Dual-Rate Diffusion: Accelerating diffusion models with an interleaved heavy-light network

Grigory Bartosh[1,2]* **David Ruhe**[2] **Emiel Hoogeboom**[2] **Jonathan Heek**[2]

Thomas Mensink[2] **Tim Salimans**[2]

[1]University of Amsterdam [2]Google DeepMind Amsterdam

Abstract

Diffusion models achieve state-of-the-art generative performance but suffer from high computational costs during inference due to the repeated evaluation of a heavy neural network. In this work, we propose Dual-Rate Diffusion, a method to accelerate sampling by interleaving the execution of a heavy high-capacity context encoder and a light efficient denoising model. The context encoder is evaluated sparsely to extract high-dimensional features, which the light denoising model reuses at every step to refine the sample. This approach significantly accelerates inference without compromising sample quality. On ImageNet benchmarks, Dual-Rate Diffusion matches the performance of standard baselines while reducing computational cost by a factor of 2–4. Furthermore, we demonstrate that our method is compatible with distillation techniques, such as Moment Matching Distillation, which enables further efficiency gains in few-step generation.

1 Introduction

Diffusion models have recently demonstrated remarkable performance across various domains, including image (Dhariwal and Nichol, 2021), video (Ho et al., 2022b), and audio generation (Kong et al., 2021). Despite their impressive sample quality, these models are computationally expensive during inference, necessitating numerous function evaluations to generate a single sample.

To reduce inference time, recent research has proposed techniques such as distillation and direct training of few-step diffusion models. While these methods decrease the number of function evaluations, the underlying function typically remains parameterized by the same computationally heavy neural network at each step. At the same time, other studies (Chen et al., 2023; Liu et al., 2025a; Ma et al., 2024) suggest that computations from earlier denoising steps can be efficiently reused, implying that a high-capacity network may not be necessary at every step.

In this work, we introduce Dual-Rate Diffusion, a simple technique to accelerate diffusion models by interleaving a heavy context encoder and a light denoising model, see Figure 1. The context encoder is evaluated at selected steps to compute high-dimensional features of the noisy samples. These features are then utilized by the denoising model to handle the denoising steps more efficiently.

This approach is motivated by two perspectives. First, it generalizes self-conditioning (Chen et al., 2023), where the model's own prediction of the clean data from the previous step is fed back as an additional input: in Dual-Rate Diffusion, the conditioning is not restricted to the form or dimensionality of the data, and it is produced by a separate network, which decouples the per-step

*Work done while interning at Google DeepMind Amsterdam.
Corresponding authors: bartosh@google.com, ruhe@google.com, mensink@google.com

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

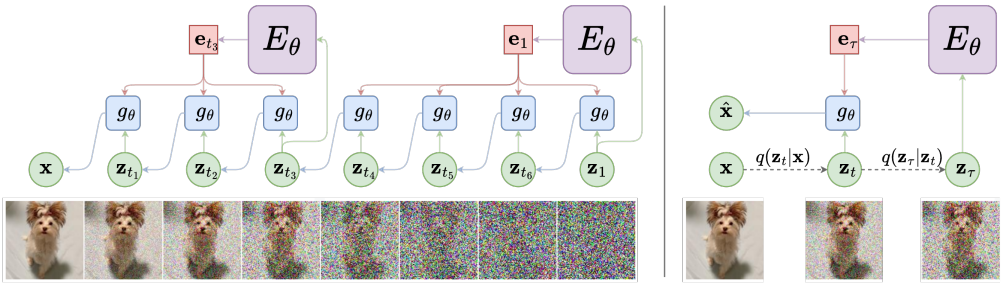

Figure 1: Overview of Dual-Rate Diffusion generative process (on the left) and training (on the right). The context encoder E_Θ (heavy network) is evaluated at sparse intervals to extract high-dimensional features, which are then reused by the light denoising model g_θ at every step to refine the sample.

computation from the network that extracts the global structure. Second, we can view the denoising process in the frequency domain, where the global structure (low frequencies) is established in early steps and remains stable, while local details emerge later (Dieleman, 2024; Rissanen et al., 2023). Evaluating the denoising model independently at every time step leads to repeated, redundant processing of this global structure, which consumes the model capacity. Dual-Rate Diffusion delegates this task to the heavy context encoder, evaluated once per interval of steps; in the steps that follow, the light denoising model reuses the extracted information and focuses on local structure.

We demonstrate that Dual-Rate Diffusion significantly reduces inference time for standard diffusion models without compromising sample quality. On ImageNet benchmarks, Dual-Rate Diffusion matches the sample quality of standard baselines while reducing the computational cost by a factor of 2–4. Furthermore, we show that this method is compatible with distillation techniques such as MMD (Salimans et al., 2024), where Dual-Rate students sample at a lower cost than standard MMD students and improve on them on ImageNet 64×64.

2 Background

Diffusion Models Diffusion models (Ho et al., 2020) consist of a forward process and a reverse process. The forward process gradually corrupts a data point $\mathbf{x}$ by injecting Gaussian noise. Intermediate noisy states $\mathbf{z}_t \sim q(\mathbf{z}_t|\mathbf{x})$ at time step $t \in [0, 1]$ can be generated as:

$$\mathbf{z}_t = \alpha_t \mathbf{x} + \sigma_t \varepsilon, \quad \text{where} \quad \varepsilon \sim \mathcal{N}(0, \mathbf{I}). \tag{1}$$

We employ a variance-preserving noise schedule (Song et al., 2021b), where $\alpha_t^2 + \sigma_t^2 = 1$, defined by the log signal-to-noise ratio (log-SNR) $\lambda_t = \log(\alpha_t^2/\sigma_t^2)$, which is a monotonic function of t.

The reverse process is an iterative refinement process. It starts from pure noise $\mathbf{z}_1$ and gradually denoises it to recover a data sample $\mathbf{z}_0$. At each iteration, the model predicts the clean data $\hat{\mathbf{x}} = g_\theta(\mathbf{z}_t, t)$ and then samples a less noisy state $\mathbf{z}_s$ (with $s < t$) from the posterior distribution $q(\mathbf{z}_s|\mathbf{z}_t, \mathbf{x} = \hat{\mathbf{x}})$. Finally, $\mathbf{x}$ is set to $\mathbf{z}_0$ exactly or quantized.

As the number of steps approaches infinity, a variational bound on the model log-likelihood is given by Ho et al. (2020):

$$\begin{aligned} \mathrm{D_{KL}}\big(q(\mathbf{x})\|p_\theta(\mathbf{x})\big) &\leq \mathrm{D_{KL}}\big(q(\mathbf{x}, \mathbf{z}_0, \ldots, \mathbf{z}_1)\|p_\theta(\mathbf{x}, \mathbf{z}_0, \ldots, \mathbf{z}_1)\big) \\ &\approx \tfrac{1}{2}\mathbb{E}_{u(t)q(\mathbf{x},\mathbf{z}_t)}\Big[-\tfrac{\mathrm{d}\lambda_t}{\mathrm{d}t} e^{\lambda_t} w(\lambda_t)\big\|\mathbf{x} - g_\theta(\mathbf{z}_t, t)\big\|^2\Big] + \underbrace{\mathrm{D_{KL}}\big(q(\mathbf{z}_1)\|p(\mathbf{z}_1)\big)}_{\approx 0}, \end{aligned} \tag{2}$$

where $u(t)$ is a uniform distribution over $t \in [0, 1]$ and $w(\lambda_t)$ is a weighting function. The objective is minimized when $g_\theta(\mathbf{z}_t, t) = \mathbb{E}_q[\mathbf{x}|\mathbf{z}_t]$. The bound holds strictly for $w(\lambda_t) \equiv 1$. In practice, however, a re-weighted ELBO (e.g., $w(\lambda_t) = \mathrm{sigmoid}(\lambda_t - b)$) is often used to improve sample quality (Kingma and Gao, 2023).

Few-Step Generation and Moment Matching Distillation Few-step generative models are commonly obtained by distilling a pretrained diffusion model, and existing methods fall into two

broad families. Trajectory-based methods, such as progressive distillation (Salimans and Ho, 2022) and consistency models (Song et al., 2023), train the student to reproduce the deterministic sampling trajectory of the teacher in fewer steps. Distribution matching methods, such as DMD (Yin et al., 2024b) and Moment Matching Distillation (MMD) (Salimans et al., 2024), instead train the student so that the distribution of its samples matches the distribution induced by the teacher. In this work, we focus on MMD.

