ROMS-IMLE: A MINIMALIST APPROACH TO COMPETITIVE SINGLE-STEP GENERATIVE MODELLING

Anonymous authors
Paper under double-blind review

ABSTRACT

Generative models have undergone many generations of evolution, from VAEs/GANs to diffusion/flow matching. Along the way, the underlying techniques have become more complicated and various beliefs about what drives strong empirical performance have taken hold. Due to the success of diffusion models and flow matching, one of the more common beliefs is the importance of transforming the noise distribution to the data distribution gradually through many small transformations. We ask whether this is truly necessary, and take a minimalist approach to designing a competitive generative model. We start with the bare-bones essentials, namely just a training objective and a model. We purposefully make both simple. For the training objective, we choose Implicit Maximum Likelihood Estimation (IMLE), and eschew more complicated alternatives such as variational inference, adversarial training and numerical integration. For the model, we eschew transformers and instead choose a moderately sized convolutional network. Then we judiciously added elements that are truly essential, which surprisingly do *not* include iterative denoising. The result is a single-step parameter-efficient generative model that produces high quality samples at fast speed: it achieves an FID of 2.56 on ImageNet 256 and simultaneously attains good precision and recall.

1 INTRODUCTION

Image generation has advanced through a succession of paradigms over the past decade. Variational autoencoders (VAEs) (Kingma & Welling, 2022) and generative adversarial networks (GANs) (Goodfellow et al., 2014) were among the first models to generate realistic images in a single forward pass, yet each approach carries a characteristic weakness. GANs are unstable to train and prone to mode collapse, which limits the diversity of their samples. VAEs train stably but tend to produce blurry, low-fidelity samples. Stochastic interpolant models (SI for short), including diffusion and flow-matching models (Ho et al., 2020; Song et al., 2021; Lipman et al., 2023; Albergo & Vanden-Eijnden, 2023), resolved many of these issues and now define the state of the art in image synthesis. Inspired by this success, subsequent work has grown steadily more complicated, introducing elaborate architectures (Peebles & Xie, 2023; Ma et al., 2024), sophisticated sampling schemes (Karras et al., 2022; Song et al., 2022), and intricate training objectives (Song et al., 2023; Geng et al., 2025a; Zhou et al., 2025; Frans et al., 2024).

We ask whether all of the complex machinery introduced in recent years is truly necessary. Rather than adding to it, we take a minimalist approach: we start from the simplest generative model we can and add only the components from modern approaches that prove essential. We choose to build on Implicit Maximum Likelihood Estimation (IMLE) (Li & Malik, 2018), which offers a stable and mode-covering training objective that is simple and intuitive. IMLE works by matching each data point to its nearest sample from the model, and then training the model to reduce the distance between them. IMLE optimizes a likelihood-based objective akin to regression, without the variational bounds and encoders of VAEs, the adversarial training of GANs, or the iterative numerical integration used by stochastic interpolant models.

The advantage of modern SI models over earlier single-step models is commonly attributed to how they decompose generation (Xiao et al., 2022; Dhariwal & Nichol, 2021): instead of mapping the prior to the data distribution in one large transformation, they break it into a sequence of smaller, simpler transformations, which is thought to be the key to producing high-quality samples. Each of

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

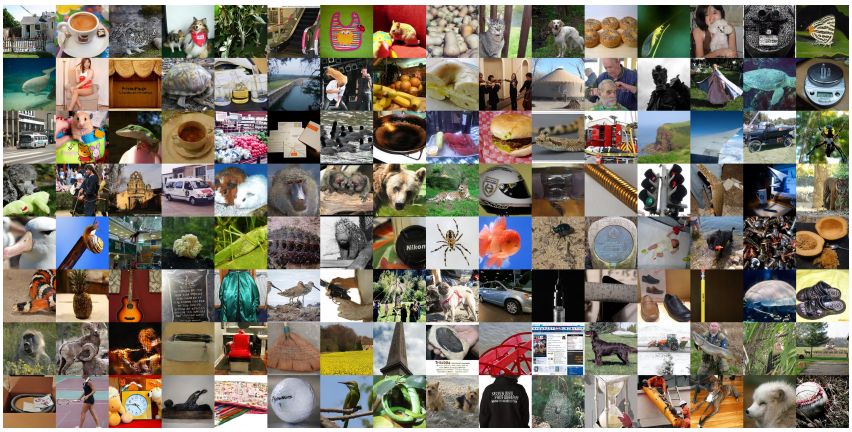


Figure 1: Random samples from our class-conditional model trained on ImageNet

these transformations is typically carried out by a separate step in the sampling process, and as a result sampling takes many steps to iterate over all the transformations. Iterative sampling is, however, computationally expensive, since it involves multiple forward passes through the model. If we can achieve high-quality generation without the cost of iterative sampling, we can make image generation both cheaper and simpler.

To this end, we start with a single-step model and add only a minimal set of modifications that are necessary to achieve competitive performance. To devise the modifications, we find the mechanisms behind SI models that are truly important, abstract them at a high level to enable application to the single-step setting and add them to our single-step model. Rather than viewing SI models as performing iterative sampling, we take a testing-centric view. We hypothesize that the ingredient that powers their success is per-stage supervision and add it to our single-step model. Our second contribution is specific to the matching phase in IMLE, which can occasionally produce poor matches due to the randomness in the sampling process. We introduce a simple change to the algorithm that makes it robust to these outliers. In line with our minimalist design philosophy, we use a ConvNeXt-based architecture (Liu et al., 2022) rather than the transformer-based architectures that now dominate.

Together, these design choices yield **Ro**bust **M**ulti-**S**tage IMLE (ROMS-IMLE), a single-step generative model that combines per-stage supervision with a robust training loss. Across three diverse datasets, our model attains competitive FID and strong precision and recall. In pixel space, ROMS-IMLE attains the highest precision and recall compared to baselines on CIFAR-10 and CelebA-HQ. For latent generation on ImageNet 256, it reaches an FID of 2.56 in a single forward pass, with 54% fewer parameters and 250$\times$ fewer function evaluations than DiT-XL/2 and SiT-XL/2. Through our results, we demonstrate that a simple single-step generative model can indeed retain the sample quality and diversity commonly exhibited by popular SI models without resorting to decomposing generation into gradual transformations.

2 Background

2.1 IMLE

The goal of generative modeling is to transform a simple, easy-to-sample prior distribution into the complex data distribution we wish to model. One way to approach this problem is to learn the transformation directly, using a neural generator that maps samples from the prior to data samples in a single pass. In general, this approach is difficult because the likelihood of the data under the generator is intractable (refer to Appendix A for more details). Implicit Maximum Likelihood Estimation (IMLE) (Li & Malik, 2018; Aghabozorgi et al., 2023) provides a workaround by implicitly maximizing the likelihood without requiring knowledge of its explicit form.

