Heterogeneous Decentralized Diffusion Models

Supplementary Material

Zhiying Jiang Raihan Seraj Marcos Villagra Bidhan Roy

Bagel Labs

{gin, raihan, marcos, bidhan}@bagel.com

1. Training Details

1.1. Data Preprocessing and Clustering

Dataset. We train on the LAION-Aesthetics subset. For DiT-B/2, we utilize LAION-Art whose aesthetic score is ≥ 8, containing around 3.9M image-text pairs. For DiT-XL/2, we filter LAION-Aesthetic for aesthetic score ≥ 4.5 and resolution $\geq 256 \times 256$. Images are center-cropped to square aspect ratio and resized to 256×256 pixels before VAE encoding.

Feature Extraction. We extract semantic features using the pretrained DINOv2-ViT-L/14 model [7], which outputs 1024-dimensional embeddings for each image. Features are computed from the [CLS] token of the final layer without finetuning.

Hierarchical Clustering. We apply hierarchical k-means clustering with $K = 8$ clusters using cosine distance as the similarity metric. The clustering is performed in two stages: first partitioning into 1024 fine-grained groups using standard k-means, then grouping them into 8 coarse clusters. This produces semantically coherent partitions (e.g., portraits, landscapes, architecture, abstract art, animals). Each image in the dataset is then assigned to its nearest cluster based on DINOv2 features.

Latent Encoding. All images are encoded using the pretrained VAE encoder from Stable Diffusion [9], producing $32 \times 32 \times 4$ latent representations with scaling factor 0.18215. We pre-calculated the encoded latents and save them to disk to avoid redundant encoding during training.

1.2. Expert Training

Architecture. Each expert uses the DiT-XL/2 architecture with 28 transformer blocks, hidden dimension 1152, 16 attention heads. Text conditioning uses frozen CLIP-ViT-L/14 text encoder (768-dimensional embeddings, maximum 77 tokens).

Objective Assignment. In the experiment of homogeneous experts, we assign Flow Matching objectives to all $K = 8$ experts. For heterogeneous training, we assign 2 experts to DDPM (ϵ-prediction) and 6 experts to Flow Matching (velocity prediction). We specifically assign DDPM experts to cluster 0 and cluster 3 as they contain high-fidelity subjects like cars and flowers. DDPM experts use cosine noise schedule [6]. Flow Matching experts use linear interpolation schedule $x_t = (1 - t)x_0 + t\epsilon$ with $t \in [0, 1]$.

Initialization. We initialize all experts from the same pretrained ImageNet DiT-XL/2 checkpoint trained with DDPM objective [8]. Following our conversion procedure (Sec. 3.5 of the main paper), we transfer patch embeddings, timestep embeddings, and per-block self-attention and FFN weights. The final linear layer is loaded with variance-prediction channels truncated (DiT uses `learn_sigma`). Per-block and final-layer modulation tables are reparameterized from DiT's adaLN MLPs at a reference timestep (t=500). Positional embeddings use fixed sinusoidal encoding following PixArt-α [1] and are not loaded. The AdaLN-Single $\text{MLP}_{\text{global}}$ and text projection are initialized with $\mathcal{N}(0, 0.02)$; cross-attention output projections are zero-initialized. For Flow Matching experts, we implement runtime timestep scaling $t_{\text{DiT}} = 999 \cdot t$ to maintain compatibility with the pretrained sinusoidal timestep encoding, which naturally handles continuous-valued inputs.

