



Tech Talk

CVPR 2026 Highlight

Image Diffusion Preview with Consistency Solver

Faster, consistent previews for interactive image generation



Fu-Yun Wang

Google DeepMind • The Chinese University of Hong Kong

with H. Zhou L. Yuan S. Woo B. Gong B. Han M.-H. Yang H. Zhang Y. Zhu T. Liu L. Zhao



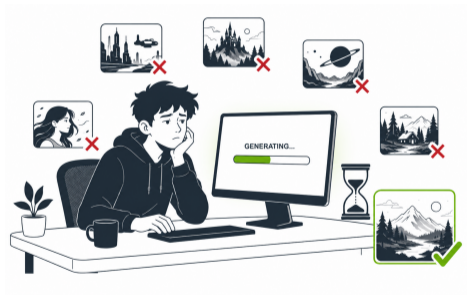
Find me at  Booth #211

 github.com/G-U-N/consolver

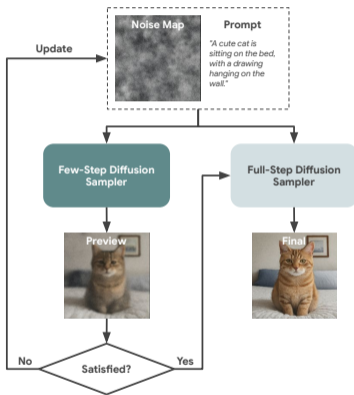
Diffusion is slow. *That hurts creators.*

- ▶ Each image takes **20–40 denoising steps**.
- ▶ Users iterate on prompts & seeds: most rolls are **thrown away**.
- ▶ Today's workflow: wait → bad result → restart.

Goal: decide *before* paying for full sampling.



The idea: Preview-and-Refine



Few-step *preview* for iteration; full-step refinement only after the user is happy.

Fidelity

Efficiency

Consistency

– what a good preview must give you

Why existing acceleration falls short

Training-free ODE solvers

- ▶ DDIM, DPM-Solver, UniPC, ...
- ▶ Theoretical assumptions \neq model's real dynamics.
- ▶ Low-step previews look *wrong*.

efficient, but bad fidelity

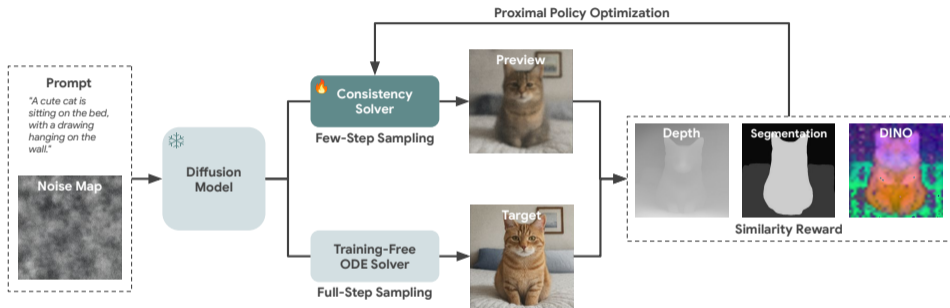
Distillation

- ▶ LCM, DMD2, PCM, ...
- ▶ Bakes acceleration *into* the weights.
- ▶ Breaks the deterministic noise \rightarrow image map.

fast, but preview \neq final

We need: few-step previews that *stay aligned* with the full-step output.

ConsistencySolver: a trainable multistep ODE solver



$$\mathbf{y}_{t_{i+1}} = \mathbf{y}_{t_i} + (n_{t_{i+1}} - n_{t_i}) \sum_{j=1}^m \mathbf{w}_j(t_i, t_{i+1}) \boldsymbol{\epsilon}_{i+1-j}$$

- ▶ **Adapts** the solver, *not* the diffusion model.
- ▶ Weights from a **tiny MLP** on (t_i, t_{i+1}) .
- ▶ Keeps the original PF-ODE mapping intact.

