SnapFusion: Text-to-Image Diffusion Model on Mobile Devices within Two Seconds

추성재, DGIST

Background – Diffusion Model & DDIM sampling

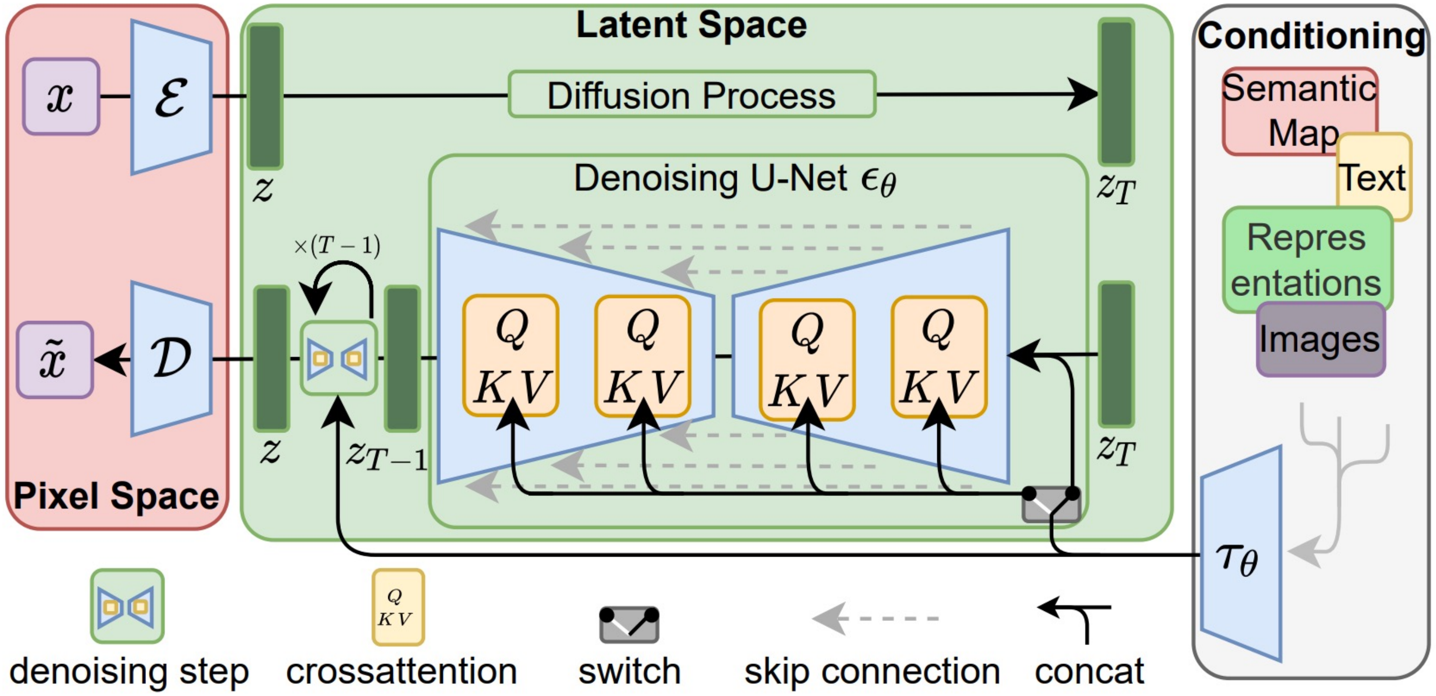
$$\min_{\boldsymbol{\theta}} \mathbb{E}_{t\sim U[0,1],\mathbf{x}\sim p_{\text{data}}(\mathbf{x}),\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I})} ||\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(t,\mathbf{z}_t) - \boldsymbol{\epsilon}||_2^2,$$

Diffusion Model Learning Objective

$$\mathbf{z}_{t'} = \alpha_{t'} \frac{\mathbf{z}_t - \sigma_t \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(t,\mathbf{z}_t)}{\alpha_t} + \sigma_{t'} \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(t,\mathbf{z}_t),$$