Optimization. Each expert trains independently with no communication:

- **Optimizer:** AdamW with $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$
- **Learning rate:** 1×10^{-4}
- **Weight decay:** 0.0 (following Chen et al. [1])
- **Warmup:** Linear warmup over first 5,000 steps
- **Batch size:** 128 per expert
- **Training steps:** 500,000 steps per expert
- **Gradient clipping:** Max norm 1.0
- **Mixed precision:** We use mixed precision training (FP16) with gradient scaling and TF32 acceleration on

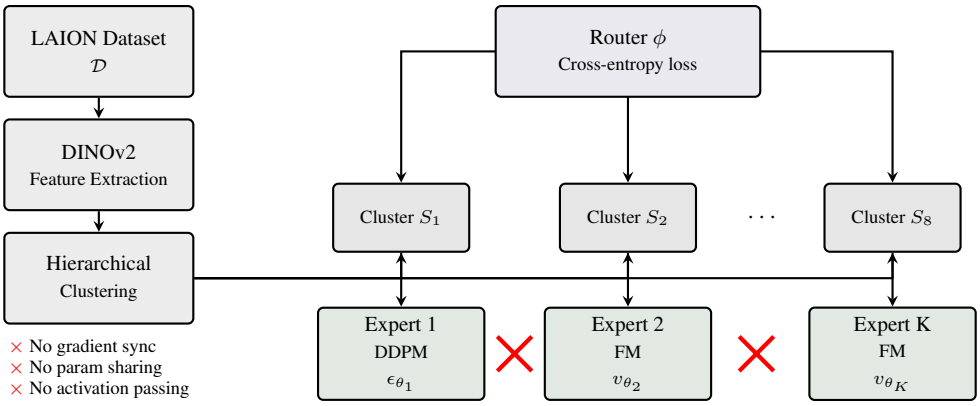


Figure 1. **Training Pipeline for Decentralized Heterogeneous Experts.** LAION dataset $\mathcal{D}$ is partitioned into K semantic clusters $\{S_1, S_2, \ldots, S_K\}$ using DINOv2 feature extraction and hierarchical clustering. Each expert trains independently on its assigned cluster with heterogeneous objectives: DDPM experts predict noise $\epsilon_{\theta_k}(x_t, t)$ while Flow Matching experts predict velocity $v_{\theta_k}(x_t, t)$. The router network ϕ trains on all data to predict cluster assignments via cross-entropy loss. Crucially, there is zero gradient synchronization, parameter sharing, or activation passing between experts during training.

NVIDIA A40 GPUs.

Exponential Moving Average. We maintain EMA weights with decay $\mu = 0.9999$ for generation, updating after each training step: $\theta_{\text{EMA}} \leftarrow \mu\theta_{\text{EMA}} + (1-\mu)\theta$.

Classifier-Free Guidance. During training, we randomly drop text conditioning with probability $p_{\text{cfg}} = 0.1$, replacing text embeddings with null embeddings obtained by encoding the empty string through the frozen CLIP text encoder ($\mathbf{e}_\varnothing \in \mathbb{R}^{77\times768}$). This enables classifier-free guidance [2] during inference.

Numerical Stability. For DDPM-to-velocity conversion at inference, we clamp predicted clean latents $\hat{x}_0$ to $[-20, 20]$ and apply adaptive velocity scaling $s(t)$ that dampens converted predictions at elevated noise levels (0.88 for $t > 0.85$, 0.93 for $0.6 < t \leq 0.85$, 0.96 for $t \leq 0.6$; see Section 3.3 for details).

1.3. Router Training

Architecture. The router uses DiT-B/2 architecture with 12 transformer blocks, hidden dimension 768, 12 attention heads, and 129M parameters. Unlike experts, the router is trained from scratch without text conditioning, processing only noisy latents x_t and timestep t.

Training Data. The router trains on the full LAION dataset (all clusters combined) with ground-truth cluster assignments from the clustering stage serving as labels. Each training sample (x_0, k) consists of a clean latent and its cluster ID $k \in \{1, \ldots, K\}$.

