MIXTURE-OF-DEPTH RECTIFIED FLOW FOR HETEROGENEOUS MULTIMODALITY

Anonymous authors
Paper under double-blind review

ABSTRACT

Hierarchical Rectified Flow models data generation through a hierarchy of coupled ordinary differential equations (ODEs), where each depth solves a flow-matching problem over progressively transformed target directions. We show that the ambiguity of the conditional target-direction distribution varies across both time and hierarchy depth, and that increasing depth progressively concentrates this distribution, making the corresponding regression problem simpler. This heterogeneous multimodality implies that a fixed hierarchy depth can allocate computation inefficiently: some space-time regions require deeper hierarchical modeling, while others can be sufficiently modeled at shallow depth. To address this, we propose *Mixture-of-Depth Rectified Flow* (MoD-RF), which adaptively selects the hierarchy depth at each outer integration step through a sequential stopping mechanism trained from depth-dependent velocity reconstruction quality, without ground-truth depth labels. Under a fixed number of function evaluations (NFEs), MoD-RF allocates deeper computation only where needed while preserving cheaper shallow updates elsewhere. Experiments on synthetic 1D and 2D distributions, MNIST, CIFAR-10, and ImageNet-32 demonstrate improved quality-compute trade-offs over standard rectified flow and fixed-depth hierarchical inference, with particularly strong gains on CIFAR-10 and ImageNet-32.

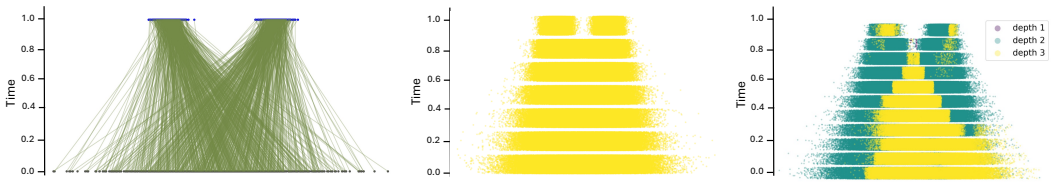
1 INTRODUCTION

Diffusion models (Ho et al., 2020; Song et al., 2021a;b) and flow-matching methods (Liu et al., 2023; Lipman et al., 2023; Albergo & Vanden-Eijnden, 2023; Albergo et al., 2023) have recently gained significant attention, with strong results across computer vision (Ho et al., 2020), medical imaging (Song et al., 2022), robotics (Kapelyukh et al., 2023), and computational biology (Guo et al., 2024). Flow matching models complex, potentially multimodal data distributions by transporting samples from a simple source distribution, such as a standard Gaussian, to the target distribution. At inference time, this is done by solving an ordinary differential equation (ODE) that evolves samples along a learned velocity field over time. The velocity field is typically learned with an MSE objective, which recovers the conditional mean velocity but not the underlying multimodal velocity distribution (Liu et al., 2023). Recent work therefore models this multimodality explicitly: Guo & Schwing (2025) use a latent-variable variational formulation, while Zhang et al. (2025) use hierarchically coupled ODEs to model velocity and higher-order distributions.

Despite recent efforts to model multimodal velocity distributions, existing methods largely overlook that the degree of multimodality varies across location and time during rectified-flow inference. For example, Zhang et al. (2025) explicitly characterize the velocity distribution as a function of location and time, revealing substantial variation in its complexity along the integration trajectory. Treating this complexity uniformly can therefore lead to inefficient modeling and unnecessary computation.

In this paper, we ask whether the variation in velocity multimodality can be modeled explicitly and exploited during inference. Doing so offers two benefits. First, capturing the full multimodal velocity distribution allows integration paths to intersect during generation. Second, under a fixed budget of neural function evaluations (NFEs), computation can be allocated adaptively, with more NFEs devoted to regions where the velocity distribution is more complex, such as areas with dense trajectory intersections, as can be seen in Figure 1. Such adaptive computation has been shown to improve the compute-performance trade-off in related generative settings (Moon et al., 2024; Tang

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.



(a) Linear Interpolation (b) HRF Depth Selection (c) MoD-RF Depth Selection

Figure 1: (a) Rectified flow matching randomly couples source and target samples, producing trajectory regions with varying intersection density and velocity ambiguity. (b) Hierarchical rectified flow. applies a fixed hierarchy depth uniformly across all space-time states, potentially allocating unnecessary computation to simpler regions. (c) Our MoD-RF adaptively allocates NFEs across depths, focusing computation on complex regions while skipping unnecessary deeper HRF integrations, thereby enabling more outer integration steps under the same compute budget.

et al., 2024). Importantly, the local degree of multimodality is not directly observed and generally must be inferred from samples or model predictions (Silverman, 1981; Hartigan & Hartigan, 1985). This motivates learning to allocate hierarchy depth without explicit supervision, in the spirit of adaptive-computation methods (Banino et al., 2021).

To exploit this variation, we study how velocity multimodality changes along the inference trajectory. Building on the hierarchical formulation of Zhang et al. (2025), we observe that the target-direction distribution varies with both time and hierarchy depth, and becomes progressively more concentrated at deeper levels, as analyzed in Section 3.1. Hence, different space-time regions may require different hierarchy depths: highly multimodal regions benefit from deeper flows, while simpler regions can be modeled with shallower ones. Motivated by this observation, we replace the fixed-depth hierarchy with depth-specific flows that are selected adaptively during inference. Under a fixed NFE budget, this allocates more computation to difficult regions and avoids unnecessary nested integration in simpler ones (see Figure 1). We refer to this formulation as *Mixture-of-Depth Rectified Flow* (MoD-RF), detailed in Section 3.2.

Empirically, we evaluate mixture-of-depth rectified flow across a range of datasets in Section 4. On synthetic data, our method more accurately models the target distribution and better captures ambiguity in the velocity field. On high-dimensional image datasets, it consistently outperforms both classic and hierarchical rectified flow under the same NFE budget, with the performance gap becoming more pronounced as the total number of NFEs increases.

2 Preliminaries

Given a dataset $D = \{x_1\}$ of observations x_1, such as images, drawn from an unknown target distribution ρ_1, the goal of generative modeling is to learn a parameterized distribution that approximates ρ_1 and enables to efficiently generate new samples.

2.1 Rectified Flow Matching

Rectified flow transports samples from a known source distribution ρ_0 to a target distribution ρ_1. Let $x_0 \sim \rho_0$ denote a source sample, $x_1 \sim \rho_1$ a target sample, and $t \sim U[0, 1]$ the interpolation time. Rectified flow constructs the interpolant

$$x_t = (1 - t)x_0 + tx_1. \tag{1}$$

The corresponding target direction is $f = x_1 - x_0$, which is the "ground-truth" velocity in standard rectified flow. A neural predictor $f_\theta(x_t, t)$ is trained using the standard ℓ_2 objective

$$\min_\theta \mathbb{E}\left[\|f - f_\theta(x_t, t)\|_2^2\right]. \tag{2}$$

At inference time, the learned target direction defines the velocity field used to transport a source sample $z_0 \sim \rho_0$:

$$dz_t = f_\theta(z_t, t)\, dt, \qquad z_0 \sim \rho_0. \tag{3}$$

Under the independent coupling in Eq. 1, different source-target pairs can pass through the same state x_t at the same time t while inducing different target directions. Hence, the conditional target-direction distribution $p(f \mid x_t, t)$ is generally multimodal. Since rectified flow regresses a deterministic target direction with an ℓ_2 objective, the optimal predictor learns the conditional mean of this distribution rather than its individual modes. Although this is sufficient to preserve the target marginals, averaging competing directions can result in curved, non-intersecting trajectories.

2.2 Hierarchical Rectified Flow

Next, we discuss the extension of the standard depth-2 hierarchical rectified flow (HRF2) to arbitrary depth d. Let $x_1 \sim \rho_1$ denote a target sample and expand the single source x_0, described in Section 2.1, to independent source samples $x_0^{(1)}, \ldots, x_0^{(d)} \overset{\text{iid}}{\sim} \rho_0$ across d hierarchy levels. For a time vector $\mathbf{t} = \begin{bmatrix} t^{(1)} & \cdots & t^{(d)} \end{bmatrix}^\top \in [0,1]^d$, the state at hierarchy depth r is defined by

$$x_t^{(r)} = (1 - t^{(r)}) x_0^{(r)} + t^{(r)} \left(x_1 - \sum_{j=1}^{r-1} x_0^{(j)} \right), \qquad r = 1, \ldots, d. \tag{4}$$

The corresponding target direction at depth r is $f^{(r)} = x_1 - \sum_{j=1}^{r} x_0^{(j)}$. As depth d increases, the hierarchical states are collected in $\mathbf{x}_t^{(1:d)} = [x_t^{(1)} \cdots x_t^{(d)}]^\top$. Using this notation, the conditional *target-direction distribution* at depth d is

$$p\Big(f^{(d)} \mid \mathbf{x}_t^{(1:d)}, \mathbf{t} \Big). \tag{5}$$

This formulation makes explicit that the target-direction distribution varies jointly with hierarchy depth and the associated time variables, causing its degree of multimodality to change substantially along the inference trajectory. We refer to this spatially and temporally varying ambiguity as *heterogeneous multimodality*.