DDIM sampling

Background – LDM, Stable Diffusion

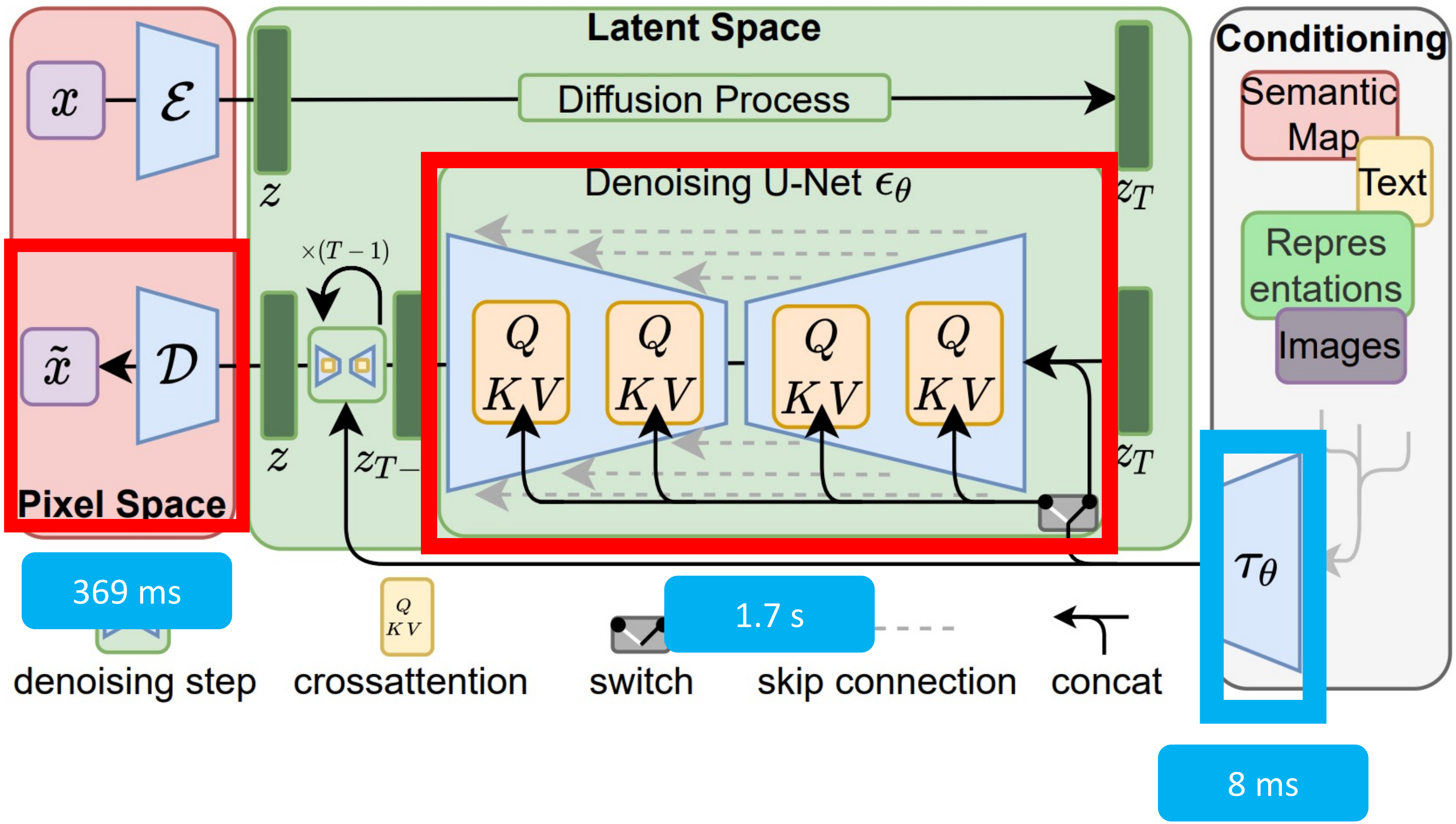


Latent Diffusion Model (LDM) Architecture

$$\tilde{\epsilon}_\theta(t, \mathbf{z}_t, \mathbf{c}) = w\hat{\epsilon}_\theta(t, \mathbf{z}_t, \mathbf{c}) - (w-1)\hat{\epsilon}_\theta(t, \mathbf{z}_t, \varnothing),$$