Optimization. Router training hyperparameters:

- **Optimizer:** AdamW with $\beta_1 = 0.9$, $\beta_2 = 0.999$
- **Learning rate:** 5×10^{-5} with cosine annealing to 5×10^{-7}
- **Weight decay:** 1×10^{-2}
- **Warmup:** None (0 steps)
- **Batch size:** 64 per GPU with gradient accumulation of 4 (effective batch: 256 per GPU)
- **Training:** 25 epochs
- **Loss:** Cross-entropy $\mathcal{L}_{\text{router}} = -\log p_\phi(k|x_t, t)$

Timestep Sampling. During router training, all samples are noised using the Flow Matching interpolation path with $t \sim \mathcal{U}(0, 1)$, regardless of their cluster's assigned training objective. This matches the inference-time convention where the denoising trajectory always operates in $t \in [0, 1]$. Timesteps are scaled to the DiT range via $t_{\text{DiT}} = 999 \cdot t$ before being fed to the router's backbone.

1.4. Computational Resources

Hardware. All experiments use NVIDIA A40 48GB GPUs, 1 GPU per expert. Router training uses 1 24GB GPU.

Training Time. With batch size 128 and 500K steps, total training requires approximately 120 A40 GPU-days across all 8 experts and router training (approximately 72 A100-equivalent GPU-days when normalized by measured FP16 training throughput). When fully parallelized across 8 GPUs, wall-clock time is approximately 15 days.

1.5. Convergence with Pretrained Initialization

Figure 2 shows validation loss curves comparing pretrained checkpoint initialization against training from scratch. The

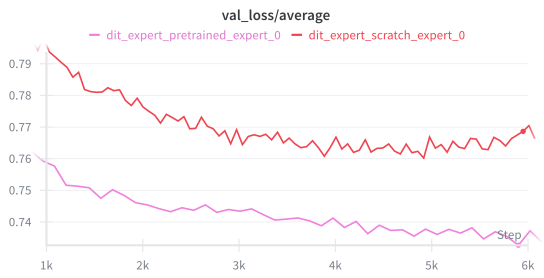


Figure 2. **Validation Loss: Pretrained vs. Scratch Initialization.** Average validation loss over early training steps for a Flow Matching expert initialized from a converted ImageNet-DDPM checkpoint versus training from scratch. Pretrained initialization yields 1.2× faster loss reduction (0.030 vs. 0.025 drop over 5K steps in the steady-improvement regime) and consistently lower loss throughout training.

pretrained expert starts at a substantially lower loss and improves 1.2× faster during steady-state training, demonstrating effective transfer of visual priors across diffusion objectives.

2. Additional Qualitative Analysis

In this section, we provide extensive qualitative results to demonstrate the capabilities of our heterogeneous decentralized diffusion framework. All images are generated at 256×256 resolution with 75 Euler sampling steps and CFG scale 6. Importantly, all text prompts used for generation are either from a held-out test set or synthetically generated by large language models, ensuring they were unseen during training. The samples showcase the diversity and quality achieved by our system trained with mixed DDPM and Flow Matching objectives across 8 specialized experts.

2.1. Diverse Generation Examples

Figures 3–6 present a wide variety of generated samples demonstrating our framework's versatility across different semantic categories, styles, and subjects. The samples shown here are generated from three representative configurations: (1) eight Flow Matching experts, (2) one DDPM expert plus seven Flow Matching experts, and (3) two DDPM experts plus six Flow Matching experts. All configurations demonstrate consistent high-quality generation, validating that our framework maintains visual fidelity across different objective mixtures while achieving significant computational efficiency gains.

2.2. Heterogeneous vs. Homogeneous Objectives

To validate the effectiveness of mixing different diffusion objectives, we compare our heterogeneous approach (combining DDPM and Flow Matching experts) against a homogeneous baseline where all experts use the same Flow Matching objective. Figures 7 and 8 show side-by-side comparisons on identical text prompts across diverse semantic categories. Our heterogeneous framework achieves comparable or superior visual quality in these examples. The results demonstrate that mixing objectives does not compromise generation fidelity and may even improve diversity and detail coherence by leveraging complementary strengths of different formulations.

