



Efficient Inference for Flow Matching via **Unified Path** CFG

Kosuke Ukita , Tsuyoshi Okita

Kyushu Institute of Technology



Background

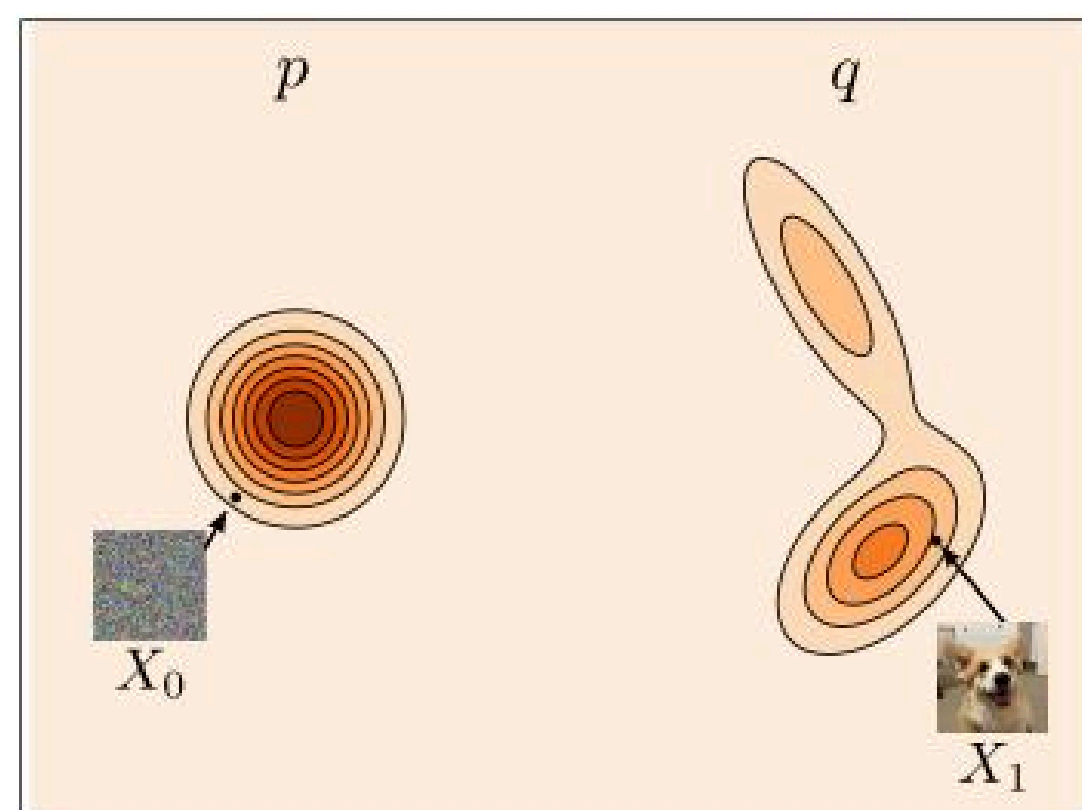
| Background - Flow Matching

Flow Matching is Generative model

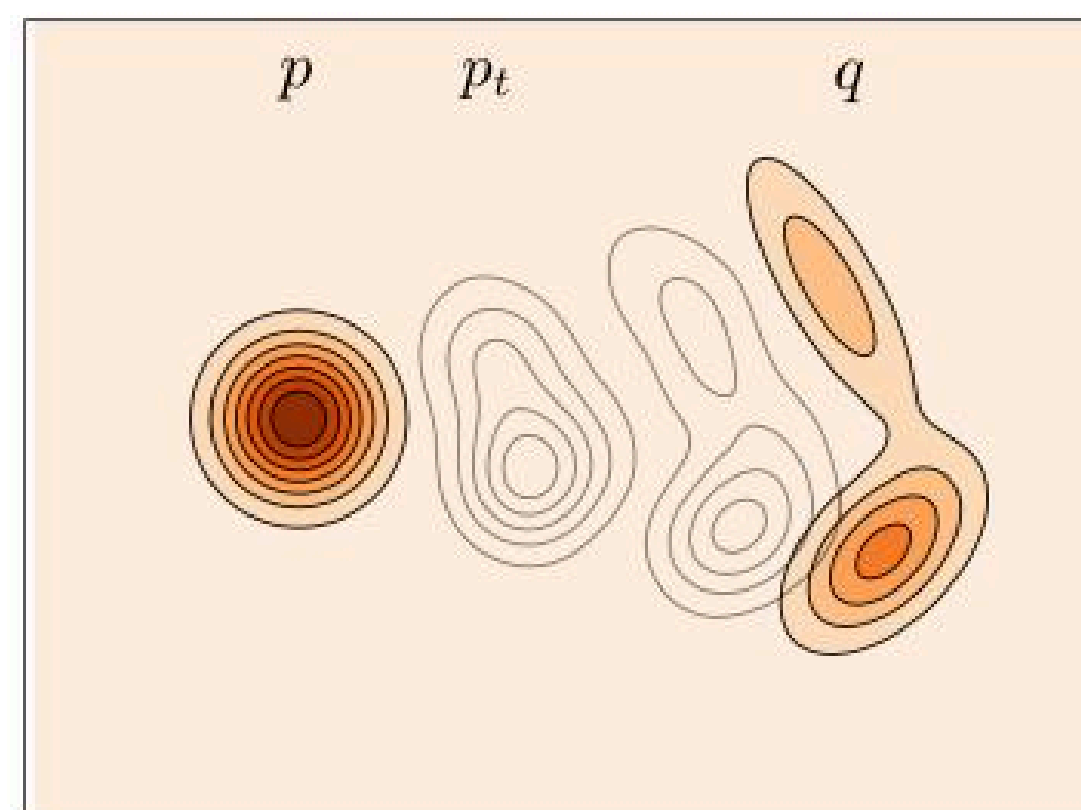
| Flow Matching for image generation



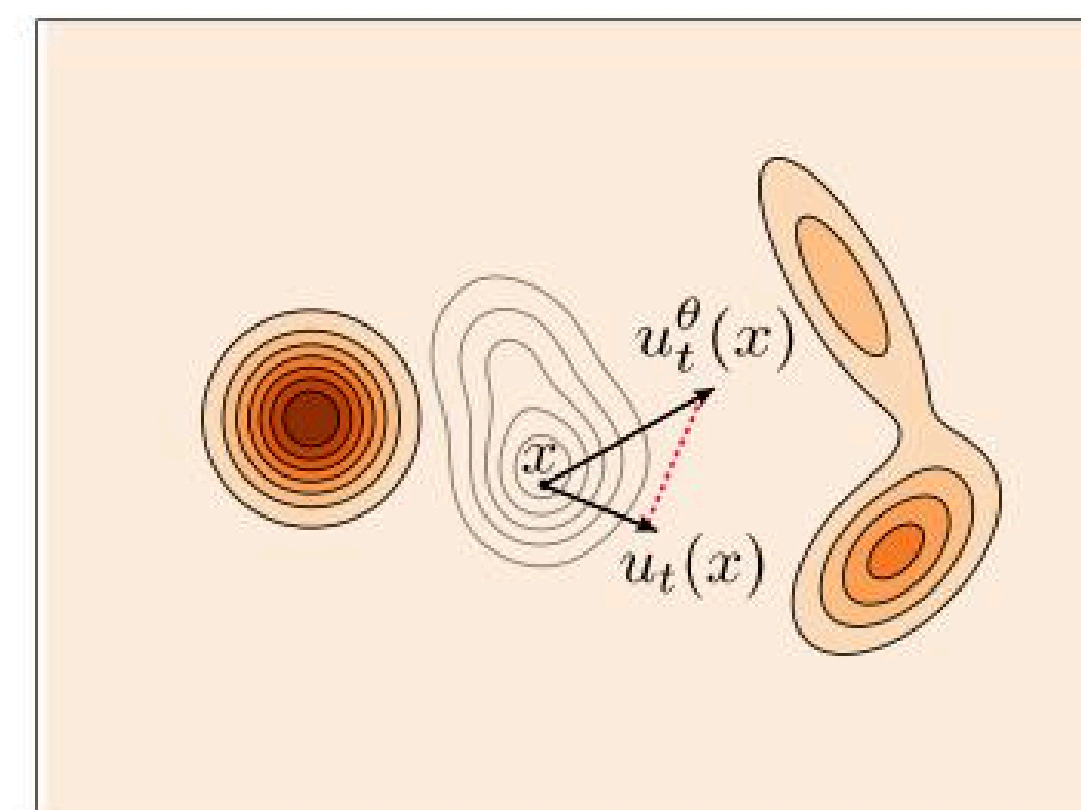
[Nanye+, 2024]