Classifier-Free Guidance (CFG)

Analysis – Macro Perspective



Analysis – Breakdown UNet

$$CrossAttention(Q_{\mathbf{z}_t}, K_{\mathbf{c}}, V_{\mathbf{c}}) = Softmax(\frac{Q_{\mathbf{z}_t} K_{\mathbf{c}}^T}{\sqrt{d}})V_{\mathbf{c}}$$

$$\hat{\epsilon}_\theta(t, \mathbf{z}_t) = \Pi\{Cross-attention(\mathbf{z}_t, \mathbf{c}), ResNet(\mathbf{z}_t, t)\}$$

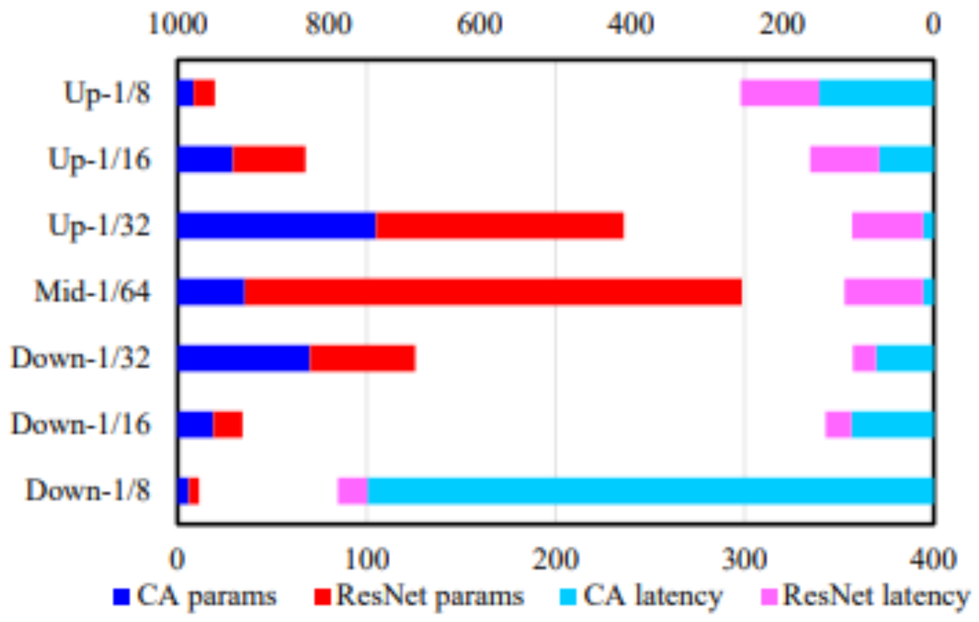
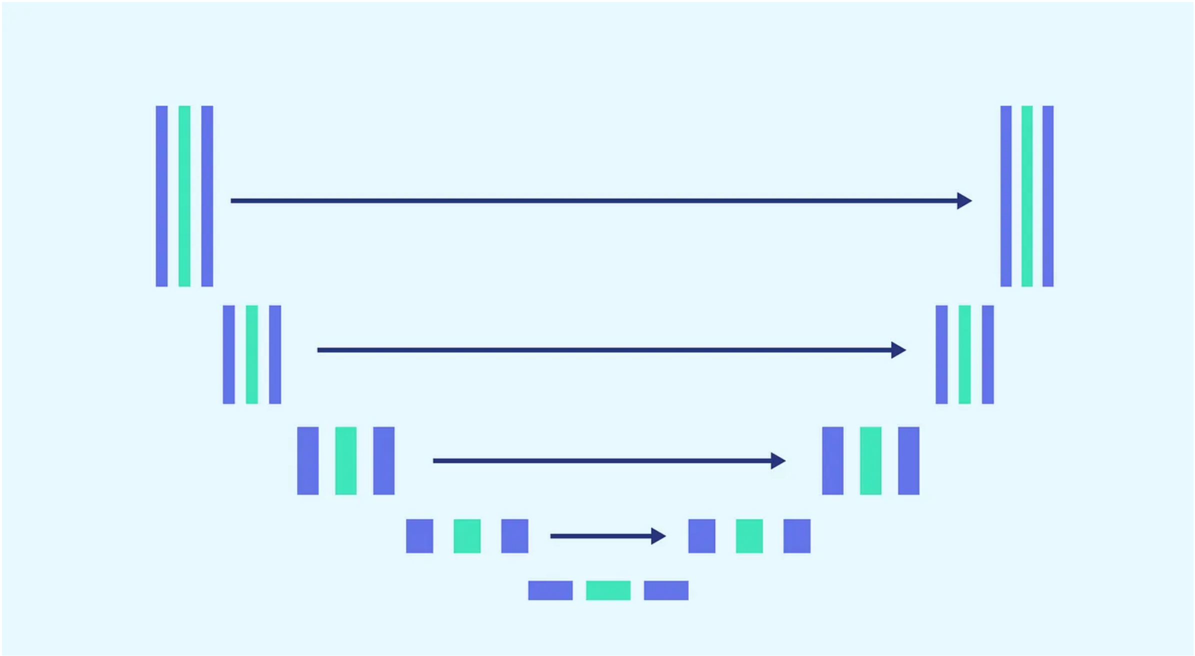