MMD adapts the standard iterative refinement process for few-step generation. Given a noisy sample $\mathbf{z}_t$, rather than estimating the expectation $\mathbb{E}_q[\mathbf{x}|\mathbf{z}_t]$, MMD generates samples $\tilde{\mathbf{x}}$ from a learnable distribution $p_\eta(\tilde{\mathbf{x}}|\mathbf{z}_t)$ and subsequently samples $\mathbf{z}_s$ from the forward posterior $q(\mathbf{z}_s|\mathbf{z}_t, \mathbf{x} = \tilde{\mathbf{x}})$. This procedure implicitly defines the distribution $p_\eta(\tilde{\mathbf{x}}|\mathbf{z}_s)$. It can be shown that if $p_\eta(\tilde{\mathbf{x}}|\mathbf{z}_s) \equiv q(\mathbf{x}|\mathbf{z}_s)$, the resulting process yields correct marginals $p_\eta(\mathbf{z}_s) = q(\mathbf{z}_s)$ for any $s < t$. Matching these conditional distributions is thus the proper objective; MMD approximates it by matching their first moments, minimizing the L2-distance:

$$\tilde{\mathcal{L}} = \frac{1}{2}\mathbb{E}_{q(\mathbf{z}_t)p_\eta(\mathbf{z}_s|\mathbf{z}_t)}\left[\left\|\mathbb{E}_{p_\eta}[\tilde{\mathbf{x}}|\mathbf{z}_s] - \mathbb{E}_q[\mathbf{x}|\mathbf{z}_s]\right\|^2\right]. \tag{3}$$

This objective can be reformulated to simplify optimization while preserving gradients $\nabla_\eta\tilde{\mathcal{L}} = \nabla_\eta\mathcal{L}$:

$$\mathcal{L} = \mathbb{E}_{q(\mathbf{z}_t)p_\eta(\tilde{\mathbf{x}},\mathbf{z}_s|\mathbf{z}_t)}\left[\tilde{\mathbf{x}}^{\mathrm{T}}\mathrm{sg}\left(\mathbb{E}_{p_\eta}[\tilde{\mathbf{x}}|\mathbf{z}_s] - \mathbb{E}_q[\mathbf{x}|\mathbf{z}_s]\right)\right], \tag{4}$$

where $\mathrm{sg}(\cdot)$ denotes the stop-gradient operator. The intractable moments in this objective are approximated using an auxiliary model $g_\varphi(\mathbf{z}_t, t) \approx \mathbb{E}_{p_\eta}[\tilde{\mathbf{x}}|\mathbf{z}_s]$ and a teacher model $g_\theta(\mathbf{z}_t, t) \approx \mathbb{E}_q[\mathbf{x}|\mathbf{z}_s]$. Both g_φ and g_θ can be trained via the standard diffusion loss (Equation (2)), with g_θ potentially being a pretrained diffusion model. In practice, MMD uses a deterministic student model $\tilde{\mathbf{x}} = g_\eta(\mathbf{z}_t, t)$ which learns to generate samples matching the teacher model's moments.

3 Dual-Rate Diffusion

Dual-Rate Diffusion accelerates denoising by employing a dual-network architecture: a computationally heavy network (context encoder) and a light network (denoising model). The context encoder is evaluated only at sparse, fixed intervals (e.g., 16 times along a 512-step sampling trajectory) and extracts high-dimensional features that capture the global structure of the sample. The denoising model runs at every step and reuses the most recent features instead of recomputing the global structure, which lets it focus its computation on refining local details (see Figure 1). In this section, we present the sampling and training procedures of Dual-Rate Diffusion in the standard diffusion setting and then show how the construction applies to distillation, using MMD as an example.

3.1 Diffusion

The context encoder processes a noisy sample $\mathbf{z}_\tau$ at a specific time step τ to generate a high-dimensional feature vector $\mathbf{e}_\tau = E_\Theta(\mathbf{z}_\tau, \tau)$. The denoising model then predicts the clean data $\hat{\mathbf{x}} = g_\theta(\mathbf{z}_t, t, \mathbf{e}_\tau)$ using the current noisy sample $\mathbf{z}_t$, the current time step $t \leq \tau$, and the most recently computed feature $\mathbf{e}_\tau$, where τ refers to the closest preceding evaluation step of the context encoder. If the denoising model is evaluated k times during sampling, the context encoder is evaluated $K \leq k$ times, once every k/K steps of the denoising model, where K is a divisor of k. Whenever the context encoder is evaluated, it is evaluated on the current state, so $\mathbf{z}_\tau = \mathbf{z}_t$ at that step, and the features $\mathbf{e}_\tau$ are then held fixed for the following $k/K - 1$ denoising steps. Each denoising step is otherwise a standard ancestral step: the denoising model predicts the clean data, and the next state is sampled from the forward posterior $q(\mathbf{z}_s|\mathbf{z}_t, \hat{\mathbf{x}})$. The cost of a trajectory is therefore K evaluations of the heavy network plus k evaluations of the light one, and the ratio k/K sets the trade-off between the two. This sampling procedure is summarized in Algorithm 1.

Conditioning mechanism. We parameterize both networks with hierarchical UViT architectures (Hoogeboom et al., 2025); Figure 2 depicts how the two are connected. The features $\mathbf{e}_\tau$ are the outputs of the last layer at each spatial resolution level of the context encoder, so they retain the spatial layout of the sample at every level rather than being pooled into a single vector. At the beginning of the corresponding level of the denoising model, they are linearly projected to match the dimensions and concatenated with the denoising model's own features, which lets each level of the light network

Algorithm 1 Ancestral sampling algorithm for Dual-Rate Diffusion.

Require: Context encoder E_Θ, denoising model g_θ, number of context encoder evaluations K and denoising steps k.

Initialize noisy data $\mathbf{z}_1 \sim \mathcal{N}(0, \mathbf{I})$
for $t \in \{1, (k-1)/k, \dots, 1/k\}$ **do**
  if $t \in \{1, (K-1)/K, \dots, 1/K\}$ **then**
    Evaluate context encoder $\mathbf{e}_\tau = E_\Theta(\mathbf{z}_t, t)$
  end if
  Predict clean data with denoising model $\hat{\mathbf{x}} = g_\theta(\mathbf{z}_t, t, \mathbf{e}_\tau)$
  Set next time step $s = t - 1/k$
  Sample next noisy point $\mathbf{z}_s \sim q(\mathbf{z}_s | \mathbf{z}_t, \hat{\mathbf{x}})$
end for
return Approximate sample $\hat{\mathbf{x}}$

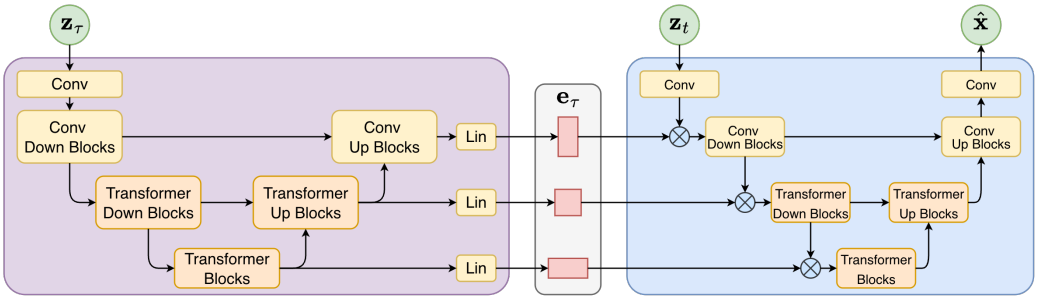

Figure 2: Overview of the Dual-Rate Diffusion conditioning mechanism. The context encoder (on the left) generates high-dimensional features at each level of the UViT architecture, which are then reused at the corresponding level of the denoising UViT (on the right). The $\otimes$ symbol denotes concatenation.

start from the representation the heavy network computed at the same level. The denoising model is additionally conditioned on the time step τ, which tells it how stale the features are; we omit this from the notation $g_\theta(\mathbf{z}_t, t, \mathbf{e}_\tau)$.