Formally, IMLE learns a direct mapping f_θ from a prior $\pi(\mathbf{z})$ to the data distribution $p_{\text{data}}(\mathbf{x})$ by minimizing the following objective:

$$\theta_{\text{IMLE}} = \arg\min_\theta \mathbb{E}_{\mathbf{z}_1,\ldots,\mathbf{z}_m \sim \pi(\mathbf{z})} \left[\sum_{i=1}^{n} \min_{j \in [m]} d\Big(\mathbf{x}_i, f_\theta(\mathbf{z}_j) \Big) \right], \tag{1}$$

where $d(\cdot,\cdot)$ is a distance function. To optimize the objective, the IMLE algorithm alternates between two phases. In the matching phase, m latent codes $\mathbf{z}_1,\ldots,\mathbf{z}_m$ are sampled from the prior and each real image $\mathbf{x}_i$ is matched to its nearest generated sample in the set $\{f_\theta(\mathbf{z}_1),\ldots,f_\theta(\mathbf{z}_m)\}$. This establishes a coupling between $\pi(\mathbf{z})$ and $p_{\text{data}}(\mathbf{x})$. Next, in the optimization phase, the parameters θ are updated to bring each generated sample closer to its matched real image. We now take a look at SI models to understand how their construction differs from IMLE, and whether any of their properties can be leveraged in a single-step model.

2.2 Flow Matching and Testing-Centric View

Stochastic interpolant (SI) methods, such as diffusion and flow matching, are commonly understood to perform iterative denoising. At training time, SI methods train a model, commonly known as a denoising network, to convert a noisy image to a clean image. At test time, the denoising network is repeatedly applied to convert pure noise to a generated sample. For the purpose of this section, we consider flow matching as an example of SI methods. Given the data distribution p_{data} and prior p_T distribution, the general objective takes the following form:

$$\mathcal{L}(\theta) = \mathbb{E}_{t\sim\mathcal{U}(0,T],\, \mathbf{x}_0\sim p_{\text{data}},\, \mathbf{x}_T\sim p_T,\, \mathbf{x}_t\sim q_t(\mathbf{x}_t|\mathbf{x}_0,\mathbf{x}_T)} \left[\ell_t(g^t(\mathbf{x}_0,\mathbf{x}_t), g_\theta^t(\mathbf{x}_t)) \right] \tag{2}$$

The function g_θ^t is the model's prediction at time t, g^t is the supervision target it is trained to match, and ℓ_t compares the two. q_t characterizes the interpolation between p_{data} and p_T.

A key distinguishing feature of SI methods is the iterative application of the denoising network, and this is what SI methods are believed to owe their success to. This need for multiple steps creates a mismatch between training and test; the denoising network is applied once for each sample during training, whereas it is applied iteratively to generate a sample during test. We consider an alternative view of this mismatch. Rather than viewing the model as one instance of the denoising network, we unroll the different iterations of the sampling procedure and view the model as a composition of the different iterations. Under this view, at test time, the model is just applied once to convert pure noise to a generated sample, and at training time, only a part of the model is evaluated at a time. This is in some sense the flip side of the conventional view; under the conventional view, the test time procedure is what deviates from the norms, whereas under the alternative view, the training time procedure is what deviates from the norms. Hence, we will term the conventional view the *training-centric view* and the alternative view the *testing-centric view*.

We now concretely construct the model under the testing-centric view. Consider a deterministic sampler in an SI method, such as those used in DDIM or flow matching. Starting from Gaussian noise, the sampler applies a succession of updates parameterized by the denoising network evaluated at different time steps. More precisely, the sampler starts by sampling $\mathbf{x}_T \sim p_T(\mathbf{x})$ and produces a sequence $\mathbf{x}_T \to \hat{\mathbf{x}}_{T-1} \to \cdots \to \hat{\mathbf{x}}_0$, where at each iteration $\hat{\mathbf{x}}_{t-1}$ is obtained from $g_\theta^t(\hat{\mathbf{x}}_t)$.

Under the testing-centric view, the model is the composition of all the updates done by the sampler, namely $g_\theta^T,\ldots,g_\theta^1$.

$$\hat{\mathbf{x}}_0 \;=\; \underbrace{g_\theta^1 \circ g_\theta^2 \circ \cdots \circ g_\theta^T}_{h_\theta}(\mathbf{x}_T), \tag{3}$$

This allows us to view the reverse process as a *single* compositional model h_θ, that maps a sample from the prior $\mathbf{x}_T$ to a data sample $\hat{\mathbf{x}}_0$. Hence, what the training-centric view treats as an iteration over time steps is now viewed as an iteration over depth in the compositional model. Under this view, each $\hat{\mathbf{x}}_{t-1} = g_\theta^t(\hat{\mathbf{x}}_t)$ is an intermediate layer of h_θ. Let us revisit the training objective. If we restrict the support of t to integer time steps in Equation 2, we can rewrite the objective as:

$$\tilde{\mathcal{L}}(\theta) = \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{\mathbf{x}_0\sim p_{\text{data}},\, \mathbf{x}_T\sim p_T,\, \mathbf{x}_t\sim q_t(\mathbf{x}_t|\mathbf{x}_0,\mathbf{x}_T)} \left[\ell_t(g^t(\mathbf{x}_0,\mathbf{x}_t), g_\theta^t(\mathbf{x}_t)) \right] \tag{4}$$

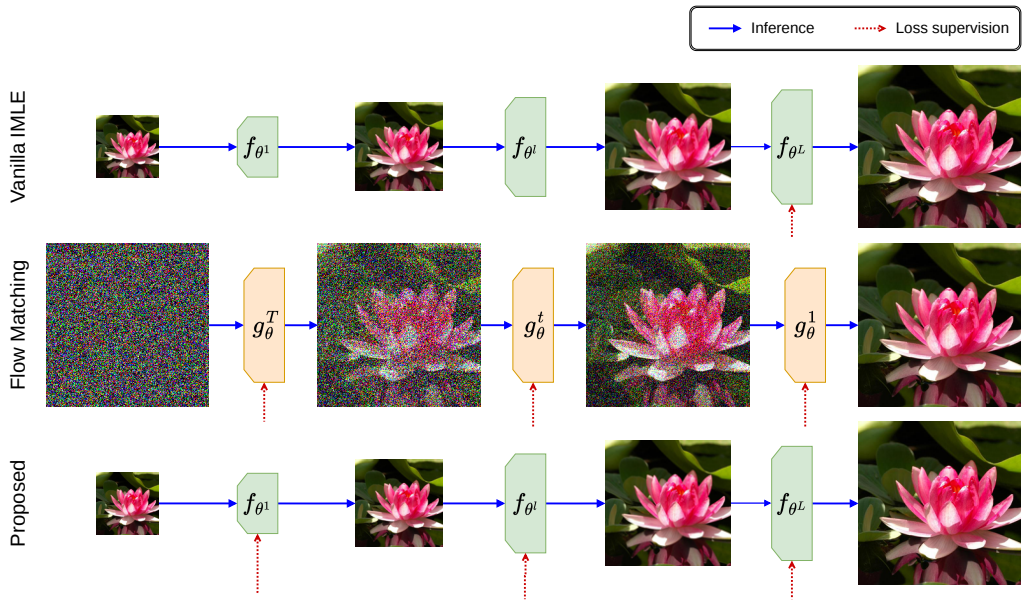


Figure 2: **Multi-stage supervision:** Each row is a generator drawn as a composition of stages; blue arrows are the inference path and red dotted arrows show where the loss is applied. Vanilla IMLE (top) supervises only its final stage, whereas flow matching (middle) supervises every denoising stage. Our proposed method (bottom) retains IMLE's architecture and one-step inference while extending direct supervision to every stage.