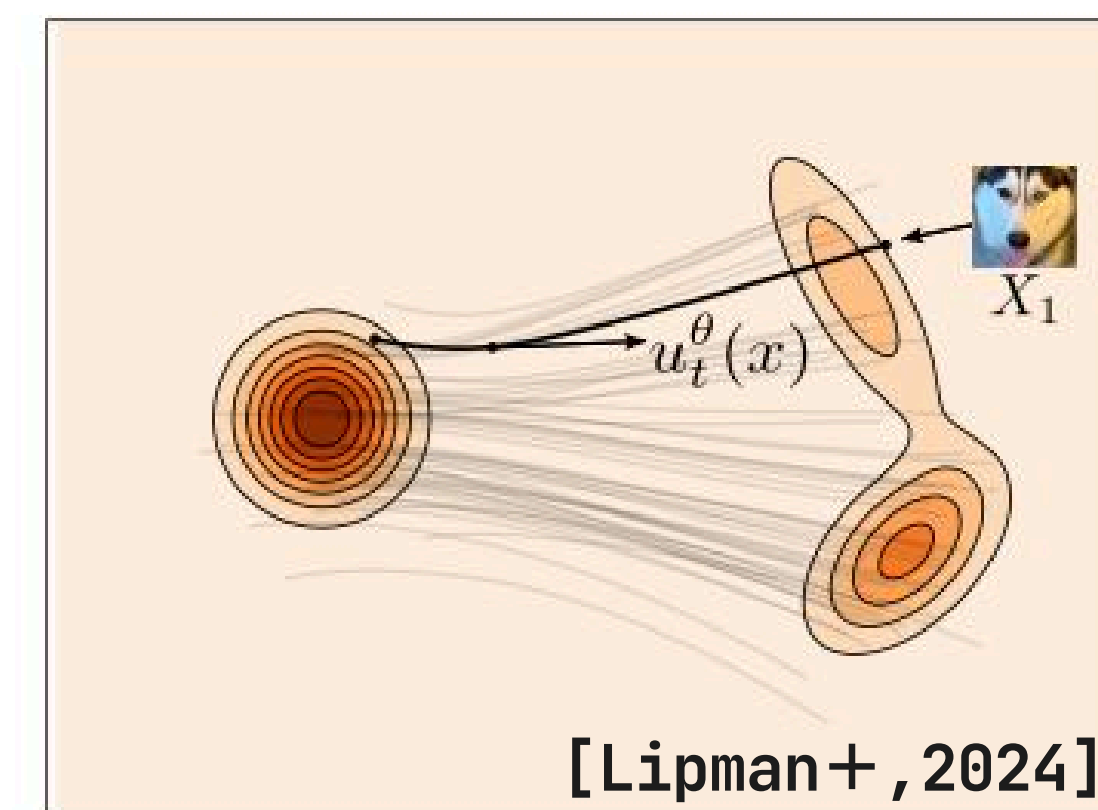
(a) Data.



(b) Path design.



(c) Training.



(d) Sampling.

Flow Matching learns to move **noise** $\rightarrow$ **data** along straight, deterministic paths (ODEs)

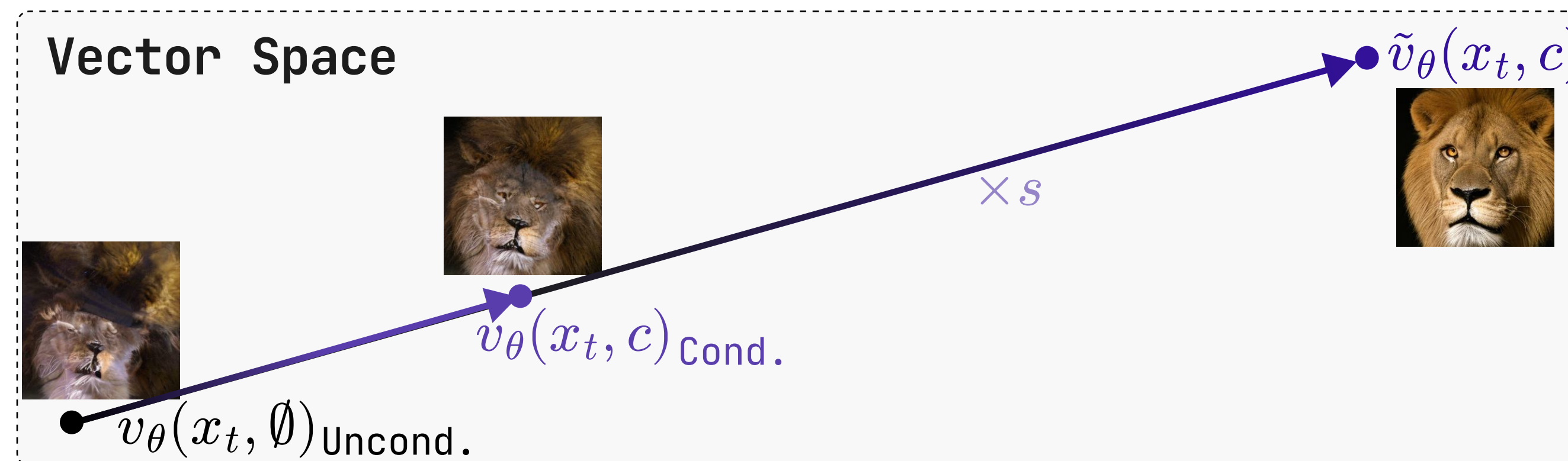
| Background - Classifier-Free Guidance (CFG)

CFG is essential for quality — but doubles inference cost

| CFG in Flow Matching

CFG computes a modified velocity field using both **conditional** and **unconditional** predictions:

$$\underbrace{\tilde{v}_{\theta}(x_t, c)}_{\text{Unconditional}} = \underbrace{v_{\theta}(x_t, \emptyset)}_{\text{Unconditional}} + \underbrace{s}_{\text{Guidance scale}} \cdot (\underbrace{v_{\theta}(x_t, c)}_{\text{Conditional}} - \underbrace{v_{\theta}(x_t, \emptyset)}_{\text{Unconditional}})$$



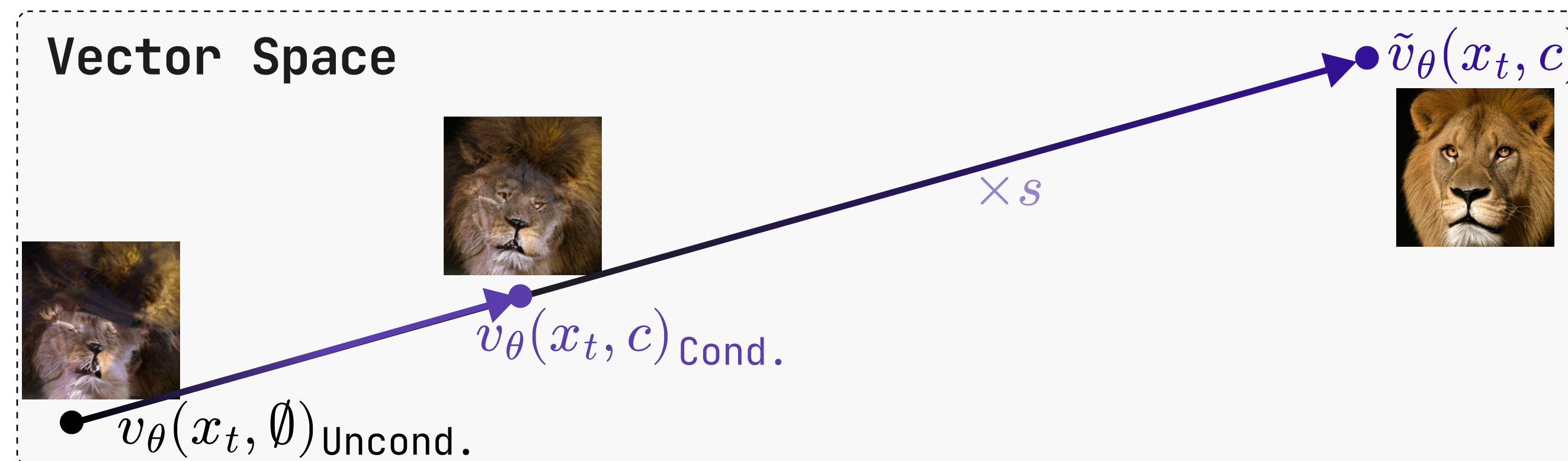
| Background - Classifier-Free Guidance (CFG)