We can then train depth-d HRF by learning a parametrized velocity predictor for the deepest target direction using an ℓ_2 objective:

$$\min_\theta \mathbb{E}\left[\Big\| x_1 - \sum_{j=1}^{d} x_0^{(j)} - f_\theta(\mathbf{x}_t^{(1:d)}, \mathbf{t}) \Big\|_2^2 \right]. \tag{6}$$

At inference time, HRF recursively couples d ODEs, where the deepest process is driven by the learned predictor and each higher-level process is driven by the terminal state of the level below:

$$dz_{t^{(d)}}^{(d)} = f_\theta\Big(\mathbf{z}_t^{(1:d)}, \mathbf{t} \Big)\, dt^{(d)};\; dz_{t^{(d-1)}}^{(d-1)} = z_1^{(d)}\, dt^{(d-1)},\; \cdots;\; dz_{t^{(1)}}^{(1)} = z_1^{(2)}\, dt^{(1)}. \tag{7}$$

Each hierarchy level is initialized from its corresponding source distribution. Thus, the deepest ODE generates the direction required by level $d-1$, whose terminal state in turn drives level $d-2$, and so on, until the outermost ODE transports the data sample from the source to the target distribution.

3 Mixture-of-Depth Rectified Flow for Heterogeneous Multimodality

This section presents our proposed *Mixture-of-Depth Rectified Flow* (MoD-RF) for modeling heterogeneous multimodality. In Section 3.1, we first analyze how target-direction ambiguity varies across time and hierarchy depth through a Gaussian-mixture case study. Building on this analysis, we introduce MoD-RF to adapt the hierarchy depth to local ambiguity, and then describe its training objective and adaptive inference procedure in Sections 3.2 and 3.3.

3.1 Heterogeneous Multimodality and A Case Study with Gaussian Mixtures

Intuitively, HRF hierarchically moves to a deeper level by introducing an additional Gaussian latent variable and integrating toward a progressively transformed target distribution. As the hierarchy

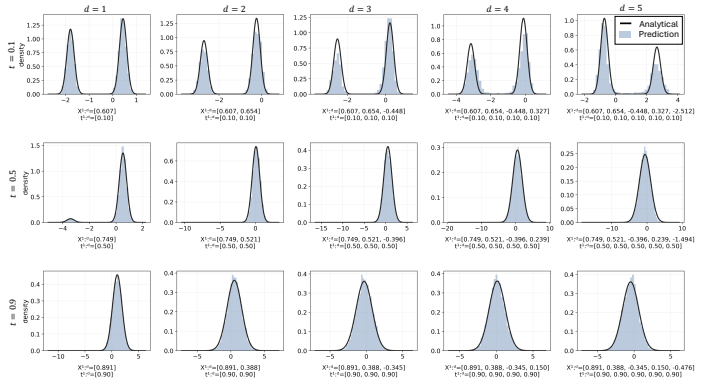


Figure 2: Conditional target-direction distributions have heterogeneous multimodality across both time and depth axes.

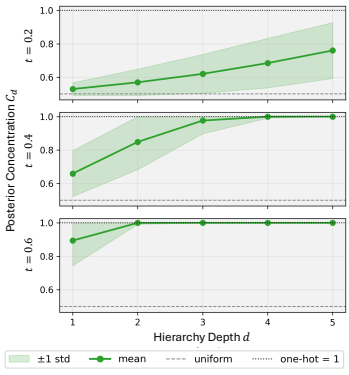


Figure 3: Posterior concentration C_d over hierarchy depth d.

deepens, the target distribution becomes increasingly smoothed, with distinct modes becoming more overlapped and the distribution approaching a simpler, near-unimodal form. Consequently, a more complex or strongly multimodal velocity distribution may require greater depth before it can be represented by a target that is sufficiently simple to learn effectively with a standard MSE objective. We formalize this intuition through an analysis of a Gaussian mixture case study as follows and provide full roof in Appendix B.1.

Corollary 1 (Conditional target-direction distribution at depth d)**.** *Let the target distribution be a finite Gaussian mixture* $\rho_1(x_1) = \sum_{k=1}^{K} w_k \mathcal{N}(x_1; \mu_k, \sigma_k^2)$*, and* $\rho_0 = \mathcal{N}(0, 1)$*.*

Then the conditional target-direction distribution at depth d *is*

$$p\Big(f^{(d)} \mid \mathbf{x}_t^{(1:d)}, \mathbf{t}\Big) = \sum_{k=1}^{K} \widetilde{w}_{d,k} \mathcal{N}\Big(f^{(d)}; m_{d,k}, q_{d,k}\Big), \tag{8}$$

where the posterior weight is $\widetilde{w}_{d,k} = \frac{w_k \mathcal{N}_d(\mathbf{x}_t^{(1:d)}; \mu_k \mathbf{t}, G_{d,k})}{\sum_{\ell=1}^{K} w_\ell \mathcal{N}_d(\mathbf{x}_t^{(1:d)}; \mu_\ell \mathbf{t}, G_{d,\ell})}$*, the component mean and variance are* $m_{d,k} = \mu_k + \mathbf{c}_{d,k}^\top G_{d,k}^{-1}(\mathbf{x}_t^{(1:d)} - \mu_k \mathbf{t})$ *and* $q_{d,k} = \sigma_k^2 + d - \mathbf{c}_{d,k}^\top G_{d,k}^{-1} \mathbf{c}_{d,k}$*, respectively, and* $[\mathbf{c}_{d,k}]_r = t^{(r)}(\sigma_k^2 + r) - 1$*. Here,* $G_{d,k}$ *denotes the covariance matrix of* $\mathbf{x}_t^{(1:d)}$ *conditioned on mixture component* k*. Its diagonal entries are*

$$[G_{d,k}]_{rr} = (1 - t^{(r)})^2 + (t^{(r)})^2(\sigma_k^2 + r - 1), \tag{9}$$

and for $r < s$*,*

$$[G_{d,k}]_{rs} = [G_{d,k}]_{sr} = t^{(s)} \left[t^{(r)}(\sigma_k^2 + r) - 1\right]. \tag{10}$$

Remark 1. Corollary 1 reveals that the conditional target-direction distribution is heterogeneous along both time and hierarchy depth. Its mixture weights, means, and variances vary with the time vector $\mathbf{t}$, while increasing depth introduces additional hierarchical states into the conditioning set. Consequently, the ambiguity of the target direction can change substantially across both time and depth, rather than remaining uniform throughout the hierarchical flow.

To understand how multimodality evolves, we further analyze its behavior across hierarchy depth. We observe that the hierarchy also progressively resolves ambiguity over the underlying mixture component, as stated in Corollary 2. The proof is deferred to Appendix B.2.

Corollary 2 (Increasing posterior concentration across depth)**.** *Given* $\widetilde{w}_{d,k}$ *defined in Corollary 1 and the corresponding posterior concentration at depth* d*, i.e.,* $\mathcal{C}_d = \|\widetilde{\mathbf{w}}_d\|_2^2 = \sum_{k=1}^{K} \widetilde{w}_{d,k}^2$*, the posterior concentration is non-decreasing with hierarchy depth in expectation:*

$$\mathbb{E}[\mathcal{C}_{d+1}] \geq \mathbb{E}[\mathcal{C}_d]. \tag{11}$$

Remark 2. Corollary 2 suggests that increasing hierarchy depth progressively reduces ambiguity among competing target modes. While shallow levels may exhibit strongly multimodal target-direction distributions, deeper levels can become increasingly dominated by a single component and

are therefore closer to a unimodal regression problem. This makes the conditional mean predicted by an ℓ_2 objective increasingly representative of the dominant target direction.

3.2 Mixture-of-Depth Rectified Flow

The analysis in Section 3.1 reveals two complementary properties of hierarchical rectified flow. First, the ambiguity of the target-direction distribution varies across space-time states (see Corollary 1). Second, increasing hierarchy depth progressively reduces this ambiguity in expectation (see Corollary 2). Importantly, however, the depth required to obtain a sufficiently concentrated target-direction distribution need not be the same for every state. Regions with nearly unimodal directions may already be well modeled at a shallow depth, whereas highly ambiguous regions may require additional levels, which is clearly shown in Figure 2.

This motivates replacing the fixed hierarchy depth of HRF with an adaptive depth variable $D \in \{1, \dots, D_{\max}\}$. We refer to the resulting formulation as *Mixture-of-Depth Rectified Flow* (MoD-RF). Conceptually, for a training tuple $(x_1, x_0^{(1:d)})$ with the observed velocity $v = x_1 - x_0^{(1)}$, MoD-RF represents the velocity distribution using depth-specific components,

$$p(v \mid x_t, t) = \sum_{d=1}^{D_{\max}} \pi_{\eta,d}\, p_\theta^{(d)} \left(v \mid \mathbf{x}_t^{(1:d)}, \mathbf{t}^{(1:d)} \right), \tag{12}$$

where $p_\theta^{(d)}$ denotes the velocity distribution induced by a hierarchy of depth d, and $\pi_{\eta,d}$ denotes the probability of using that depth for the current hierarchical state.

