iteration.



Training Under More Realistic Settings

A more realistic setting where fresh real images are incorporated into each iteration.



Model	A	B	C
FID	28.0	30.8	27.5

The method can outperform the original model trained on the original real images.

Summary

- Diffusion models collapse from **generalization to memorization** in the self-consuming loop.
- The entropy of the training dataset can serve as a robust predictor of memorization.
- Through the **entropy-based selection** methods, we mitigate the memorization issue and slow down the quality degradation.

Summary

- Diffusion models collapse from **generalization to memorization** in the self-consuming loop.
- The entropy of the training dataset can serve as a robust predictor of memorization.
- Through the **entropy-based selection** methods, we mitigate the memorization issue and slow down the quality degradation.

From this perspective, many questions need to be addressed:

- What caused the entropy of the dataset to decrease? (sampling or architecture bias?)
- Theoretically, how can we characterize the decaying rate based on simplified models?
- How can we further design methods for mitigating model collapse?

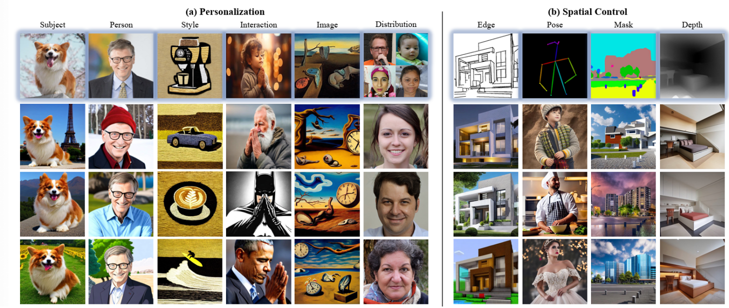
Summary

- Diffusion models collapse from **generalization to memorization** in the self-consuming loop.
- The entropy of the training dataset can serve as a robust predictor of memorization.
- Through the **entropy-based selection** methods, we mitigate the memorization issue and slow down the quality degradation.

1. Lianghe Shi, Meng Wu, Huijie Zhang, Zekai Zhang, Molei Tao, Qing Qu. A Closer Look at Model Collapse: From a Generalization-to-Memorization Perspective. *Neural Information Processing Systems (NeurIPS'25)*, 2025. (**spotlight, top 3.2%**)

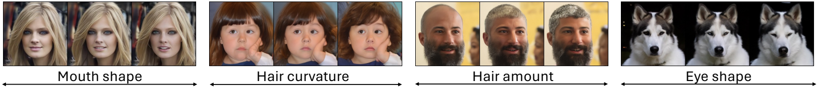
Low-Rank Image Editing & Watermarking

Controlled Generation is Challenging



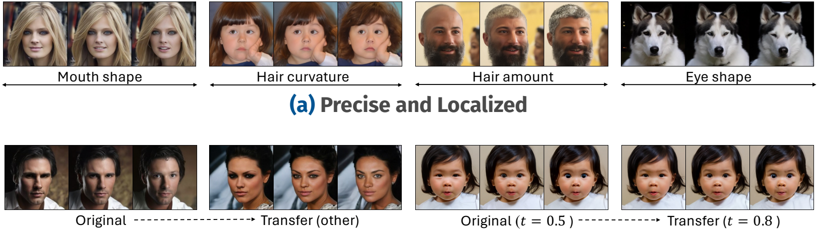
- Text prompt control is mostly global, they are not precise and they cannot do **local editing**.
- ControlNet is **expensive** and it relies on an extra neural network.
- Most methods remain heuristic and they lack **lack interpretability**.

LOw-rank COntrollable Image Editing (LOCO Edit)



(a) Precise and Localized

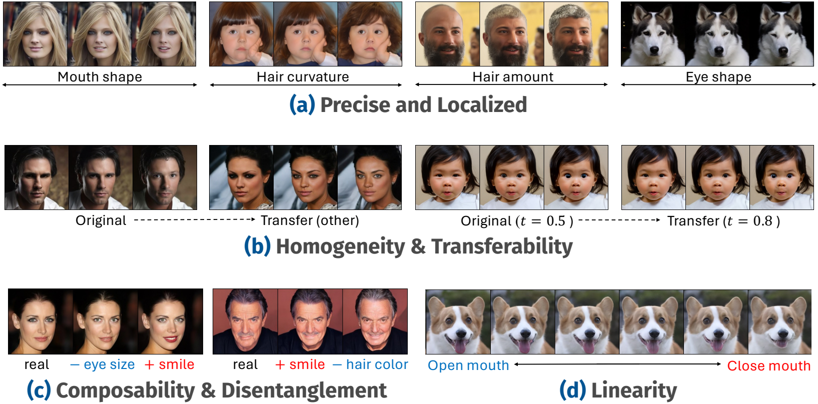
LOw-rank COntrollable Image Editing (LOCO Edit)



(a) Precise and Localized

(b) Homogeneity & Transferability

LOw-rank COntrollable Image Editing (LOCO Edit)



Editing in Text-to-image Diffusion Models

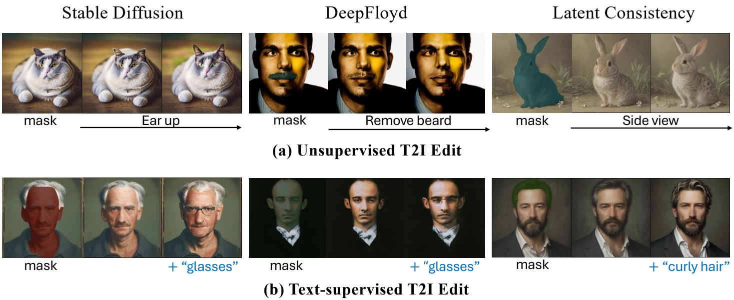


Figure 5: T-LOCO Edit on T2I diffusion models.

How does LOCO Edit Work?

Consider a unconditional diffusion model $\boldsymbol{s}_{\boldsymbol{\theta}}$:

- **Posterior mean predictor (PMP)** for the image x_0:

$$\boldsymbol{x}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t; t) := \frac{\boldsymbol{x}_t + (1-\alpha_t)\,\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)}{\sqrt{\alpha_t}} \approx \mathbb{E}[\boldsymbol{x}_0 | \boldsymbol{x}_t],$$

How does LOCO Edit Work?