CFG is essential for quality — but doubles inference cost

| CFG in Flow Matching

CFG computes a modified velocity field using both **conditional** and **unconditional** predictions:

$$\underbrace{\tilde{v}_{\theta}(x_t, c)}_{\text{Unconditional}} = \underbrace{v_{\theta}(x_t, \emptyset)}_{\text{Unconditional}} + \underbrace{s}_{\text{Guidance scale}} \cdot (\underbrace{v_{\theta}(x_t, c)}_{\text{Conditional}} - \underbrace{v_{\theta}(x_t, \emptyset)}_{\text{Unconditional}})$$



→ This requires running the network **TWICE** per step:

$$NFE_{CFG} = 2 \times \frac{NFE}{\text{Number of Function Evaluations}}$$

Related Works

Existing methods still require large tuning costs

| **CFG distillation** (Guidance Distillation[Meng+,2023], FLUX.1 [Black Forest Labs,2024])

- Guidance scale fixed during training → **no flexibility** at test time
- **Requires full fine-tuning** of large backbone → not practical

Existing methods still require large tuning costs

| CFG distillation (Guidance Distillation[Meng+,2023], FLUX.1 [Black Forest Labs,2024])

- Guidance scale fixed during training → **no flexibility** at test time
- **Requires full fine-tuning** of large backbone → not practical

| Adapter-based distillation (PnP [Hsiao+,2024], AGD [Jensen+,2025])

- Promising — **no backbone tuning needed**
- But needs **~40%** adapter size to work well → **tuning cost grows as backbone gets larger**
- 1% adapter has been tried — but FID **drops significantly**

Existing methods still require large tuning costs

| CFG distillation (Guidance Distillation[Meng+,2023], FLUX.1 [Black Forest Labs,2024])

- Guidance scale fixed during training → **no flexibility** at test time
- **Requires full fine-tuning** of large backbone → not practical

| Adapter-based distillation (PnP [Hsiao+,2024], AGD [Jensen+,2025])

- Promising — **no backbone tuning needed**
- But needs **~40%** adapter size to work well → **tuning cost grows as backbone gets larger**
- 1% adapter has been tried — but FID **drops significantly**

→ By targeting **only the unconditional prediction**, we achieve high-quality CFG with **<1% adapter**

Hypothesis

| Hypothesis

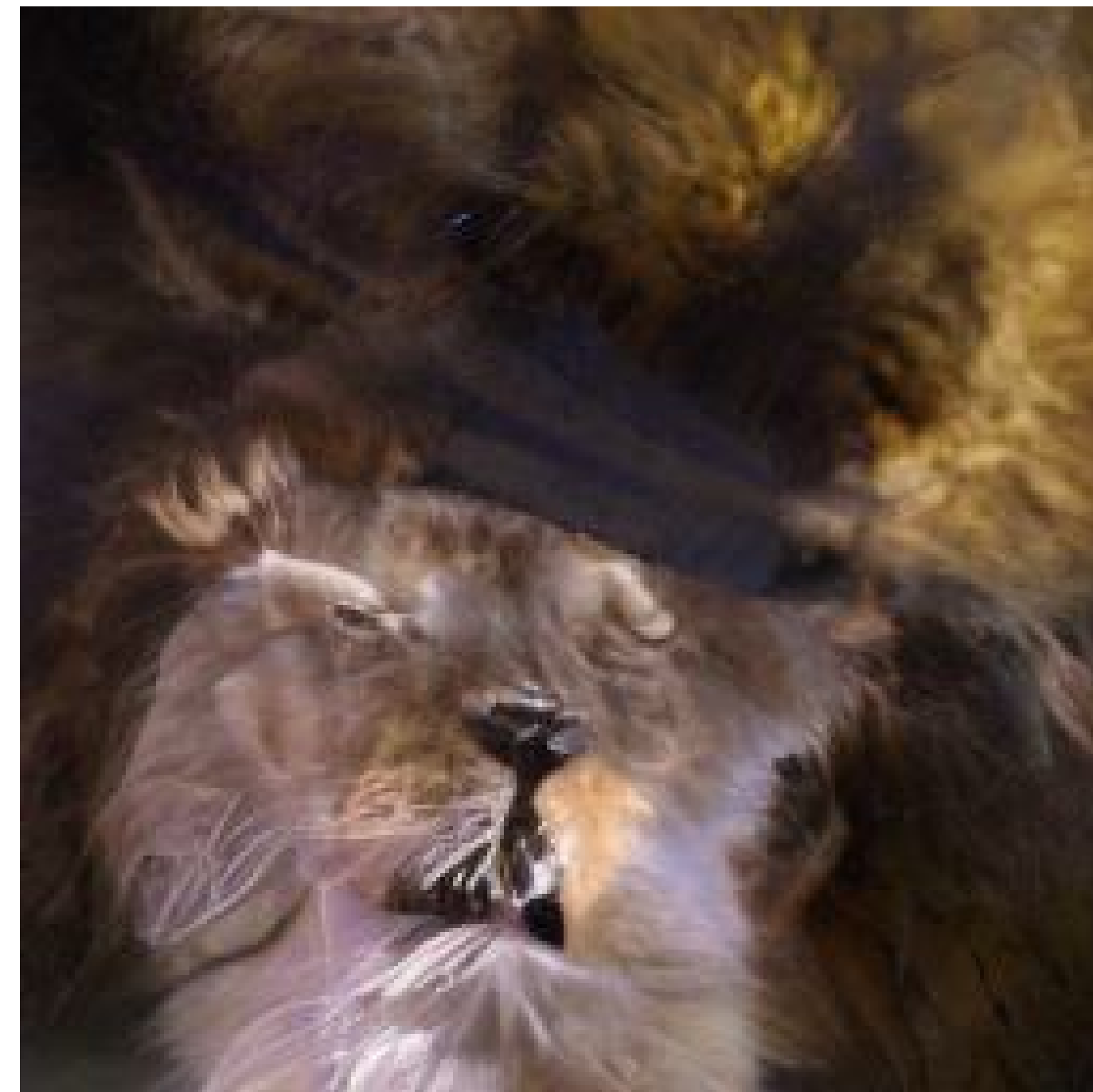
Conditional features already contain enough information **to reconstruct unconditional** predictions

| Evidence 1: Generated Image at same seed $\rightarrow$ nearly identical outputs

Conditional



Unconditional



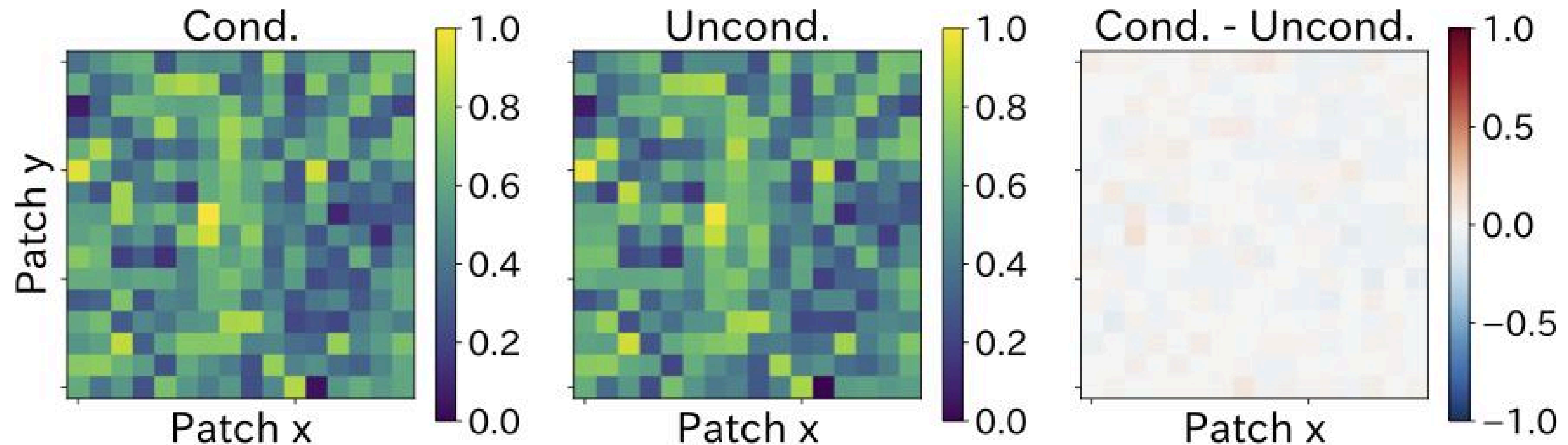
$\rightarrow$ The same structural information is shared