Depth distribution. Intuitively, the appropriate hierarchy depth must depend on the local ambiguity of the target-direction distribution. However, directly predicting the depth-selection distribution is nontrivial because the context required to assess a depth-d decision, $\mathbf{x}_t^{(1:d)}$ and $\mathbf{t}^{(1:d)}$, is only available after the hierarchy has been evaluated up to depth d. Thus, deeper hierarchical states cannot be used to determine the required depth before they are constructed.

We therefore parameterize the depth-selection probability through a sequential stopping process:

$$\pi_{\eta,d} = s_{\eta,d} \prod_{j=1}^{d-1} (1 - s_{\eta,j}), \qquad s_{\eta,d} = p_\eta \Big(\text{stop at d} \mid \mathbf{x}_t^{(1:d)}, \mathbf{t}^{(1:d)} \Big), \tag{13}$$

where $s_{\eta,d}$ denotes the probability of stopping at depth d given the hierarchical context observed so far, and we set $s_{\eta,D_{\max}} = 1$ to ensure termination. Hence, selecting depth d corresponds to continuing through all preceding depths and stopping once the depth-d representation is available. This formulation allows the model to adaptively allocate greater hierarchy depth to more ambiguous regions while terminating early for states that are already sufficiently resolved.

3.3 Training and Inference

3.3.1 Training

We emphasize that no ground-truth label is available for the appropriate hierarchy depth of each sample. We therefore supervise the depth selector using the prediction quality achieved at each depth. A straightforward choice is the local target-direction error used to train HRF-d, but this only reflects the accuracy of the depth-d direction $f^{(d)}$ and does not directly measure how well that depth reconstructs the velocity of interest. Since the selector ultimately determines which hierarchy depth is used to model the velocity distribution, we instead assess each depth through a coarse reconstruction of the original velocity.

Specifically, for a training tuple $(x_1, x_0^{(1:d)})$ with $v = x_1 - x_0^{(1)}$, we first predict the depth-d target direction at the initial hierarchical state and fold it back to velocity space:

$$\widehat{v}^{(d)} = \widehat{f}^{(d)} + \sum_{j=2}^{d} x_0^{(j)}, \text{ with } \widehat{f}^{(d)} = f_\theta^{(d)} \left(\mathbf{x}_0^{(1:d)}, \mathbf{0}^d \right), \tag{14}$$

Algorithm 1 Adaptive Inference with Mixture-of-Depth Rectified Flow

Require: Source sample z_0, maximum depth $D_{\max}$, stopping threshold δ
for each outer integration step **do**
  Initialize the current outer state and sample deeper source states
  Evaluate $s_{\eta,d}$ for $d = 1, \ldots, D_{\max}$ {Eq. 13}
  Select terminal depth d if $s_{\eta,d} > \delta$
  Lock the selected depth d for the current outer step
  if $d = 1$ **then**
    Predict the velocity using the depth-one component (standard RF)
  else
    Solve the nested depth-d HRF to obtain the velocity
  end if
  Update the outer data state using the predicted velocity
end for
return Generated sample z_1

We then train the depth selector with a likelihood-weighted cross-entropy objective,

$$\mathcal{L}(\eta) = -\sum_{d=1}^{D_{\max}} r_d \log \pi_{\eta,d}, \text{ with } r_d = \exp\left(-\frac{\|v - \widehat{v}^{(d)}\|_2^2}{2\sigma_d^2}\right), \tag{15}$$

where σ_d^2 is a fixed scale parameter ($\sigma_d = 0.5$ in this work). The score r_d provides a soft measure of how well depth d reconstructs the final velocity: depths with smaller reconstruction error receive larger supervision weights. As a result, the selector is encouraged to assign higher probability to the depths that already provide an accurate velocity prediction, while reserving deeper computation for more ambiguous cases. We note that, although the velocity is only coarsely reconstructed, the reconstruction requires only inexpensive additions of already available intermediate states and therefore remains simulation-free, as in prior work.

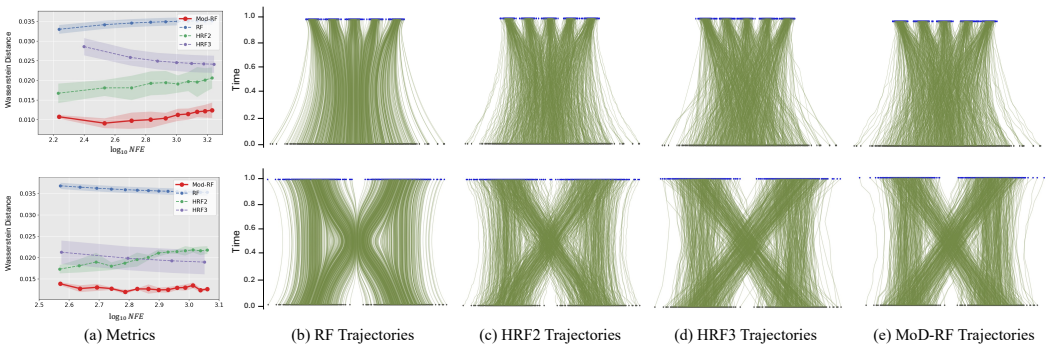
3.3.2 Adaptive Inference

At inference time, MoD-RF selects the hierarchy depth independently at each outer, depth-one integration step (see Algorithm 1). Given the current outer state $z_t^{(1)}$, we sample source states $z_0^{(2:D_{\max})}$ for the deeper hierarchy levels and evaluate the stopping probabilities sequentially as the hierarchical context becomes available. For each sample, we select the first depth d such that $s_{\eta,d} > \delta$, where δ is a fixed stopping threshold; if no earlier depth satisfies the criterion, we use $D_{\max}$.

Once a depth is selected, both the depth and the sampled source states are fixed for the remainder of that outer step. If $d = 1$, the depth-one predictor directly provides the velocity update. If $d > 1$, we solve the corresponding nested HRF to the selected depth using the prescribed inner integration steps, and recursively propagate the resulting direction back to the outermost level. The stopping network is not re-evaluated during this nested solve. In this way, MoD-RF adapts computational depth across samples and outer integration steps, allocating deeper computation only where additional hierarchy is needed while preserving a consistent HRF trajectory within each step.

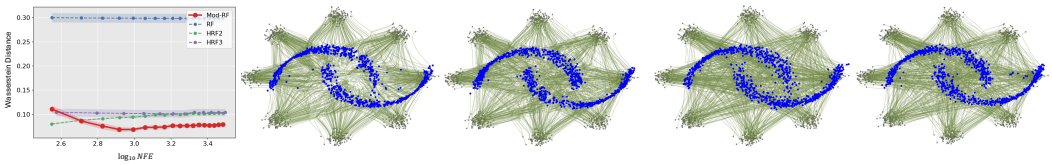
4 Experiments

We evaluate MoD-RF on synthetic and image-generation tasks to study whether adaptive hierarchy depth improves the allocation of computation under a fixed inference budget. We consider three settings: synthetic 1D distributions, synthetic 2D distributions, and high-dimensional image datasets. Across all experiments, we compare against standard Rectified Flow (RF) (Liu et al., 2023) and fixed-depth Hierarchical Rectified Flow (HRF) (Zhang et al., 2025), including HRF2 and HRF3 when applicable. Following Zhang et al. (2025), we report the total number of neural function evaluations (NFEs), accounting for evaluations across all hierarchy levels, and compare methods under matched NFE budgets. This allows us to isolate the benefit of allocating deeper computation selectively rather than applying a fixed hierarchy depth uniformly throughout inference.



(a) Metrics (b) RF Trajectories (c) HRF2 Trajectories (d) HRF3 Trajectories (e) MoD-RF Trajectories

Figure 4: Results on 1D example. Top row: ρ_0 is a standard Gaussian and ρ_1 is a mixture of 5 Gaussians. Bottom row: Both ρ_0 and ρ_1 are a mixture of 2 Gaussians. (a) The 1-Wasserstein distance with respect to NFE. (c), (d) and (e) The trajectories (green) of sample particles flowing from the source distribution (grey) to target distribution (blue).



(a) Metrics (b) RF Trajectories (c) HRF2 Trajectories (d) HRF3 Trajectories (e) MoD-RF Trajectories

Figure 5: Results on 2D data, where ρ_0 is a mixture of 8 Gaussians and ρ_1 is the moons data. (a) Sliced 2-Wasserstein distance with respect to NFE. (b), (c), (d) and (e) show the trajectories (green) of sample particles flowing from source distribution (grey) to target distribution (blue).