Figure 2: Latency (iPhone 14 Pro, ms) and parameter (M) analysis for cross-attention (CA) and ResNet blocks in the UNet of Stable Diffusion.



Architecture Optimization – Efficient UNet

$$\hat{\epsilon}_\theta(t, \mathbf{z}_t) = \Pi\{p(Cross - Attention(\mathbf{z}_t, \mathbf{c}), I), p(ResNet(\mathbf{z}_t, t), I)\}$$

Robust Training

$$A \in \{A^{+,-}_{Cross-Attention[i,j]}, A^{+,-}_{ResNet[i,j]}\}$$

Evolution Action Set

- Latency (built at analyzing Unet)
- Generative Performance (CLIP score)
- $\frac{\Delta CLIP}{\Delta Latency}$: Block with lower latency & high CLIP score will be remain, Block with high latency & low CLIP score will be removed

- Proposed Efficient Image Decoder
  - obtained with Channel Reduction
  - 3.8x fewer parameters, 3.2x faster than SD-v1.5 Image Decoder

Architecture Optimization – Efficient UNet

Algorithm 1 Optimizing UNet Architecture

Require: UNet: $\hat{\epsilon}_\theta$; validation set: $\mathbb{D}_{val}$; latency lookup table $\mathbb{T}$: $\{\textit{Cross-Attention}[i,j], \textit{ResNet}[i,j]\}$.
Ensure: $\hat{\epsilon}_\theta$ converges and satisfies latency objective S.

- **while** $\hat{\epsilon}_\theta$ not converged **do**
  - **Perform robust training.**
  - $\rightarrow$ **Architecture optimization:**
  - **if** perform architecture evolving at this iteration **then**
    - $\rightarrow$ **Evaluate blocks:**
    - **for** each $block[i,j]$ **do**
      - $\Delta CLIP \leftarrow \text{eval}(\hat{\epsilon}_\theta, A^{-}_{block[i,j]}, \mathbb{D}_{val})$,
      - $\Delta Latency \leftarrow \text{eval}(\hat{\epsilon}_\theta, A^{-}_{block[i,j]}, \mathbb{T})$
    - **end for**
    - $\rightarrow$ **Sort actions based on $\frac{\Delta CLIP}{\Delta Latency}$, execute action, and evolve architecture to get latency T:**
    - **if** latency objective S is not satisfied **then**
      - $\{\hat{A}^{-}\} \leftarrow \arg\min_{A^{-}} \frac{\Delta CLIP}{\Delta Latency}$,
    - **else**
      - $\{\hat{A}^{+}\} \leftarrow \text{copy}(\arg\max_{A^{-}} \frac{\Delta CLIP}{\Delta Latency})$,
      - $\hat{\epsilon}_\theta \leftarrow \text{evolve}(\hat{\epsilon}_\theta, \{\hat{A}\})$
    - **end if**
  - **end if**
- **end while**

