



Exploring Diffusion Models' Corruption Stage in Few-Shot Fine-tuning and Mitigating with Bayesian Neural Networks

Xiaoyu Wu^{1,*} Jiaru Zhang^{1,*} Yang Hua² Bohan Lyu³ Hao Wang⁴ Tao Song^{1,†} Haibing Guan¹

¹Shanghai Jiao Tong University ²Queen's University Belfast ³Tsinghua University ⁴Stevens Institute of Technology

*Equal contribution [†]Corresponding author Code: github.com/Nicholas0228/BNN-Finetuning-DMs



arXiv:2405.19931

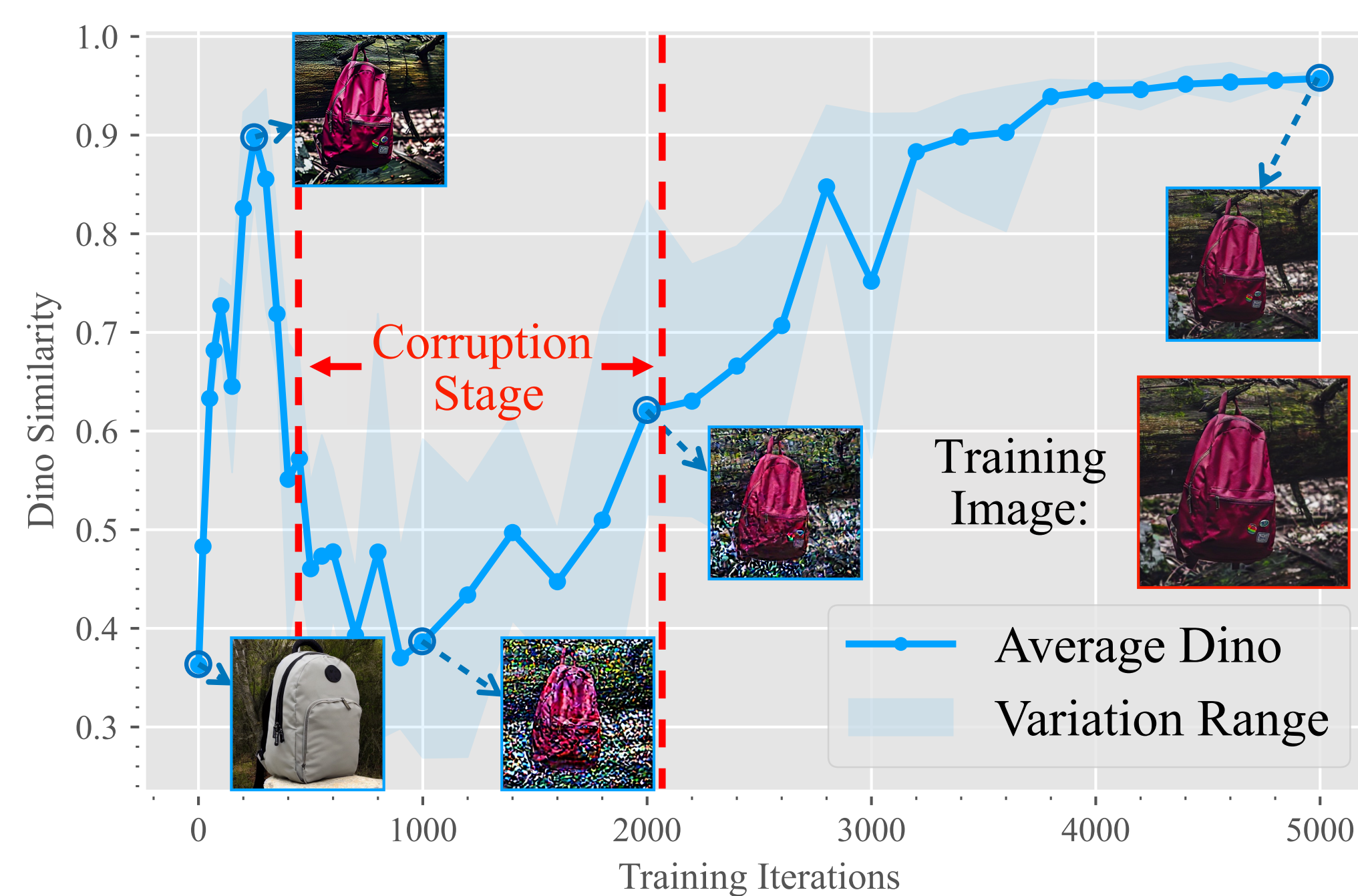


GitHub Code

Motivation: The Corruption Stage

Few-shot fine-tuning (DreamBooth, LoRA, OFT) personalizes Diffusion Models from just a few images. We reveal a surprising phenomenon during fine-tuning:

fidelity **improves** $\rightarrow$ **corrupts (noisy)** $\rightarrow$ **overfits**