4.1 Synthetic 1D Data

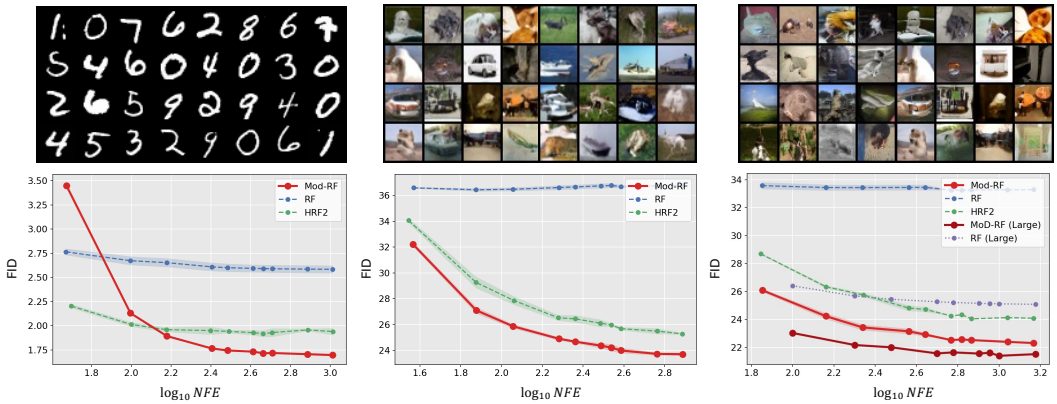
Setup. We first evaluate MoD-RF in one-dimensional settings by considering two source-target configurations. In the first, the source distribution ρ_0 is a standard Gaussian and the target distribution ρ_1 is a mixture of five Gaussians. In the second, both ρ_0 and ρ_1 are mixtures of two Gaussians. Here, the generation quality is measured using the 1-Wasserstein distance between generated samples and the target distribution, and all methods are compared under the same total NFE budget.

Results. As shown in Figure 4, MoD-RF consistently achieves the lowest Wasserstein distance across the evaluated NFE budgets in both 1D settings. In the Gaussian-to-five-Gaussian setting, the advantage is already visible at small NFE budgets and remains as additional computation is provided, indicating that adaptive depth can efficiently resolve the substantially more multimodal target distribution without requiring uniformly deep hierarchical integration. A similar trend is observed in the mixture-to-mixture setting, where MoD-RF maintains a clear advantage over RF, HRF2, and HRF3 throughout the evaluated NFE range.

The trajectory visualizations provide a complementary view of this behavior. Standard rectified flow produces non-intersecting trajectories that bend around ambiguous regions, whereas hierarchical methods permit trajectory intersections by modeling the multimodal velocity distribution. MoD-RF preserves this benefit while adapting the hierarchy depth across outer integration steps. Intuitively, deeper hierarchy progressively simplifies the modeling of the same depth-one velocity distribution, and different space-time locations may require different depths before the target-direction distribution becomes sufficiently easy to predict. MoD-RF therefore allocates deeper computation only where needed, rather than applying a fixed depth uniformly along the trajectory.

4.2 Synthetic 2D Data

Setup. We next consider a 2D data generation problem where the source distribution ρ_0 is a mixture of eight Gaussians and the target distribution ρ_1 is the two-moons distribution. We compare RF,



(a) MNIST (b) CIFAR-10 (c) ImageNet-32

Figure 6: Experimental results on (a) MNIST, (b) CIFAR-10 and (c) ImageNet-32 datasets. Top row: samples of generated images, bottom row: FID scores with respect to total NFEs

HRF2, HRF3, and MoD-RF under matched total NFE budgets, using sliced 2-Wasserstein distance for evaluation and particle trajectories for qualitative comparison. More details are in Appendix C.1.

Results. As shown in Figure 5, HRF2 is competitive at the smallest NFE budget, but MoD-RF improves rapidly as more computation becomes available and outperforms all fixed-depth variants in the moderate- and high-NFE regimes. While the fixed-depth models largely plateau, MoD-RF continues to reduce the sliced Wasserstein distance. The trajectory visualizations show that all hierarchical methods permit intersecting transport paths toward the two-moons target. The performance gap therefore suggests that the main advantage of MoD-RF comes from adapting hierarchy depth to allocate computation more effectively, rather than from trajectory intersection alone.

4.3 Image Data

Setup. Finally, we evaluate MoD-RF on high-dimensional image generation using MNIST (LeCun et al., 1998), CIFAR-10 (Krizhevsky & Hinton, 2009), and ImageNet-32 (Deng et al., 2009). These experiments examine whether the benefits of adaptive hierarchy depth extend beyond synthetic distributions to substantially higher-dimensional generative tasks. We compare MoD-RF against standard rectified flow (RF) and fixed-depth HRF2, and evaluate generation quality using Fréchet Inception Distance (FID). All function evaluations incurred by nested hierarchical solves are included in the reported NFE budget. More details are presented in Appendix C.2.

Results. Figure 6 reveals a consistent pattern across all three datasets. At very small NFE budgets, HRF2 provides a strong quality-compute trade-off and can outperform MoD-RF on MNIST, where MoD-RF is less competitive in the lowest-compute regime. This is because MoD-RF allocates computation adaptively at each outer velocity step, which provides limited benefit when only a very small number of outer steps is available. As the available budget increases, however, MoD-RF improves substantially and eventually surpasses HRF2 on both datasets. On ImageNet-32, MoD-RF already outperforms both RF and HRF2 at small NFE budgets and maintains this advantage throughout the evaluated range.

The advantage becomes clearer in the moderate- and high-NFE regimes. On MNIST, the FID of HRF2 improves quickly at first but then changes only modestly, whereas MoD-RF continues to improve after crossing the fixed-depth baseline. A similar trend is observed on CIFAR-10, where HRF2 performs well with limited computation but its gains diminish as more NFEs are introduced. On ImageNet-32, MoD-RF consistently achieves lower FID than both RF and HRF2 across all evaluated NFE budgets, while continuing to improve as additional computation becomes available. Standard rectified flow remains substantially worse than the hierarchical approaches throughout most of the evaluated range.

Ablation on model capacity. We further test whether MoD-RF remains effective with a stronger RF baseline by increasing the parameter count of RF by $15.9\times$ (see Appendix C), denoted as RF (Large), and evaluating both RF (Large) and MoD-RF (Large) while keeping HRF2 unchanged. Here, MoD-RF (Large) uses RF (Large) and HRF2 as the depth-specific flows. As shown in Figure 6, the larger backbone substantially improves RF, while MoD-RF (Large) consistently achieves lower FID than both RF (Large) and the standard MoD-RF across the evaluated NFE range. This shows that adaptive hierarchy depth remains beneficial as the underlying predictor becomes stronger and is complementary to increased model capacity. More generally, MoD-RF allocates additional computation across hierarchy depths and outer integration steps rather than uniformly increasing computation within a fixed hierarchy, leading to a more effective use of the available NFE budget.

5 Related Work

Generative Modeling. Deep generative modeling has been studied through generative adversarial networks (GANs) (Goodfellow et al., 2014; Arjovsky et al., 2017), variational autoencoders (VAEs) (Kingma & Welling, 2014), and normalizing flows (Tabak & Turner, 2013; Rezende & Mohamed, 2015; Dinh et al., 2017; Huang et al., 2018; Durkan et al., 2019). Continuous normalizing flows extend invertible transformations to neural ODEs (Chen et al., 2018; Grathwohl et al., 2019), while diffusion models learn to reverse a stochastic noising process (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2021b) and have achieved strong performance in high-dimensional generative tasks (Rombach et al., 2022; Hoogeboom et al., 2022; Saharia et al., 2022; Nguyen et al., 2025). These methods motivate increasingly efficient continuous-time generative models, but generally do not adapt their computation according to the local multimodal complexity encountered along a generation trajectory. In contrast, our method explicitly exploits such local variation to adapt the computational depth of rectified-flow inference.

Rectified Flow and Velocity Distribution. Flow matching learns a time-dependent velocity field through an ODE (Lipman et al., 2023; Albergo & Vanden-Eijnden, 2023; Albergo et al., 2023), while rectified flow uses linear interpolation between source and target samples (Liu et al., 2023). Under the standard ℓ_2 objective, the model predicts the conditional mean velocity rather than the full multimodal velocity distribution. Prior work improves rectified-flow trajectories through straighter or better source–target couplings (Liu et al., 2023; Pooladian et al., 2023; Tong et al., 2024). More recent methods explicitly model velocity multimodality using latent variables (Guo & Schwing, 2025) or hierarchical ODEs (Zhang et al., 2025). In contrast to fixed-depth HRF, MoD-RF adapts the hierarchy depth to the local complexity of the velocity distribution.

Adaptive Computation for Multimodal Modeling. Multimodality can be identified from samples using classical statistical tools, such as kernel-density-based mode analysis (Silverman, 1981) and the dip test (Hartigan & Hartigan, 1985). More broadly, adaptive-computation methods vary inference cost based on input or prediction difficulty. PonderNet learns how many computation steps to use (Banino et al., 2021), Mixture-of-Depths Transformer dynamically allocates computation across network depth (Raposo et al., 2024), and Switch Transformers route inputs through different experts under sparse computation (Fedus et al., 2022). In generative modeling, Moon et al. (2024) accelerate diffusion sampling through early exiting, while Tang et al. (2024) adapt computation across denoising steps. Unlike these methods, MoD-RF adapts the depth of hierarchical velocity modeling according to whether additional hierarchy improves local velocity reconstruction.