Consider a unconditional diffusion model $s_{\boldsymbol{\theta}}$:

- **Posterior mean predictor (PMP)** for the image x_0:

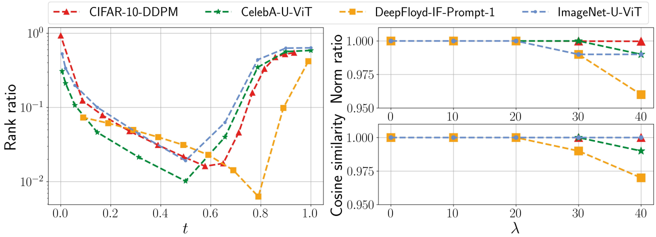
$$\boldsymbol{x}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t;t) := \frac{\boldsymbol{x}_t + (1-\alpha_t)\,\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_t,t)}{\sqrt{\alpha_t}} \approx \mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t],$$

- The **1st order Taylor expansion** of $\boldsymbol{x}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t + \lambda\Delta\boldsymbol{x})$ at $\boldsymbol{x}_t$:

$$\boxed{\boldsymbol{l}_{\boldsymbol{\theta}}(\boldsymbol{x}_t;\lambda\Delta\boldsymbol{x}) \;:=\; \boldsymbol{x}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t) + \lambda\boldsymbol{J}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t)\cdot\Delta\boldsymbol{x},}$$

where $\boldsymbol{J}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t) = \nabla_{\boldsymbol{x}_t}\boldsymbol{x}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t)$ is the Jacobian of $\boldsymbol{x}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t)$

Inductive Bias Towards "Simple" Solutions[3]



(a) **Low-rankness of the Jacobian** (b) **Local linearity of PMP**

The trained network via Adam tends to have simple structures:

- **Low-rankness** of the Jacobian $\boldsymbol{J}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t) = \nabla_{\boldsymbol{x}_t} \boldsymbol{x}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t)$:

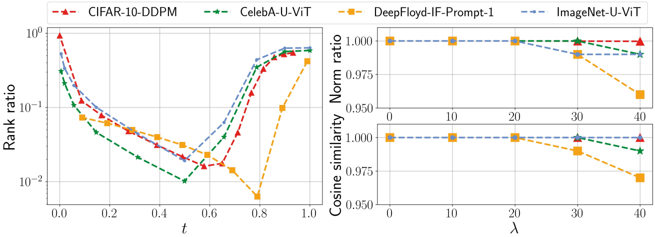
$$\boldsymbol{J}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t) = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{U}^\top = \sum_{i=1}^{r} \sigma_i \boldsymbol{u}_i \boldsymbol{u}_i^\top.$$

- **Local linearity** of the DAE:

$$\boldsymbol{x}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t + \lambda \Delta \boldsymbol{x}) \approx \boldsymbol{x}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t) + \lambda \boldsymbol{J}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t) \cdot \Delta \boldsymbol{x}$$

[3] X. Li, Y. Dai, Q. Qu. Understanding Generalizability of Diffusion Models Requires Rethinking the Hidden Gaussian Structure. *NeurIPS*, 2024.

How does LOCO Edit Work?



(a) **Low-rankness of the Jacobian** (b) **Local linearity of PMP**

Two key properties:

- **Local linearity** of the PMP $x_{\theta,t}(x_t) \approx l_\theta(x_t; \lambda \Delta x)$.
- **Low-rankness** of the Jacobian $J_{\theta,t}(x_t) = U\Sigma V^\top = \sum_{i=1}^{r} \sigma_i u_i v_i^\top$;

How does LOCO Edit Work?

$$\boldsymbol{J}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t) = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top = \sum_{i=1}^{r} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^\top$$

- **Local linearity** of the PMP with $\Delta \boldsymbol{x} = \boldsymbol{v}_i$, one column of $\boldsymbol{V}$:

$$\begin{aligned} \boldsymbol{x}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t + \lambda \boldsymbol{v}_i) &\approx \boldsymbol{x}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t) + \lambda \boldsymbol{J}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t)\boldsymbol{v}_i \\ &= \boldsymbol{x}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t) + \lambda \sum_{j=1}^{r} \sigma_j \boldsymbol{u}_j \boldsymbol{v}_j^\top \boldsymbol{v}_i \\ &= \hat{\boldsymbol{x}}_{0,t} + \lambda \sigma_i \boldsymbol{u}_i. \end{aligned}$$

How does LOCO Edit Work?

$$\boldsymbol{J}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t) = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top = \sum_{i=1}^{r} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^\top$$

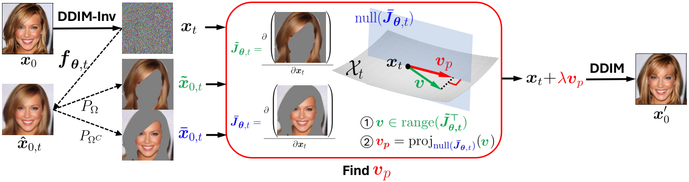
- **Local linearity** of the PMP with $\Delta \boldsymbol{x} = \boldsymbol{v}_i$, one column of $\boldsymbol{V}$:

$$\begin{aligned} \boldsymbol{x}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t + \lambda \boldsymbol{v}_i) &\approx \boldsymbol{x}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t) + \lambda \boldsymbol{J}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t)\boldsymbol{v}_i \\ &= \boldsymbol{x}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t) + \lambda \sum_{j=1}^{r} \sigma_j \boldsymbol{u}_j \boldsymbol{v}_j^\top \boldsymbol{v}_i \\ &= \hat{\boldsymbol{x}}_{0,t} + \lambda \sigma_i \boldsymbol{u}_i. \end{aligned}$$

- **Low rankness of the Jacobian** $J_{\boldsymbol{\theta},t}(\boldsymbol{x}_t)$ (e.g., $t = 0.7$):
  - $\boldsymbol{V}$ can be computed efficiently via generalized power method!

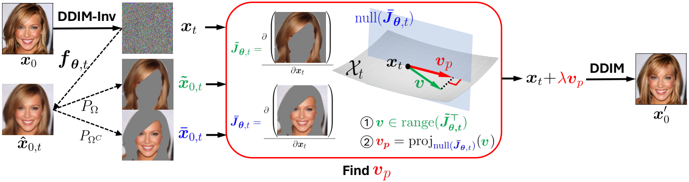
Overview of LOCO Edit

- Illustration of LOCO Edit for unconditional diffusion models:

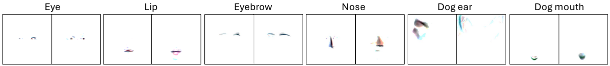


Overview of LOCO Edit

- Illustration of LOCO Edit for unconditional diffusion models:

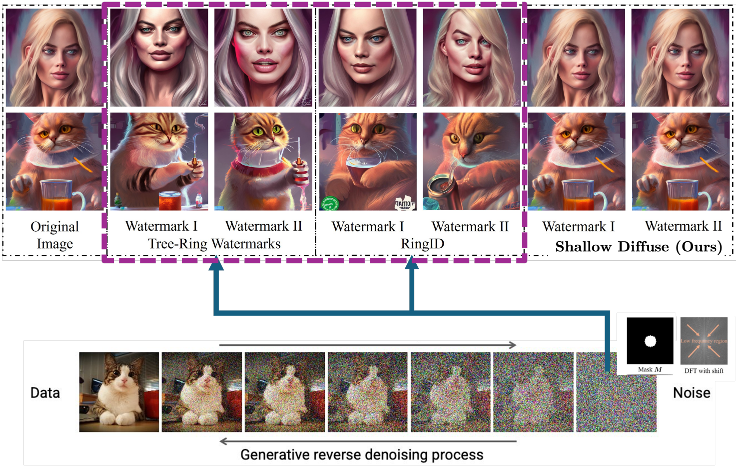


- Visualizing **editing directions** identified via LOCO Edit:

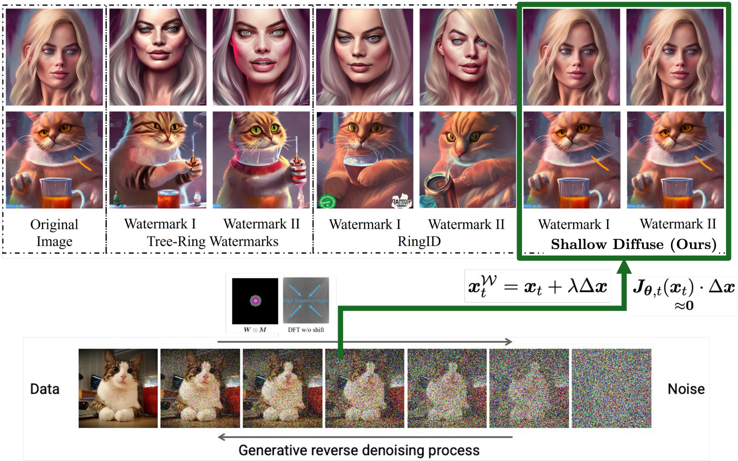


Visual Comparison with Existing Methods

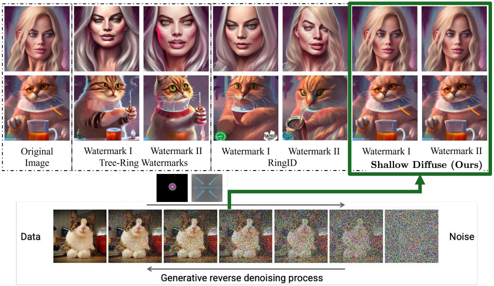
Shallow Diffuse: Robust and Invisible Watermarking



Shallow Diffuse: Robust and Invisible Watermarking



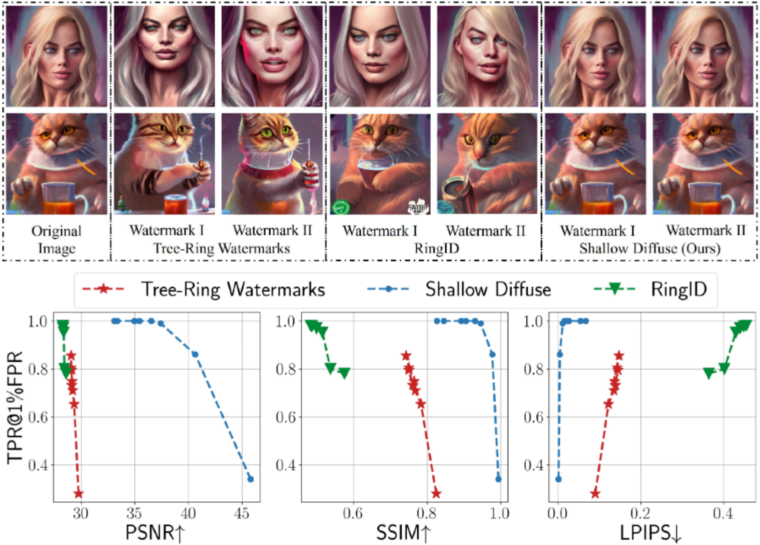
Shallow Diffuse: Robust and Invisible Watermarking



Key idea: Inject the watermark Δx in the **Null Space** of $J_{\boldsymbol{\theta},t}(x_t)$:

$$x_{\boldsymbol{\theta},t}(x_t^{\mathcal{W}}) = x_{\boldsymbol{\theta},t}(x_t) + \boxed{\underbrace{\lambda J_{\boldsymbol{\theta},t}(x_t) \cdot \Delta x}_{\approx \mathbf{0}}} \approx x_{\boldsymbol{\theta},t}(x_t)$$

Shallow Diffuse: Comparison



Shallow Diffuse: Comparison

Dataset	Method	Generation Consistency			Watermark Robustness (AUC ↑/TPR@1%FPR↑)				
		PSNR ↑	SSIM ↑	LPIPS ↓	Clean	Distortion	Regeneration	Adversarial	Average
COCO	SD w/o WM	32.28	0.78	0.06	-	-	-	-	-
COCO	DwtDct	37.88	0.97	**0.02**	0.83	0.54	0.00	0.82	0.36
COCO	DwtDctSvd	38.06	**0.98**	**0.02**	**1.00**	0.76	0.06	0.00	0.38
COCO	RivaGAN	**40.57**	**0.98**	0.04	**1.00**	0.93	0.05	**1.00**	0.59
COCO	Stegastamp	31.88	0.86	0.08	**1.00**	0.97	0.47	0.26	0.68
COCO	Gaussian Shading	10.17	0.23	0.65	**1.00**	0.99	**1.00**	0.47	0.92
COCO	Tree-Ring	28.22	0.57	0.41	**1.00**	0.90	0.95	0.31	0.84
COCO	RingID	12.21	0.38	0.58	**1.00**	0.98	**1.00**	0.79	**0.96**
COCO	**Shallow Diffuse**	32.11	0.84	0.05	**1.00**	**1.00**	0.96	0.62	0.93
DiffusionDB	SD w/o WM	33.42	0.85	0.03	-	-	-	-	-
DiffusionDB	DwtDct	37.77	0.96	**0.02**	0.76	0.34	0.01	0.78	0.27
DiffusionDB	DwtDctSvd	37.84	0.97	**0.02**	**1.00**	0.74	0.04	0.00	0.36
DiffusionDB	RivaGAN	**40.6**	**0.98**	0.04	0.98	0.88	0.04	**0.98**	0.56
DiffusionDB	Stegastamp	32.03	0.85	0.08	**1.00**	0.96	0.46	0.26	0.67
DiffusionDB	Gaussian Shading	10.61	0.27	0.63	**1.00**	0.99	**1.00**	0.46	0.92
DiffusionDB	Tree-Ring	28.3	0.62	0.29	**1.00**	0.81	0.87	0.26	0.76
DiffusionDB	RingID	12.53	0.45	0.53	**1.00**	0.99	**1.00**	0.79	**0.97**
DiffusionDB	**Shallow Diffuse**	33.07	0.89	0.03	**1.00**	**1.00**	0.93	0.59	0.92

Discussion

- Training diffusion models exhibits implicit bias towards low-dimensional structures (low-rank Jacobian and linearity).
- We can leverage the benign structures to manipulate the generation and protect the copyright in principled manners.

Discussion

- Training diffusion models exhibits implicit bias towards low-dimensional structures (low-rank Jacobian and linearity).
- We can leverage the benign structures to manipulate the generation and protect the copyright in principled manners.