We emphasize how the compositional model h_θ is trained by the loss in Equation 4, noting that each stage g_θ^t in the composition is directly supervised. Under the testing-centric view, the training objective therefore performs *intermediate per-stage supervision*: every intermediate layer g_θ^t of the compositional model h_θ receives a direct training signal (Figure 2). This training-time property, rather than iterative sampling, is what we believe is the reason for high-quality generation in SI models.

3 Method

Our goal in this paper is to improve IMLE, a single-step generator, so that it can match the sample quality of modern generative models that generate over many sequential steps. The two contributions presented in the paper improve on the optimization and matching phases of IMLE, respectively. The first contribution, *multi-stage supervision*, improves the optimization phase of IMLE by providing a direct training signal to each stage of the generator. Our second contribution, *robust training loss*, improves the matching phase of IMLE by making it more robust to outliers. We describe them in greater detail below.

3.1 Optimization Phase: Adapting Multi-Stage Supervision for IMLE

In the previous section, we presented the SI sampler as a single compositional model h_θ. We can now contrast this compositional model with the single-step generation setup of vanilla IMLE. The SI objective supervises every stage of the composition, whereas vanilla IMLE supervises only the final output of f_θ. Supervision at every stage is the property we want to carry over. The unrolled view introduced in Section 2.2 applies naturally to IMLE: its generator can also be seen as a prior-to-data network composed of stages, only executed in a single forward pass rather than through iterative sampling. What it lacks is *multi-stage supervision*, which we add through a change to the training objective.

Compositional scaffold We choose to build the IMLE generator f_θ as a stack of upsampling blocks. We adopt this design deliberately; similar to Section 2.2, it lets us write f_θ as a composition

$$f_\theta = f_{\theta^L} \circ f_{\theta^{L-1}} \circ \cdots \circ f_{\theta^1},$$

in which each stage f_{θ^l} operates at a progressively higher resolution, with per-stage parameters $\theta = (\theta^1, \ldots, \theta^L)$. This recasts IMLE in a form similar to the diffusion construction, with each f_{θ^l} now analogous to g_θ^t, only indexed by resolution rather than noise level.

The architecture of f_θ in existing IMLE methods (Vashist et al., 2024; Aghabozorgi et al., 2023) is compositional, but the training is not. The loss in Equation 1 is computed only at the final output $f_\theta(\mathbf{z})$, which directly supervises only stage L. Stages $1, \ldots, L-1$ receive a training signal only through gradients that flow back from this final loss. We address this by supervising every stage directly, at the resolution at which it operates.

Concretely, two things are needed: a way to read out a prediction at each stage, and a way to obtain a target at the matching resolution. We add a per-stage output head (see Figure 2), so that stage f_{θ^l} outputs an image $\tilde{\mathbf{x}}^l$ at its own resolution. We obtain the target by downsampling the real image $\mathbf{x}_i$ to the same resolution with a kernel K_l. At each stage l, the per-stage loss term compares $\tilde{\mathbf{x}}^l$ against $K_l(\mathbf{x}_i)$. Each per-stage loss is averaged over the elements at that stage's resolution, so the L terms are on a comparable scale. The training objective is the average of all L per-stage loss terms.

Algorithm 1 ROMS-IMLE training procedure

Require: Dataset $\{\mathbf{x}_i\}_{i=1}^n$, generator $f_\theta = f_{\theta^L} \circ \cdots \circ f_{\theta^1}$ with parameters $\theta = (\theta^1, \ldots, \theta^L)$, downsampling kernel $K_l(\cdot)$, distance function $d(\cdot, \cdot)$ averaged over the elements of its arguments

Initialize θ to a random vector
for $t = 1$ **to** T **do**
  Draw i.i.d. latent codes $\mathbf{z}_1, \ldots, \mathbf{z}_m \sim \mathcal{N}(0, I)$
  $\tilde{\mathbf{x}}_j \leftarrow f_\theta(\mathbf{z}_j)$ for all $j \in [m]$ ▷ Generate samples
  $\sigma(i) \leftarrow \arg\min_{j \in [m]} d(\mathbf{x}_i,\ \tilde{\mathbf{x}}_j) \quad \forall i \in [n]$ ▷ Nearest-neighbour matching
  for each minibatch $S \subseteq [n]$ **do**
    $\tilde{\mathbf{x}}^l_{\sigma(i)} \leftarrow (f_{\theta^l} \circ \cdots \circ f_{\theta^1})(\mathbf{z}_{\sigma(i)})$ for all $i \in S, l \in [L]$ ▷ Per-stage outputs
    $\mathcal{J} \leftarrow 0$
    for $l = 1$ **to** L **do**
      $\mathcal{J} \leftarrow \mathcal{J} + \frac{1}{|S|} \sum_{i \in S} d\Big(K_l(\mathbf{x}_i),\ \tilde{\mathbf{x}}^l_{\sigma(i)}\Big)$ ▷ Per-stage supervision
    end for
    $\theta \leftarrow \theta - \frac{\eta}{L} \nabla_\theta \mathcal{J}$ ▷ Gradient step
  end for
end for
return θ

We test this new training procedure on Oxford Flowers (Nilsback & Zisserman, 2008), an image dataset of roughly 1000 images at 256×256 resolution. We use this dataset for the ablation because it is small enough to train and iterate over quickly, yet visually diverse enough that mode collapse is easy to spot in the generated samples. Table 1 reports the results and Figure 3 shows random samples from each configuration: our proposed training paradigm with multi-stage supervision (B) substantially outperforms vanilla IMLE (A). Additional experiment details can be found in Appendix B.

3.2 Matching Phase: Robust Loss for Mismatched Pairs

SI models interpolate between the noise distribution and the empirical data distribution along a predetermined trajectory, and so the flow between them does not need to be recomputed during training. On the other hand, in a single-step model like IMLE, the model distribution changes during training, so the matching between model samples and ground truth data points needs to be recomputed. Due to randomness in the sampling, there may be no samples drawn from a mode of the model distribution, and as a result, a ground truth data point that could otherwise be explained by that mode would not be well matched. Note that this is different from mode collapse: the poor matching results from randomness in the sampling, not a deficiency in the model distribution.

Table 1: Ablation study on Oxford Flowers (Nilsback & Zisserman, 2008) (256×256), evaluated at the same training iteration. We report the mean ± standard deviation over 3 seeds. Row B adds multi-stage supervision to the baseline. Rows C-E each replace the loss function in Row B with a different robust loss.

	Configuration	FID ↓	Precision ↑	Recall ↑
A	Vanilla IMLE	62.61 ± 2.28	0.45 ± 0.03	0.46 ± 0.01
B	+ multi-stage supervision	31.24 ± 1.08	0.79 ± 0.01	0.72 ± 0.02
C	(B) + Cauchy loss	19.80 ± 0.58	0.88 ± 0.01	**0.74 ± 0.01**
D	(B) + Pseudo-Huber loss	16.49 ± 0.09	0.91 ± 0.00	0.72 ± 0.02
E	(B) + Geman-McClure loss **(ours)**	**13.90 ± 0.08**	**0.93 ± 0.00**	0.73 ± 0.02



(a) **Conf-A**: Vanilla IMLE (b) **Conf-B**: multi-stage supervision (c) **Conf-E**: multi-stage supervision + Geman-McClure loss

Figure 3: **Qualitative ablation**: Random samples from IMLE models trained on Oxford Flowers (256×256) for selected configurations in Table 1. The multi-stage supervision in Conf-B substantially improves sample quality over the vanilla IMLE in Conf-A. The robust loss in Conf-E further improves sample quality, yielding sharper and more realistic images.

