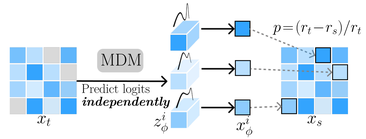
Soft-Di[M]O: Improving One-Step Discrete Image Generation with Soft Embeddings

Yuanzhi Zhu[1] Xi Wang[1] Stéphane Lathuilière[2] Vicky Kalogeiton[1]
[1]LIX, École Polytechnique, CNRS, IPP [2]Inria at Univ. Grenoble Alpes, CNRS, LJK



yuanzhi-zhu.github.io/about/

TL;DR

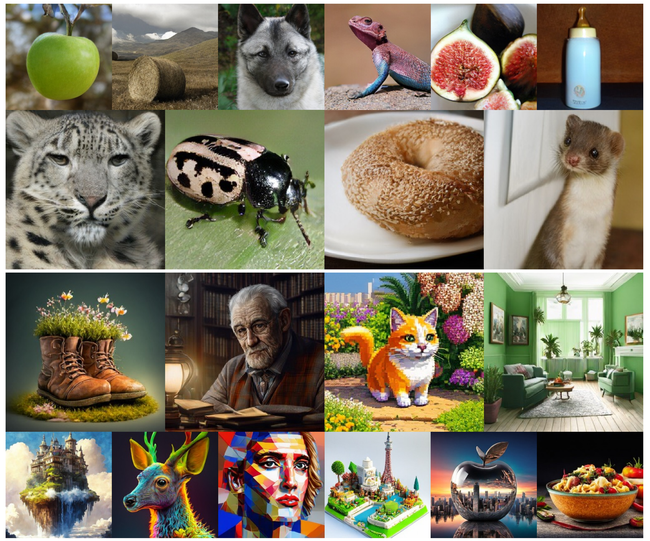
Soft-Di[M]O improves one-step discrete image generation by replacing hard tokens with continuous soft embeddings, enabling differentiable adversarial and reward-based fine-tuning.

Motivation & Highlights

Motivation: One-step discrete diffusion models struggle with fine-tuning due to non-differentiable hard token sampling.

Highlights:

- **Soft Embeddings**: Replace hard tokens for differentiable optimization.
- **Adversarial Tuning**: Enables GAN training for discrete models.
- **Reward Tuning**: Direct gradient flow for aesthetic/CLIP rewards.
- **Test-Time Emb. Optimization**: TTS for one-step generator.
- **SOTA Results**: *(i)* **1.56 FID** on ImageNet 256×256 in one step.
  (ii) higher than teacher GenEval & HPS scores on T2I.



Di[M]O Framework

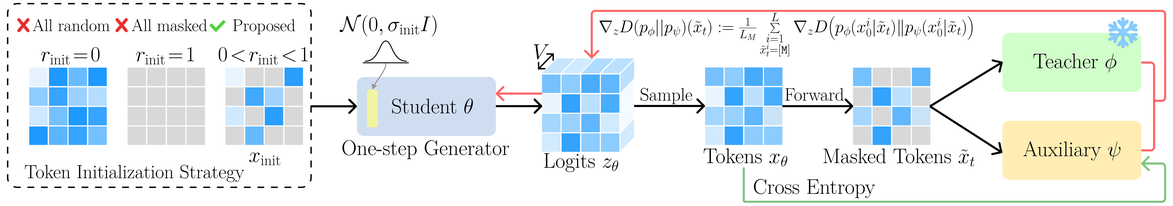


Figure: **Di[M]O Pipeline.** From x_{init}, the student θ produces logits z_θ and sampled tokens x_θ. A forward mask diffusion process yields intermediate states $\tilde{x}_t$. For each $\tilde{x}_t$, we alternately update θ and auxiliary model ψ: θ minimizes the token-level conditional divergence $D(p_\phi,|,p_\psi)(\tilde{x}_t)$, while ψ is trained via cross-entropy on x_θ to provide gradients for θ. The teacher ϕ remains frozen.

Soft-Di[M]O Framework

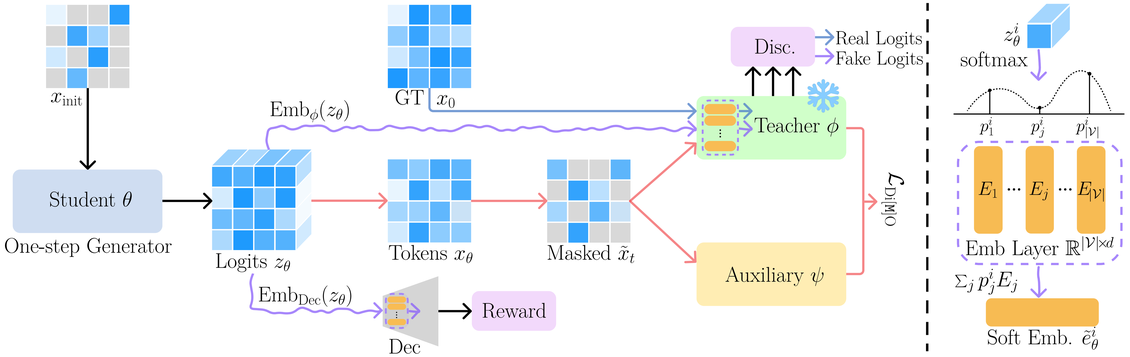
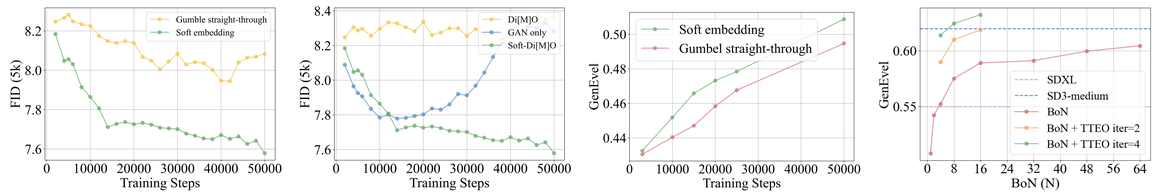


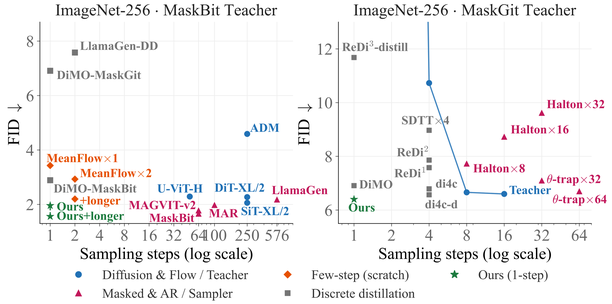
Figure: **Soft-Di[M]O Pipeline.** From x_{init}, the one-step generator (student θ) produces logits z_θ. Discrete tokens x_θ sampled from these logits define the Di[M]O distillation loss (red path). Concurrently, logits are transformed into *soft embeddings* (purple path), enabling differentiable supervision for post-training. For each position i, $z_\theta^i \xrightarrow{\text{softmax}} p^i$ and $\tilde{e}_\theta^i = \Sigma_j p_j^i E_j$.

Ablations on Key Designs



Experimental Results

1. Class-to-Image Generation



2. Text-to-Image Generation