Contributions:

- **First** to identify & characterize the *corruption stage* in few-shot fine-tuning of DMs.
- **Explain** it via a heuristic modeling of the fine-tuning dynamics.
- **Mitigate** it with **Bayesian Neural Networks**, with no extra inference cost and compatible with existing methods.

$\Rightarrow$ **Root cause:** the learned distribution is **narrowed** onto the few training images.

Image fidelity (Dino) rises, then **collapses into noisy "corruption"**, then recovers only by overfitting to the training images.

Quantitative Results

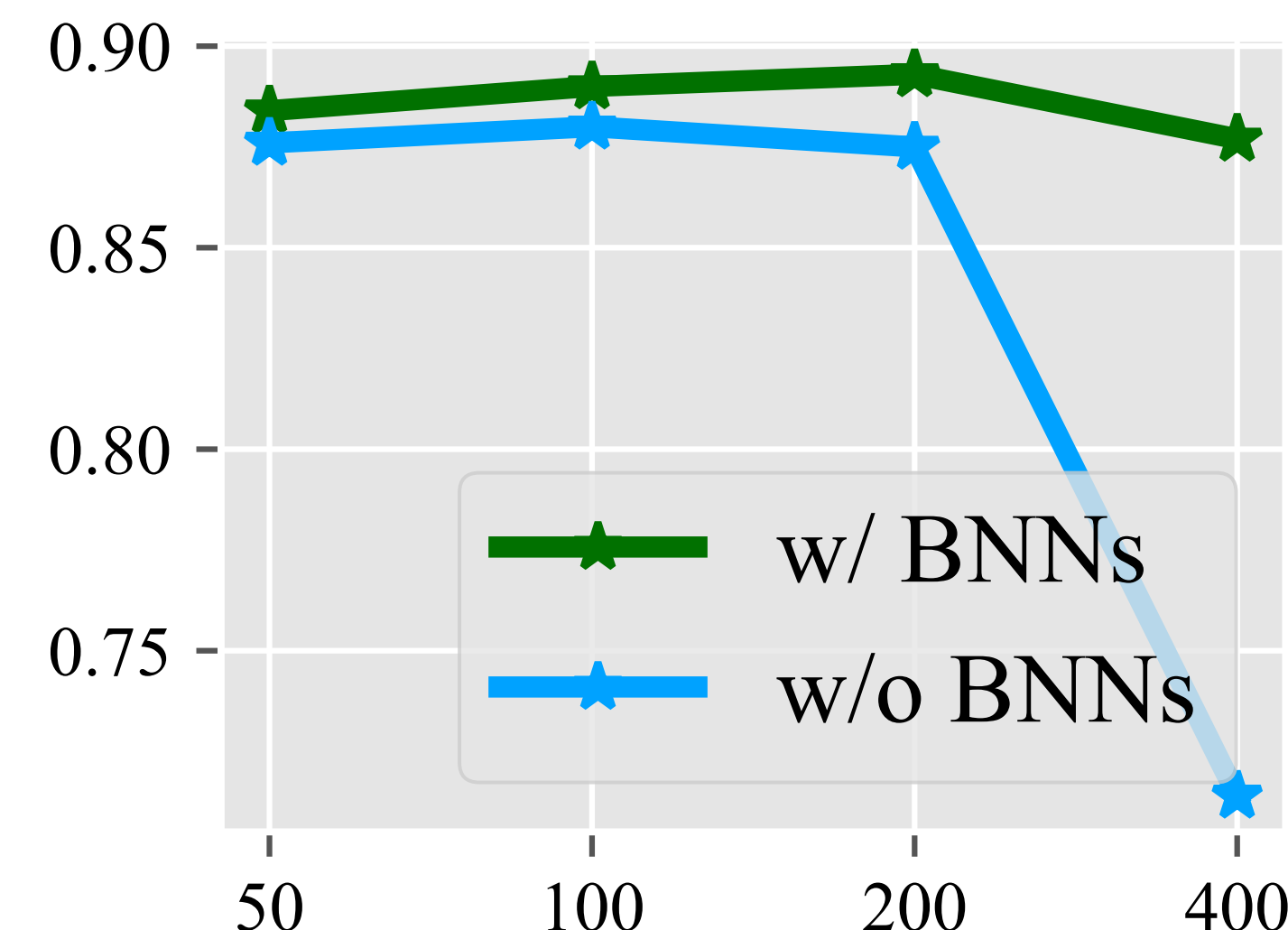
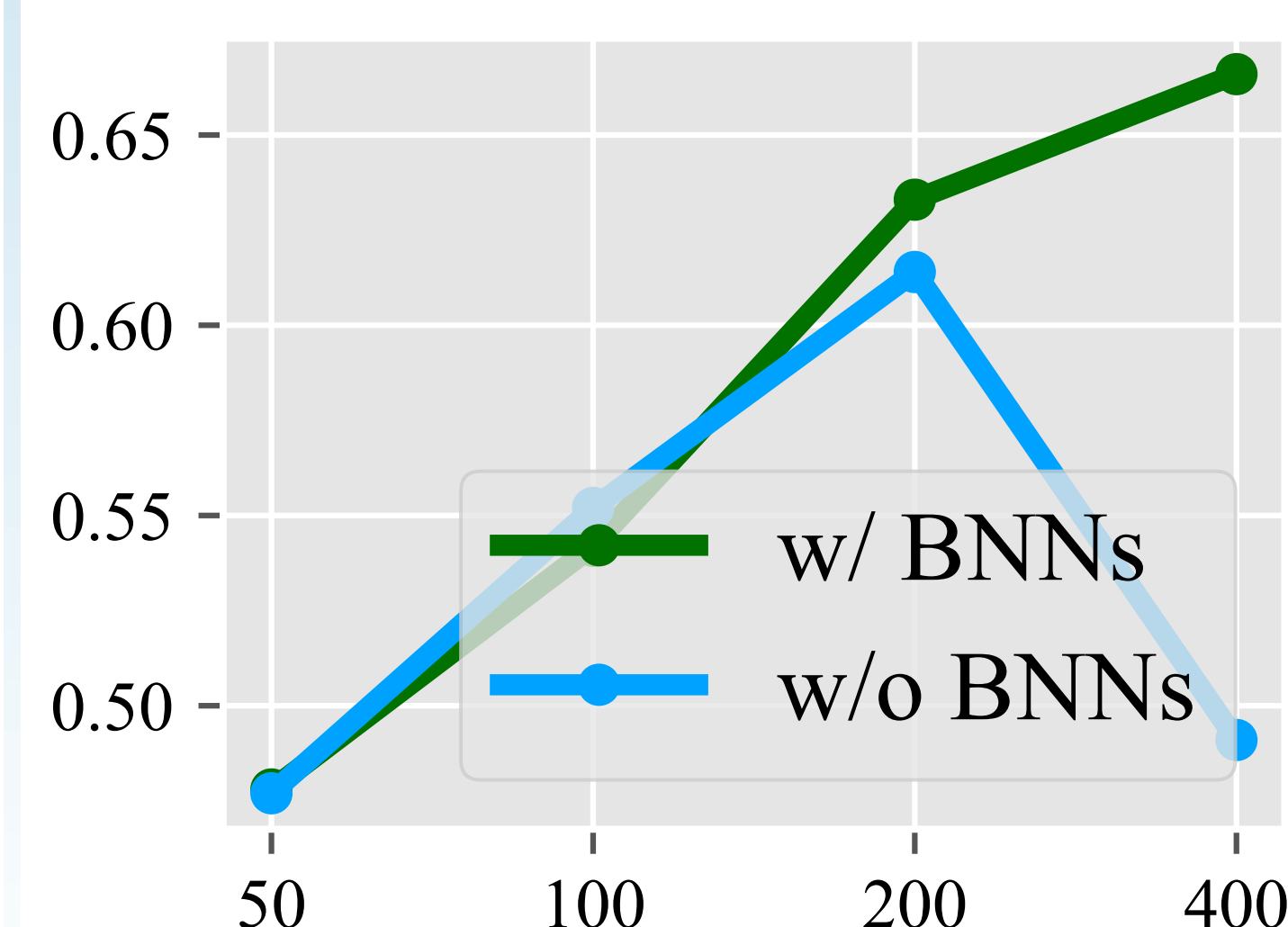
BNNs improve every metric, across all methods and both tasks (SD v1.5, avg. of 5 seeds).