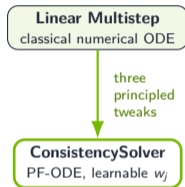
Not just a black box: it's a **principled** multistep solver

Classical m -step **LMM** for $\dot{\mathbf{x}}_t = f(t, \mathbf{x}_t)$:

$$\mathbf{x}_{t_{i+1}} = \sum_{j=0}^{m-1} \mu_j \mathbf{x}_{t_{i-j}} + \Delta t \sum_{j=0}^m w_j f_{t_{i+1-j}}$$

Adapted to PF-ODE sampling with three changes:

1. **Explicit only** ($w_0 = 0$) – PF-ODE is smooth.
2. **Anchor to current state** ($\mu_0 = 1$) – no history storage.
3. **Time-conditioned** $w_j(t_i, t_{i+1})$ – adapts to the denoising trajectory.



Recovers DDIM, iPNDM,
DPM-Solver, DPM-Solver++ as special cases.

g-u-n.github.io/blogs/consolver-theory.html

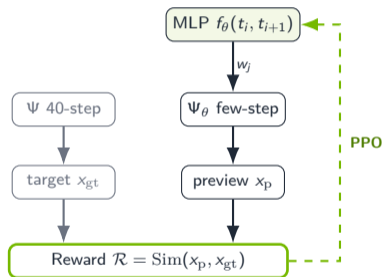
Full derivation & proofs on the blog.

Train the solver with **RL**, not distillation

- ▶ Treat solver coefficients as a **policy**.
- ▶ Reward = perceptual similarity (depth, DINO, segmentation, ...) between preview and 40-step target.
- ▶ Optimize via **PPO** on a fixed offline set of (prompt, noise, x_{gt}) triples.

Non-differentiable rewards OK

Compact MLP, low memory



Results: 47% fewer steps for the same quality

47%

fewer NFE vs. Multistep DPM-Solver
at matching FID on COCO

18.5 FID @ 12 steps (SD1.5)

0.86 Edit Reward @ 5 steps
(FLUX.1-Kontext)

Method	Steps	FID↓	CLIP↑	DINO↑
DDIM	5	52.59	87.8	73.2
Multistep DPM	5	25.87	93.1	85.5
UniPC	5	23.15	93.2	85.5
LCM (distill.)	4	22.00	90.0	75.1
ConsistencySolver	5	20.39	94.2	86.5
Multistep DPM	12	18.95	97.7	94.5
ConsistencySolver	12	18.53	97.9	95.0

Stable Diffusion 1.5 / COCO 2017 – bold = best per step budget.

Previews look like the final image



Stable Diffusion 1.5 – 5-step previews vs. 40-step full sampling.

Real impact: **1.88× faster**, users actually keep the result

End-to-end user study

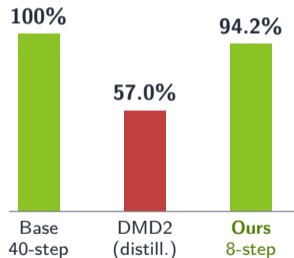
preview → refine vs. full sampling, H100

Set	Full	Preview	Speedup
GenEval	2.88s	1.74s	1.66×
COCO	3.64s	1.85s	1.97×
LAION	6.35s	2.87s	2.21×

Plug-and-play

SD1.5→SDXL / DreamShaper / FLUX.1-Kontext
no retraining.

User satisfaction (within 10 tries)



GenEval prompts, judged by Claude/.

Take-aways

- ▶ **Preview-and-refine** is the right interaction loop for diffusion.
- ▶ **ConsistencySolver**: a tiny, learnable multistep solver – *not* a distilled model.
- ▶ Trained with **RL** \Rightarrow high fidelity *and* consistency at few steps.
- ▶ Drops into existing pipelines (SD, SDXL, FLUX.1-Kontext) with no retraining.

Find me at the NVIDIA Booth #211