2. Siyi Chen*, Huijie Zhang*, Minzhe Guo, Yifu Lu, Peng Wang, Qing Qu. Exploring Low-Dimensional Subspaces in Diffusion Models for Controllable Image Editing. *Neural Information Processing Systems (NeurIPS'24)*, 2024.
3. Wenda Li, Huijie Zhang, Qing Qu. Shallow Diffuse: Robust and Invisible Watermarking through Low-Dimensional Subspaces in Diffusion Models. *NeurIPS*, 2025 (**spotlight, top 3.2 %**).

Understanding Classifier-Free Guidance (CFG)

Conditional Generation and Classifier Guidance

- In practice, we often want to generate **specific types** of images (e.g., “a dog,” “a cat”).
- To achieve this, we have to sample using a conditional score

$$\underbrace{\nabla \log p(\boldsymbol{x}_t \mid \boldsymbol{c})}_{\textbf{conditional score}} = \underbrace{\nabla \log p(\boldsymbol{x}_t)}_{\textbf{unconditional score}} + \underbrace{\nabla p(\boldsymbol{c} \mid \boldsymbol{x}_t)}_{\textbf{classifier score}}$$

  so that the denoising process can be conditioned on the input c (e.g., a class label, a text prompt, an image embedding).

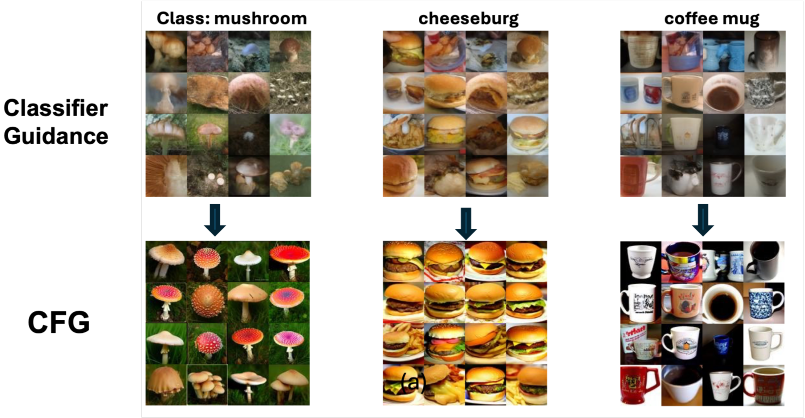
Conditional Generation and Classifier Guidance

- In practice, we often want to generate **specific types** of images (e.g., “a dog,” “a cat”).
- To achieve this, we have to sample using a conditional score

$$\underbrace{\nabla \log p(\boldsymbol{x}_t \mid \boldsymbol{c})}_{\textbf{conditional score}} = \underbrace{\nabla \log p(\boldsymbol{x}_t)}_{\textbf{unconditional score}} + \underbrace{\nabla p(\boldsymbol{c} \mid \boldsymbol{x}_t)}_{\textbf{classifier score}}$$

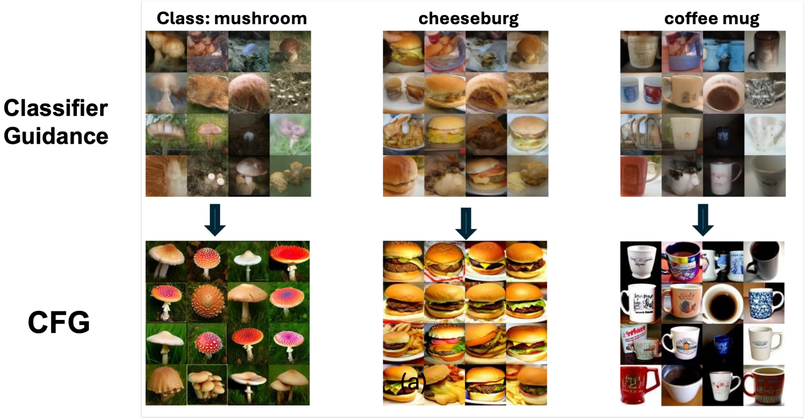
  so that the denoising process can be conditioned on the input c (e.g., a class label, a text prompt, an image embedding).
- **Classifier guidance** achieve this by training a **separate classifier** to approximate $p(\boldsymbol{c} \mid \boldsymbol{x}_t)$ across noise levels t.

Classifier Guidance vs. Classifier Free Guidance



- **Classifier guidance:** low-quality with similar patterns;

Classifier Guidance vs. Classifier Free Guidance



- **Classifier guidance:** low-quality with similar patterns;
- **CFG:** Significantly improved visual quality and distinctiveness.

Classifier Free Guidance (CFG)

The CFG operates by conditional sampling from

$$
\begin{aligned}
&\nabla \log p_{\texttt{CFG}}(\boldsymbol{x}_t \mid \boldsymbol{c}) \\
&= \nabla \log p(\boldsymbol{x}_t \mid \emptyset) + \gamma' \left(\nabla \log p(\boldsymbol{x}_t \mid \boldsymbol{c}) - \nabla \log p(\boldsymbol{x}_t \mid \emptyset)\right) \\
&= \nabla \log p(\boldsymbol{x}_t \mid \boldsymbol{c}) + \underbrace{(\gamma' - 1)}_{\gamma} \underbrace{\left(\nabla \log p(\boldsymbol{x}_t \mid \boldsymbol{c}) - \nabla \log p(\boldsymbol{x}_t \mid \emptyset)\right)}_{g(\boldsymbol{x}_t, \boldsymbol{c})}
\end{aligned}
$$

Classifier Free Guidance (CFG)

The CFG operates by conditional sampling from

$$
\begin{aligned}
&\nabla \log p_{\texttt{CFG}}(\boldsymbol{x}_t \mid \boldsymbol{c}) \\
&= \nabla \log p(\boldsymbol{x}_t \mid \emptyset) + \gamma' \left(\nabla \log p(\boldsymbol{x}_t \mid \boldsymbol{c}) - \nabla \log p(\boldsymbol{x}_t \mid \emptyset)\right) \\
&= \nabla \log p(\boldsymbol{x}_t \mid \boldsymbol{c}) + \underbrace{(\gamma' - 1)}_{\gamma} \underbrace{\left(\nabla \log p(\boldsymbol{x}_t \mid \boldsymbol{c}) - \nabla \log p(\boldsymbol{x}_t \mid \emptyset)\right)}_{g(\boldsymbol{x}_t, \boldsymbol{c})}
\end{aligned}
$$

- Essentially, $g(\boldsymbol{x}_t, \boldsymbol{c}) = \nabla \log p(\boldsymbol{c} \mid \boldsymbol{x}_t)$, where conditional $\log p(\boldsymbol{x}_t \mid \boldsymbol{c})$ and unconditional $\nabla \log p(\boldsymbol{x}_t \mid \emptyset)$ are trained jointly.