Evaluate Latancy, CLIP score

Use evaluated score
to add/remove blocks
in model

Step Distillation – Overview

$$\mathcal{L}_{\text{ori}} = \mathbb{E}_{t \sim U[0,1], \mathbf{x} \sim p_{\text{data}}(\mathbf{x}), \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})} \, ||\hat{\mathbf{v}}_{\boldsymbol{\theta}}(t, \mathbf{z}_t, \mathbf{c}) - \mathbf{v}||_2^2,$$

$$\mathbf{v} = \alpha_t \epsilon - \sigma_t \mathbf{x}$$

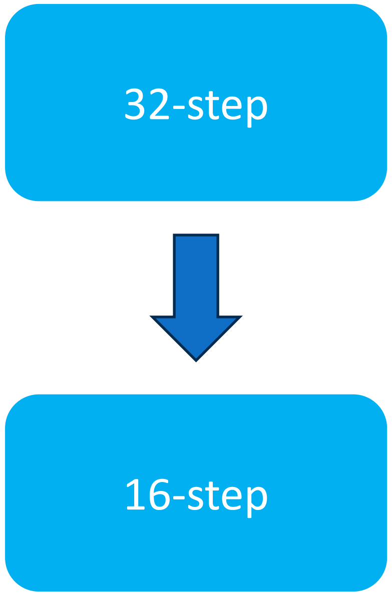
Original Loss of UNet predicts velocity

1. Obtain 16-step SD-v1.5 with step distillation
2. Obtain 16-step efficient UNet with step distillation
3. use 16-step SD-v1.5 as teacher, 16-step efficient UNet as student, do step distillation and obtain final 8-step efficient UNet