| Hypothesis

Conditional features already contain enough information **to reconstruct unconditional** predictions

| Evidence 1: Generated Image at same seed $\rightarrow$ nearly identical outputs

| Evidence 2: Attention maps are almost identical – difference ≈ 0



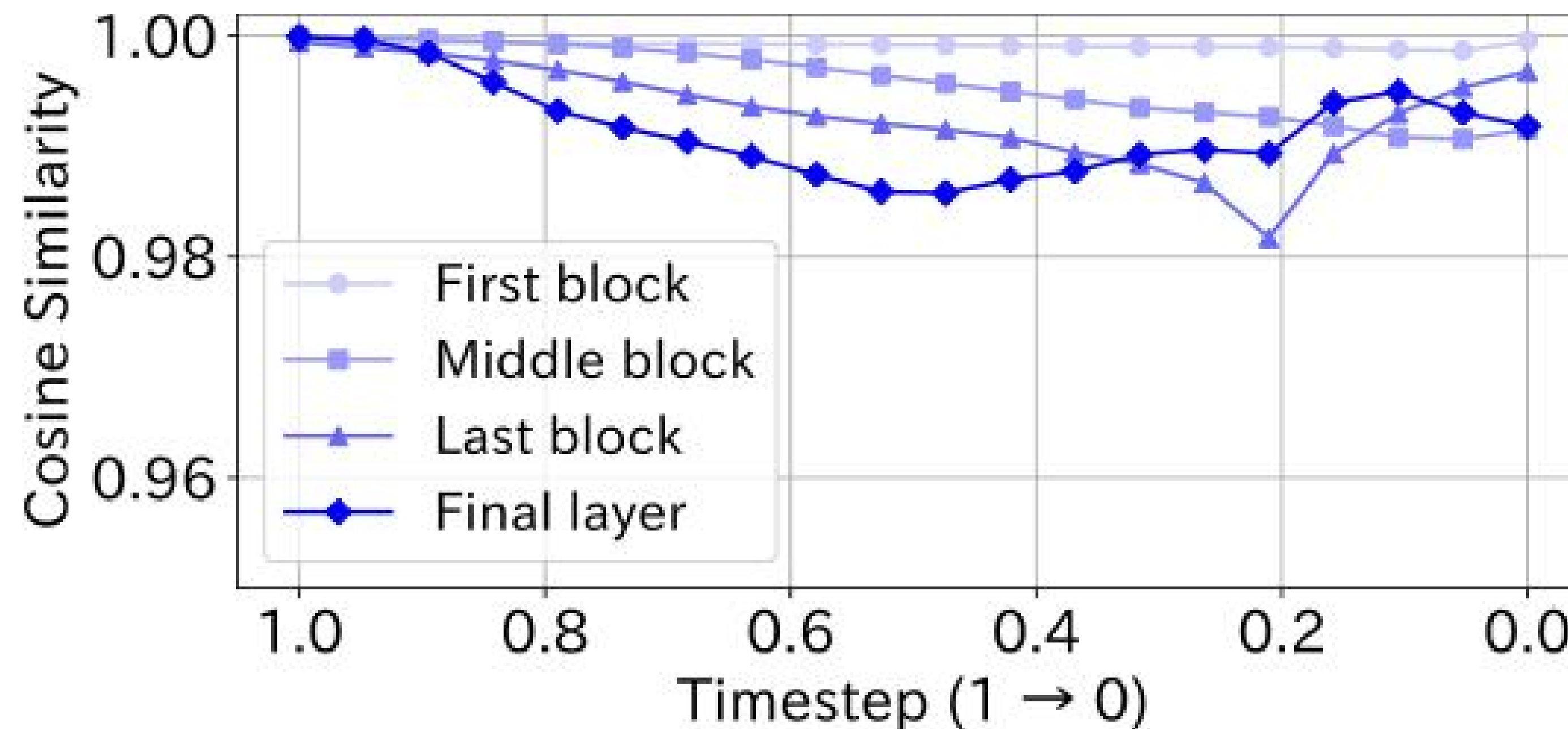
| Hypothesis

Conditional features already contain enough information **to reconstruct unconditional** predictions

| Evidence 1: Generated Image at same seed $\rightarrow$ nearly identical outputs

| Evidence 2: Attention maps are almost identical – difference ≈ 0

| Evidence 3: Cosine similarity > 0.98 across all layers and timesteps



| Hypothesis

Conditional features already contain enough information **to reconstruct unconditional** predictions

| **Evidence 1:** Generated Image at same seed $\rightarrow$ nearly identical outputs

| **Evidence 2:** Attention maps are almost identical – difference ≈ 0

| **Evidence 3:** Cosine similarity > 0.98 across all layers and timesteps

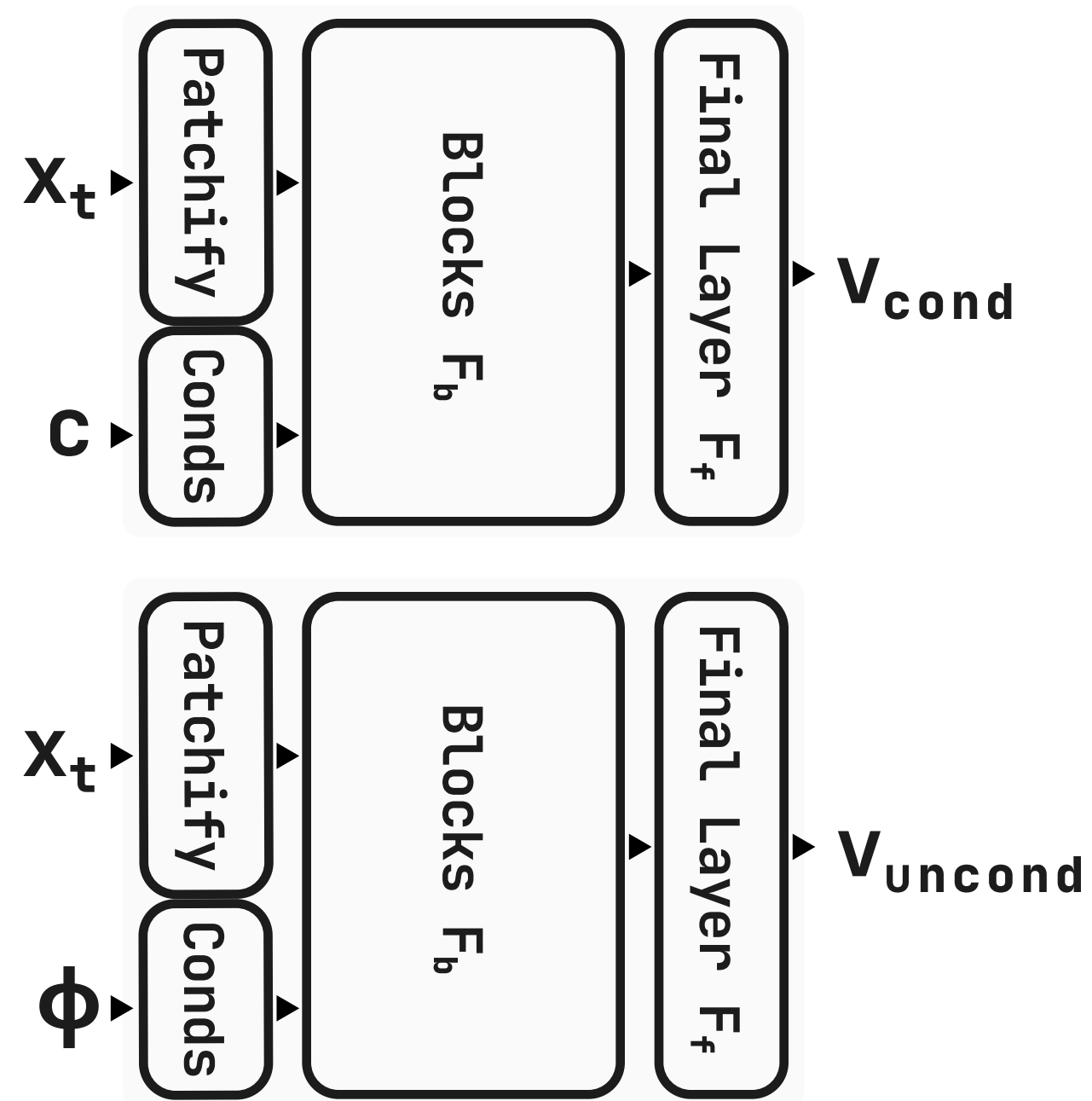
$\rightarrow$ A **small adapter** ($<1\%$) is enough to bridge the gap