Training. The Dual-Rate Diffusion objective extends the standard diffusion loss (Equation (2)):

$$\mathrm{D}_{\mathrm{KL}}\big(q(\mathbf{x}) \| p_{\Theta,\theta}(\mathbf{x})\big) \leq \frac{1}{2}\mathbb{E}_{u(t,\tau)q(\mathbf{x},\mathbf{z}_t,\mathbf{z}_\tau)}\left[-\frac{\mathrm{d}\lambda_t}{\mathrm{d}t}e^{\lambda_t} w(\lambda_t)\left\|\mathbf{x} - g_\theta\big(\mathbf{z}_t, t, E_\Theta(\mathbf{z}_\tau, \tau)\big)\right\|^2\right], \quad (5)$$

where $u(t, \tau)$ denotes a uniform distribution over the joint time steps. In our experiments, we sample τ uniformly from the interval $[1/K, 1]$ and subsequently sample t uniformly from $(\tau - 1/K, \tau]$. This objective remains a valid variational bound on the model likelihood when $w(\lambda_t) = 1$. Algorithm 2 details the training procedure.

Markovian processes. Unlike standard diffusion, correct sampling in Dual-Rate Diffusion generally requires matching the joint distribution $p_{\Theta,\theta}(\mathbf{z}_t, \mathbf{z}_\tau) \approx q(\mathbf{z}_t, \mathbf{z}_\tau)$ during inference. Since the denoising model g_θ depends on both $\mathbf{z}_t$ and $\mathbf{z}_\tau$, using non-Markovian posteriors $q(\mathbf{z}_s|\mathbf{x}, \mathbf{z}_t)$ (e.g., DDIM (Song et al., 2021a)) causes a distribution mismatch between training and inference. Training with the same non-Markovian posterior does not resolve this. With a Markovian posterior, the generative transitions $p(\mathbf{z}_s|\mathbf{z}_t)$ approach the forward transitions $q(\mathbf{z}_s|\mathbf{z}_t)$ as the model approaches the data posterior, and the joint distribution of $(\mathbf{z}_t, \mathbf{z}_\tau)$ is determined by these transitions, exactly as in standard diffusion. With a non-Markovian posterior, every state between $\mathbf{z}_\tau$ and $\mathbf{z}_t$ is tied to the same clean sample $\mathbf{x}$ under the forward process, whereas the generative process replaces $\mathbf{x}$ by a new prediction $\hat{\mathbf{x}}$ at every step, so the two induce different joint distributions of $(\mathbf{z}_t, \mathbf{z}_\tau)$ regardless of how the model is trained. Therefore, we use Markovian processes for both training and sampling; the effect of applying a deterministic sampler to a trained model is quantified in Section 4.1.

The Markovian requirement concerns the target process, not the model's own. Each generative step depends on two previous states, $\mathbf{z}_t$ and $\mathbf{z}_\tau$, so the process the model defines is itself non-Markovian; its target is Markovian: $\mathbf{z}_\tau$ carries no information about the next state beyond what $\mathbf{z}_t$ already contains,

Algorithm 2 Dual-Rate Diffusion training algorithm.

Require: Context encoder E_Θ, denoising model g_θ, number of context encoder evaluations K and denoising steps k, loss weight $w(\lambda_t)$, and dataset $\mathcal{D}$.

for $n = 0{:}N$ **do**
  Sample clean data $\mathbf{x} \sim \mathcal{D}$
  Sample time of context encoder evaluation $\tau \sim u[1/K, 1]$
  Sample time delta $\delta_t \sim u(0, 1/K]$ and set $t = \tau - \delta_t$
  Sample noisy state $\mathbf{z}_\tau \sim q(\mathbf{z}_\tau|\mathbf{x})$ and less noisy $\mathbf{z}_t \sim q(\mathbf{z}_t|\mathbf{z}_\tau, \mathbf{x})$
  Minimize $\mathcal{L} = -\frac{\mathrm{d}\lambda_t}{\mathrm{d}t} e^{\lambda_t} w(\lambda_t) \left\| \mathbf{x} - g_\theta\big(\mathbf{z}_t, t, E_\Theta(\mathbf{z}_\tau, \tau)\big) \right\|^2$ with respect to θ and Θ
end for

Algorithm 3 Dual-Rate MMD training algorithm with alternating optimization of auxiliary denoising model g_φ and student models E_H and g_η. Differences from standard MMD are highlighted in blue.

Require: Pretrained denoising model g_θ, student context encoder E_H and denoising model g_η, auxiliary denoising model g_φ, number of context encoder evaluations K and denoising steps k, loss weight $w(\lambda_t)$, dataset $\mathcal{D}$.

for $n = 0{:}N$ **do**
  Sample clean data $\mathbf{x} \sim \mathcal{D}$
  Sample time of context encoder evaluation $\tau \sim U(\{1/K, \ldots, (K-1)/K, 1\})$
  Sample step delta $\Delta_t \sim U(\{0, 1, \ldots, k/K - 1\})$ and set $t = \tau - \frac{\Delta_t}{k}$
  Sample time delta $\delta_s \sim u(0, 1/k]$ and set $s = t - \delta_s$
  Sample noisy state $\mathbf{z}_\tau \sim q(\mathbf{z}_\tau|\mathbf{x})$ and calculate $\mathbf{e}_\tau = E_H(\mathbf{z}_\tau, \tau)$
  Sample $\mathbf{z}_t \sim p_{H,\eta}(\mathbf{z}_t|\mathbf{z}_\tau)$ with a rollout of the denoising model g_η
  Sample $\mathbf{z}_s$ as $\tilde{\mathbf{x}} = g_\eta(\mathbf{z}_t, t, \mathbf{e}_\tau)$, $\mathbf{z}_s \sim q(\mathbf{z}_s|\mathbf{z}_t, \tilde{\mathbf{x}})$
  if n is even **then**
    Minimize $\mathcal{L} = -\frac{\mathrm{d}\lambda_s}{\mathrm{d}s} e^{\lambda_s} w(\lambda_s) \left(\|\tilde{\mathbf{x}} - g_\varphi(\mathbf{z}_s, s)\|^2 + \|g_\theta(\mathbf{z}_s, s) - g_\varphi(\mathbf{z}_s, s)\|^2 \right)$ w.r.t. φ
  else
    Minimize $\mathcal{L} = -\frac{\mathrm{d}\lambda_s}{\mathrm{d}s} e^{\lambda_s} w(\lambda_s) \tilde{\mathbf{x}}^\mathrm{T} \mathrm{sg}\left(g_\varphi(\mathbf{z}_s, s) - g_\theta(\mathbf{z}_s, s)\right)$ w.r.t. η and H
  end if
end for

and a sufficiently flexible denoising model alone would suffice. The context encoder is therefore a computational device rather than a modelling one: it extracts, once per interval, the information that a light denoising model cannot afford to recompute at every step, and packages it for reuse. The combined model thus spends its capacity and compute where they are needed while remaining as flexible as a larger denoising model alone.

3.2 Distillation

Although distillation and direct training can yield one-step models, achieving high sample quality often requires maintaining a multi-step process. However, even for these few-step models, evaluating a deep neural network at every step can be computationally prohibitive. Dual-Rate Diffusion addresses this bottleneck by amortizing the cost of heavy global computations across multiple lighter steps, replacing the heavy network with the light denoising model for the intermediate refinements.

A generic approach would be to take any model (distilled or not), sample generative trajectories, and perform a secondary distillation step by training a Dual-Rate model on these trajectories. While valid, this strategy complicates the training pipeline and may introduce compounding errors from the additional distillation stage. Instead, we demonstrate that Dual-Rate Diffusion can be trained directly as a few-step model within Moment Matching Distillation (MMD), without this extra stage.

To construct a Dual-Rate MMD model, we simply substitute the Dual-Rate architecture for the student model of MMD. We denote the student context encoder as E_H and the student denoising model as g_η, with H and η their parameters; the teacher g_θ and the auxiliary model g_φ remain standard single-network denoising models as in Section 2. For sampling, we employ the exact same procedure as in standard Dual-Rate Diffusion (see Algorithm 1).