Classifier Free Guidance (CFG)

The CFG operates by conditional sampling from

$$\begin{aligned}
&\nabla \log p_{\texttt{CFG}}(\boldsymbol{x}_t \mid \boldsymbol{c}) \\
&= \nabla \log p(\boldsymbol{x}_t \mid \emptyset) + \gamma' \left(\nabla \log p(\boldsymbol{x}_t \mid \boldsymbol{c}) - \nabla \log p(\boldsymbol{x}_t \mid \emptyset)\right) \\
&= \nabla \log p(\boldsymbol{x}_t \mid \boldsymbol{c}) + \underbrace{(\gamma' - 1)}_{\gamma} \underbrace{\left(\nabla \log p(\boldsymbol{x}_t \mid \boldsymbol{c}) - \nabla \log p(\boldsymbol{x}_t \mid \emptyset)\right)}_{g(\boldsymbol{x}_t, \boldsymbol{c})}
\end{aligned}$$

- Essentially, $g(\boldsymbol{x}_t, \boldsymbol{c}) = \nabla \log p(\boldsymbol{c} \mid \boldsymbol{x}_t)$, where conditional $\log p(\boldsymbol{x}_t \mid \boldsymbol{c})$ and unconditional $\nabla \log p(\boldsymbol{x}_t \mid \emptyset)$ are trained jointly.
- We have $\nabla \log p_{\texttt{CFG}}(\boldsymbol{x}_t \mid \boldsymbol{c}) = \nabla \log p(\boldsymbol{x}_t \mid \boldsymbol{c})$ only when $\gamma = 0$.

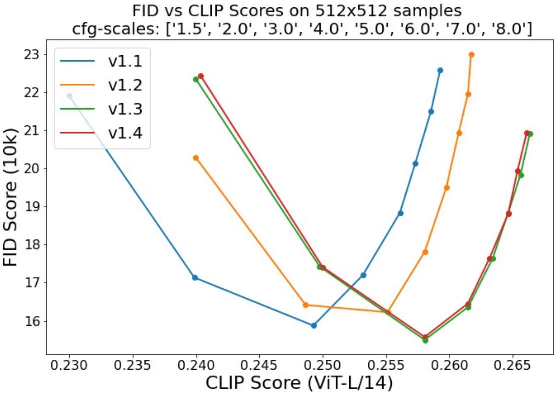
Classifier Free Guidance (CFG)

The CFG operates by conditional sampling from

$$
\begin{aligned}
&\nabla \log p_{\texttt{CFG}}(\boldsymbol{x}_t \mid \boldsymbol{c}) \\
&= \nabla \log p(\boldsymbol{x}_t \mid \emptyset) + \gamma' \left(\nabla \log p(\boldsymbol{x}_t \mid \boldsymbol{c}) - \nabla \log p(\boldsymbol{x}_t \mid \emptyset)\right) \\
&= \nabla \log p(\boldsymbol{x}_t \mid \boldsymbol{c}) + \underbrace{(\gamma' - 1)}_{\gamma} \underbrace{\left(\nabla \log p(\boldsymbol{x}_t \mid \boldsymbol{c}) - \nabla \log p(\boldsymbol{x}_t \mid \emptyset)\right)}_{g(\boldsymbol{x}_t, \boldsymbol{c})}
\end{aligned}
$$

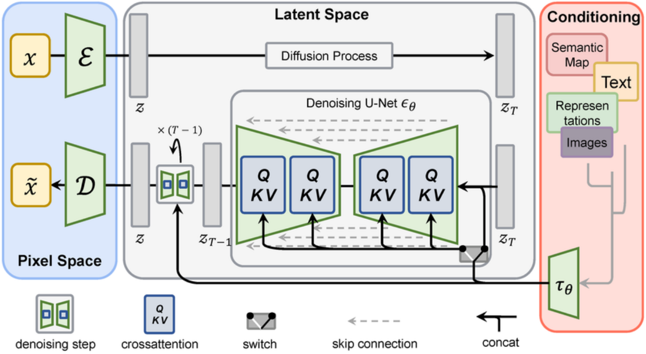
- Essentially, $g(\boldsymbol{x}_t, \boldsymbol{c}) = \nabla \log p(\boldsymbol{c} \mid \boldsymbol{x}_t)$, where conditional $\log p(\boldsymbol{x}_t \mid \boldsymbol{c})$ and unconditional $\nabla \log p(\boldsymbol{x}_t \mid \emptyset)$ are trained jointly.
- We have $\nabla \log p_{\texttt{CFG}}(\boldsymbol{x}_t \mid \boldsymbol{c}) = \nabla \log p(\boldsymbol{x}_t \mid \boldsymbol{c})$ only when $\gamma = 0$.
- However, the guidance strength $\gamma \geq 0$ is typically chosen to be quite **large** (e.g., $\gamma \in [5, 8]$) for CFG to work.

Ablation Studies of Strength γ



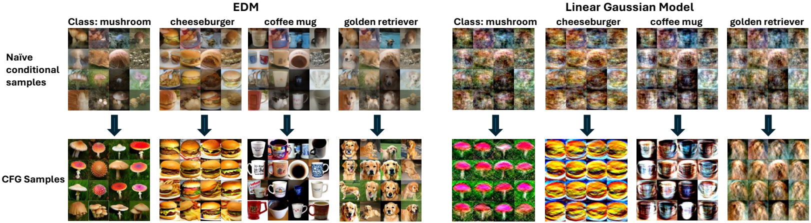
Why does large γ in CFG work really well in practice?

Importance of Understanding CFG



CFG is the fundamental technique in modern text-to-image (T2I) diffusion models in the latent space.

Why does CFG Improve Sample Quality?

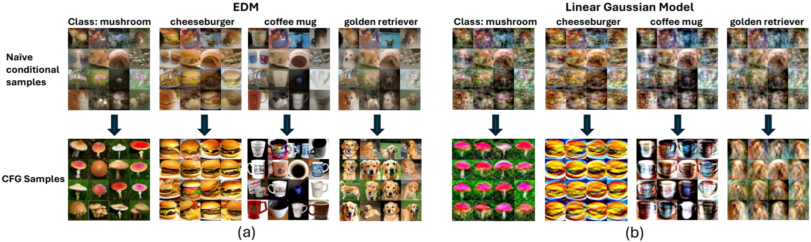


(a) (b)

Questions

- Why naive conditional sampling is subpar?
- How CFG with large γ improves image quality?

Why does CFG Improve Sample Quality?



Questions

- Why naive conditional sampling is subpar?
- How CFG with large γ improves image quality?

We study these questions on **linear** diffusion models, capturing the essential insights on real-world nonlinear models.