Figure 4 shows this phenomenon, where the ground truth image (top row) is shown alongside its nearest-neighbour matches in the current round (middle) and the previous round (bottom). For most images, both matches found in consecutive rounds of sampling are close to the ground truth, indicating that the generator has learned these targets well. For the highlighted image, however, the current-round match does not look like the ground truth, while the previous-round match does.

The standard squared ℓ_2 loss lets these mismatches dominate training, since its gradient with respect to the generated sample is proportional to the residual. A mismatched pair has a large residual and therefore contributes a correspondingly large gradient. The problem is, in essence, one of outliers: a small number of large-residual pairs distort the gradient computed from a much larger pool of well-matched pairs.

The classical remedy in robust statistics is to replace the squared loss with a function that grows sub-quadratically in the residual. We adopt this approach: we replace $d(\cdot, \cdot)$ with a robust loss



Figure 4: **Mismatch in nearest-neighbour pairing.** Ground-truth image (top) and its nearest neighbours among generated samples in the current round (middle) and the previous round (bottom). For most images, both are close to the ground truth. For the highlighted image, the current-round nearest neighbour does not resemble the ground truth.

function. When a mismatch occurs and produces a large residual, the robust loss reduces its influence on the gradient, allowing the remaining well-matched pairs to dominate the update. We apply the robust loss at each decoder stage and ablate three candidates (Barron, 2019): Cauchy, Pseudo-Huber, and Geman-McClure. As shown in Table 1, all three improve performance over the non-robust loss, with Geman-McClure performing best. Figure 3 shows that Geman-McClure with multi-stage supervision also yields the best qualitative performance, producing sharper and more realistic images than the other configurations.

3.3 Round-Trip Rejection for Latent-Space Generation

For high-resolution datasets, a common paradigm is to generate in the latent space (Vahdat et al., 2021; Rombach et al., 2022) of a frozen, pretrained autoencoder and decode the generated latents into images. In the literature on SI models and GANs, inference-time tradeoffs of quality against diversity are popular and are enabled by techniques such as classifier-free guidance (CFG) (Ho & Salimans, 2022) and the truncation trick (Brock et al., 2019). To enable the same tradeoff, we design a simple technique, which we call round-trip rejection.

Latent space generation introduces a distribution mismatch: the autoencoder faithfully reconstructs only the images it was trained on, but our generator is not constrained to stay within that distribution and occasionally produces images the autoencoder does not represent well, which decode into visibly degraded samples. We add an inference-time step that identifies and discards these samples. We detect them using a *round-trip cost*: the distance between a generated image and the image obtained by encoding and then decoding it again. This round trip projects an image onto the distribution the autoencoder can represent, so a well-represented image changes little (low cost) while a poorly-represented one changes substantially (high cost). We reject generated samples whose round-trip cost is high. We include further implementation details in Appendix D.

Putting it together. Combining multi-stage supervision with the robust Geman-McClure loss yields our final training recipe, which we refer to as Robust Multi-Stage Implicit Maximum Likelihood Estimation (ROMS-IMLE). The architecture and inference path are unchanged from vanilla IMLE; the changes are entirely in the training objective. Algorithm 1 states the full procedure.

4 Related Works

One-step generative models GANs remain strong one-step image generators (Brock et al., 2019; Sauer et al., 2022; Zhang et al., 2022; Park & Kim, 2022). Several works (Karras et al., 2018; Denton et al., 2015; Karnewar & Wang, 2020) have tried to improve GAN by incorporating multi-scale supervision. GANs are however difficult to train (Arjovsky et al., 2017; Arjovsky & Bottou, 2017) and prone to mode collapse (Bau et al., 2019; Durall et al., 2020). A separate line of research tries to make iterative stochastic interpolant models fast by reducing the number of sampling steps, through distillation (Salimans & Ho, 2022), consistency training (Song et al., 2023), or redesigned objectives such as shortcut models (Frans et al., 2024), MeanFlow (Geng et al., 2025a;b), and inductive moment matching (Zhou et al., 2025). Although these methods sample in few steps at inference, their training relies on the underlying SI formulation. During training, the transformation from noise to data is still modelled as a sequence of smaller, simpler transformations, even though only a few of them are applied at inference. Moreover, their training objective typically pairs each data sample with an independently drawn noise sample, so the coupling between the prior and the data distribution is arbitrary during training. IMLE in contrast, chooses this coupling through an explicit matching phase that pairs each data point with its nearest generated sample.

In recent years, IMLE has received attention for its mode-covering objective and strong performance in few-shot image synthesis (Vashist et al., 2024; Aghabozorgi et al., 2023). IMLE has also been adopted beyond image generation, in areas such as visuomotor policy learning (Rana et al., 2025), trajectory planning (Lee et al., 2026), and shape completion (Arora et al., 2021).

5 Experiments

Datasets We demonstrate the performance of our method on three datasets: unconditional generation on CIFAR-10 (50K images, 32×32 resolution) (Krizhevsky, 2009) and CelebA-HQ (30K images,



Figure 5: Random samples from our unconditional model trained on CelebA-HQ

Table 2: Results for unconditional generation on CIFAR-10

Method	NFE	FID ↓	Pr. ↑	Re. ↑
DDIM	100	4.16	0.62	0.57
CT	1	8.79	<u>0.70</u>	0.42
Mean Flows	1	**2.87**	0.69	0.59
IMM	1	3.16	0.66	<u>0.60</u>
StyleFormer	1	<u>2.88</u>	0.64	0.54
ROMS-IMLE (Pixel)	1	3.93	**0.91**	**0.80**

Table 3: Results for unconditional generation on CelebA-HQ

Method	NFE	FID ↓	Pr. ↑	Re. ↑
Shortcut	1	23.65	<u>0.76</u>	0.16
Shortcut	128	10.79	0.71	<u>0.41</u>
StyleSwin	1	**5.32**	0.70	0.39
ROMS-IMLE (Pixel)	1	<u>6.70</u>	**0.96**	**0.61**

256×256 resolution) (Karras et al., 2018), and class-conditional generation on ImageNet 256 (Deng et al., 2009).

Evaluation metrics We use the FID-50k metric (Heusel et al., 2017) to assess the perceptual quality of the generated images, as is standard in prior work. Since FID is not always a perfect indicator of perceptual quality (Jayasumana et al., 2024), we additionally evaluate fidelity and coverage by computing precision and recall using the metrics defined by Kynkäänniemi et al. (2019). More details on the evaluation metrics are provided in Appendix E.