The Dual-Rate MMD objective directly extends the standard MMD loss (Equation (3)) by sampling not two but three noisy states at $s < t \leq \tau$:

$$\mathcal{L} = \mathbb{E}_{q(\mathbf{z}_\tau) p_{H,\eta}(\tilde{\mathbf{x}}, \mathbf{z}_s, \mathbf{z}_t | \mathbf{z}_\tau)} \left[\tilde{\mathbf{x}}^{\mathrm{T}} \mathrm{sg} \left(\mathbb{E}_{p_{H,\eta}}[\tilde{\mathbf{x}} | \mathbf{z}_s] - \mathbb{E}_q[\mathbf{x} | \mathbf{z}_s] \right) \right]. \tag{6}$$

Similar to regular MMD, training requires a pretrained denoising teacher model g_θ and an auxiliary denoising model g_φ to approximate the intractable moments $\mathbb{E}_q[\mathbf{x}|\mathbf{z}_s]$ and $\mathbb{E}_{p_{H,\eta}}[\tilde{\mathbf{x}}|\mathbf{z}_s]$, respectively.

While the MMD objective targets marginal distribution matching, $p_{H,\eta}(\mathbf{z}_s) \approx q(\mathbf{z}_s)$, it does not guarantee that the student's generated trajectory distribution matches the ground truth forward process. Consequently, if the student is trained using intermediate states sampled from the true forward process $q(\mathbf{z}_t, \mathbf{z}_\tau)$, it will encounter out-of-distribution inputs during inference, leading to compounding errors.

To ensure that the student stays in-distribution during training, intermediate states $\mathbf{z}_t$ must be generated by the model itself, mirroring the inference dynamics. Specifically, rather than relying on the teacher's forward process, we sample $\mathbf{z}_t \sim p_{H,\eta}(\mathbf{z}_t|\mathbf{z}_\tau)$ by unrolling the student's iterative sampling procedure. Thus, whenever t and τ span multiple denoising steps, we perform a sequence of generative steps using the denoising model g_η to traverse from $\mathbf{z}_\tau$ to $\mathbf{z}_t$. We refer to this process as a rollout of the denoising model g_η. The complete training procedure is summarized in Algorithm 3. The same principle extends to $\mathbf{z}_\tau$: instead of sampling it from the forward process (line 6), the student can generate it as well, by a full rollout from pure noise. This aligns the input distribution of the student during training even more closely with inference, and since every state is then generated by the student, distillation no longer requires training data; all reported results use this variant (Appendix A.2).

Following the original MMD setup, we alternate between optimizing the auxiliary model g_φ and the student models E_H and g_η. The auxiliary model is trained to predict the samples $\tilde{\mathbf{x}}$ while being regularized to remain close to the teacher model g_θ. This regularization helps stabilize the training process and should not bias the optimum point.

In summary, Dual-Rate MMD differs from the original MMD formulation in three key aspects. First, it parameterizes the student model with a Dual-Rate architecture. Second, intermediate states $\mathbf{z}_t$ are generated via a rollout of the student's denoising model to prevent distribution shift. Finally, whereas the original MMD samples the time steps t from a continuous uniform distribution, we sample both t and τ from discrete uniform distributions defined over the inference grids of denoising steps and context encoder evaluations. This follows the same in-distribution principle as the rollout: at inference, the student is evaluated only on the fixed grid of k denoising steps and K context encoder evaluations (Algorithm 1), and sampling t and τ continuously exposes it to time pairs that never occur at inference, which in our experiments destabilized the early phase of distillation. The time s, at which only the teacher and the auxiliary model are evaluated, remains continuous.

Other distillation methods. The construction above is not specific to moment matching: the three modifications are defined for any objective evaluated on sampled states of a stochastic few-step trajectory, such as other distribution matching methods (Yin et al., 2024b); we test only MMD. Objectives built on a deterministic teacher trajectory, such as progressive distillation (Salimans and Ho, 2022) and consistency models (Song et al., 2023), or on a differential characterization of the flow map (Boffi et al., 2025), do not satisfy the Markovian requirement of Section 3.1; models trained with them can be converted into Dual-Rate models by the trajectory distillation described above.

4 Experiments

We evaluate Dual-Rate Diffusion on the standard class-conditional ImageNet (Deng et al., 2009) benchmarks at 64×64 and 128×128 in two settings: (1) training a standard diffusion model to demonstrate the efficiency gains of the interleaving architecture, and (2) applying Dual-Rate Diffusion to Moment Matching Distillation (MMD) to show its compatibility with few-step distillation methods.

Metrics. In all experiments, we report the Fréchet Inception Distance (FID) (Heusel et al., 2017) as the primary metric for sample quality, computed on 50k generated samples against the training set. We measure the computational cost of sampling in FLOPs, which are hardware-independent, and also report the number of function evaluations (NFE).

Baselines. We compare against pixel-space diffusion models on the same benchmarks. VDM++ (Kingma and Gao, 2023) and EDM2 (Karras et al., 2024) are standard diffusion models with many-

Table 1: Ablation of different training techniques for Dual-Rate Diffusion on ImageNet 64×64 with $K = 16$ context encoder evaluations and $k = 512$ denoising steps.

Multi-level	Data Aug	Feature Drop	FID
✗	✗	✗	1.30
✓	✗	✗	1.24
✗	✓	✗	1.26
✗	✗	✓	1.14
✓	✓	✗	1.20
✓	✗	✓	1.13
✓	✓	✓	**1.12**

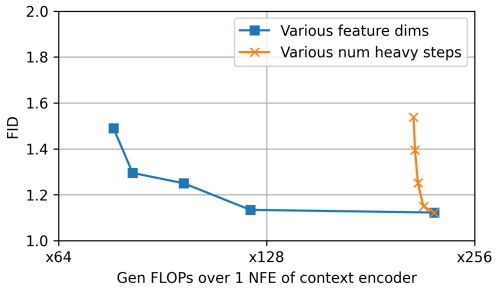

Figure 4: Ablation of the number of context encoder evaluations K versus the feature dimension of the denoising model (ImageNet 64×64).

step samplers, the latter using a deterministic second-order Heun solver; SiD2 (Hoogeboom et al., 2025) is the model whose training recipe our base model follows. In the few-step setting, CTM (Kim et al., 2024) and sCD (Lu and Song, 2025) are consistency-type models trained along the deterministic trajectory of a teacher, DMD2 (Yin et al., 2024a) is a distribution matching method, and Moment Matching Distillation (Salimans et al., 2024) is the distribution matching method of Section 2 on which our distilled models build.

Experimental setup. For the regular diffusion experiments, we follow the setup from Hoogeboom et al. (2025), including sigmoid weighting of the objective and an optimized amount of sampling noise, following Salimans and Ho (2022). For the distillation experiments, we follow the original MMD setup from Salimans et al. (2024), using a pretrained standard diffusion model as the teacher. We parameterize the context encoder E_Θ with the UViT architecture of the base model, and the denoising model g_θ with the same architecture with fewer blocks and channels, approximately two times lighter in FLOPs; the exact architectures are given in Table 3, further details in Appendix B, and sample outputs in Appendix D.

4.1 Standard Diffusion

We first validate Dual-Rate Diffusion in the standard diffusion setting.

Three training techniques improve Dual-Rate Diffusion. First, we condition the denoising model on features from every level of the context encoder hierarchy (Section 3.1) rather than only the last level. Second, we use extra data augmentation (random translation) during training. Third, during training we randomly replace the features $\mathbf{e}_\tau$ with a zero vector with probability $p = 0.5$. This feature dropout mirrors the training procedure of self-conditioning (Chen et al., 2023); our hypothesis is that, because the denoising model must also work without them, the context features it learns to use complement its own computation rather than replace it, which makes them more robust. Table 1 ablates the three techniques: all of them contribute to the performance improvement, and feature dropout has the largest effect on the FID score.

We also ablate the number of context encoder evaluations and the capacity allocation within the denoising model. Figure 4 compares reducing K with narrowing the denoising model (512 down to 128 channels per level): reducing K gives a worse performance-compute trade-off, and the FID degrades sharply below 8 evaluations. Figure 3 ablates depth against width: removing blocks hurts markedly, whereas narrowing offers a much better trade-off. Appendix A.1 gives the exact ranges of these ablations and a comparison with fully convolutional denoising models, which performed considerably worse.

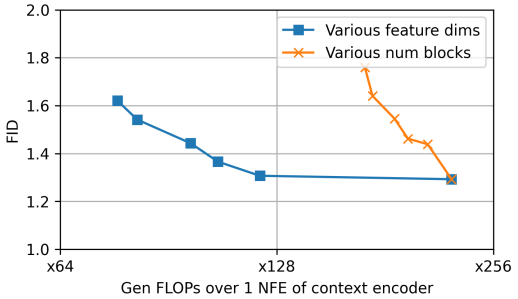

Figure 3: Ablation of the denoising model depth versus its feature dimension on ImageNet 64×64.