Linear Models with Gaussian Data Assumption

Lemma (Linear Score with Gaussian Data)

Assume the data $p_0(\boldsymbol{x})$ is Gaussian with $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, with the mean $\boldsymbol{\mu}$ and the covariance $\boldsymbol{\Sigma} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^\top$. The optimal solution of the score function $\nabla \log p(\boldsymbol{x}_t)$ at time-step t can be derived as

$$\nabla \log p(\boldsymbol{x}_t) = \frac{1}{\sigma_t^2}(\tilde{\boldsymbol{\Sigma}}_t - \boldsymbol{I})(\boldsymbol{x}_t - \boldsymbol{\mu})$$

where $\tilde{\boldsymbol{\Sigma}}_t = \boldsymbol{U}\tilde{\boldsymbol{\Lambda}}_t\boldsymbol{U}^\top$ with $\tilde{\boldsymbol{\Lambda}}_t = \mathrm{diag}\left(\frac{\lambda_1}{\lambda_1+\sigma_t^2}, \cdots, \frac{\lambda_d}{\lambda_d+\sigma_t^2}\right)$.

With Tweedie's formula, we have the relationship:

$$\nabla \log p(\boldsymbol{x}_t) \approx \frac{\boldsymbol{x}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t) - \boldsymbol{x}_t}{\sigma_t^2}.$$

Class Condition Score and CFG

If we let the conditional and unconditional data distributions be $\mathcal{N}(\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$ and $\mathcal{N}(\boldsymbol{\mu}_{uc}, \boldsymbol{\Sigma}_{uc})$ with overlapping bases $\boldsymbol{U}_c$ and $\boldsymbol{U}_{uc}$,

$$\nabla \log p_{\texttt{CFG}}(\boldsymbol{x}_t \mid \boldsymbol{c}) = \nabla \log p(\boldsymbol{x}_t \mid \boldsymbol{c}) + \gamma \cdot g(\boldsymbol{x}_t, \boldsymbol{c})$$

- **Class condition score** $\nabla \log p(\boldsymbol{x}_t \mid \boldsymbol{c})$**:**

$$\nabla \log p(\boldsymbol{x}_t \mid \boldsymbol{c}) = \frac{1}{\sigma_t^2}(\tilde{\boldsymbol{\Sigma}}_{c,t} - \boldsymbol{I})(\boldsymbol{x}_t - \boldsymbol{\mu}_c)$$

Class Condition Score and CFG

If we let the conditional and unconditional data distributions be $\mathcal{N}(\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$ and $\mathcal{N}(\boldsymbol{\mu}_{uc}, \boldsymbol{\Sigma}_{uc})$ with overlapping bases $\boldsymbol{U}_c$ and $\boldsymbol{U}_{uc}$,

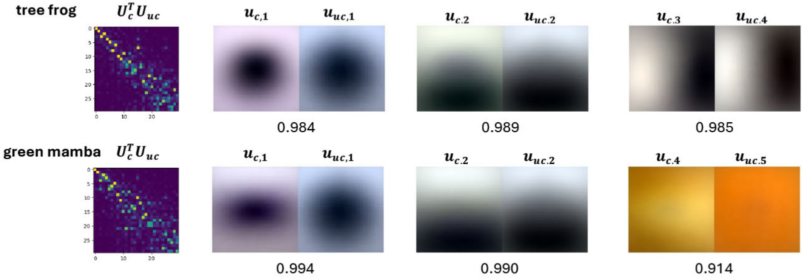
$$\nabla \log p_{\texttt{CFG}}(\boldsymbol{x}_t \mid \boldsymbol{c}) = \nabla \log p(\boldsymbol{x}_t \mid \boldsymbol{c}) + \gamma \cdot g(\boldsymbol{x}_t, \boldsymbol{c})$$

- **Class condition score** $\nabla \log p(\boldsymbol{x}_t \mid \boldsymbol{c})$**:**

$$\nabla \log p(\boldsymbol{x}_t \mid \boldsymbol{c}) = \frac{1}{\sigma_t^2}(\tilde{\boldsymbol{\Sigma}}_{c,t} - \boldsymbol{I})(\boldsymbol{x}_t - \boldsymbol{\mu}_c)$$

  - Classifier guidance only uses $\nabla \log p(\boldsymbol{x}_t \mid \boldsymbol{c})$, which is shaped by the covariance structure $\boldsymbol{\Sigma}_c$ (Principal Components).
  - PCs of $\boldsymbol{\Sigma}_c$ do not necessarily capture class-specific patterns.

Why Classifier Guidance Does Not Work



- Sampling only with class condition score $\nabla \log p(\boldsymbol{x}_t \mid \boldsymbol{c})$:

$$\boldsymbol{x}_t = \boldsymbol{\mu}_c + \sum_{i=1}^{d} \sqrt{\frac{\sigma_t^2 + \lambda_i}{\sigma_T^2 + \lambda_i}} \boldsymbol{u}_{c,i}^T (\boldsymbol{x}_T - \boldsymbol{\mu}) \boldsymbol{u}_{c,i}.$$

- PCs of Σ_c do not necessarily capture class-specific patterns.

Decomposition of CFG: Positive CPC

If we let $\mathcal{N}(\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$ and $\mathcal{N}(\boldsymbol{\mu}_{uc}, \boldsymbol{\Sigma}_{uc})$ be the data distributions of conditional and unconditional data, then

$$\nabla \log p_{\texttt{CFG}}(\boldsymbol{x}_t \mid \boldsymbol{c}) = \nabla \log p(\boldsymbol{x}_t \mid \boldsymbol{c}) + \gamma \cdot g(\boldsymbol{x}_t, \boldsymbol{c})$$
$$g(\boldsymbol{x}_t, \boldsymbol{c}) = \mathcal{T}_{\texttt{Pos-CPC}} \; + \; \mathcal{T}_{\texttt{Neg-CPC}} \; + \; \mathcal{T}_{\texttt{Mean-Shift}}$$

- **The positive contrastive principal component (Pos-CPC):**

$$\mathcal{T}_{\texttt{Pos-CPC}} = \frac{1}{\sigma_t^2} \boldsymbol{V}_{t,+} \hat{\boldsymbol{\Lambda}}_{t,+} \boldsymbol{V}_{t,+}^\top (\boldsymbol{x}_t - \boldsymbol{\mu}_c),$$

  where $\boldsymbol{V}_{t,+}$ is the eigenvector matrix of $\tilde{\boldsymbol{\Sigma}}_{c,t} - \tilde{\boldsymbol{\Sigma}}_{uc,t}$ with positive eigenvalues $\hat{\boldsymbol{\Lambda}}_{t,+}$, such that $\boldsymbol{v}_{+,i}^\top \tilde{\boldsymbol{\Sigma}}_{c,t} \boldsymbol{v}_{+,i} > \boldsymbol{v}_{+,i}^\top \tilde{\boldsymbol{\Sigma}}_{uc,t} \boldsymbol{v}_{+,i}$.
- $\mathcal{T}_{\texttt{Pos-CPC}}$ **enhances** components of $\boldsymbol{x}_t - \boldsymbol{\mu}_c$ that align with $\boldsymbol{V}_{t,+}$.