Implementation details Our network architecture comprises a fully-connected mapping network inspired by (Karras et al., 2019) and a synthesis network constructed using ConvNeXt (Liu et al., 2022) modules. We use bicubic interpolation for downsampling images/latents. We use the AdamW optimizer (Loshchilov & Hutter, 2019) with β_1, β_2 set to 0.9. Our method works in both pixel and latent space; to demonstrate that, we train our model in pixel space for CIFAR-10 and CelebA-HQ, and in the EQ-VAE (Kouzelis et al., 2025) latent space for ImageNet 256.

5.1 Results and Analysis

CIFAR-10 Table 2 shows that our model attains the best precision and recall in the table by wide margins: precision of 0.91 against 0.70 for the next best method (Song et al., 2023), and recall of 0.80 against 0.60 (Zhou et al., 2025). Notably, these margins hold even against the 100-step DDIM (Song et al., 2022) baseline (0.62 / 0.57), so a single forward pass of our model covers the data distribution better than a hundred evaluations of a diffusion sampler. At one NFE (number of function evaluations), our FID is competitive with the strongest one-step baselines (Geng et al., 2025a; Park & Kim, 2022).

CelebA-HQ Table 3 shows that our model has the best precision and recall, while being the second-best on FID. The 128-step Shortcut model (Frans et al., 2024) is outperformed on every metric despite

Table 4: Results for class-conditional image generation on ImageNet 256

Model	Params (M)	NFE	FID ↓	Pr. ↑	Re. ↑
BigGAN(Brock et al., 2019)	112	1	7.03	0.87	0.28
StyleGAN-XL(Sauer et al., 2022)	166	1	2.45	0.81	0.49
DiT-L/2 (Peebles & Xie, 2023)	458	250	23.30	-	-
DiT-XL/2	675	250	9.60	-	-
DiT-XL/2 (cfg = 1.5)	675	250	2.27	0.83	0.57
SiT-L/2 (Ma et al., 2024)	458	250	18.80	-	-
SiT-XL/2	675	250	8.30	-	-
SiT-XL/2 (cfg = 1.5)	675	250	2.06	0.83	0.59
iMeanFlow (Geng et al., 2025b)	610	1	1.72	0.78	0.63
IMM (1 step, cfg = 1.5) (Zhou et al., 2025)	675	1	8.05	0.67	0.64
IMM (2 steps, cfg = 1.5)	675	2	3.99	0.74	0.65
VAR-$d16$ (Tian et al., 2024)	310	10	3.30	0.84	0.51
VAR-$d30$-re	2000	10	1.73	0.82	0.60
ROMS-IMLE (Latent)	310	1	4.16	0.77	0.49
ROMS-IMLE (Latent) + Round-trip rejection	310	1	2.56	0.79	0.50

it using more evaluation steps. Compared to StyleSwin (Zhang et al., 2022), which achieves the best FID, our model produces more diverse samples (recall 0.61 vs. 0.39).

ImageNet 256 Table 4 shows that our latent-space model reaches an FID of 4.16 at one NFE with 310M parameters. With round-trip rejection, our FID improves to 2.56. Our FID is competitive with strong one-NFE and multi-step baselines such as iMeanFlow, StyleGAN-XL, DiT-XL/2, and SiT-XL/2.

6 Conclusion

Conclusion We introduced ROMS-IMLE, a single-step IMLE-based generative model. The method modifies IMLE training in two ways: multi-stage supervision at each upsampling resolution, and a robust loss that handles mismatched nearest-neighbour pairs. Across CIFAR-10, CelebA-HQ, and ImageNet 256, these changes make ROMS-IMLE competitive in FID while attaining strong precision and recall among one-step methods. More broadly, our results show that a single-step generator like IMLE can match much of the quality and diversity of iterative models.

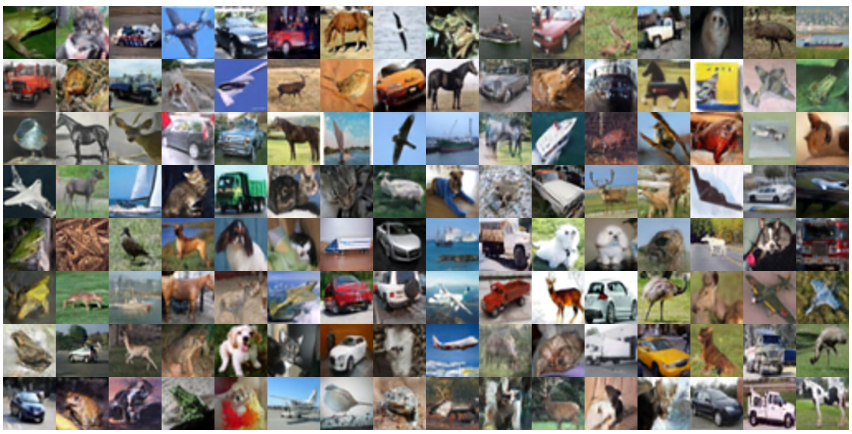


Figure 6: Random samples from our unconditional model trained on CIFAR-10

AI USE STATEMENT

We used AI tools for the following tasks:

- To format tables and figures in LaTeX.
- Implement methods: Coding agents suggested changes to optimize the code for increased throughput.
- Give feedback for the content of the paper to simulate peer review process.

The research ideas, methods, experiments, and technical content are the authors' own. We have reviewed and validated all AI-assisted work and take responsibility for the final content of the work.

ETHICS STATEMENT

This work adheres to the ICLR Code of Ethics. It involves no human subjects and no new data collection. All experiments use publicly available research datasets: CIFAR-10, CelebA-HQ, ImageNet, and Oxford Flowers. CelebA-HQ consists of face images of celebrities, and we use it only as a standard benchmark for image generation, following prior work. The main risk of this work is the misuse of generative models to create misleading images, that can be exploited for disinformation or fraudulent activities.

REPRODUCIBILITY STATEMENT

We describe the training procedure in Algorithm 1 and the experimental setup, datasets, and evaluation metrics in Section 5. Appendix D reports further details required for reproducing the results.

REFERENCES

Mehran Aghabozorgi, Shichong Peng, and Ke Li. Adaptive IMLE for few-shot pretraining-free generative modelling. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 248–264. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/aghabozorgi23a.html.

Michael S. Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants, 2023. URL https://arxiv.org/abs/2209.15571.

Martin Arjovsky and Léon Bottou. Towards principled methods for training generative adversarial networks, 2017. URL https://arxiv.org/abs/1701.04862.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan, 2017. URL https://arxiv.org/abs/1701.07875.

Himanshu Arora, Saurabh Mishra, Shichong Peng, Ke Li, and Ali Mahdavi-Amiri. Multimodal shape completion via imle, 2021. URL https://arxiv.org/abs/2106.16237.

Jonathan T Barron. A general and adaptive robust loss function. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4331–4339, 2019.

David Bau, Jun-Yan Zhu, Jonas Wulff, William Peebles, Hendrik Strobelt, Bolei Zhou, and Antonio Torralba. Seeing what a gan cannot generate, 2019. URL https://arxiv.org/abs/1910.11626.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=B1xsqj09Fm.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Emily Denton, Soumith Chintala, Arthur Szlam, and Rob Fergus. Deep generative image models using a laplacian pyramid of adversarial networks, 2015. URL https://arxiv.org/abs/1506.05751.

Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis, 2021. URL https://arxiv.org/abs/2105.05233.