Proposed Method

| Proposed Method - cfgUP

CFG in a single forward pass — with a <1% adapter

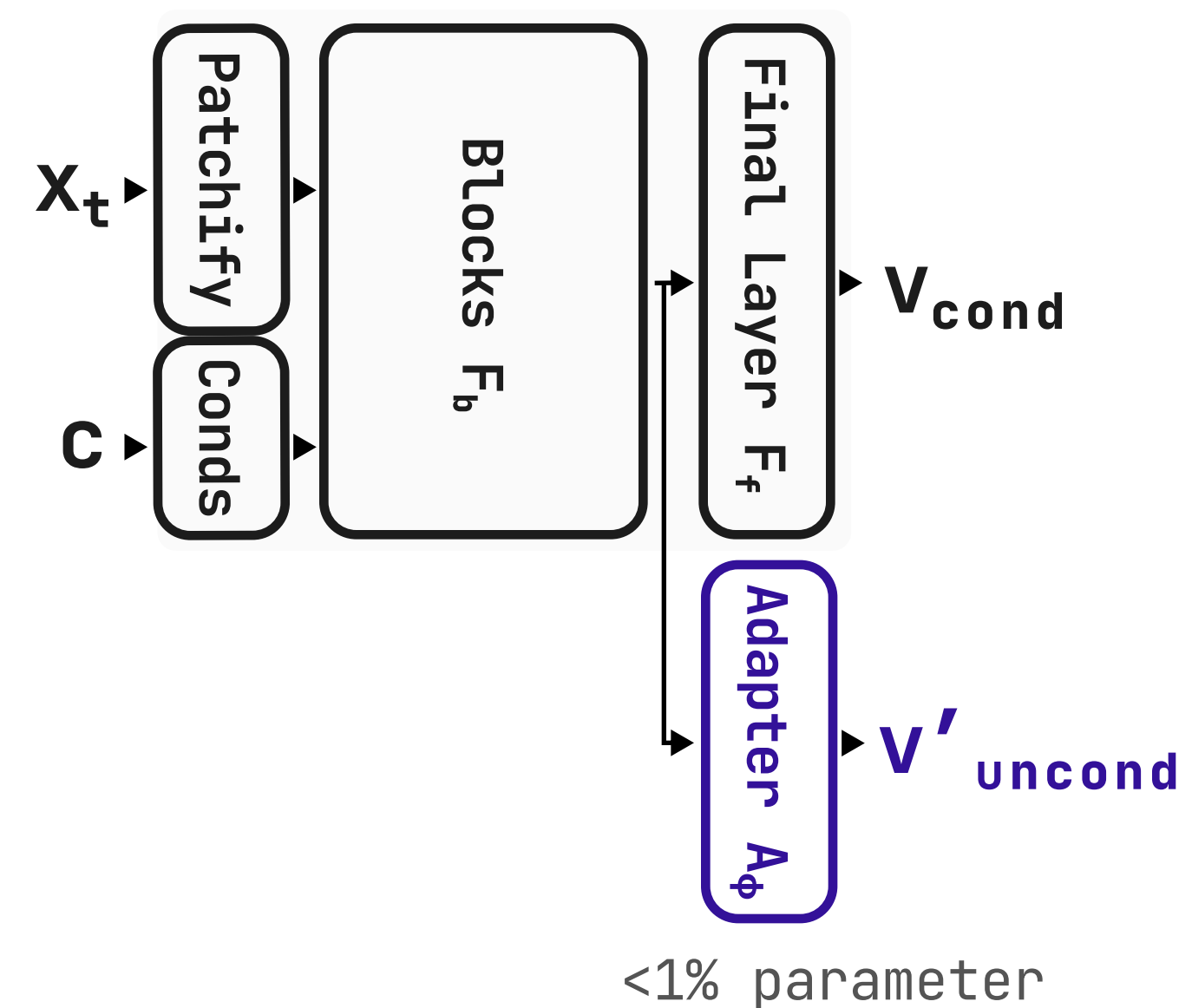
| Baseline (standard CFG)



Run full network **twice** per step

- once with condition $c \rightarrow V_{\text{cond}}$
- once without condition $\rightarrow V_{\text{uncond}}$

| cfgUP (ours)



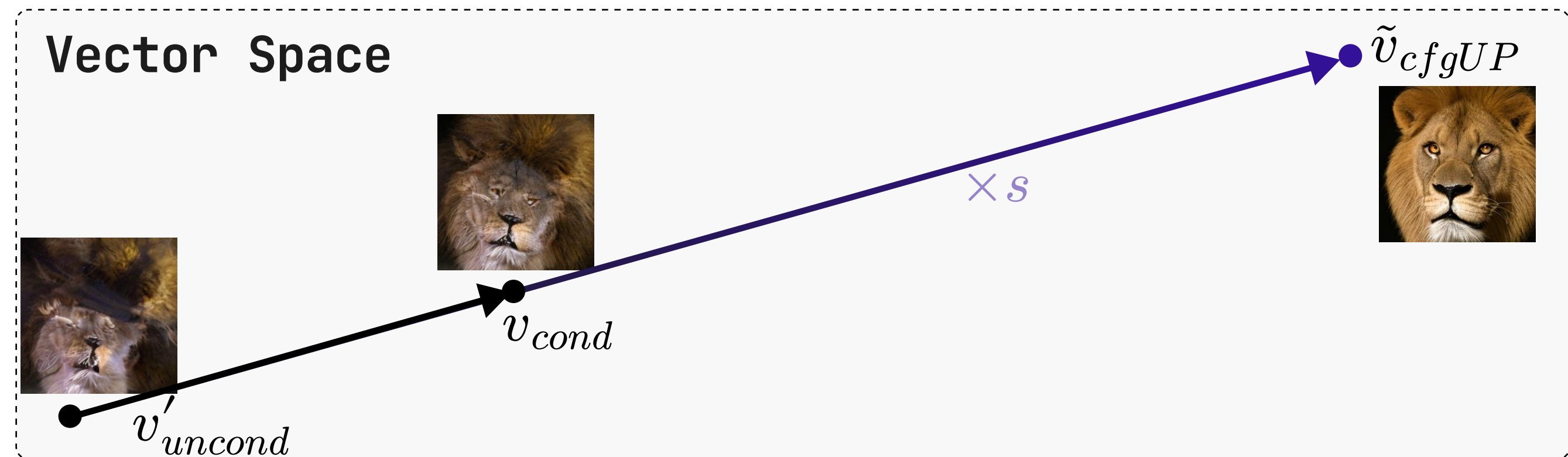
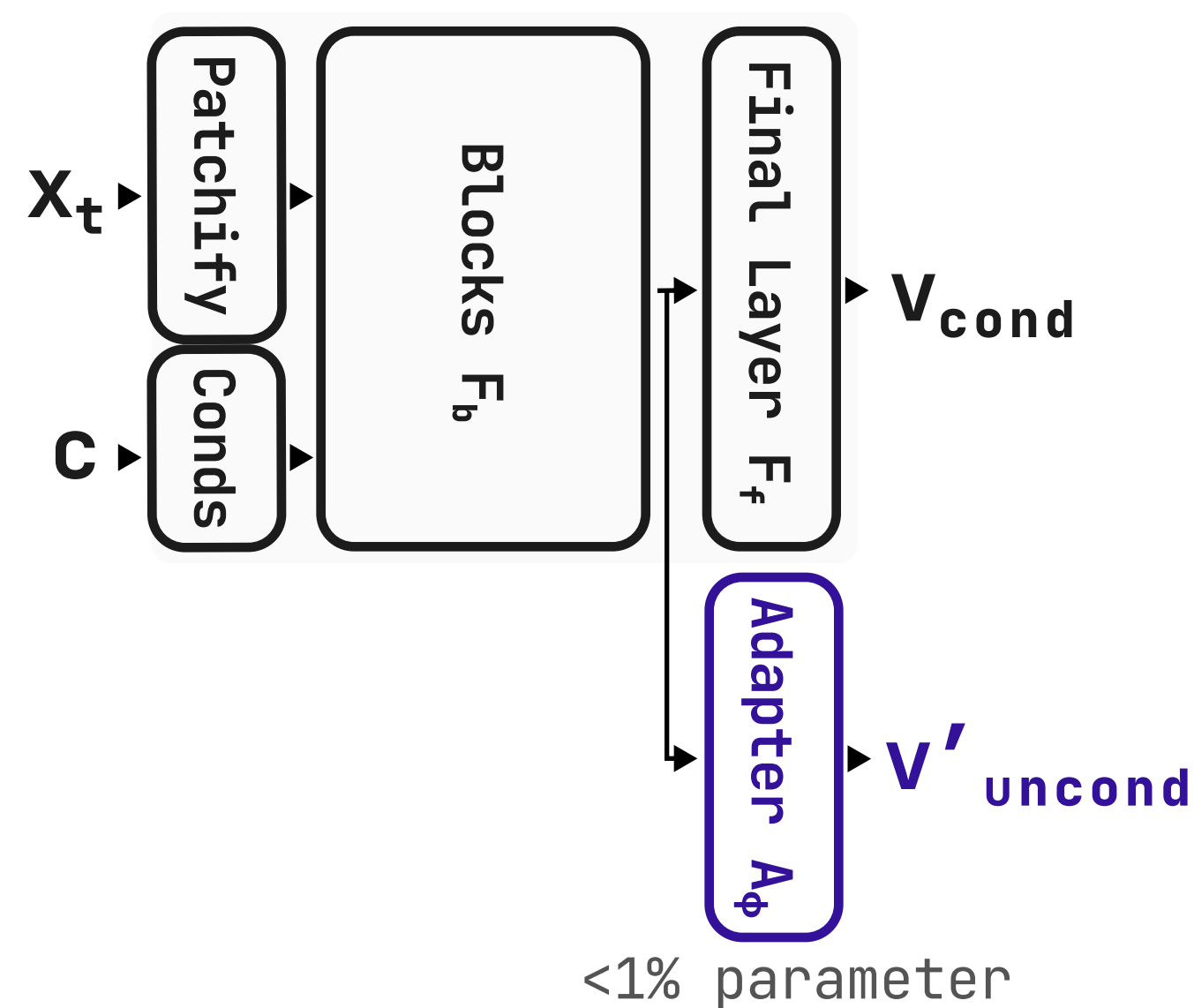
Run full network **once**