Decomposition of CFG: Negative CPC

If we let the conditional and unconditional data distributions be $\mathcal{N}(\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$ and $\mathcal{N}(\boldsymbol{\mu}_{uc}, \boldsymbol{\Sigma}_{uc})$ with overlapping bases $\boldsymbol{U}_c$ and $\boldsymbol{U}_{uc}$,

$$\nabla \log p_{\texttt{CFG}}(\boldsymbol{x}_t \mid \boldsymbol{c}) = \nabla \log p(\boldsymbol{x}_t \mid \boldsymbol{c}) + \gamma \cdot g(\boldsymbol{x}_t, \boldsymbol{c})$$
$$g(\boldsymbol{x}_t, \boldsymbol{c}) = \mathcal{T}_{\texttt{Pos-CPC}} + \mathcal{T}_{\texttt{Neg-CPC}} + \mathcal{T}_{\texttt{Mean-Shift}}$$

- **The negative contrastive principal component (Pos-CPC):**

$$\mathcal{T}_{\texttt{Neg-CPC}} = \frac{1}{\sigma_t^2} \boldsymbol{V}_{t,-} \hat{\boldsymbol{\Lambda}}_{t,-} \boldsymbol{V}_{t,-}^{\top} (\boldsymbol{x}_t - \boldsymbol{\mu}_c).$$

where $\boldsymbol{V}_{t,-}$ is the eigenvectors of $\tilde{\boldsymbol{\Sigma}}_{c,t} - \tilde{\boldsymbol{\Sigma}}_{uc,t}$ with negative eigenvalues $\hat{\boldsymbol{\Lambda}}_{t,-}$, such that $\boldsymbol{v}_{-,i}^{\top} \tilde{\boldsymbol{\Sigma}}_{c,t} \boldsymbol{v}_{-,i} < \boldsymbol{v}_{-,i}^{\top} \tilde{\boldsymbol{\Sigma}}_{uc,t} \boldsymbol{v}_{-,i}$.

Decomposition of CFG: Negative CPC

If we let the conditional and unconditional data distributions be $\mathcal{N}(\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$ and $\mathcal{N}(\boldsymbol{\mu}_{uc}, \boldsymbol{\Sigma}_{uc})$ with overlapping bases $\boldsymbol{U}_c$ and $\boldsymbol{U}_{uc}$,

$$\nabla \log p_{\texttt{CFG}}(\boldsymbol{x}_t \mid \boldsymbol{c}) = \nabla \log p(\boldsymbol{x}_t \mid \boldsymbol{c}) + \gamma \cdot g(\boldsymbol{x}_t, \boldsymbol{c})$$
$$g(\boldsymbol{x}_t, \boldsymbol{c}) = \mathcal{T}_{\texttt{Pos-CPC}} \; + \; \mathcal{T}_{\texttt{Neg-CPC}} \; + \; \mathcal{T}_{\texttt{Mean-Shift}}$$

- **The negative contrastive principal component (Pos-CPC):**

$$\mathcal{T}_{\texttt{Neg-CPC}} = \frac{1}{\sigma_t^2} \boldsymbol{V}_{t,-} \hat{\boldsymbol{\Lambda}}_{t,-} \boldsymbol{V}_{t,-}^\top (\boldsymbol{x}_t - \boldsymbol{\mu}_c).$$

  where $\boldsymbol{V}_{t,-}$ is the eigenvectors of $\tilde{\boldsymbol{\Sigma}}_{c,t} - \tilde{\boldsymbol{\Sigma}}_{uc,t}$ with negative eigenvalues $\hat{\boldsymbol{\Lambda}}_{t,-}$, such that $\boldsymbol{v}_{-,i}^\top \tilde{\boldsymbol{\Sigma}}_{c,t} \boldsymbol{v}_{-,i} < \boldsymbol{v}_{-,i}^\top \tilde{\boldsymbol{\Sigma}}_{uc,t} \boldsymbol{v}_{-,i}$.
- $\mathcal{T}_{\texttt{Neg-CPC}}$ **suppresses** components of $\boldsymbol{x}_t - \boldsymbol{\mu}_c$ that align with $\boldsymbol{V}_{t,-}$.

Decomposition of CFG: Mean-Shift

If we let the conditional and unconditional data distributions be $\mathcal{N}(\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$ and $\mathcal{N}(\boldsymbol{\mu}_{uc}, \boldsymbol{\Sigma}_{uc})$ with overlapping bases $\boldsymbol{U}_c$ and $\boldsymbol{U}_{uc}$,

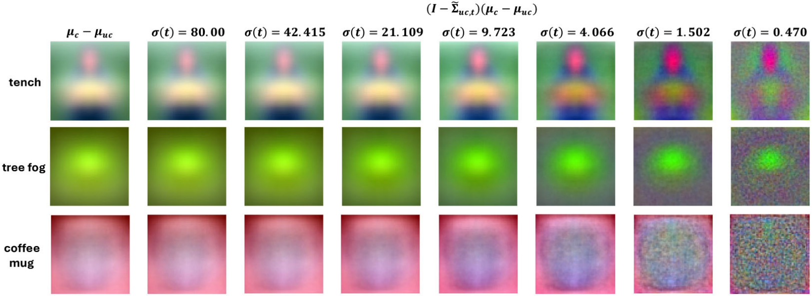
$$\nabla \log p_{\texttt{CFG}}(\boldsymbol{x}_t \mid \boldsymbol{c}) = \nabla \log p(\boldsymbol{x}_t \mid \boldsymbol{c}) + \gamma \cdot g(\boldsymbol{x}_t, \boldsymbol{c})$$
$$g(\boldsymbol{x}_t, \boldsymbol{c}) = \mathcal{T}_{\texttt{Pos-CPC}} \; + \; \mathcal{T}_{\texttt{Neg-CPC}} \; + \; \mathcal{T}_{\texttt{Mean-Shift}}$$

- **The mean-shift component:**

$$\mathcal{T}_{\texttt{Mean-Shift}} = \frac{1}{\sigma_t^2}(\boldsymbol{I} - \tilde{\boldsymbol{\Sigma}}_{uc,t})(\boldsymbol{\mu}_c - \boldsymbol{\mu}_{uc}) \approx \frac{\gamma}{\sigma_t^2}(\boldsymbol{\mu}_c - \boldsymbol{\mu}_{uc})$$

- $\mathcal{T}_{\texttt{Mean-Shift}}$ is independent of $\boldsymbol{x}_t$, i.e., it adds a **constant perturbation** to all trajectories, leading to low diversity.