Ricard Durall, Avraam Chatzimichailidis, Peter Labus, and Janis Keuper. Combating mode collapse in gan training: An empirical analysis using hessian eigenvalues, 2020. URL https://arxiv.org/abs/2012.09673.

Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One step diffusion via shortcut models, 2024. URL https://arxiv.org/abs/2410.12557.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, J. Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling, 2025a. URL https://arxiv.org/abs/2505.13447.

Zhengyang Geng, Yiyang Lu, Zongze Wu, Eli Shechtman, J Zico Kolter, and Kaiming He. Improved mean flows: On the challenges of fastforward generative models. *arXiv preprint arXiv:2512.02012*, 2025b.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014. URL https://arxiv.org/abs/1406.2661.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NIPS*, 2017.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance, 2022. URL https://arxiv.org/abs/2207.12598.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020. URL https://arxiv.org/abs/2006.11239.

Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization, 2017. URL https://arxiv.org/abs/1703.06868.

Sadeep Jayasumana, Srikumar Ramalingam, Andreas Veit, Daniel Glasner, Ayan Chakrabarti, and Sanjiv Kumar. Rethinking fid: Towards a better evaluation metric for image generation, 2024. URL https://arxiv.org/abs/2401.09603.

Animesh Karnewar and Oliver Wang. Msg-gan: Multi-scale gradients for generative adversarial networks, 2020. URL https://arxiv.org/abs/1903.06048.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation, 2018. URL https://arxiv.org/abs/1710.10196.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks, 2019. URL https://arxiv.org/abs/1812.04948.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models, 2022. URL https://arxiv.org/abs/2206.00364.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2022. URL https://arxiv.org/abs/1312.6114.

Theodoros Kouzelis, Ioannis Kakogeorgiou, Spyros Gidaris, and Nikos Komodakis. Eq-vae: Equivariance regularized latent space for improved generative image modeling, 2025. URL https://arxiv.org/abs/2502.09509.

Alex Krizhevsky. Learning multiple layers of features from tiny images. pp. 32–33, 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. *Advances in Neural Information Processing Systems*, 32, 2019.

Grayson Lee, Minh Bui, Shuzi Zhou, Yankai Li, Mo Chen, and Ke Li. Implicit maximum likelihood estimation for real-time generative model predictive control, 2026. URL https://arxiv.org/abs/2603.13733.

Ke Li and Jitendra Malik. Implicit maximum likelihood estimation, 2018. URL https://arxiv.org/abs/1809.09087.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling, 2023. URL https://arxiv.org/abs/2210.02747.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s, 2022. URL https://arxiv.org/abs/2201.03545.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. URL https://arxiv.org/abs/1711.05101.

Nanye Ma, Mark Goldstein, Michael S Albergo, Nicholas M Boffi, Eric Vanden-Eijnden, and Saining Xie. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In *European Conference on Computer Vision*, pp. 23–40. Springer, 2024.

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing*, Dec 2008.

Jeeseung Park and Younggeun Kim. Styleformer: Transformer based generative adversarial networks with style vector, 2022. URL https://arxiv.org/abs/2106.07023.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4195–4205, 2023.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. URL https://arxiv.org/abs/2103.00020.

Krishan Rana, Robert Lee, David Pershouse, and Niko Suenderhauf. Imle policy: Fast and sample efficient visuomotor policy learning via implicit maximum likelihood estimation, 2025. URL https://arxiv.org/abs/2502.12371.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022. URL https://arxiv.org/abs/2112.10752.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models, 2022. URL https://arxiv.org/abs/2202.00512.

Axel Sauer, Katja Schwarz, and Andreas Geiger. Stylegan-xl: Scaling stylegan to large diverse datasets, 2022. URL https://arxiv.org/abs/2202.00273.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models, 2022. URL https://arxiv.org/abs/2010.02502.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations, 2021. URL https://arxiv.org/abs/2011.13456.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models, 2023. URL https://arxiv.org/abs/2303.01469.

George Stoica, Vivek Ramanujan, Xiang Fan, Ali Farhadi, Ranjay Krishna, and Judy Hoffman. Contrastive flow matching, 2025. URL https://arxiv.org/abs/2506.05350.

Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction, 2024. URL https://arxiv.org/abs/2404.02905.

Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space, 2021. URL https://arxiv.org/abs/2106.05931.

Chirag Vashist, Shichong Peng, and Ke Li. Rejection sampling imle: Designing priors for better few-shot image synthesis, 2024. URL https://arxiv.org/abs/2409.17439.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. URL https://arxiv.org/abs/1706.03762.

Zhisheng Xiao, Karsten Kreis, and Arash Vahdat. Tackling the generative learning trilemma with denoising diffusion gans, 2022. URL https://arxiv.org/abs/2112.07804.

Bowen Zhang, Shuyang Gu, Bo Zhang, Jianmin Bao, Dong Chen, Fang Wen, Yong Wang, and Baining Guo. Styleswin: Transformer-based gan for high-resolution image generation, 2022. URL https://arxiv.org/abs/2112.10762.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric, 2018. URL https://arxiv.org/abs/1801.03924.

Linqi Zhou, Stefano Ermon, and Jiaming Song. Inductive moment matching. In *Forty-second International Conference on Machine Learning*, 2025.

A Note on Maximum Likelihood Estimation

Maximum Likelihood Estimation (MLE) is widely used in generative modeling because it provides a principled method for fitting models to data by maximizing the likelihood that observed samples originate from the model's distribution. Consequently, samples generated from models optimized with MLE (and its variants) are, by design, expected to closely resemble the data points. Directly optimizing the likelihood is often computationally intractable, since the data is generally high-dimensional and the model's mapping from the prior to the data distribution is typically not surjective. Different generative models take different approaches to get around this issue.

Diffusion models, for example, circumvent this by maximizing a variational lower bound known as the Evidence Lower Bound (ELBO). Specifically, diffusion models factorize the generation process into multiple small, iterative transitions from a known prior distribution $p(\mathbf{x}_T)$. This can be written as follows:

$$\begin{aligned} p(\mathbf{x}_0 \mid \mathbf{x}_T) &= \int \cdots \int p_\theta(\mathbf{x}_0 \mid \mathbf{x}_1)\, p_\theta(\mathbf{x}_1 \mid \mathbf{x}_2) \cdots p_\theta(\mathbf{x}_{T-1} \mid \mathbf{x}_T)\, \mathrm{d}\mathbf{x}_1 \cdots \mathrm{d}\mathbf{x}_{T-1} \\ &= \int \cdots \int \left[\prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) \right] \mathrm{d}\mathbf{x}_1 \cdots \mathrm{d}\mathbf{x}_{T-1} \end{aligned} \tag{5}$$

In contrast, Implicit Maximum Likelihood Estimation (IMLE) models the transition from the prior to the data in a single step, implicitly optimizing the MLE without explicitly factorizing into intermediate distributions.