Table 2: Results on ImageNet 64×64 and 128×128 for Dual-Rate Diffusion and Dual-Rate MMD. We report the computational cost of sampling in terms of teraflops (TFLOPs) and number of function evaluations (NFE). For the Dual-Rate experiments NFE refers to the number of denoising steps and K refers to the number of context encoder evaluations. The Moment Matching results are those reported by Salimans et al. (2024) and were obtained with their teacher model (see Section 4.2).

Methods	NFE	**ImageNet** 64×64 TFLOPs	FID↓	**ImageNet** 128×128 TFLOPs	FID↓
BASE MODELS					
VDM++ (Kingma and Gao, 2023)	1024	429.25	1.43	544.26	1.75
EDM2-XL (Heun) (Karras et al., 2024)	63	25.58	1.33	-	-
SiD2 (Hoogeboom et al., 2025)	512	-	-	70.37	1.26
Our base model (guidance $w = 1$)	512	55.53	1.25	70.37	1.50
Our base model (guidance $w = 0$)	512	55.53	2.38	70.37	2.66
Dual-Rate Diffusion ($K = 16$)					
	128	3.53	1.52	8.05	3.21
	256	7.44	**1.12**	13.98	1.89
	512	13.14	**1.12**	28.23	**1.48**
DISTILLATION					
CTM (Kim et al., 2024)	1	-	1.73	-	-
sCD-XXL (Lu and Song, 2025)	1	-	2.04	-	-
	2	-	1.48	-	-
Moment Matching (Salimans et al., 2024)	2	0.22	3.86	0.27	3.14
	4	0.43	1.50	0.55	1.72
	8	0.87	1.24	1.10	1.49
DMD2 (Yin et al., 2024a)	1	-	1.28	-	-
Dual-Rate MMD					
$K = 1$	2	0.20	1.56	0.24	2.60
$K = 2$	4	0.39	1.44	0.48	2.58
$K = 4$	8	0.79	1.17	0.92	1.90

We summarize the evaluation results and compare with other pixel-space generative baselines in Table 2. With a significantly lower sampling budget, Dual-Rate Diffusion achieves FID scores comparable to the base model at both resolutions. On ImageNet 64×64, it achieves a better FID score than the base model at the same number of denoising steps with a 4 times smaller budget, and reaches the same quality with half the denoising steps at more than 7 times lower cost.

Capacity and compute. The Dual-Rate model consists of the context encoder, which has the architecture of the base model, and the denoising model on top of it, so it has more parameters than the base model (Table 3) but spends fewer FLOPs per sample. We attribute the improvement over the base model on ImageNet 64×64 to this combination: a higher effective capacity, used more efficiently during sampling. A single network with the per-step cost of our denoising model does not give the same trade-off. Figure 5 compares the base model, this lighter single network, and Dual-Rate Diffusion over a range of denoising steps: the lighter network is worse than Dual-Rate Diffusion at every budget, and falls behind the base model once the budget allows the base model enough steps.

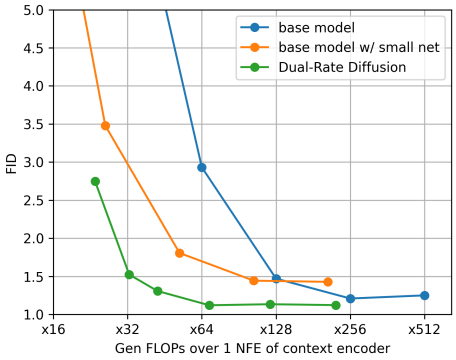

Figure 5: FID versus sampling cost on ImageNet 64×64 for the base model, a lighter single network, and Dual-Rate Diffusion. Cost in multiples of one context encoder evaluation (108.45 GFLOPs).

Training cost. The efficiency gains of Dual-Rate Diffusion are realized at inference; training is more expensive than for the base model, because every training iteration evaluates both the context encoder and the

denoising model (Algorithm 2). Each network is evaluated once per iteration with the same forward-backward structure as the base model, so the overhead per iteration is the ratio of the summed per-network FLOPs to those of the base model: about 1.4 times on ImageNet 64×64 and 1.5 times on ImageNet 128×128 (Table 4). Both models are trained for the same number of iterations with the same batch size, so the same factor applies to the total training compute. Table 4 also compares the two models at matched training compute, by evaluating Dual-Rate Diffusion at the iteration at which it has consumed the total training compute of the fully trained base model. At that point it surpasses the base model on ImageNet 64×64 and is slightly behind it on ImageNet 128×128, while sampling 2–4 times cheaper.

Deterministic sampling. Dual-Rate Diffusion is trained for the Markovian ancestral sampler, and a deterministic sampler such as DDIM (Song et al., 2021a) changes the joint distribution of $\mathbf{z}_t$ and $\mathbf{z}_\tau$ that the denoising model is evaluated on (Section 3.1). To measure its effect, we sampled the trained models with an untuned DDIM sampler on the same grid of $K = 16$ context encoder evaluations and $k = 512$ denoising steps. The mismatch costs 0.2–0.3 FID at both resolutions (Table 5); deterministic sampling therefore remains usable, and the ancestral sampler is our default.

4.2 Distillation

Next, we evaluate the compatibility of Dual-Rate Diffusion with distillation, specifically Moment Matching Distillation (MMD). We chose MMD as a state-of-the-art representative of distribution matching methods whose iterative refinement structure matches Dual-Rate Diffusion most closely: its student samples a stochastic multi-step trajectory, which is exactly the setting in which the Dual-Rate architecture amortizes the context encoder over the intermediate steps. For the student context encoder E_H and the student denoising model g_η, we use the same parameterization as in the standard diffusion experiments. The teacher model g_θ is our base model, which also initializes the auxiliary model g_φ.

MMD is sensitive to the initialization of the student model, and training a student from scratch leads to instabilities in the early stages. Unlike in regular MMD, the Dual-Rate student cannot be initialized with the teacher model directly, so we pretrain it briefly as a Dual-Rate diffusion model with the context encoder initialized from the teacher (Appendix A.2).

Following the original MMD setup, we do not use classifier-free guidance for sampling or in the auxiliary model during training. MMD allows guidance in the teacher model during training, but in our experiments it does not improve the student and in some cases destabilizes training, for standard MMD and Dual-Rate MMD alike (Figure 6 shows the former). We therefore train all distilled models with an unguided teacher, which matters for the comparison with published results below.

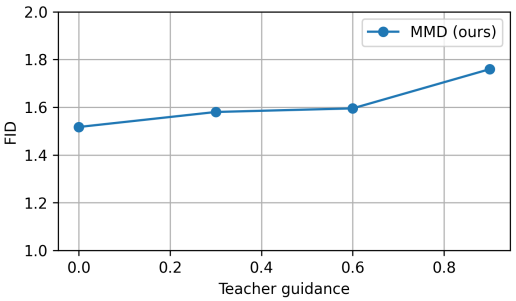

Figure 6: Effect of classifier-free guidance in the teacher model on the FID of a standard MMD student trained in our setup on ImageNet 64×64.

On ImageNet 64×64, Dual-Rate MMD outperforms standard MMD at every step count, reaching a better FID score with a lower computational budget. On ImageNet 128×128, Dual-Rate MMD is better at 2 steps but behind the published MMD results at 4 and 8 steps. With only 2–8 denoising steps, there are few intermediate steps over which the cost of the context encoder can be amortized. In the distilled setting the Dual-Rate architecture therefore gains less than in the standard setting, and the gain takes the form of better quality at a comparable budget rather than a lower budget at matched quality.

The published MMD results and ours were also obtained with different teachers (Table 6). The teacher of Salimans et al. (2024) is similar in size to our base model but is used with slight guidance, which makes it roughly one FID point stronger than the unguided base model from which our students are distilled. With 8 steps, our students nonetheless surpass this stronger teacher at both resolutions, as Salimans et al. (2024) also observe for standard MMD students relative to theirs. A direct comparison of the two distillation methods at 128×128 would require a shared teacher.