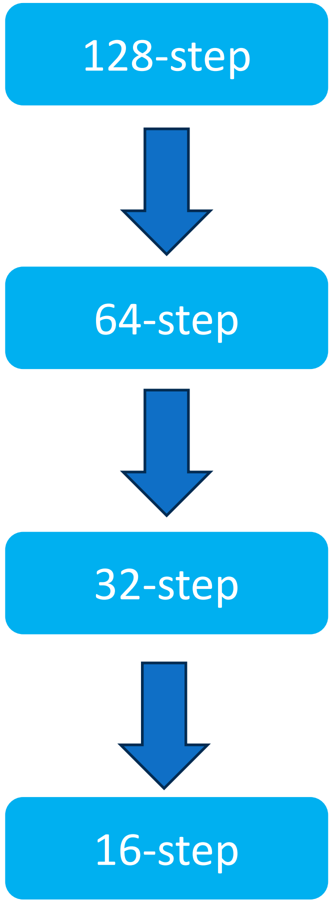
step distillation pipeline

Step Distillation – Direct vs.progressively

: step distillation

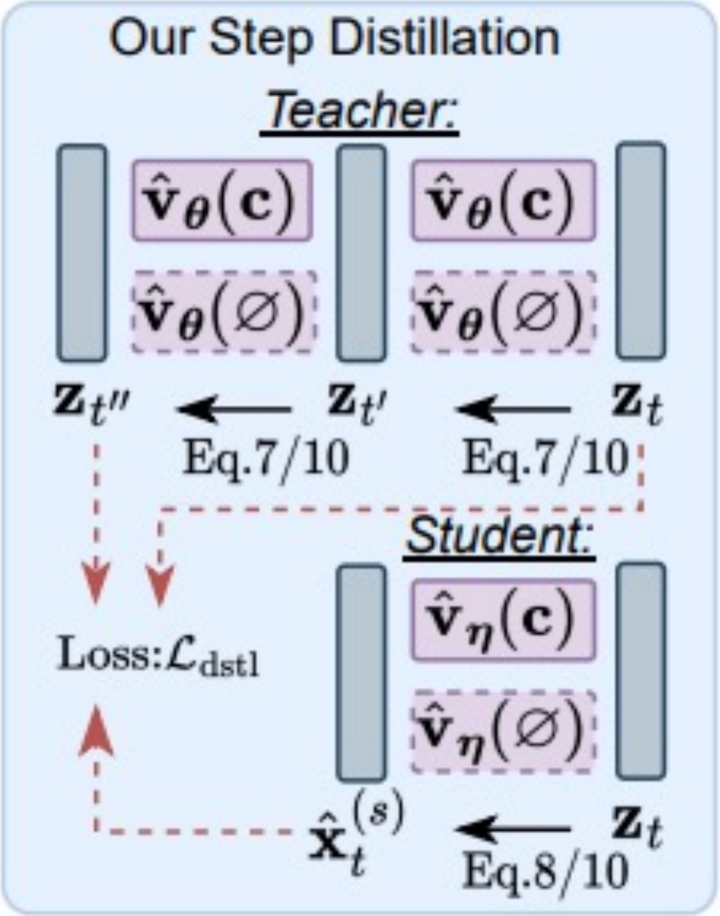


Direct



Progressively

Step Distillation – Vanila Step Distillation



$$\hat{\mathbf{v}}_t = \hat{\mathbf{v}}_\theta(t, \mathbf{z}_t, \mathbf{c}) \Rightarrow \mathbf{z}_{t'} = \alpha_{t'}(\alpha_t \mathbf{z}_t - \sigma_t \hat{\mathbf{v}}_t) + \sigma_{t'}(\sigma_t \mathbf{z}_t + \alpha_t \hat{\mathbf{v}}_t),$$
$$\hat{\mathbf{v}}_{t'} = \hat{\mathbf{v}}_\theta(t', \mathbf{z}_{t'}, \mathbf{c}) \Rightarrow \mathbf{z}_{t''} = \alpha_{t''}(\alpha_{t'} \mathbf{z}_{t'} - \sigma_{t'} \hat{\mathbf{v}}_{t'}) + \sigma_{t''}(\sigma_{t'} \mathbf{z}_{t'} + \alpha_{t'} \hat{\mathbf{v}}_{t'}).$$

teacher’s DDIM step

$$\hat{\mathbf{v}}_t^{(s)} = \hat{\mathbf{v}}_\eta(t, \mathbf{z}_t, \mathbf{c}) \Rightarrow \hat{\mathbf{x}}_t^{(s)} = \alpha_t \mathbf{z}_t - \sigma_t \hat{\mathbf{v}}_t^{(s)},$$

Student’s DDIM step

$$\mathcal{L}_{\text{vani\_dstl}} = \varpi(\lambda_t)\,||\,\hat{\mathbf{x}}_t^{(s)} - \frac{\mathbf{z}_{t''} - \frac{\sigma_{t''}}{\sigma_t}\mathbf{z}_t}{\alpha_{t''} - \frac{\sigma_{t''}}{\sigma_t}\alpha_t}\,||_2^2,$$

Loss of vanila step distillation

Problem : CLIP score becomes worse

Step Distillation – CFG-aware Step Distillation

$$\tilde{\mathbf{v}}_t^{(s)} = w\hat{\mathbf{v}}_{\boldsymbol{\eta}}(t, \mathbf{z}_t, \mathbf{c}) - (w-1)\hat{\mathbf{v}}_{\boldsymbol{\eta}}(t, \mathbf{z}_t, \varnothing),$$