B Experiment details for ablation study on Oxford Flowers

This section provides additional details for the ablation study on the Oxford Flowers dataset, presented in Table 1 of the main paper. All configurations (A through E) use the identical architecture and the exact training setup; the only variable changed is the training objective. The only minor difference is that multi-resolution configurations have a trainable 1×1 convolution layer for intermediate outputs per stage. This change slightly increases the number of parameters for configurations B-E by network width $\times$ output channels $\times$ number of stages. In practice, it is close to roughly 4K extra parameters (varies slightly depending on the dataset), on the order 10^{-5} of the model size. In our actual implementation, we just set these parameters to not be trainable for vanilla IMLE (configuration A).

All the configurations were trained for exactly the same number of steps. Comparing method variants for the same number of training iterations (epochs) under an identical training setting is standard practice in the literature (Kouzelis et al., 2025; Stoica et al., 2025).

Table 5: FID ↓ over training for the five configurations of Table 1, reported as mean ± standard deviation over 3 seeds.

	Epoch							
Conf.	500	1000	1500	2000	2500	3000	3500	3900
A	169.66 ± 4.17	134.31 ± 5.30	115.20 ± 3.65	101.40 ± 2.44	88.91 ± 4.48	75.71 ± 1.71	68.03 ± 1.40	62.61 ± 2.28
B	142.41 ± 2.25	105.56 ± 6.23	82.57 ± 3.72	65.16 ± 3.29	51.86 ± 2.17	43.66 ± 1.93	34.77 ± 1.06	31.24 ± 1.08
C	108.94 ± 5.00	67.56 ± 2.92	51.12 ± 9.02	40.74 ± 1.13	30.10 ± 1.73	25.12 ± 1.14	21.30 ± 0.66	19.80 ± 0.58
D	97.02 ± 4.45	55.79 ± 4.53	36.96 ± 1.45	28.84 ± 0.63	24.01 ± 1.61	20.59 ± 1.61	16.44 ± 0.58	16.49 ± 0.09
E	**83.52 ± 5.43**	**46.26 ± 3.55**	**35.24 ± 2.32**	**24.57 ± 1.45**	**18.77 ± 0.17**	**15.97 ± 0.91**	**15.42 ± 0.60**	**13.90 ± 0.08**

Our proposed method converges roughly $5\times$ faster than the original under identical training conditions: it reaches the baseline's final FID (62.61 at 3900 epochs) in ~ 800 epochs. Crucially, at every matched step count our proposed method's FID is strictly lower. Table 5 reports FID at 500-epoch intervals for all five configurations, each as mean ± standard deviation over 3 seeds, and Figure 7 plots the full FID trajectory over training.

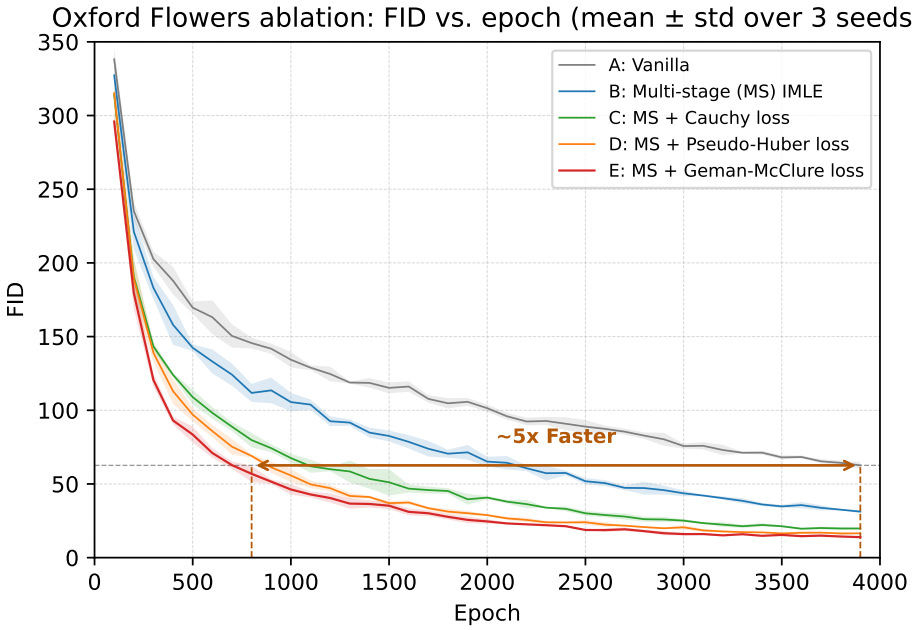


Figure 7: FID vs. epoch on Oxford Flowers for the five configurations of Table 1, reported as mean ± standard deviation over 3 seeds. Our proposed configuration (E) reaches the baseline's (A) final FID roughly 5x faster.

C Ablation on Sampling Budget

The IMLE matching phase selects nearest neighbours for n data points among m generated samples. In this section, we ablate the sampling budget m/n. For our configuration E from Table 1, we report the effect of the sampling budget on FID for Oxford-Flowers. We vary the ratio m/n and report FID at different training epochs in Table 6. Even with a small sample factor (2), our method converges and reaches reasonable performance. The gains show clear diminishing returns: moving from 4 to 10 improves the final FID only marginally. These results show that IMLE does not need to generate a large number of samples in the matching phase to achieve good performance.

Table 6: Effect of the nearest-neighbour sampling budget on FID for Oxford-Flowers.

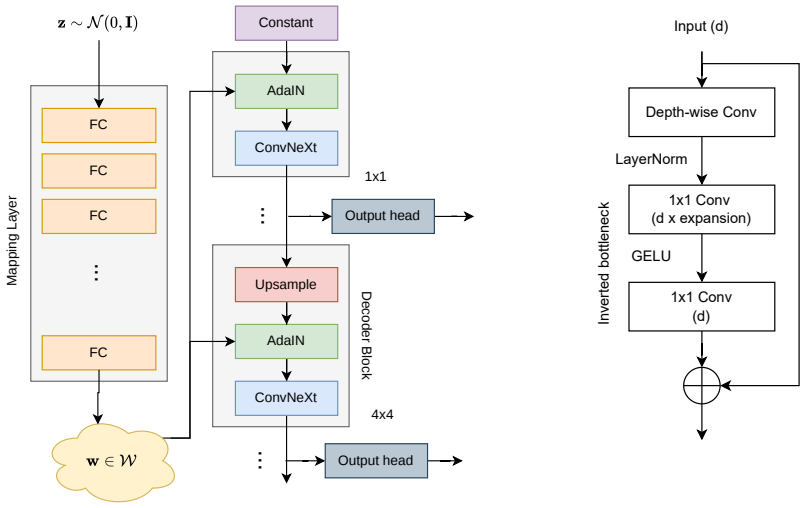
Epoch	$m/n = 2$	$m/n = 4$	$m/n = 10$
500	94.42	78.65	66.37
1000	64.60	49.38	39.54
1500	46.04	35.92	26.76
2000	36.04	25.80	20.59
2500	26.58	18.88	17.68
3000	27.34	16.79	15.00
3500	23.56	14.85	13.41

For the experiments in Section 5, we use a budget of $m/n = 5$ across all three datasets (CIFAR-10, CelebA-HQ, ImageNet 256).