5 Related Work

Accelerating Sampling. A significant body of research reduces the number of sampling steps in diffusion models, with advanced ODE solvers (Dockhorn et al., 2022; Karras et al., 2022; Lu et al., 2022), distillation methods that compress the sampling process into fewer steps (Salimans and Ho, 2022; Song et al., 2023; Yin et al., 2024b), and approaches that learn a direct mapping from noise to data (Deng et al., 2026; Geng et al., 2025). Our work is orthogonal to these strategies: rather than reducing the number of steps, we lower the computational cost per step, and the two can be combined, as Section 4.2 demonstrates for Moment Matching Distillation (Salimans et al., 2024).

Efficient Architectures and Caching. The idea of reusing computation across time steps has been explored in several forms. Self-conditioning (Chen et al., 2023) feeds the model's estimate of the clean data from the previous step back as an input to the next step, which improves sample quality without adding parameters; the same heavy network is still evaluated at every step, so the sampling cost is unchanged. Cascaded Diffusion Models (Ho et al., 2022a) decompose generation into a sequence of models operating at increasing resolutions. Most closely related to our work is Clockwork Diffusion (Habibian et al., 2024), which updates different layers of a single U-Net at different rates by reusing low-resolution feature maps from preceding steps, motivated by the slow evolution of semantic content. DeepCache (Ma et al., 2024) similarly reuses high-level U-Net features across steps, and a growing line of work applies caching to Diffusion Transformers, caching tokens by importance scores (Lou et al., 2024; Zou et al., 2025) or at random (Zou et al., 2026), exploiting layer-wise similarity (Liu et al., 2025b), or adapting the caching schedule (Kahatapitiya et al., 2025). The caching methods are training-free and accelerate a fixed, already trained network by reusing its internal activations; Clockwork Diffusion trains a light replacement for part of such a network. Dual-Rate Diffusion instead decouples the generative process into two separate networks with independently chosen capacities and trains them jointly. These methods were developed for latent-space U-Nets and Diffusion Transformers rather than for pixel-space UViT models, so we do not compare against them directly; the two approaches are complementary, since caching could be applied on top of a trained Dual-Rate model, to both of its networks.

6 Limitations and Future Work

Dual-Rate Diffusion increases the training cost per iteration by a factor of 1.4–1.5 in our experiments, since both networks are evaluated at every training step (Section 4.1). An alternating training scheme for the two networks could reduce this overhead, which we leave for future work.

Dual-Rate Diffusion conditions each denoising step on two states of the noisy sample, $\mathbf{z}_t$ and $\mathbf{z}_\tau$, whose joint distribution must match between training and inference. This restricts training and sampling to stochastic Markovian processes: deterministic samplers degrade the FID (Section 4.1), and distillation requires rollouts of the student during training (Section 3.2), which add to its cost.

Several directions remain open. The context encoder is evaluated on a uniform grid in our experiments, but this is a choice of the sampler rather than of the model: Algorithm 2 samples τ continuously and only requires an encoder evaluation to be at most $1/K$ ahead of the denoising steps that use it, so a trained model admits non-uniform and adaptive schedules without retraining. In the distilled setting, where times are sampled on the inference grid, such schedules would have to be included in training. Other directions include alternative architectures and conditioning mechanisms for the two networks, applying the construction to latent-space and transformer-based models, and extending it to modalities with pronounced temporal redundancy, such as video and audio, where we expect the gains to be larger.

7 Conclusion

We introduced Dual-Rate Diffusion, which accelerates diffusion models by interleaving a heavy context encoder, evaluated at sparse intervals, with a light denoising model that reuses its features at every step. In the standard diffusion setting, it matches or improves on the base model at 2–4 times lower sampling cost. Combined with Moment Matching Distillation, it yields few-step students that sample more cheaply than standard MMD students, improve on them on ImageNet 64×64, and at 8 steps surpass even the stronger teacher of Salimans et al. (2024).

Acknowledgments and Disclosure of Funding

This work was carried out at Google DeepMind, which also provided the computational resources. The authors received no third-party funding for this work and declare no competing interests beyond their listed affiliations.

References

N. M. Boffi, M. S. Albergo, and E. Vanden-Eijnden. How to build a consistency model: Learning flow maps via self-distillation. In *Advances in Neural Information Processing Systems*, 2025.

T. Chen, R. Zhang, and G. Hinton. Analog bits: Generating discrete data using diffusion models with self-conditioning. In *International Conference on Learning Representations*, 2023.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

M. Deng, H. Li, T. Li, Y. Du, and K. He. Generative modeling via drifting. *Advances in Neural Information Processing Systems*, 2026.

P. Dhariwal and A. Nichol. Diffusion models beat GANs on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021.

S. Dieleman. Diffusion is spectral autoregression, 2024. URL `https://sander.ai/2024/09/02/spectral-autoregression.html`.

T. Dockhorn, A. Vahdat, and K. Kreis. GENIE: Higher-order denoising diffusion solvers. *Advances in Neural Information Processing Systems*, 35:30150–30166, 2022.

Z. Geng, M. Deng, X. Bai, J. Z. Kolter, and K. He. Mean flows for one-step generative modeling. *Advances in Neural Information Processing Systems*, 2025.

A. Habibian, A. Ghodrati, N. Fathima, G. Sautiere, R. Garrepalli, F. Porikli, and J. Petersen. Clockwork diffusion: Efficient generation with model-step distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8352–8361, 2024.

M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. *Advances in Neural Information Processing Systems*, 30, 2017.

J. Ho and T. Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

J. Ho, C. Saharia, W. Chan, D. J. Fleet, M. Norouzi, and T. Salimans. Cascaded diffusion models for high fidelity image generation. *Journal of Machine Learning Research*, 23(47):1–33, 2022a.

J. Ho, T. Salimans, A. Gritsenko, W. Chan, M. Norouzi, and D. J. Fleet. Video diffusion models. *Advances in Neural Information Processing Systems*, 35:8633–8646, 2022b.

E. Hoogeboom, T. Mensink, J. Heek, K. Lamerigts, R. Gao, and T. Salimans. Simpler diffusion: 1.5 FID on ImageNet512 with pixel-space diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18062–18071, 2025.

K. Kahatapitiya, H. Liu, S. He, D. Liu, M. Jia, C. Zhang, M. S. Ryoo, and T. Xie. Adaptive caching for faster video generation with diffusion transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15240–15252, 2025.

T. Karras, M. Aittala, T. Aila, and S. Laine. Elucidating the design space of diffusion-based generative models. *Advances in Neural Information Processing Systems*, 35:26565–26577, 2022.

T. Karras, M. Aittala, J. Lehtinen, J. Hellsten, T. Aila, and S. Laine. Analyzing and improving the training dynamics of diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24174–24184, 2024.

D. Kim, C.-H. Lai, W.-H. Liao, N. Murata, Y. Takida, T. Uesaka, Y. He, Y. Mitsufuji, and S. Ermon. Consistency trajectory models: Learning probability flow ODE trajectory of diffusion. In *International Conference on Learning Representations*, 2024.

D. Kingma and R. Gao. Understanding diffusion objectives as the ELBO with simple data augmentation. *Advances in Neural Information Processing Systems*, 36:65484–65516, 2023.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Z. Kong, W. Ping, J. Huang, K. Zhao, and B. Catanzaro. DiffWave: A versatile diffusion model for audio synthesis. In *International Conference on Learning Representations*, 2021.

H. Liu, W. Zhang, J. Xie, F. Faccio, M. Xu, T. Xiang, M. Z. Shou, J.-M. Perez-Rua, and J. Schmidhuber. Faster diffusion via temporal attention decomposition. *Transactions on Machine Learning Research*, 2025a.

J. Liu, J. Geddes, Z. Guo, H. Jiang, and M. K. Nandwana. SmoothCache: A universal inference acceleration technique for diffusion transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 3254–3263, 2025b.

J. Lou, W. Luo, Y. Liu, B. Li, X. Ding, W. Hu, Y. Li, and C. Ma. Token caching for diffusion transformer acceleration. *arXiv preprint arXiv:2409.18523*, 2024.

C. Lu and Y. Song. Simplifying, stabilizing and scaling continuous-time consistency models. In *International Conference on Learning Representations*, 2025.

C. Lu, Y. Zhou, F. Bao, J. Chen, C. Li, and J. Zhu. DPM-Solver: A fast ODE solver for diffusion probabilistic model sampling in around 10 steps. *Advances in Neural Information Processing Systems*, 35:5775–5787, 2022.