- Blocks F_b (frozen)
- Final layer $F_f \rightarrow V_{\text{cond}}$
- Adapter A_ϕ (<1%) $\rightarrow V'_{\text{uncond}}$

| Proposed Method - cfgUP

CFG in a single forward pass — with a <1% adapter

| cfgUP (ours)



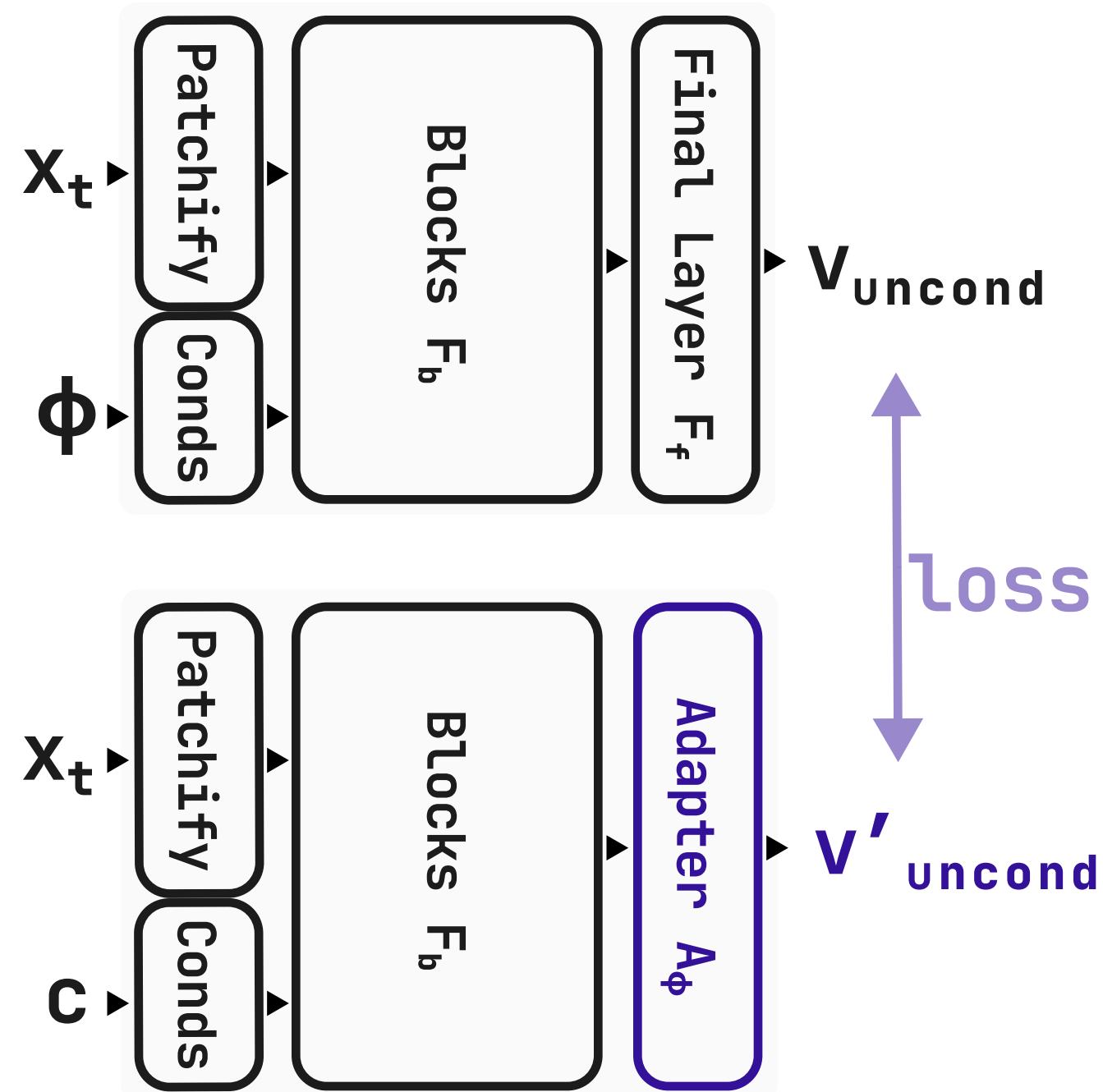
$$\tilde{v}_{cfgUP} = v'_{uncond} + s \cdot (v_{cond} - v'_{uncond})$$

- ✓ **No backbone tuning**
- ✓ Guidance scale **flexible** at test time
- ✓ Adapter: **only 0.39%-0.94%** of parameters