D Architecture and Training Details

We provide the detailed network architecture in Figure 8.
Our generator maps a latent code ($\mathbf{z} \sim \mathcal{N}(0, \mathbf{I}), \mathbf{z} \in \mathbb{R}^{1024}$) to a style vector $\mathbf{w}$ via an MLP (mapping layer), which modulates each layer of a ConvNeXt-style decoder using Adaptive Instance

Normalization (AdaIN)(Huang & Belongie, 2017). Starting from a learned constant feature map at low resolution, the decoder progressively upsamples the representation through a stack of residual ConvNeXt(Liu et al., 2022) blocks with per-resolution width control.



(a) Network Architecture (b) Standard ConvNeXt block

Figure 8: (a) Network architecture, which comprises a mapping network, upsampling layers and ConvNeXt blocks. (b) Inner workings of a standard ConvNeXt block.

Training Details The training was done on NVIDIA L40S GPUs. The training time depends primarily on the size of the dataset. Default PyTorch implementation of bicubic interpolation was used for all resizing operations, because it achieved the best results. We used a cosine scheduler with warmup for 100 iterations.

Pixel-based models We used a weighted combination of LPIPS (Zhang et al., 2018) loss and pixel loss for optimization, with weighting factors of 1.0 and 0.1 respectively. LPIPS requires images to be of at least 32×32 resolution; hence images below this resolution were resized to 32×32 before being passed into the LPIPS network. The 32×32 variant (used for CIFAR-10) contains 110 million trainable parameters, while the 256×256 variant (used for CelebA-HQ) has 138 million. We do not use any data augmentation.

Latent-based models For ImageNet, we train our generator in the latent space of an EQ-VAE (Kouzelis et al., 2025), which is kept frozen throughout training. The model contains 310 million trainable parameters. The per-stage training loss is the Geman-McClure loss applied to elementwise residuals computed directly on the latent representations. Per-stage targets are obtained by encoding the real image with the frozen EQ-VAE encoder and downsampling the resulting latent map with K_l to each stage's resolution.

We use rejection sampling to remove low-quality samples, with a different rejection criterion: at evaluation time, we compute for each generated sample the round-trip reconstruction cost of encoding and decoding it with the EQ-VAE. The distance between the generated sample and its round-trip reconstruction is measured in LPIPS (Zhang et al., 2018) space. Low-quality samples incur a high round-trip cost, and we reject samples whose cost exceeds a fixed threshold, which removes roughly 5% of generated samples. We find that removing these samples improves FID. Reported ImageNet metrics are computed on samples that pass this filter.

Distance metric for nearest-neighbour selection. The distance $d(\cdot, \cdot)$ used in the $\arg\min_j$ step of Algorithm 1 differs between pixel-space and latent-space models. For our pixel-space models (CIFAR-10, CelebA-HQ), we use LPIPS (Zhang et al., 2018) as the selection distance. For our

latent-space model on ImageNet 256, we use the squared ℓ_2 distance in the EQ-VAE (Kouzelis et al., 2025) latent space.

Nearest-neighbour search at scale. We use FAISS-GPU to accelerate exact nearest-neighbour search across the m candidate samples.

E Note on Evaluation Metrics

In order to evaluate a generative model, we need a holistic picture across complementary metrics. While FID remains the most widely reported, it is known to have serious flaws: it compresses everything into a single scalar and cannot indicate what a model is lacking. In particular, Jayasumana et al. (2024) shows that two very different distributions can be considered identical under the Gaussian feature assumption of the Fréchet distance. Hence metrics like precision and recall, which separately measure fidelity (how realistic samples are) and coverage (how well the data distribution is covered), give a more complete assessment. For our case, high precision and recall indicate that individual samples are realistic and that the generated distribution covers the real data well: the two properties one ultimately cares about in a generative model. We read the combination of strong precision/recall with competitive FID as evidence that the method produces realistic, diverse samples, rather than optimizing a single distributional statistic.

Protocols also vary across papers, most notably in the Inception weights and numbers of samples used in the evaluation. These choices can shift the resulting values. To have consistency, we use the most popular protocol described by Dhariwal & Nichol (2021) and apply it uniformly to every number we computed.

For FID, we calculate the Fréchet distance between the real and generated distributions in the feature space of a pre-trained InceptionV3 network using 50K generated images. Some implementations, most notably StyleSwin (Zhang et al., 2022), report their metrics using different number of samples (30K instead of 50K). In such cases, we used the official weights to recalculate the metric.

We report precision and recall using the standard implementation of Kynkäänniemi et al. (2019), with the InceptionV3 feature extractor and $k = 3$ for the k-nearest-neighbour radius. This matches the protocol most commonly used in prior image generation literature. Not all baselines report precision and recall, so we computed the missing numbers ourselves.

F Ablation on a Transformer-Based Generator

Our motivation was minimalist generative modeling, which is why we used and refined a convolution-based decoder for the paper. A natural question arises: are our contributions architecture-agnostic and can we transfer our training recipe to a different architecture? To answer this question, we repeat the ablation from Table 1 on Oxford Flowers using a transformer-based generator. We construct this variant by replacing the ConvNeXt blocks in our decoder with attention blocks (Vaswani et al., 2023), keeping the rest of the pipeline unchanged. The resulting model has 65M parameters. We train all configurations of this model for 4000 epochs, without any architecture-specific tuning. The results are reported in Table 7. Starting from this attention-based generator without either contribution (Vanilla IMLE), we add multi-stage supervision, and then the Geman-McClure robust loss. The trend from Table 1 holds on this different backbone: multi-stage supervision improves over vanilla IMLE, and the robust loss improves further, confirming that our training recipe transfers across architectures.

Table 7: Ablation on Oxford Flowers with a transformer-based decoder. Single run, without architecture-specific tuning.

	Configuration	FID ↓	Pr. ↑	Re. ↑
A	Vanilla IMLE	61.30	0.46	0.38
B	(A) + multi-stage supervision	33.32	0.76	0.72
E	(B) + Geman-McClure loss	**19.27**	**0.87**	**0.76**

We did not tune the attention architecture extensively. Our aim in this ablation is only to show that the contributions of ROMS-IMLE can be transferred to a different architecture, and that the same



Figure 9: **Prior-space interpolation for CelebA-HQ**. Interpolating linearly between two prior samples $\mathbf{z}$ and decoding each intermediate point produces plausible images that vary smoothly, suggesting the generator has learned a coherent mapping from the prior rather than memorizing the training data.

expected trend holds. In this ablation, we did not aim to design an attention-based model that achieves state-of-the-art performance; doing so would require its own iteration on architectural design choices, which is outside the scope of this ablation and left for future work.

G NEAREST-NEIGHBOUR TEST

A natural concern for a generative model trained with a nearest-neighbour matching objective is that the generator could memorize training images rather than synthesize novel ones. To check for this, we retrieve the top-8 nearest neighbours of each generated sample from the CelebA-HQ training set, using cosine similarity in CLIP embedding space (Radford et al., 2021). Figure 10 shows this comparison for a set of generated samples. These results, along with interpolation results in Figure 9, indicate that our model synthesizes novel images rather than reproducing individual training examples.



Figure 10: **Nearest-neighbour analysis on CelebA-HQ.** Each row shows a generated sample together with its nearest neighbours from the training set, retrieved by cosine similarity using CLIP encoder (Radford et al., 2021). Our generated samples are visually distinct from their nearest neighbours, indicating that the model is not memorizing training images.