X. Ma, G. Fang, and X. Wang. DeepCache: Accelerating diffusion models for free. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15762–15772, 2024.

A. Q. Nichol and P. Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171, 2021.

E. Perez, F. Strub, H. de Vries, V. Dumoulin, and A. Courville. FiLM: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

S. Rissanen, M. Heinonen, and A. Solin. Generative modelling with inverse heat dissipation. In *International Conference on Learning Representations*, 2023.

T. Salimans and J. Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022.

T. Salimans, T. Mensink, J. Heek, and E. Hoogeboom. Multistep distillation of diffusion models via moment matching. *Advances in Neural Information Processing Systems*, 37:36046–36070, 2024.

J. Song, C. Meng, and S. Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021a.

Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021b.

Y. Song, P. Dhariwal, M. Chen, and I. Sutskever. Consistency models. In *International Conference on Machine Learning*, 2023.

T. Yin, M. Gharbi, T. Park, R. Zhang, E. Shechtman, F. Durand, and W. T. Freeman. Improved distribution matching distillation for fast image synthesis. *Advances in Neural Information Processing Systems*, 37:47455–47487, 2024a.

T. Yin, M. Gharbi, R. Zhang, E. Shechtman, F. Durand, W. T. Freeman, and T. Park. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024b.

C. Zou, X. Liu, T. Liu, S. Huang, and L. Zhang. Accelerating diffusion transformers with token-wise feature caching. In *International Conference on Learning Representations*, 2025.

C. Zou, S. Zheng, E. Zhang, R. Guo, H. Xu, Z. Shi, C. He, X. Hu, and L. Zhang. Rethinking token-wise feature caching: Accelerating diffusion transformers with dual feature caching. *IEEE Transactions on Image Processing*, 35:6211–6220, 2026.

A Additional results

In this section, we present additional results to motivate our design choices and demonstrate the contributions of different components of Dual-Rate Diffusion.

A.1 Diffusion

For the context encoder, we use a UViT architecture based on Hoogeboom et al. (2025). We believe this is a reasonable choice, as this architecture is designed to capture long-range dependencies effectively, which is crucial for extracting global structural information from the noisy input.

For the denoising model, we use the same UViT architecture as for the context encoder but with a reduced number of blocks and channels. As the initial configuration, we use the denoising model described in Table 3. The depth and width ablation of Figure 3 explores various combinations of channels across the three spatial resolution levels, ranging from $[128, 128, 128]$ to $[256, 256, 512]$, and numbers of transformer blocks at the two transformer levels, ranging from $[1 + 1, 1]$ to $[3 + 3, 8]$.

Table 3: Experimental setup.

	ImageNet 64×64		**ImageNet** 128×128	
Context encoder architecture				
Channels	$[256, 512, 1024]$		$[128, 256, 512, 1024]$	
Patch size	1×1		2×2	
Block types	[Res, Transf, Transf]		[Res, Res, Transf, Transf]	
Number of blocks	$[3 + 3, 3 + 3, 16]$		$[3 + 3, 3 + 3, 3 + 3, 16]$	
Parameters	394M		397M	
GFLOPs per inference	108.45		137.44	
Denoising model architecture				
Channels	$[256, 256, 512]$		$[128, 256, 256, 512]$	
Patch size	1×1		2×2	
Block types	[Res, Transf, Transf]		[Res, Res, Transf, Transf]	
Number of blocks	$[3 + 3, 3 + 3, 8]$		$[3 + 3, 3 + 3, 3 + 3, 8]$	
Parameters	88M		94M	
GFLOPs per inference	44.02		73.01	
Hyperparameters				
Setup	Diffusion	Distillation	Diffusion	Distillation
Log-SNR boundaries	$(-12, 12)$	$(-12, 12)$	$(-15, 15)$	$(-15, 15)$
Sigmoid loss bias	1	1	0	0
Classifier-free guidance	1	0	1	0
Guidance interval	$(1.5, 5)$	N/A	$(0, 5)$	N/A
Number of TPUs	16	64	16	64
Batch size	1024	1024	1024	512
Training steps	10^6	1.5×10^5	10^6	1.5×10^5
Network dropout	0.1	0.0	0.1	0.0
Feature dropout	0.5	0.0	0.5	0.0
EMA decay	0.9999	0.999	0.9999	0.999
Learning rate	10^{-4}	10^{-6}	10^{-4}	10^{-6}
Adam betas	$[0.9, 0.99]$	$[0.0, 0.99]$	$[0.9, 0.99]$	$[0.0, 0.99]$
Warmup steps	0	10^3	0	10^3

We also experimented with lighter architectures, such as fully convolutional networks, for the denoising model. While it is possible to reduce the computational cost of model inference by up to 20 times compared to the context encoder, this leads to a significant drop in performance. In our experiments, convolutional denoising models do not reach FID scores below 3 on ImageNet 64×64, which is significantly worse than UViT-based architectures. We note that although convolutional blocks are more parameter-efficient than transformer blocks, they remain computationally expensive. A single 3×3 convolutional layer requires $9D^2S^2$ FLOPs, where D is the feature dimension and

S is the spatial size. Overall, in our experiments, fully convolutional models do not yield a better performance-compute trade-off than UViT-based architectures.

A.2 Distillation

Initialization of the student. The Dual-Rate student cannot be initialized with the teacher model directly, because the two have different architectures. We therefore pretrain the student with the standard Dual-Rate objective (Algorithm 2), with the context encoder E_H initialized from the teacher and kept frozen, so that only the student denoising model g_η is trained. The teacher is a suboptimal feature extractor for the student denoising model, so the pretrained student is weaker than one pretrained from scratch, but this pretraining is considerably cheaper to train, and in our experiments it initializes the distillation equally well.

Algorithm 4 describes the training procedure for Dual-Rate MMD with a full rollout of the student model. It is identical to Algorithm 3, except that $\mathbf{z}_\tau$ is generated by a rollout of the student models g_η and E_H starting from pure noise $\mathbf{z}_1 \sim \mathcal{N}(0, \mathbf{I})$ instead of being sampled from the forward process. Distillation with full rollout therefore does not require data samples.

Algorithm 4 Dual-Rate MMD training algorithm with full rollout of student and alternating optimization of auxiliary denoising model g_φ and student models E_H and g_η.

Require: Pretrained denoising model g_θ, student context encoder E_H and denoising model g_η, auxiliary denoising model g_φ, number of context encoder evaluations K and denoising steps k, loss weight $w(\lambda_t)$.

for $n = 0{:}N$ **do**
  Sample time of context encoder evaluation $\tau \sim U(\{1/K, \dots, (K-1)/K, 1\})$
  Sample step delta $\Delta_t \sim U(\{0, 1, \dots, k/K - 1\})$ and set $t = \tau - \frac{\Delta_t}{k}$
  Sample time delta $\delta_s \sim u(0, 1/k]$ and set $s = t - \delta_s$
  Sample $\mathbf{z}_\tau \sim p_{H,\eta}(\mathbf{z}_\tau)$ with a full rollout of student models g_η and E_H
  Detach $\mathbf{z}_\tau = \mathrm{sg}\,(\mathbf{z}_\tau)$ and calculate $\mathbf{e}_\tau = E_H(\mathbf{z}_\tau, \tau)$
  Sample $\mathbf{z}_t \sim p_{H,\eta}(\mathbf{z}_t|\mathbf{z}_\tau)$ with a rollout of the denoising model g_η
  Sample $\mathbf{z}_s$ as $\tilde{\mathbf{x}} = g_\eta(\mathbf{z}_t, t, \mathbf{e}_\tau), \mathbf{z}_s \sim q(\mathbf{z}_s|\mathbf{z}_t, \tilde{\mathbf{x}})$
  if n is even **then**
    Minimize $\mathcal{L} = -\frac{\mathrm{d}\lambda_s}{\mathrm{d}s} e^{\lambda_s} w(\lambda_s) \left(\|\tilde{\mathbf{x}} - g_\varphi(\mathbf{z}_s, s)\|^2 + \|g_\theta(\mathbf{z}_s, s) - g_\varphi(\mathbf{z}_s, s)\|^2 \right)$ w.r.t. φ
  else
    Minimize $\mathcal{L} = -\frac{\mathrm{d}\lambda_s}{\mathrm{d}s} e^{\lambda_s} w(\lambda_s) \tilde{\mathbf{x}}^{\mathrm{T}} \mathrm{sg}\,(g_\varphi(\mathbf{z}_s, s) - g_\theta(\mathbf{z}_s, s))$ w.r.t. η and H
  end if
end for

For the setup with $K = 4$ context encoder evaluations and $k = 8$ denoising steps, standard training of Dual-Rate MMD reaches an FID score of 1.61 on ImageNet 64×64, while training with full rollout reaches an FID score of 1.17 with the same setup.