Method	Object (DreamBooth)			Subject (CelebA-HQ)		
	Dino	Clip-I	IQA	Dino	Clip-I	IQA
DreamBooth	0.614	0.771	0.875	0.642	0.723	0.789
+ BNNs	0.633	0.785	0.893	0.696	0.757	0.811
LoRA	0.542	0.722	0.864	0.602	0.656	0.804
+ BNNs	0.618	0.769	0.890	0.604	0.663	0.824
OFT	0.649	0.786	0.861	0.675	0.728	0.784
+ BNNs	0.661	0.791	0.884	0.696	0.743	0.798

Dino / Clip-I: fidelity IQA: image quality. Text fidelity (Clip-T) & diversity (Lpips) also improve.

User study (101 participants): images from DMs with BNNs are preferred in **almost all** cases.

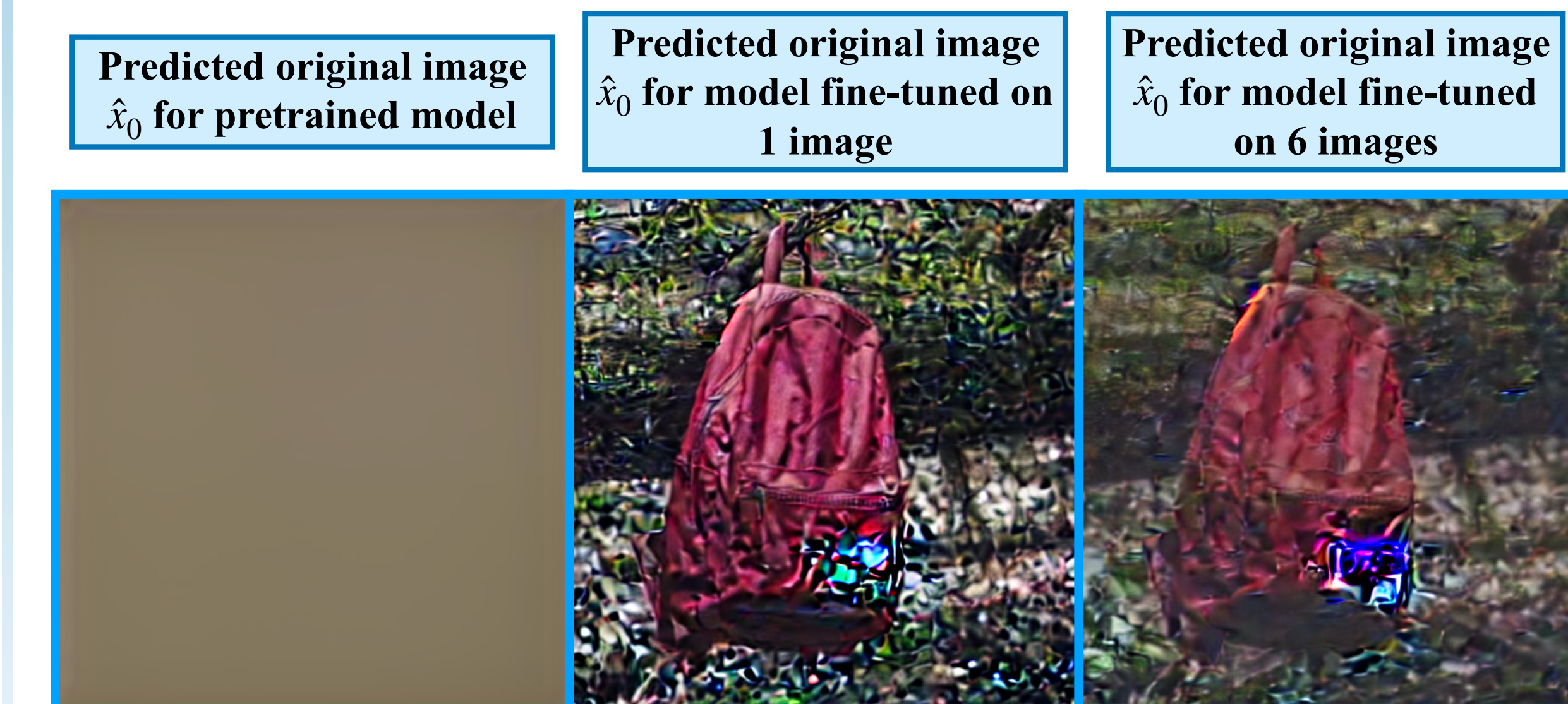
Generalizes broadly — consistent gains across DMs, training steps & # images.