6 Conclusion

In this work, we introduced *Mixture-of-Depth Rectified Flow*, an adaptive extension of hierarchical rectified flow for heterogeneous multimodality. We showed that target-direction ambiguity varies across time and hierarchy depth, and that deeper hierarchies progressively reduce this ambiguity. MoD-RF therefore selects the hierarchy depth adaptively at each outer integration step using a sequential stopping mechanism trained from velocity reconstruction quality, without requiring ground-truth depth labels. Experiments on synthetic data, MNIST, CIFAR-10, and ImageNet-32 demonstrate improved quality-compute trade-offs over standard rectified flow and fixed-depth HRF. Overall, adaptive hierarchy depth provides a more effective allocation of inference computation according to local target-direction complexity.

AI USE STATEMENT

In this work, we used generative AI tools for implementing methods. We have not used generative AI tools for: help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, design or provide feedback on research methodology or experiments, interpret results. Generate synthetic data sets, assist with translation, clean and reformat dataset, support qualitative and thematic data analysis are not applicable to this work. Additionally, we used generative AI tools for draft parts of a research paper, summarize or analyse existing literature, edit a research paper to improve readability. We have reviewed all AI-assisted work. We checked LLM-generated relevant work through a manual literature survey and analysis. LLM-generated wording was verified and tested for correctness against original draft version by 2 authors. We also double check the LLM-generated texts to ensure they do not hallucinate findings and numerical results. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide detailed information to support reproducibility of both the theoretical and empirical results. The model formulation, training objective, sequential depth-selection mechanism, and adaptive inference procedure are described in Sections 3.2-4, with additional pseudocode for the nested adaptive solver provided in Appendix A. Complete derivations and proofs for the Gaussian-mixture analysis and posterior concentration results are included in Appendix B. Detailed experimental settings, including network architectures, parameter counts, optimization hyperparameters, depth-selector configurations, integration schedules, NFE accounting, dataset configurations, and scaling protocols are provided in Appendix C. All comparisons use matched NFE budgets with the accounting rules specified in the paper and appendix. We will also release the source code upon acceptance to facilitate reproduction of the reported results.

REFERENCES

M. Albergo and E. Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In *International Conference on Learning Representations*, 2023.

M. Albergo, N. Boffi, and E. Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *Journal of Machine Learning Research*, 2023.

M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, 2017.

A. Banino, J. Balaguer, and C. Blundell. PonderNet: Learning to ponder. In *ICML Workshop on Automated Machine Learning*, 2021.

R. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, 2018.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2009.

L. Dinh, J. Sohl-Dickstein, and S. Bengio. Density estimation using Real NVP. In *International Conference on Learning Representations*, 2017.

C. Durkan, A. Bekasov, I. Murray, and G. Papamakarios. Neural spline flows. In *Advances in Neural Information Processing Systems*, 2019.

W. Fedus, B. Zoph, and N. Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 2022.

I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, 2014.

W. Grathwohl, R. Chen, J. Bettencourt, and D. Duvenaud. Ffjord: Free-form continous dynamics for scalable reversible generative models. In *International Conference on Learning Representations*, 2019.

P. Guo and A. Schwing. Variational rectified flow matching. In *International Conference on Machine Learning*, 2025.

Z. Guo, J. Liu, Y. Wang, M. Chen, D. Wang, D. Xu, and J. Cheng. Diffusion models in bioinformatics and computational biology. *Nature Reviews Bioengineering*, 2024.

J. A. Hartigan and P. M. Hartigan. The dip test of unimodality. *The Annals of Statistics*, 1985. doi: 10.1214/aos/1176346577.

J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, 2020.

E. Hoogeboom, V. G. Satorras, C. Vignac, and M. Welling. Equivariant diffusion for molecule generation in 3d. In *International Conference on Machine Learning*, 2022.

C.-W. Huang, D. Krueger, A. Lacoste, and A. Courville. Neural autoregressive flows. In *International Conference on Machine Learning*, 2018.

I. Kapelyukh, V. Vosylius, and E. Johns. DALL-E-Bot: Introducing web-scale diffusion models to robotics. *IEEE Robotics and Automation Letters*, 2023.

D. Kingma and M. Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014.

A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998.

Y. Lipman, R. Chen, H. Ben-Hamu, M. Nickel, and M. Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

X. Liu, C. Gong, and Q. Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations*, 2023.

T. Moon, M. Choi, E. Yun, J. Yoon, G. Lee, J. Cho, and J. Lee. A simple early exiting framework for accelerated sampling in diffusion models. In *International Conference on Machine Learning*, 2024.

M. D. Nguyen, D. T. Nguyen, T. V. Nguyen, H. Yamada, H. H. Pham, and P. Le Nguyen. Bridging classification and segmentation in osteosarcoma assessment via foundation and discrete diffusion models. In *IEEE International Symposium on Biomedical Imaging*, 2025.

A.-A. Pooladian, H. Ben-Hamu, C. Domingo-Enrich, B. Amos, Y. Lipman, and R. Chen. Multisample flow matching: Straightening flows with minibatch couplings. In *International Conference on Machine Learning*, 2023.

D. Raposo, S. Ritter, B. A. Richards, T. Lillicrap, P. C. Humphreys, and A. Santoro. Mixture-of-depths: Dynamically allocating compute in transformer-based language models. *arXiv preprint arXiv:2404.02258*, 2024.

D. Rezende and S. Mohamed. Variational inference with normalizing flows. In *International Conference on Machine Learning*, 2015.

R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, R. Denton, S. Ghasemipour, R. Gontijo-Lopes, B. Karagol Ayan, T. Salimans, J. Ho, D. J. Fleet, and M. Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In *Advances in Neural Information Processing Systems*, 2022.

B. W. Silverman. Using kernel density estimates to investigate multimodality. *Journal of the Royal Statistical Society: Series B (Methodological)*, 1981. doi: 10.1111/j.2517-6161.1981.tb01155.x.

J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, 2015.

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

Y. Song, J. Sohl-Dickstein, D. 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, L. Shen, L. Xing, and S. Ermon. Solving inverse problems in medical imaging with score-based generative models. In *International Conference on Learning Representations*, 2022.

E. G. Tabak and C. V. Turner. A family of nonparametric density estimation algorithms. *Communications on Pure and Applied Mathematics*, 2013.

S. Tang, Y. Wang, C. Ding, Y. Liang, Y. Li, and D. Xu. AdaDiff: Accelerating diffusion models through step-wise adaptive computation. In *European Conference on Computer Vision*, 2024.

A. Tong, N. Malkin, G. Huguet, Y. Zhang, J. Rector-Brooks, K. Fatras, G. Wolf, and Y. Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *Transactions on Machine Learning Research*, 2024.

Y. Zhang, Y. Yan, A. Schwing, and Z. Zhao. Towards hierarchical rectified flow. In *International Conference on Learning Representations*, 2025.

APPENDIX: MIXTURE-OF-DEPTH RECTIFIED FLOW FOR HETEROGENEOUS MULTIMODALITY

The appendix is organized as follows. Appendix A provides detailed pseudocode for adaptive inference, extending Algorithm 1. Appendix B presents the full theoretical derivations and proofs, including the proof of Corollary 1 in Appendix B.1 and the proof of Corollary 2 in Appendix B.2. Appendix C provides additional experimental and implementation details. Finally, Appendix D presents qualitative results, and Appendix E discusses the limitations of our work.

A DETAILED PSEUDOCODE

We present the detailed version of Algorithm 1 as follows.

Algorithm 2 Adaptive Inference with Mixture-of-Depth Rectified Flow

Require: Source sample $z_0^{(1)}$, depth predictors $\{f_\theta^{(d)}\}_{d=1}^{D_{\max}}$, stopping model s_η, integration steps $\{N_d\}_{d=1}^{D_{\max}}$, stopping threshold δ

Initialize $z^{(1)} \leftarrow z_0^{(1)}$
for $i = 0, \dots, N_1 - 1$ **do**
  Set $t^{(1)} \leftarrow i/N_1$
  Sample deeper source states $z_0^{(d)} \sim \rho_0$ for $d = 2, \dots, D_{\max}$
  Construct $\mathbf{z}^{(1:D_{\max})} = [z^{(1)}, z_0^{(2)}, \dots, z_0^{(D_{\max})}]$ and initialize $t^{(2:D_{\max})} = 0$
  Evaluate stopping probabilities $s_{\eta,1:D_{\max}}$ from the current hierarchical context
  Select

$$d^\star \leftarrow \min\{d : \ s_{\eta,d} > \delta\},$$

  with $s_{\eta,D_{\max}} = 1$
  Lock $d^\star$ and the sampled deeper source states for this outer step
  if $d^\star = 1$ **then**
    $u \leftarrow f_\theta^{(1)}(z^{(1)}, t^{(1)})$
  else
    $u \leftarrow$ LOCKEDNESTEDSOLVE$(\mathbf{z}, \mathbf{t}, 2, d^\star, \{N_d\})$
  end if
  Update the outer state: $z^{(1)} \leftarrow z^{(1)} + \frac{1}{N_1} u$
end for
return $z^{(1)}$

At the beginning of each depth-one Euler step, the source states of all deeper hierarchy levels are sampled once and the stopping model is evaluated to determine the first depth $d^\star$ satisfying $s_{\eta,d^\star} > \delta$. The selected depth and sampled hierarchical states are then fixed throughout the corresponding nested integration. For $d^\star > 1$, Algorithm 3 recursively integrates the hierarchy until the terminal depth is reached, where $f_\theta^{(d^\star)}$ provides the target direction. The resulting terminal state is propagated recursively back to depth one and used as the outer velocity update. Importantly, the stopping model is queried only once per outer step and is not re-evaluated within the nested solve.