| Proposed Method - cfgUP training

Only the adapter is trained — backbone stays frozen

| Training objective

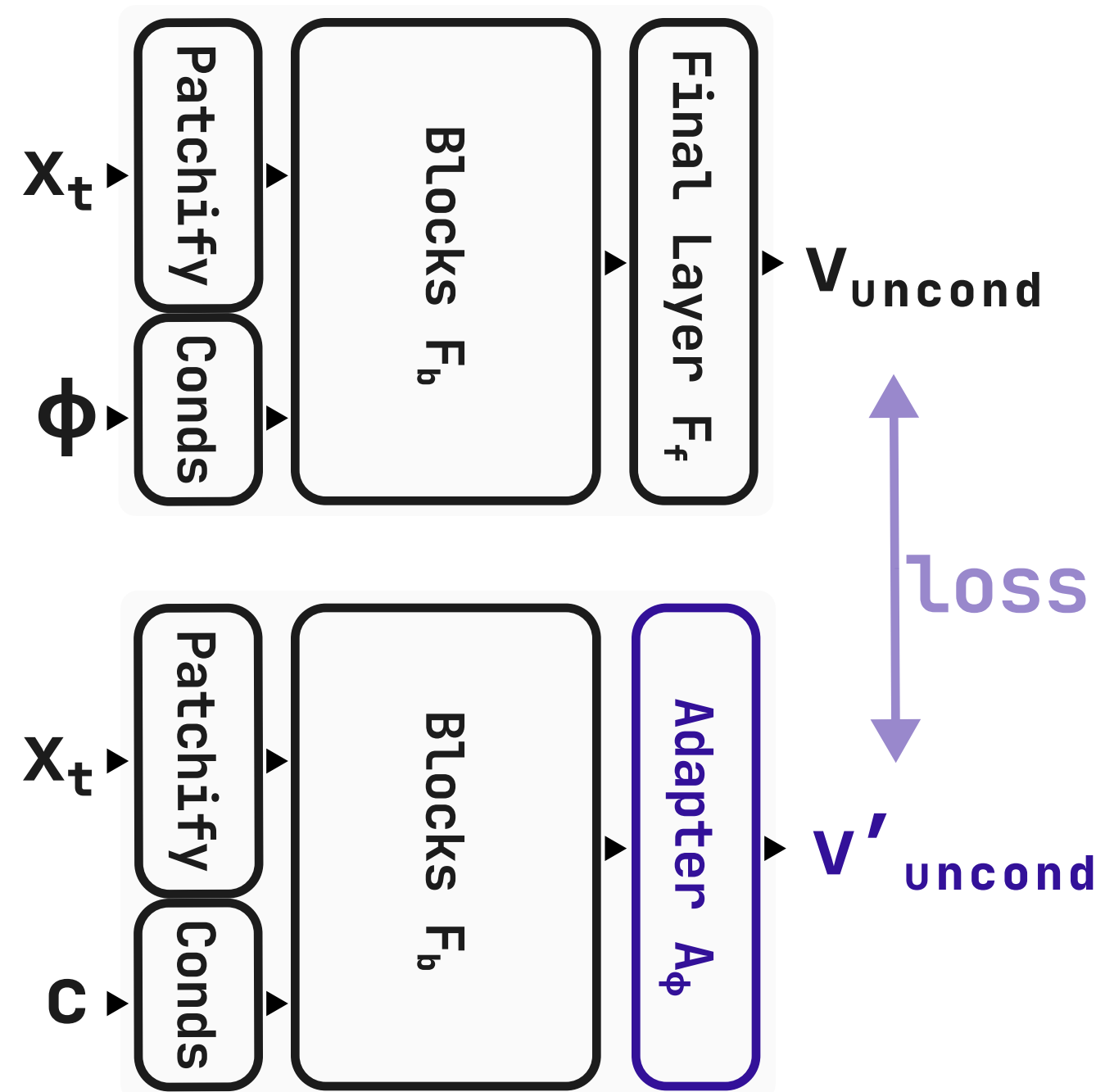


Minimize the gap between predicted and true unconditional output:

$$\mathcal{L}_{\text{adapter}} = \mathbb{E}_{x_t, c} \left[\left\| v_{\hat{\theta}}(x_t, \emptyset) - A_{\phi}(h, c) \right\|^2 \right] \\ h = F_b(x_t, c)$$

Only the adapter is trained — backbone stays frozen

| Training objective



Minimize the gap between predicted and true unconditional output:

$$\mathcal{L}_{\text{adapter}} = \mathbb{E}_{x_t, c} \left[\left\| v_{\hat{\theta}}(x_t, \emptyset) - A_{\phi}(h, c) \right\|^2 \right] \\ h = F_b(x_t, c)$$

→ **Only adapter parameters** are updated

Model size / Patch size	Backbone params	Adapter params	Adapter ratio
S/2	32.9M	0.308M	0.93%
B/2	130M	1.20M	0.92%
L/2	458M	2.13M	0.46%
XL/2	675M	2.69M	0.39%

Adapter: only 0.39%-0.94% of parameters

Experiments

| Experiments - Training Cost

10% of data, 1 GPU, ~10 hours

| Setup

Backbone : SiT-XL/2 [Nanye+, 2024]

Dataset : ImageNet 256x256 [Russakovsky+, 2015]

Optimizer : AdamW (lr=1e-4, batch=32)

GPU : Single NVIDIA A100 (40GB, peak VRAM 16.5 GB)

| Data Efficiency

Training data	FID ↓	sFID ↓	Precision ↑	Recall ↑	Training time ↓
10% subset	<u>3.47</u>	8.60	.8081	<u>.5630</u>	<u>9.86</u> hours
100% full	<u>3.47</u>	<u>4.92</u>	<u>.8163</u>	.5474	27.66 hours

→ Same FID with **65% less** training time

Experiments - Generation Quality

Large quality recovery from unguided baseline

SiT-XL/2
(cfg=1.5, ODE)



cfgUP
(ours)



Model	FID ↓	IS ↑	Precision ↑	Recall ↑
SiT-XL/2 (no CFG)	9.35	126.06	.67	<u>.68</u>
SiT-XL/2 (cfg=1.5, ODE)	<u>2.15</u>	158.09	<u>.81</u>	.60
SiT-XL/2 cfgUP (ours)	3.16	<u>211.46</u>	.80	.58

| Experiments - Inference Efficiency

NFE halved — under 1 second per image

Method	ODE solver	NFE ↓	Latency ↓	Speedup ↑
SiT-XL/2 (cfg=1.5)	Euler	500	17.71 s	1.0x
SiT-XL/2 (no CFG)	Euler	250	8.86 s	2.0x
SiT-XL/2 (cfg=1.5)	Dopri5	40-50	1.86 s	9.5x
SiT-XL/2 cfgUP (ours)	Dopri5	<u>20-25</u>	<u>0.97</u> s	<u>18.3x</u>

NFE halved → sub-second latency achieved

| Experiments - Text-to-Image

Works beyond class conditioning — tested on SD3.5

| Setup

Backbone : Stable Diffusion 3.5 Medium [Esser+,2024]

Training data : only 10k synthesized images from SD3.5

Evaluation : COCO dataset (10k prompts)

"A close up of a clear vase with flowers." "A set of plush toy teddy bears sitting in a sled."

SD3.5 M
(baseline)



cfgUP
(ours)



| Experiments - Text-to-Image

Works beyond class conditioning — tested on SD3.5

| Setup

Backbone : Stable Diffusion 3.5 Medium [Esser+,2024]

Training data : only 10k synthesized images from SD3.5

Evaluation : COCO dataset (10k prompts)

"A close up of a clear vase with flowers." "A set of plush toy teddy bears sitting in a sled."

SD3.5 M
(baseline)



cfgUP
(ours)



- **Latency halved** (9.23s $\rightarrow$ 4.77s)
- Text alignment largely maintained
(CLIP Score $\uparrow$: 31.85 $\rightarrow$ 29.92)
- Perceptual diversity preserved
(LPIPS $\uparrow$: 0.735 $\rightarrow$ 0.722)

Conclusion

| Conclusion

| Problem

CFG **doubles** inference cost

| Conclusion

| Problem

CFG **doubles** inference cost

| Our Solution: **cfgUP**

- Lightweight **adapter** (<1%) attached to frozen backbone
- Estimates **unconditional** prediction from **conditional** features
- CFG in a **single** forward pass → **NFE halved**

| Conclusion

| Problem

CFG **doubles** inference cost

| Our Solution: **cfgUP**

- Lightweight **adapter** (<1%) attached to frozen backbone
- Estimates **unconditional** prediction from **conditional** features
- CFG in a **single** forward pass → **NFE halved**

| Results

- FID degradation kept minimal while cutting **NFE in half**
- Generalizes to **Text-to-Image** (SD3.5)

| Conclusion

| Problem

CFG **doubles** inference cost

| Our Solution: **cfgUP**

- Lightweight **adapter** (<1%) attached to frozen backbone
- Estimates **unconditional** prediction from **conditional** features
- CFG in a **single** forward pass → **NFE halved**

| Results

- FID degradation kept minimal while cutting **NFE in half**
- Generalizes to **Text-to-Image** (SD3.5)

| Limitations & Future Work

- Remaining FID gap vs. standard CFG (2.15 → 3.16, ~1 point)
- Theoretical justification for adapter size

Thank you!

Appendix

Benchmark

Table 2. Benchmarking class-conditional image generation on ImageNet 256×256 . We report FID, IS, sFID, Precision, and Recall. Our model yields competitive results against leading approaches like SiT, DiT and LDM, validating the effectiveness of our proposed sampling strategy.

Class-Conditional ImageNet 256×256					
Model	FID↓	sFID↓	IS↑	Precision↑	Recall↑
BigGAN-deep [2]	6.95	7.36	171.40	.87	.28
StyleGAN-XL [25]	2.30	4.02	265.12	.78	.53
ADM [4]	10.94	6.02	100.98	.69	.63
ADM-G, ADM-U	3.94	6.14	215.84	.83	.53
CDM [7]	4.88	-	158.71	-	-
LDM-8 [22]	15.51	-	79.03	.65	.63
LDM-4	10.56	-	103.49	.71	.62
LDM-4 (cfg=1.25)	3.95	-	178.22	.81	.55
Simple Diffusion (U-Net) [9]	3.76	-	171.60	-	-
DiT-XL/2 [21]	9.62	6.85	121.50	.67	.67
DiT-XL/2 (cfg=1.5)	2.27	4.60	278.24	.83	.57
SiT-XL/2 (ODE) [17]	9.35	6.38	126.06	.67	.68
SiT-XL/2 (SDE)	8.26	6.32	131.65	.68	.67
SiT-XL/2 (cfg=1.5, ODE)	2.15	4.60	158.09	.81	.60
SiT-XL/2 (cfg=1.5, SDE)	2.06	4.49	277.50	.83	.59
SiT-XL/2 (cfgUP=2.0, ODE) (ours)	3.16	6.77	211.46	.80	.58