Decomposition of CFG: Mean-Shift

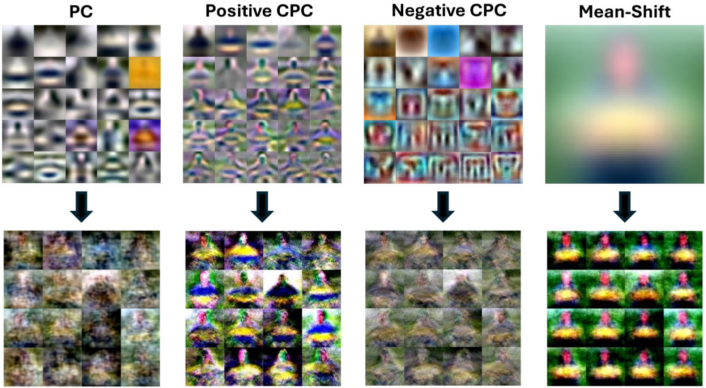


- **The mean-shift component:**

$$\mathcal{T}_{\texttt{Mean-Shift}} = \frac{1}{\sigma_t^2}(\boldsymbol{I} - \tilde{\Sigma}_{uc,t})(\boldsymbol{\mu}_c - \boldsymbol{\mu}_{uc}) \approx \frac{\gamma}{\sigma_t^2}(\boldsymbol{\mu}_c - \boldsymbol{\mu}_{uc})$$

- $\mathcal{T}_{\texttt{Mean-Shift}}$ is independent of x_t, i.e., it adds a **constant perturbation** to all trajectories, leading to low diversity.

How Does CFG Lead to High Quality Samples?

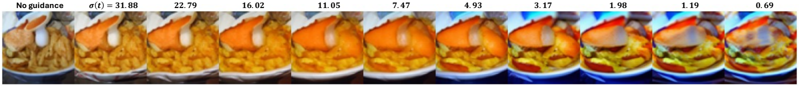


$$\nabla \log p_{\texttt{CFG}}(\boldsymbol{x}_t \mid \boldsymbol{c}) = \nabla \log p(\boldsymbol{x}_t \mid \boldsymbol{c}) + \gamma \cdot g(\boldsymbol{x}_t, \boldsymbol{c})$$

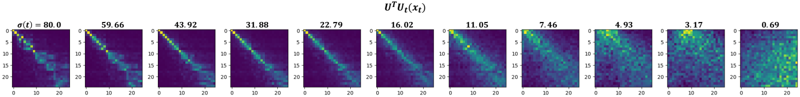
$$g(\boldsymbol{x}_t, \boldsymbol{c}) = \mathcal{T}_{\texttt{Pos-CPC}} + \mathcal{T}_{\texttt{Neg-CPC}} + \mathcal{T}_{\texttt{Mean-Shift}}$$

Linear-to-Nonlinear Transition in Real-World Models

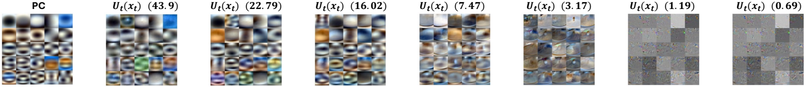
(a) Effects of nonlinear CFG at individual timesteps



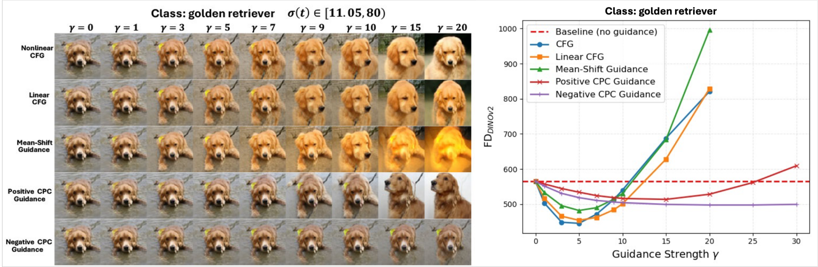
(b) Effects of linear CFG at individual timestep



(c) Correlation matrix between the PCs (U) of data covariance and singular vectors of network Jacobians $U_t(x_t)$



(d) Evolution of singular vectors of network Jacobians $U_t(x_t)$ across different noise levels $\sigma(t)$



Key observations:

- Mean-shift guidance dominates CFG's effect (in linear regime).
- CPC guidance could also lead to improved generation quality.

Discussion

Main takeaway:

- The diffusion model by itself does not adequately model the class-specific information.
- CFG identifies and enhances class-specific patterns.

4. Xiang Li, Rongrong Wang, Qing Qu. Towards Understanding the Mechanisms of Classifier-Free Guidance. *Neural Information Processing Systems (NeurIPS'25)*, 2025. (**spotlight, top 3.2%**)

Conclusion & Acknowledgement

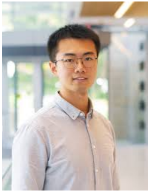
Take-Home Message

- **Training with Synthetic Data:** suffer from model collapse due to generalization-to-memorization transition, and can be mitigated through effective data selection
- **Content Manipulation:** we can leverage low-dimensional subspaces to effectively manipulate the generation
- **Classifier-free Guidance:** we explained why CFG works through contrastive subspaces.

Major References

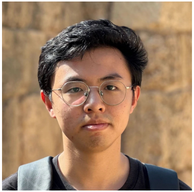
1. Lianghe Shi, Meng Wu, Huijie Zhang, Zekai Zhang, Molei Tao, Qing Qu. A Closer Look at Model Collapse: From a Generalization-to-Memorization Perspective. *Neural Information Processing Systems (NeurIPS'25)*, 2025. (**spotlight, top 3.2%**)
2. Siyi Chen*, Huijie Zhang*, Minzhe Guo, Yifu Lu, Peng Wang, Qing Qu. Exploring Low-Dimensional Subspaces in Diffusion Models for Controllable Image Editing. *Neural Information Processing Systems (NeurIPS'24)*, 2024.
3. Wenda Li, Huijie Zhang, Qing Qu. Shallow Diffuse: Robust and Invisible Watermarking through Low-Dimensional Subspaces in Diffusion Models. *NeurIPS*, 2025 (**spotlight, top 3.2 %**).
4. Xiang Li, Rongrong Wang, Qing Qu. Towards Understanding the Mechanisms of Classifier-Free Guidance. *Neural Information Processing Systems (NeurIPS'25)*, 2025. (**spotlight, top 3.2%**)

Acknowledgement

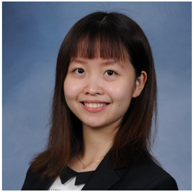


Lianghe Shi
(UMich)

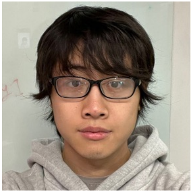
Meng Wu
(UMich)



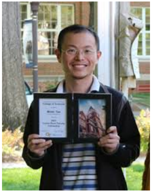
Huijie Zhang
(UMich)



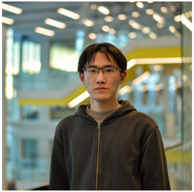
Siyi Chen
(UMich)



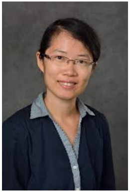
Xiang Li
(UMich)



Molei Tao
(GaTech)



Wenda Li
(UMich)



Rongrong
Wang
(MSU)

Acknowledgement

MICHIGAN INSTITUTE FOR
COMPUTATIONAL DISCOVERY & ENGINEERING
UNIVERSITY OF MICHIGAN

National Science Foundation

amazon

DARPA

Office of Naval Research
ONR
Science & Technology

Google

Thank You!