B THEORETICAL DERIVATIONS AND PROOFS

B.1 PROOF FOR COROLLARY 1

We first restate Corollary 1 in details and its proof as follows.

Corollary 1 (Conditional direction distribution for depth-d HRF) Let the target sample x_1 follow a finite Gaussian mixture,

$$\rho_1(x_1) = \sum_{k=1}^{K} w_k \, \mathcal{N}(x_1; \mu_k, \sigma_k^2), \qquad w_k > 0, \qquad \sum_{k=1}^{K} w_k = 1, \tag{16}$$

Algorithm 3 Locked Nested HRF Solve

Require: Hierarchical states $\mathbf{z}$, times $\mathbf{t}$, current depth r, selected terminal depth $d^\star$, integration steps $\{N_d\}$

for $i = 0, \dots, N_r - 1$ **do**
  Set $t^{(r)} \leftarrow i/N_r$
  if $r = d^\star$ **then**
    Evaluate the terminal depth predictor:

$$u \leftarrow f_\theta^{(d^\star)}\left(\mathbf{z}^{(1:d^\star)}, \mathbf{t}^{(1:d^\star)}\right)$$

  else
    Recursively solve the next hierarchy level:

$$u \leftarrow \textsc{LockedNestedSolve}(\mathbf{z}, \mathbf{t}, r+1, d^\star, \{N_d\})$$

  end if
  Update the current hierarchy state:

$$z^{(r)} \leftarrow z^{(r)} + \frac{1}{N_r} u$$

end for
return $z^{(r)}$

and let the source samples at different hierarchy levels satisfy

$$x_0^{(1)}, \dots, x_0^{(d)} \overset{\text{iid}}{\sim} \mathcal{N}(0,1), \tag{17}$$

independently of x_1.

For a fixed time vector

$$\mathbf{t} = \begin{bmatrix} t^{(1)} & \cdots & t^{(d)} \end{bmatrix}^\top \in [0,1]^d, \tag{18}$$

define the state at hierarchy level r as in HRF:

$$x_t^{(r)} = (1 - t^{(r)}) x_0^{(r)} + t^{(r)} \left(x_1 - \sum_{j=1}^{r-1} x_0^{(j)} \right), \qquad r = 1, \dots, d. \tag{19}$$

Collect the states through depth d as

$$\mathbf{x}_t^{(1:d)} = \begin{bmatrix} x_t^{(1)} & \cdots & x_t^{(d)} \end{bmatrix}^\top, \tag{20}$$

and define the target direction at depth d as

$$f^{(d)} = x_1 - \sum_{j=1}^{d} x_0^{(j)}. \tag{21}$$

Assume that the covariance matrices defined below are nonsingular. Then, conditioned on $\mathbf{x}_t^{(1:d)} = \mathbf{z}^{(1:d)}$, the depth-$d$ direction follows a Gaussian mixture:

$$p\left(f^{(d)} \mid \mathbf{x}_t^{(1:d)} = \mathbf{z}^{(1:d)}, \mathbf{t}\right) = \sum_{k=1}^{K} \widetilde{w}_{d,k}\left(\mathbf{z}^{(1:d)}, \mathbf{t}\right) \mathcal{N}\left(f^{(d)}; m_{d,k}\left(\mathbf{z}^{(1:d)}, \mathbf{t}\right), q_{d,k}(\mathbf{t})\right). \tag{22}$$

For mixture component k, define

$$\mathbf{c}_{d,k} = \begin{bmatrix} c_{1,k} & \cdots & c_{d,k} \end{bmatrix}^\top, \qquad c_{r,k} = t^{(r)}(\sigma_k^2 + r) - 1. \tag{23}$$

Let $G_{d,k} \in \mathbb{R}^{d \times d}$ denote the covariance matrix of $\mathbf{x}_t^{(1:d)}$ conditioned on mixture component k. Its diagonal entries are

$$[G_{d,k}]_{rr} = (1 - t^{(r)})^2 + (t^{(r)})^2(\sigma_k^2 + r - 1), \tag{24}$$

and, for $r < s$,

$$[G_{d,k}]_{rs} = t^{(s)}\left[t^{(r)}(\sigma_k^2 + r) - 1\right], \qquad [G_{d,k}]_{sr} = [G_{d,k}]_{rs}. \tag{25}$$

The posterior mixture weights are

$$\widetilde{w}_{d,k}\left(\mathbf{z}^{(1:d)}, \mathbf{t}\right) = \frac{w_k\, \mathcal{N}_d\left(\mathbf{z}^{(1:d)}; \mu_k \mathbf{t}, G_{d,k}\right)}{\sum_{\ell=1}^{K} w_\ell\, \mathcal{N}_d\left(\mathbf{z}^{(1:d)}; \mu_\ell \mathbf{t}, G_{d,\ell}\right)}. \tag{26}$$

The conditional mean and variance of component k are

$$m_{d,k}\left(\mathbf{z}^{(1:d)}, \mathbf{t}\right) = \mu_k + \mathbf{c}_{d,k}^\top G_{d,k}^{-1}\left(\mathbf{z}^{(1:d)} - \mu_k \mathbf{t}\right), \tag{27}$$

$$q_{d,k}(\mathbf{t}) = \sigma_k^2 + d - \mathbf{c}_{d,k}^\top G_{d,k}^{-1} \mathbf{c}_{d,k}. \tag{28}$$

Proof for Corollary 1

Introduce a mixture-component variable C such that

$$\mathbb{P}(C = k) = w_k, \qquad x_1 \mid C = k \sim \mathcal{N}(\mu_k, \sigma_k^2). \tag{29}$$

For a fixed component k, define

$$\mathbf{u}_d = \begin{bmatrix} x_0^{(1)} & \cdots & x_0^{(d)} & x_1 \end{bmatrix}^\top. \tag{30}$$

Then

$$\mathbf{u}_d \mid C = k \sim \mathcal{N}\Big(\begin{bmatrix} 0 & \cdots & 0 & \mu_k \end{bmatrix}^\top, \mathrm{diag}(1, \ldots, 1, \sigma_k^2)\Big). \tag{31}$$

Both the hierarchical states $\mathbf{x}_t^{(1:d)}$ and the depth-d target direction

$$f^{(d)} = x_1 - \sum_{j=1}^{d} x_0^{(j)} \tag{32}$$

are linear transformations of $\mathbf{u}_d$. Therefore,

$$\begin{bmatrix} f^{(d)} \\ \mathbf{x}_t^{(1:d)} \end{bmatrix} \Bigg| \; C = k \sim \mathcal{N}\left(\begin{bmatrix} \mu_k \\ \mu_k \mathbf{t} \end{bmatrix}, \begin{bmatrix} \sigma_k^2 + d & \mathbf{c}_{d,k}^\top \\ \mathbf{c}_{d,k} & G_{d,k} \end{bmatrix}\right). \tag{33}$$

In particular,

$$\mathrm{Var}(f^{(d)} \mid C = k) = \sigma_k^2 + d, \tag{34}$$

and direct calculation gives

$$\mathrm{Cov}\left(x_t^{(r)}, f^{(d)} \mid C = k\right) = t^{(r)}(\sigma_k^2 + r) - 1 = c_{r,k}. \tag{35}$$

Similarly,

$$\mathrm{Var}\left(x_t^{(r)} \mid C = k\right) = (1 - t^{(r)})^2 + (t^{(r)})^2(\sigma_k^2 + r - 1), \tag{36}$$

and, for $r < s$,

$$\mathrm{Cov}\left(x_t^{(r)}, x_t^{(s)} \mid C = k\right) = t^{(s)}\left[t^{(r)}(\sigma_k^2 + r) - 1\right]. \tag{37}$$

Applying the Gaussian conditioning formula yields

$$f^{(d)} \mid \mathbf{x}_t^{(1:d)} = \mathbf{z}^{(1:d)}, C = k \sim \mathcal{N}\Big(m_{d,k}(\mathbf{z}^{(1:d)}, \mathbf{t}), q_{d,k}(\mathbf{t})\Big). \tag{38}$$

Moreover,

$$\mathbf{x}_t^{(1:d)} \mid C = k \sim \mathcal{N}_d\left(\mu_k \mathbf{t}, G_{d,k}\right). \tag{39}$$

Bayes' rule gives the posterior mixture weights $\widetilde{w}_{d,k}$, and marginalizing over C gives the stated conditional direction distribution.