Our Idea: Broaden the Learned Distribution with BNNs

Why does corruption happen?

Few-shot fine-tuning **narrows** the model's learned distribution onto the few training images. Even when asked to denoise a *blank* input, a fine-tuned DM "hallucinates" a training image, indicating its distribution has collapsed.



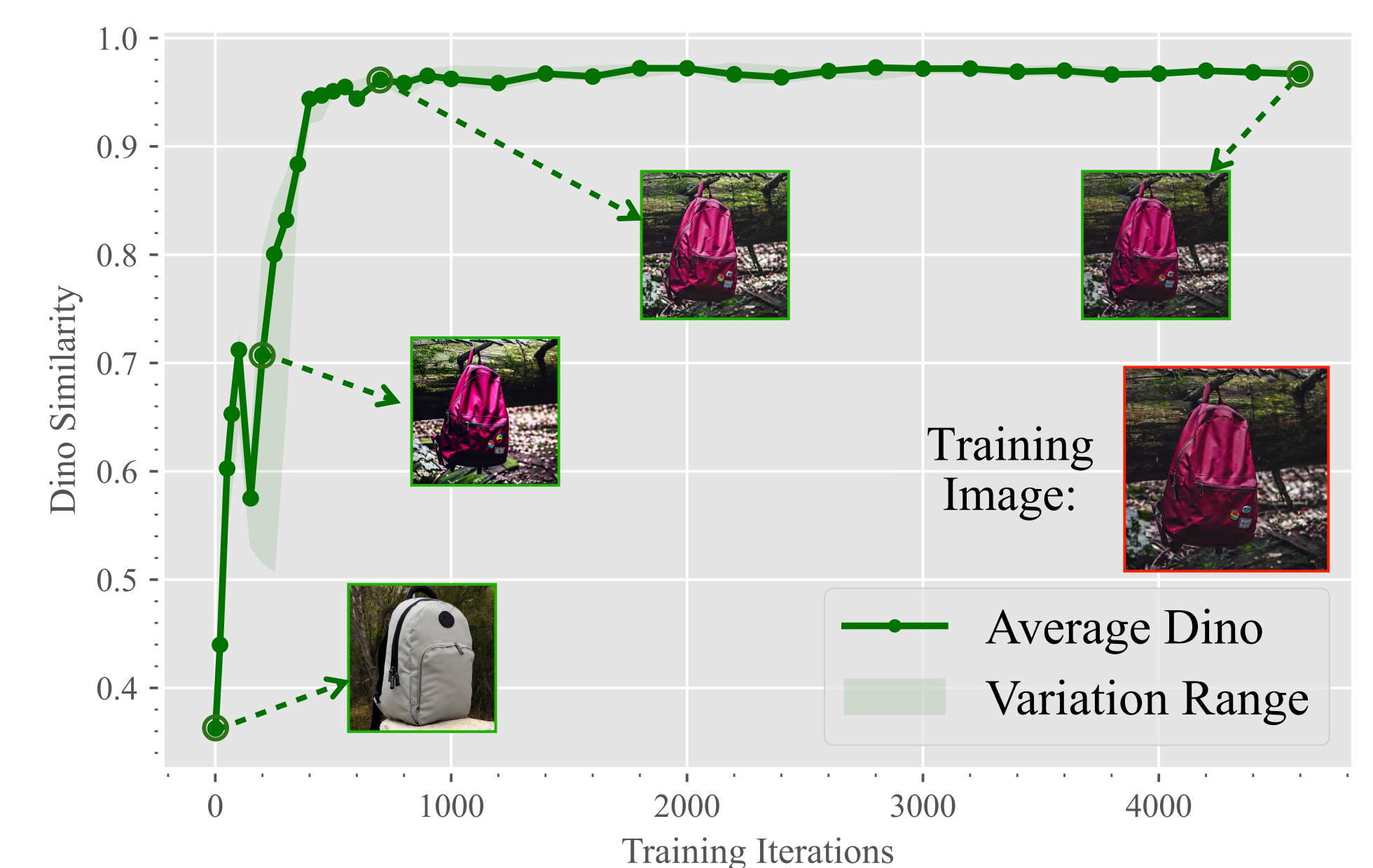
Denosing a blank input: the pretrained DM stays blank; fine-tuned DMs re-generate a training image.