Ablation on CFG scale

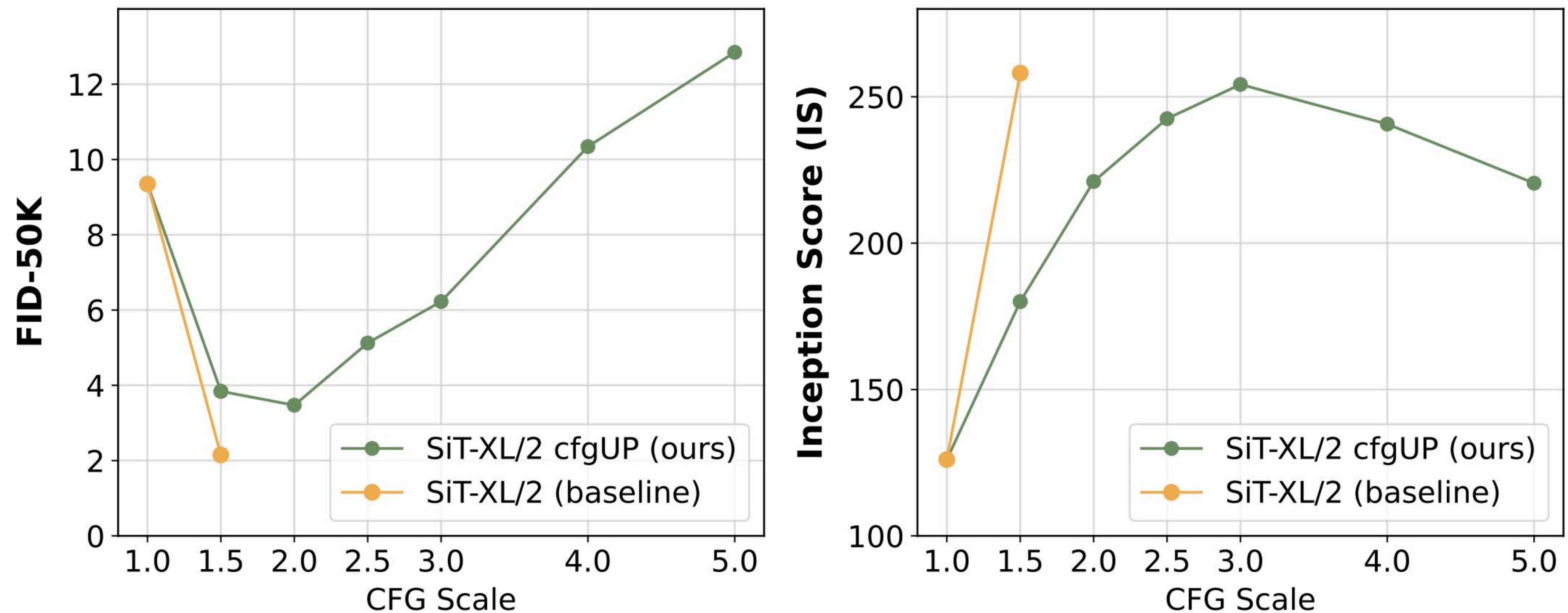


Fig. 3. Ablation study on CFG scale. FID (left) and Inception Score (right) sweeps for SiT-XL/2 with cfgUP. The results demonstrate significant improvements over the unguided baseline at moderate guidance scales.

Ablation on model scale

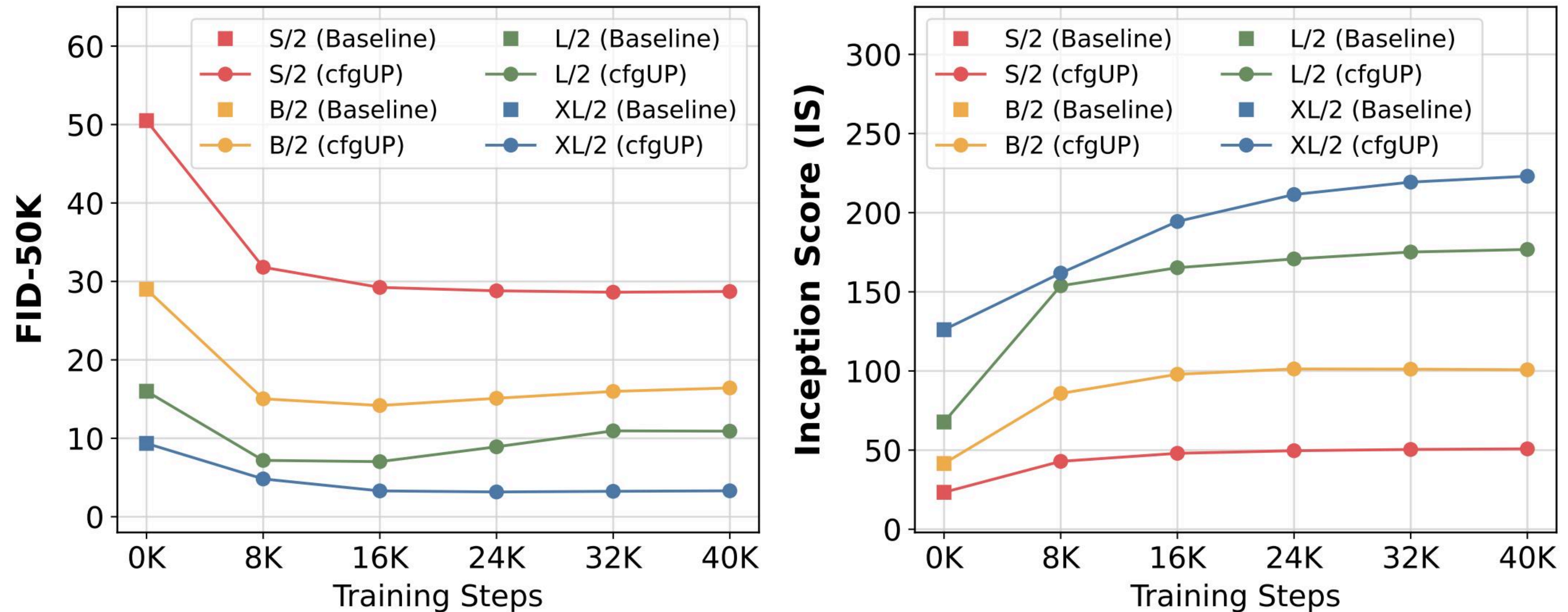


Fig. 4. Impact of training steps and model scaling. FID (left) and Inception Score (right) trajectories for different model sizes. Solid lines denote the training progress of our cfgUP method, initialized from the baseline models (squares at step 0).

Algorithm

Algorithm 1 CFG (Baseline)

Require: $v_{\hat{\theta}} = \{F_b, F_f\}$

Require: c, s

- 1: Number of Sampling Steps n
 - 2: Start noise $x_n \sim N(0, I)$
 - 3: **for** $t = n \rightarrow 0$ **do**
 - 4: $\triangleright$ *Path 1 (Cond.)*
 - 5: $h_c \leftarrow F_b(x_t, c)$
 - 6: $v_c \leftarrow F_f(h_c, c)$
 - 7: $\triangleright$ *Path 2 (Uncond.)*
 - 8: $h_{unc} \leftarrow F_b(x_t, \emptyset)$
 - 9: $v_{unc} \leftarrow F_f(h_{unc}, \emptyset)$
 - 10: $\triangleright$ *Guidance*
 - 11: $v_g \leftarrow v_{unc} + s \cdot (v_c - v_{unc})$
 - 12: $x_t \leftarrow \text{Solver}(x_t, t, v_g)$
 - 13: **end for**
 - 14: **return** x_0
-

Algorithm 2 cfgUP (Ours)

Require: $v_{\hat{\theta}} = \{F_b, F_f\}, A_\phi$

Require: c, s

- 1: Number of Sampling Steps n
 - 2: Start noise $x_n \sim N(0, I)$
 - 3: **for** $t = n \rightarrow 0$ **do**
 - 4: $\triangleright$ *Unified Path*
 - 5: $h_c \leftarrow F_b(x_t, c)$
 - 6: $v_c \leftarrow F_f(h_c, c)$
 - 7: $v'_{unc} \leftarrow A_\phi(h_c, c)$
 - 8: $\triangleright$ *Guidance*
 - 9: $v'_g \leftarrow v'_{unc} + s \cdot (v_c - v'_{unc})$
 - 10: $x_t \leftarrow \text{Solver}(x_t, t, v'_g)$
 - 11: **end for**
 - 12: **return** x_0
-