Student's DDIM step applied with CFG

$$\mathcal{L} = \mathcal{L}_{\text{dstl}} + \gamma\mathcal{L}_{\text{ori}},$$
$$\mathcal{L}_{\text{dstl}} = \mathcal{L}_{\text{cfg\_dstl}} \text{ if } P \sim U[0,1] < p \text{ else } \mathcal{L}_{\text{vani\_dstl}},$$

total loss function

vanila distillation loss makes better FID score & cfg distillation loss makes better CLIP score (trade-off)

Experiment – Text-to-Image Generation

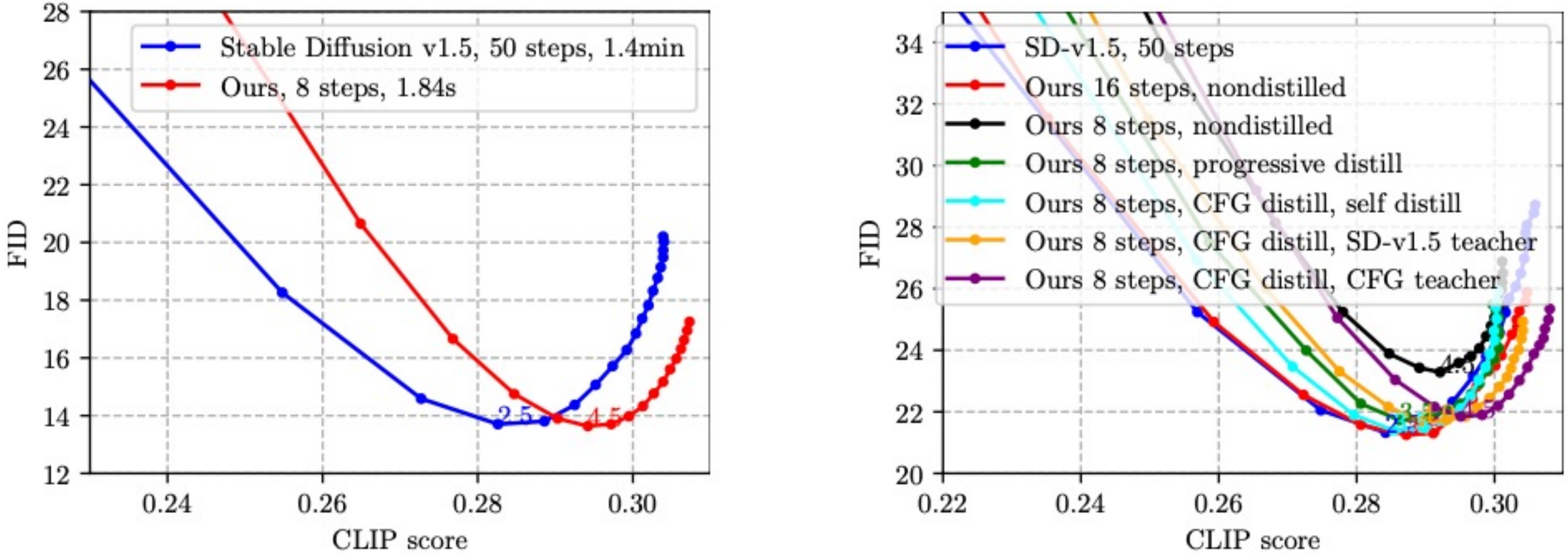


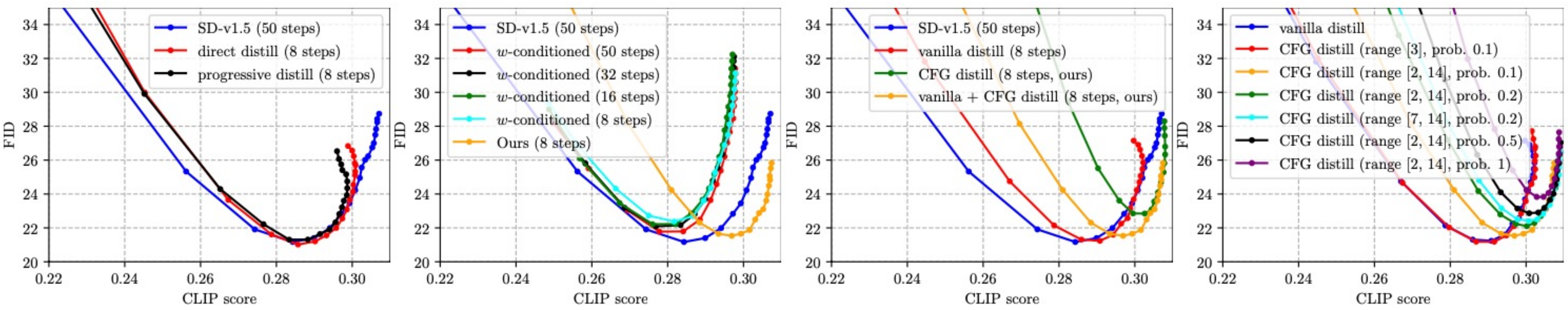
Figure 4: FID *vs.* CLIP on MS-COCO 2014 validation set with CFG scale from 1.0 to 10.0. **Left**: Comparison with SD-v1.5 on *full* set (30K). **Right**: Different settings for step and teacher models tested on 6K samples.

Experiment – Ablation Analysis

(a) SD	(b) SD removes CA	(c) Remove CA in DS	(d) Remove CA in mid.	(e) Remove CA in US	(f) Remove RB in DS	(g) Remove RB in mid.	(h) Remove RB in US