$\Rightarrow$ the learned distribution has **collapsed** onto the training set, so the model can no longer denoise $\Rightarrow$ **corruption**.

Our fix: Bayesian Neural Networks.

Treat the weights as **distributions** ($\theta \sim \mathcal{N}(\mu, \sigma^2)$) rather than fixed values. This **implicitly broadens** the learned distribution and counteracts corruption.

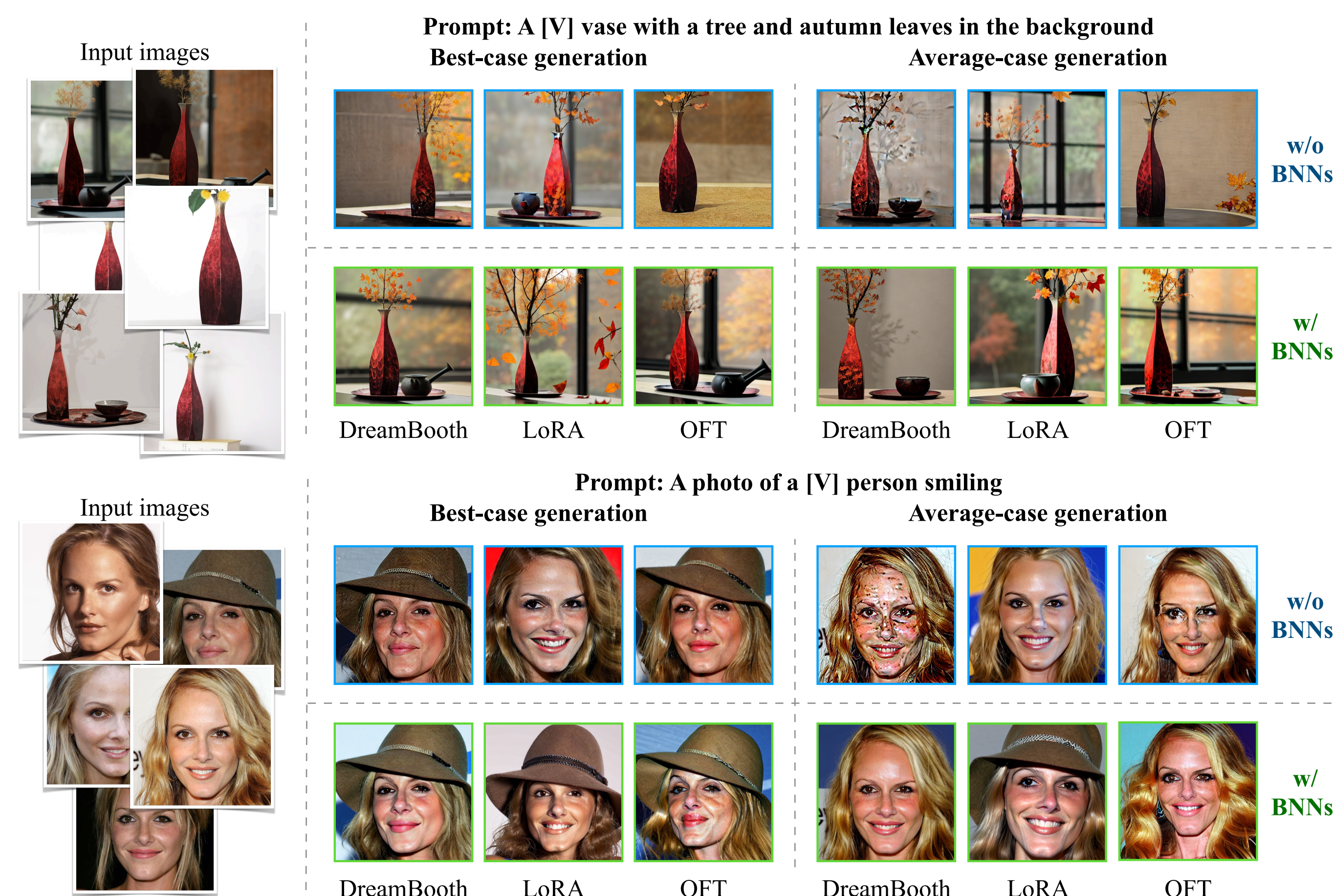
- ✓ **No extra inference cost** (use the mean μ at test time)
- ✓ **Compatible** with DreamBooth / LoRA / OFT
- ✓ Only a **subset of layers** needs to be Bayesian