Depth-one special case. For $d = 1$, the HRF state and target direction reduce to

$$x_t^{(1)} = (1-t)x_0^{(1)} + tx_1, \qquad f^{(1)} = x_1 - x_0^{(1)}. \tag{40}$$

Thus, $f^{(1)}$ is exactly the "ground-truth" velocity of standard rectified flow. Defining

$$\widetilde{\sigma}_{k,t}^2 = (1-t)^2 + t^2\sigma_k^2, \tag{41}$$

we obtain

$$p\Big(f^{(1)} \mid x_t^{(1)} = x, t\Big) = \sum_{k=1}^{K} \widetilde{w}_{k,t}(x)\mathcal{N}\left(f^{(1)}; \frac{(1-t)(\mu_k - x) + t\sigma_k^2 x}{\widetilde{\sigma}_{k,t}^2}, \frac{\sigma_k^2}{\widetilde{\sigma}_{k,t}^2}\right), \tag{42}$$

where

$$\widetilde{w}_{k,t}(x) = \frac{w_k \mathcal{N}\left(x; t\mu_k, \widetilde{\sigma}_{k,t}^2\right)}{\sum_{\ell=1}^{K} w_\ell \mathcal{N}\left(x; t\mu_\ell, \widetilde{\sigma}_{\ell,t}^2\right)}. \tag{43}$$

Hence, the depth-one case recovers the velocity distribution in Corollary 1 in (Zhang et al., 2025).

B.2 Posterior concentration across depth.

We present our proof for Corollary 2 as follows.

Proof for Corollary 2

For each mixture component k, the posterior weight at depth d is

$$\widetilde{w}_{d,k} = \mathbb{P}\Big(C = k \mid \mathbf{X}_t^{(1:d)}\Big). \tag{44}$$

At depth $d+1$,

$$\widetilde{w}_{d+1,k} = \mathbb{P}\Big(C = k \mid \mathbf{X}_t^{(1:d)}, X_t^{(d+1)}\Big). \tag{45}$$

By the tower property,

$$\mathbb{E}\Big[\widetilde{w}_{d+1,k} \mid \mathbf{X}_t^{(1:d)}\Big] = \widetilde{w}_{d,k}. \tag{46}$$

Define the posterior concentration

$$\mathcal{C}_d = \|\widetilde{\mathbf{w}}_d\|_2^2 = \sum_{k=1}^{K} \widetilde{w}_{d,k}^2. \tag{47}$$

Since $\widetilde{\mathbf{w}}_d$ is a probability vector,

$$\frac{1}{K} \le \mathcal{C}_d \le 1, \tag{48}$$

where the lower bound is attained for uniform posterior weights and the upper bound is attained for a one-hot posterior vector.

Using Eq. equation 46,

$$\mathbb{E}\Big[\widetilde{w}_{d+1,k}^2 \mid \mathbf{X}_t^{(1:d)}\Big] = \widetilde{w}_{d,k}^2 + \mathrm{Var}\Big(\widetilde{w}_{d+1,k} \mid \mathbf{X}_t^{(1:d)}\Big). \tag{49}$$

Summing over k gives

$$\mathbb{E}\Big[\mathcal{C}_{d+1} \mid \mathbf{X}_t^{(1:d)}\Big] = \mathcal{C}_d + \sum_{k=1}^{K} \mathrm{Var}\Big(\widetilde{w}_{d+1,k} \mid \mathbf{X}_t^{(1:d)}\Big) \ge \mathcal{C}_d. \tag{50}$$

Taking expectation over $\mathbf{X}_t^{(1:d)}$ yields

$$\mathbb{E}[\mathcal{C}_{d+1}] \ge \mathbb{E}[\mathcal{C}_d]. \tag{51}$$

Hence, the posterior mixture weights become increasingly concentrated with hierarchy depth in expectation. The inequality is strict whenever

$$\mathbb{P}\left(\sum_{k=1}^{K} \mathrm{Var}\left(\widetilde{w}_{d+1,k} \mid \mathbf{X}_t^{(1:d)}\right) > 0\right) > 0, \tag{52}$$

that is, whenever the additional state $X_t^{(d+1)}$ changes the posterior distribution over the latent mixture component with positive probability.

C Detailed Experiment Settings

In this section, we provide additional implementation and experimental details for the results in Section 4. We largely follow the architectures, optimization settings, and evaluation protocol of (Zhang et al., 2025), while incorporating the depth-specific predictors and adaptive stopping mechanism introduced by MoD-RF. Unless otherwise stated, all methods use the same data preprocessing and evaluation protocol within each experiment. For all experiments, we set $\delta = 0.5$.

C.1 Synthetic Experiments

Architecture. For the 1D and 2D experiments, we follow the low-dimensional architecture used in HRF. Each depth-d predictor is implemented as an independent network, rather than as a shared backbone evaluated with a masked hierarchy prefix. The hierarchical states $\mathbf{x}_t^{(1:d)}$ and time variables $\mathbf{t}^{(1:d)}$ are first processed separately. Each input is mapped through a sinusoidal positional embedding followed by a linear - GELU - linear projection. The resulting state and time representations are then concatenated and passed through three fully connected layers to predict the depth-d target direction. Because each hierarchy depth has its own independently parameterized predictor, the model capacity increases with d. For the 1D experiments, HRF-1/2/3 contain 82,689 / 304,513 / 673,793 parameters, respectively. For the 2D experiment, they contain 86,914 / 321,154 / 711,042 parameters.

Depth Selector. MoD-RF augments these depth-specific predictors with a causal Transformer hazard model described in Section 3.2. At each hierarchy level, the corresponding state representation is combined with its time information and a learned depth embedding to form a depth token. The token sequence is processed causally along the hierarchy so that the stopping decision at depth d depends only on information available up to that depth. We use a 3-layer, 8-head Transformer with feed-forward width 256 and learned depth embeddings. For the 1D experiments, the selector uses hidden width 64 and contains 154,561 parameters, while for the 2D experiment it uses hidden width 128 and contains 414,977 parameters.

Data configurations. For the 1D experiments, we consider the two source-target configurations described in Section 4.1: a standard Gaussian source transported to a mixture of five Gaussians, and a mixture of two Gaussians transported to another mixture of two Gaussians. For the 2D experiment, the source distribution is a mixture of eight Gaussians and the target distribution is the two-moons distribution.

Training. Following HRF, we generate $100{,}000$ samples for training and use another $100{,}000$ samples for evaluation for each synthetic configuration. All synthetic HRFs are trained for $50{,}000$ iterations with a batch size of 1024. We train multiple models with independent random seeds and report results averaged across runs. The source variables introduced at different hierarchy depths are sampled independently from the corresponding source distribution.

For MoD-RF, we reuse trained HRF baselines. The depth selector is trained using the likelihood-weighted velocity-reconstruction objective in Eq. equation 15, with $\sigma_d = 0.5$ for $10{,}000$ iterations with batch size of 1024. No ground-truth depth labels are used.

Inference and NFE accounting. We use Euler integration for all synthetic experiments. Following HRF, the computational cost is measured by the total number of neural function evaluations (NFEs) across all hierarchy levels. For a fixed depth-d HRF with N_r integration steps at hierarchy level r, the resulting nested integration requires

$$\mathrm{NFE}_d = \prod_{r=1}^{d} N_r$$

predictor evaluations. For example, for HRF2, the step combinations

$$(1, 5), \quad (2, 5), \quad (5, 10), \quad (10, 10)$$

correspond to total NFE budgets of 5, 10, 50, and 100, respectively.

MoD-RF instead determines the terminal hierarchy depth independently at each outer integration step. Once a depth is selected, the depth and sampled hierarchical source states are locked during the corresponding nested solve, as detailed in Algorithm 1 and Appendix A. We count every evaluation of a depth-specific predictor within the nested solve toward the total NFE, so that MoD-RF and the fixed-depth baselines are compared using the same computational measure.

NFE Scaling. We fix the number of inner steps at each depth and scale computation by increasing the number of outer, depth-one steps. This keeps the nested ODE solver unchanged across budgets, provides the same depth-specific integration schedule for both the fixed-depth baselines and MoD-RF, and avoids extensive budget-specific tuning. This choice is especially appropriate for MoD-RF, since the selected depth can vary across samples and outer steps, making per-budget step allocation difficult to define consistently. We still report the total number of function evaluations across all hierarchy levels, so all methods remain directly comparable under matched NFE budgets.

C.2 Image Experiments

Architecture. For MNIST, CIFAR-10, and ImageNet-32, we follow the U-Net family used in HRF. Each depth-d predictor is implemented as a single network that jointly processes all active hierarchy levels. Specifically, the states from the d levels are concatenated along the channel dimension, ordered from deepest to shallowest, before the first convolution. The corresponding time variables are not shared: each hierarchy level receives its own time embedding, which is injected into the network through the residual blocks. As a result, the effective input width grows with hierarchy depth while preserving level-specific temporal conditioning.