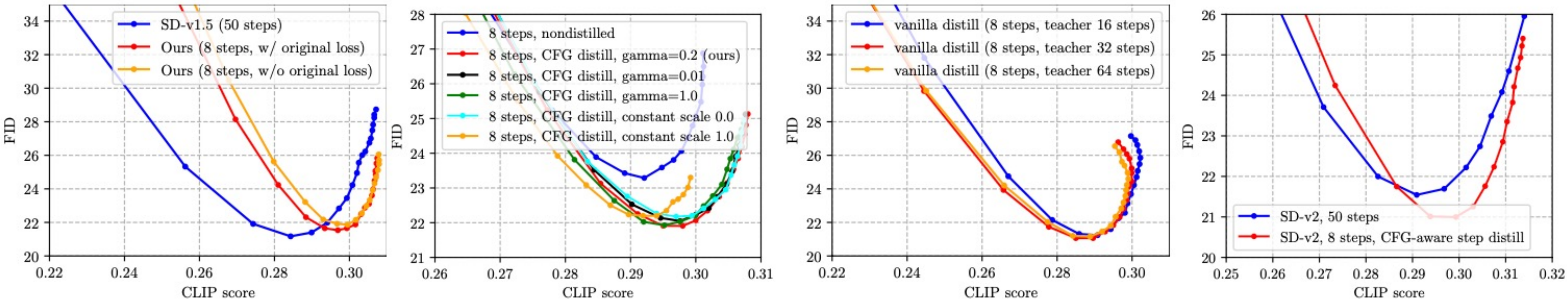
Figure 5: **Advantages of robust training**. Prompts of top row: *a photo of an astronaut riding a horse on mars* and bottom row: *A pikachu fine dining with a view to the Eiffel Tower*. **(a)** Images from SD-v1.5. **(b)** Removing cross-attention (CA) blocks in downsample stage of SD-v1.5. **(c) - (e)** Removing cross-attention (CA) blocks in {downsample (DS), middle (mid.), upsample (US)} using our model after *robust* training. **(f) - (h)** Removing ResNet blocks (RB) in different stages using our model. The model with robust training maintains reasonable performance after dropping blocks.

Experiment – Ablation Analysis



(a) Direct *vs.* progressive (b) *w*-conditioned *vs.* ours (c) Vanilla *vs.* CFG distill (d) CFG hyper-parameters

Experiment – Ablation Analysis



(a) $\mathcal{L}_{\text{ori}}$ in Eq. (11). (b) γ in Eq. (11). (c) Number of teacher steps. (d) Distillation on SD-v2.