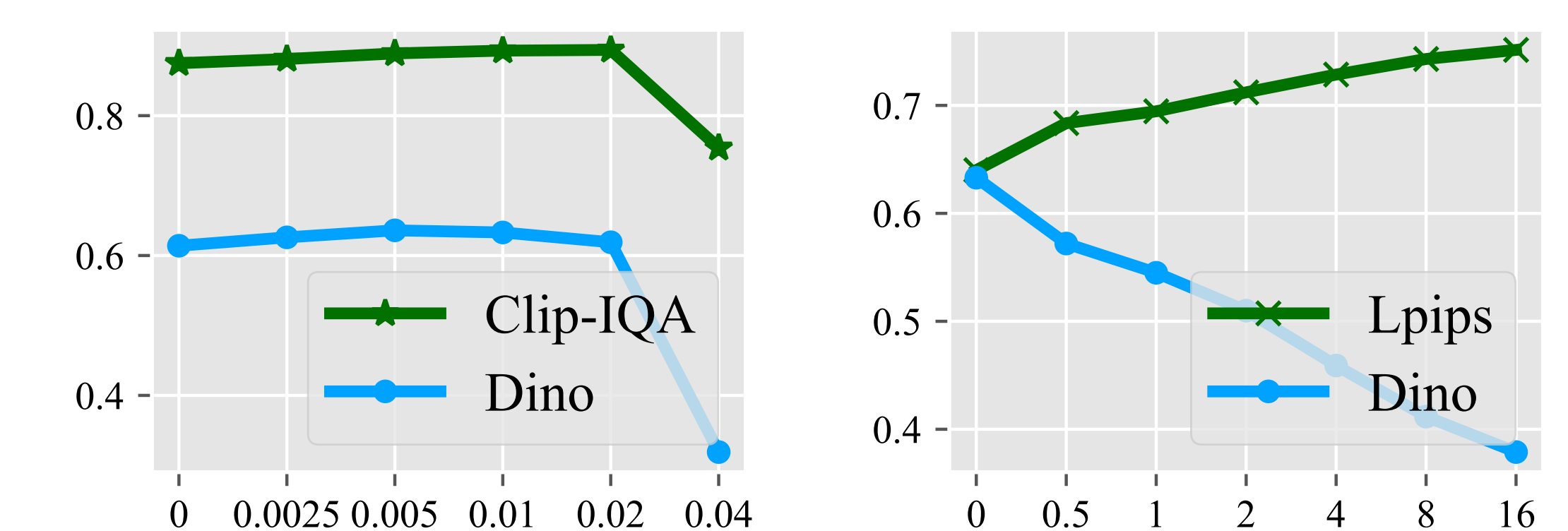
With BNNs the fidelity rises smoothly. **No corruption stage.**

Qualitative Results & Ablation

BNNs **remove corruption**, giving **sharper, more faithful and more diverse** generations across DreamBooth / LoRA / OFT (object- & subject-driven).



Ablation study.



A **moderate** randomness σ_θ is best (too large $\Rightarrow$ collapse); λ trades **diversity** (Lpips) for **fidelity** (Dino).

Where to apply BNNs?

- Applying BNNs only to **normalization layers** modifies just $\sim 0.02\%$ of parameters, yet still performs strongly.
- Cross-attention layers $\Rightarrow$ higher **text** fidelity; up-block $\Rightarrow$ good trade-off.

Cost. BNNs add training-time memory/time only; **inference is unchanged**.

Takeaway: The *corruption stage* stems from a narrowed learned distribution. **BNNs broaden it**, improving fidelity, quality and diversity at **zero extra inference cost**.