B Implementation Details

We provide the implementation details for our experiments on ImageNet 64×64 and 128×128 in Table 3. The base model has the architecture of the context encoder, so the Dual-Rate models carry the parameters of both networks in total. For both standard diffusion and distillation, we closely follow the setups from Hoogeboom et al. (2025) and Salimans et al. (2024), respectively. We always randomly flip images horizontally with a probability of 0.5. When using extra data augmentation, we also apply random translation with a probability of 0.4. We use a cosine noise schedule (Nichol and Dhariwal, 2021) for all experiments.

For sampling, we use the standard ancestral sampling algorithm adapted for Dual-Rate Diffusion (see Algorithm 1). During sampling, we also apply clipping of the $\mathbf{x}$ predictions to the range $[-1, 1]$. For the experiments with standard diffusion, we use classifier-free guidance (Ho and Salimans, 2022). To support this, during training, we drop the class conditioning with a probability of 0.1. In Dual-Rate Diffusion, to obtain conditional and unconditional predictions, we evaluate both the context encoder and the denoising model twice: once with class conditioning and once without. We also use an optimized amount of sampling noise, following Salimans and Ho (2022), for standard diffusion, but not for distillation.

We use the same UViT architecture for both the context encoder and the denoising model. To condition the denoising model on two time steps (the current time t and the context encoder evaluation time τ), we use a standard FiLM conditioning mechanism (Perez et al., 2018). To construct a two-time embedding, we calculate the Fourier features for both time steps, project them linearly, and add them together. For the transformer blocks, we use a head dimension of 128 and, for standard diffusion, dropout. We use the v-parameterization for the denoising model.

We use the Adam optimizer (Kingma and Ba, 2015) with $\epsilon = 10^{-12}$ and no weight decay. For distillation, we employ a linear learning rate warmup; we do not use learning rate annealing. We use gradient clipping with a maximum norm of 1. All our experiments are conducted on TPU v6e accelerators.

FID scores are computed on 50k generated samples against the training set (Section 4). We report the lowest FID score achieved during training, measuring every 20k iterations for standard diffusion experiments and every 5k iterations for distillation experiments.

C Additional Analysis

This section collects the measurements behind the discussions of training cost, deterministic sampling, and teacher quality in Section 4.

C.1 Training cost

Table 4: Training cost of Dual-Rate Diffusion relative to the base model. A training iteration evaluates both networks, so the overhead per iteration is the ratio of the summed per-network FLOPs to those of the base model. Both models are trained for 10^6 iterations with the same batch size. The matched-compute row reports Dual-Rate Diffusion at the iteration at which its total training compute equals that of the fully trained base model.

	ImageNet 64×64	**ImageNet** 128×128
GFLOPs per forward pass, context encoder (base model)	108.45	137.44
GFLOPs per forward pass, denoising model	44.02	73.01
Training cost per iteration, relative to base model	$1.41\times$	$1.53\times$
Iterations at matched training compute	$\approx 7.1 \times 10^5$	$\approx 6.5 \times 10^5$
FID at matched training compute	≈ 1.19	≈ 1.57
FID after 10^6 iterations	1.12	1.48
FID of the base model after 10^6 iterations	1.25	1.50

C.2 Deterministic sampling

Table 5: FID of Dual-Rate Diffusion ($K = 16$, $k = 512$) with the ancestral sampler used throughout the paper and with a DDIM sampler on the same grid, without sampler-specific tuning.

Sampler	**ImageNet** 64×64	**ImageNet** 128×128
Ancestral (Markovian)	1.12	1.48
DDIM (deterministic)	1.34	1.80

C.3 Teachers in the distillation comparison

Table 6: FID of the teacher models behind the Moment Matching results in Table 2. The published MMD results of Salimans et al. (2024) use their teacher with guidance $w = 0.1$; our distilled models use our base model without guidance, since guidance in the teacher does not improve the student (Figure 6). The last row gives the FID of the respective 8-step Dual-Rate MMD student.

	ImageNet 64×64	**ImageNet** 128×128
Teacher of Salimans et al. (2024) ($w = 0.1$)	1.42	1.76
Our base model ($w = 1$)	1.25	1.50
Our base model ($w = 0$), teacher of Dual-Rate MMD	2.38	2.66
Dual-Rate MMD student ($K = 4$, 8 steps)	1.17	1.90

D Image Samples



Figure 7: Random samples generated by Dual-Rate Diffusion on ImageNet 64×64 with $K = 16$ context encoder evaluations and $k = 512$ denoising steps.



Figure 8: Random samples generated by Dual-Rate MMD on ImageNet 64×64 with $K = 4$ context encoder evaluations and $k = 8$ denoising steps.



Figure 9: Random samples generated by Dual-Rate Diffusion on ImageNet 128×128 with $K = 16$ context encoder evaluations and $k = 512$ denoising steps.

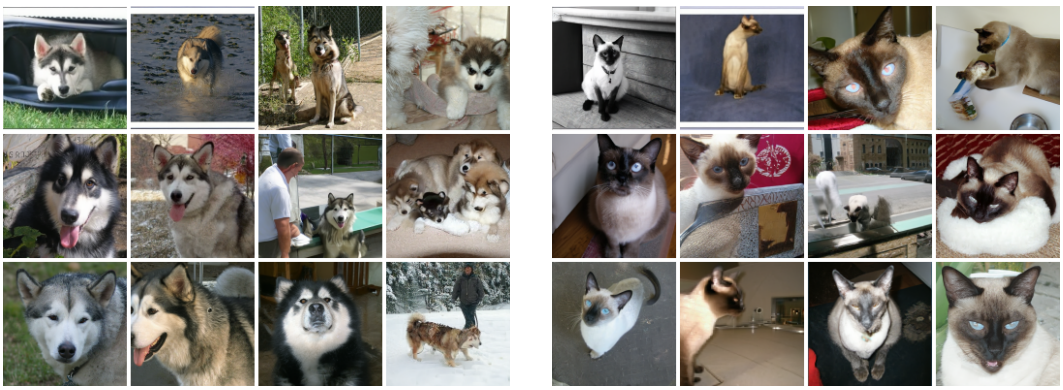

Figure 10: Random samples generated by Dual-Rate MMD on ImageNet 128×128 with $K = 4$ context encoder evaluations and $k = 8$ denoising steps.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims in the abstract and intro reflect the performance-computation trade-off and model architecture (Dual-Rate Diffusion and MMD compatibility).

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 6 explicitly outlines limitations, such as increased training step cost and challenges with non-Markovian inference steps matching the training distribution.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The method relies mostly on empirical formulation over existing proofs (diffusion ELBO, MMD distillation), not introducing novel theoretical findings.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The architectures, parameter counts, training procedures and hyperparameters are specified in Section 4 and Appendix B.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: Code is not released with this paper.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Training and evaluation details, hyperparameters, optimizers and batch sizes are given in Appendix B.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Running multiple ImageNet generation models at scale is computationally expensive, hence single runs are reported without error bars, as in prior work.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Compute resources per experiment run are documented in Appendix B, enumerating TPUs configuration and FLOPs count.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We follow standard generative model procedures and benchmark experiments that align thoroughly with NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: The work focuses on foundational algorithmic acceleration for inference within neural networks, which does not have direct societal impacts independent of the baseline generative models.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.

- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The work uses standard benchmarking datasets (ImageNet) which pose low risk for novel malicious misuses compared to public assets.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Usage of standard dataset ImageNet and pre-existing works are properly credited within references.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The research does not present or release new datasets or equivalent distinct assets requiring documentation.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper conducts methodological development strictly without any human subjects or crowdsourced assessments.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The research doesn't involve experiments with human subjects that require Institutional Review Board evaluation.

    Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are not conceptually used or declared as integral parts for methodology scaling or experimental components of this work.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.