For MNIST, we use a U-Net with channel multipliers $(1, 2, 2)$, one residual block per resolution, and attention at resolution 16. The base channel width is 32, giving 1.08M, 4.25M, and 9.51M parameters for HRF-1, HRF-2, and HRF-3, respectively. For CIFAR-10, we add one downsampling stage and use channel multipliers $(1, 2, 2, 2)$, again with one residual block per resolution and attention at resolution 16. With the same base width of 32, HRF-1/2/3 contain 1.53M / 6.04M / 13.55M parameters. For ImageNet-32, we use the same channel multipliers and attention resolution as CIFAR-10, but increase the number of residual blocks to two per resolution. The corresponding HRF-1/2/3 models contain 2.25M / 8.91M / 20.01M parameters. MoD-RF uses these same depth-specific backbones, with each predictor receiving the hierarchical states and time variables available up to its corresponding depth.

Depth Selector. Per-sample depth selection is performed using a causal Transformer hazard model. For each hierarchy level, a small encoder maps the corresponding state to a token; for image inputs, this encoder consists of strided convolutional layers. A level-specific time embedding and a learned depth embedding are then added to each token. The resulting token sequence is processed by a 5-layer, 8-head Transformer encoder with hidden width 128 and a causal mask over hierarchy depth, ensuring that the stopping decision at depth d depends only on information available up to that depth. An MLP head outputs one continuation logit for each level. The selector architecture is shared across MNIST, CIFAR-10, and ImageNet-32 and contains approximately 1.10M parameters in all cases: 1,099,265 parameters for MNIST and 1,100,417 parameters for CIFAR-10 and ImageNet-32, with the difference arising only from the input-channel dimension of the first encoder layer.

Large Model Variant. For the ImageNet-32 scaling ablation, we enlarge only the depth-1 RF backbone while keeping the deeper correction network unchanged. Specifically, the base channel width of the depth-1 model is increased from 32 to 128, increasing its parameter count from 2.25M to 35.75M, corresponding to a $15.9\times$ increase. The depth-2 correction network remains identical to the standard configuration, with base width 32 and 8.91M parameters. Thus, the Large variant strengthens the underlying RF predictor without uniformly scaling the entire hierarchy. This design allows us to test whether the benefit of adaptive hierarchical modeling persists when MoD-RF is built on top of a substantially stronger depth-1 baseline.

Optimization. We follow the optimization settings of HRF. All image models are trained using Adam with

$$\beta_1 = 0.9, \qquad \beta_2 = 0.999, \qquad \epsilon = 10^{-8},$$

and no weight decay. For MNIST, the U-Net channel multipliers are $[1, 2, 2]$, while CIFAR-10 and ImageNet-32 use $[1, 2, 2, 2]$.

For MNIST and CIFAR-10, we use a learning rate of 1×10^{-4} and a batch size of 128. For ImageNet-32, we use a learning rate of 2×10^{-4} and a batch size of 512. All datasets are trained for $100{,}000$ iterations for HRF baselines and $50{,}000$ iterations for the depth selector. We use the same training protocol for all compared methods within each dataset.

Image sampling. Generation quality is evaluated using Fréchet Inception Distance (FID). As in the synthetic experiments, we report the total number of predictor evaluations over all nested hierarchy levels as the NFE budget.

NFE Scaling. We fix the number of inner steps at each depth and scale computation by increasing the number of outer, depth-one steps. This keeps the nested ODE solver unchanged across budgets, provides the same depth-specific integration schedule for both the fixed-depth baselines and MoD-RF, and avoids extensive budget-specific tuning. This choice is especially appropriate for MoD-RF, since the selected depth can vary across samples and outer steps, making per-budget step allocation difficult to define consistently. We still report the total number of function evaluations across all hierarchy levels, so all methods remain directly comparable under matched NFE budgets.

For MoD-RF, adaptive depth selection is performed separately for each sample and each outer integration step. Consequently, different samples may incur different hierarchical computation within the same outer step. We account for all depth-specific predictor evaluations when reporting the NFE of MoD-RF and compare against RF and HRF at matched average inference budgets.

D Qualitative Results

We present qualitative results of our model trained on CIFAR-10 data in Figure 7.

E Limitations

MoD-RF improves the allocation of inference computation by adapting hierarchy depth to local target-direction ambiguity, but it has several limitations. First, the method introduces additional components beyond standard rectified flow, including deeper hierarchical predictors and a learned depth selector, which increase training complexity and memory usage. Second, the stopping mechanism is trained using reconstruction-based supervision rather than direct labels of the optimal hierarchy depth. While this avoids requiring explicit depth annotations, the selected depth can depend on the quality and calibration of the reconstruction signal. Third, adaptive depth is most beneficial when sufficient computation is available to make meaningful choices across hierarchy levels. In very low-NFE regimes, the overhead of adaptive selection may provide limited benefit compared with a shallow fixed-depth model. Finally, our experiments are limited to synthetic distributions and image-generation benchmarks up to ImageNet-32. It remains to be studied how well the proposed strategy scales to substantially larger models, higher-resolution generation, and other modalities, as well as whether alternative stopping objectives can further improve the quality-compute trade-off.

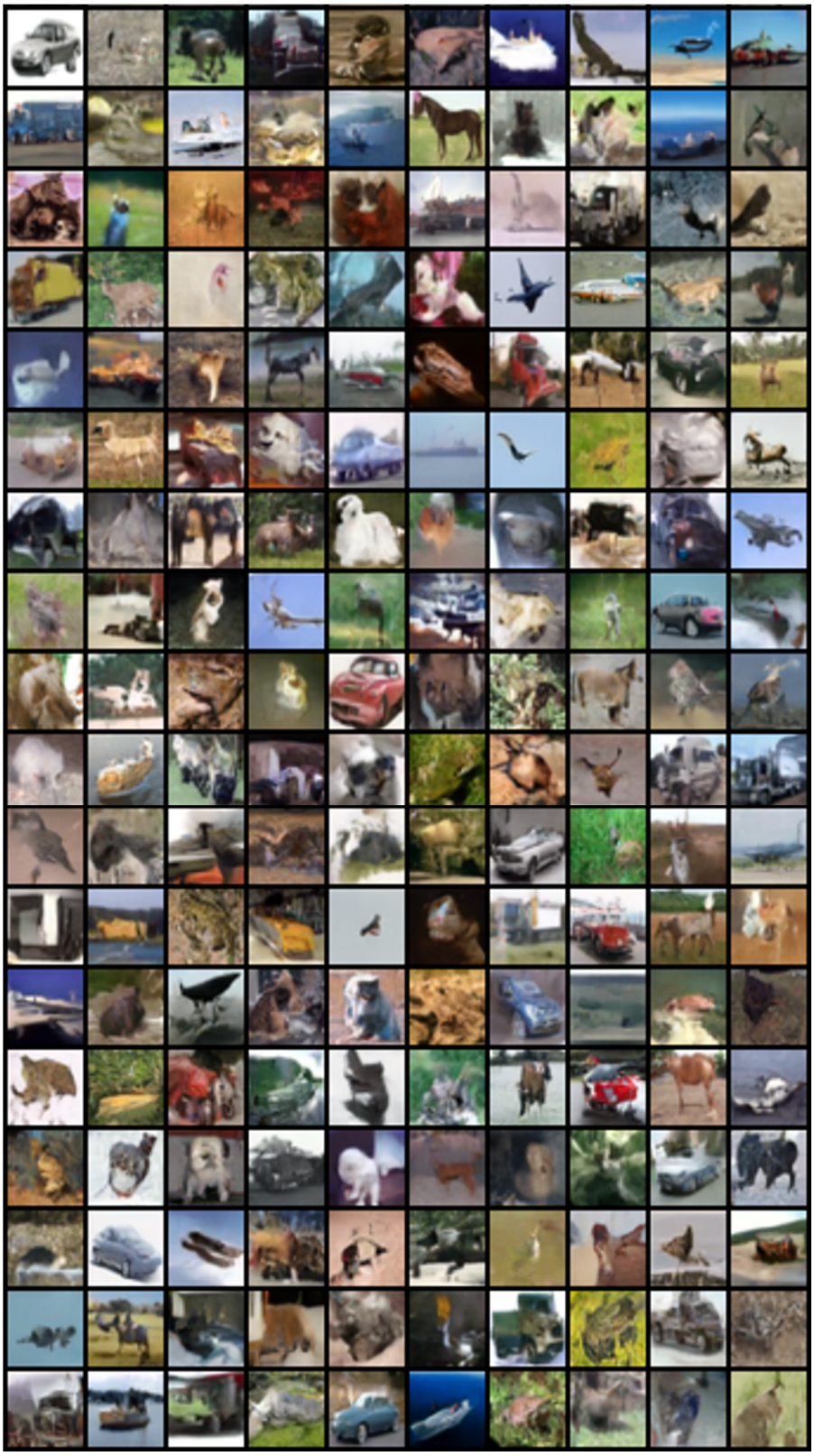


Figure 7: Random selected samples generated from our model trained on CIFAR-10 data