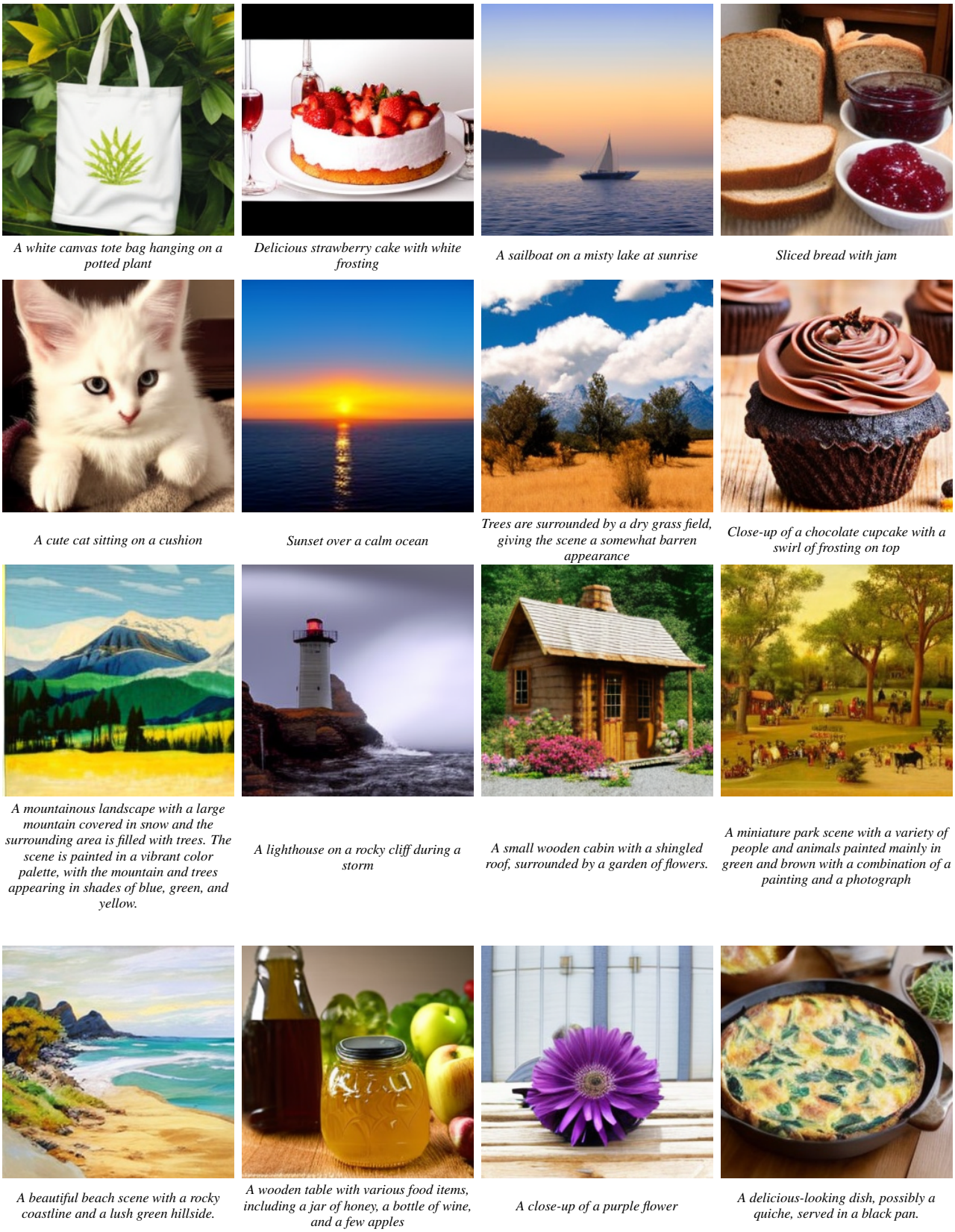


Figure 3. **Diverse Generation Examples (Set 1).** Representative samples demonstrating our framework's ability to generate high-quality, diverse images across multiple semantic categories. Images are produced by our heterogeneous ensemble combining DDPM and Flow Matching objectives.

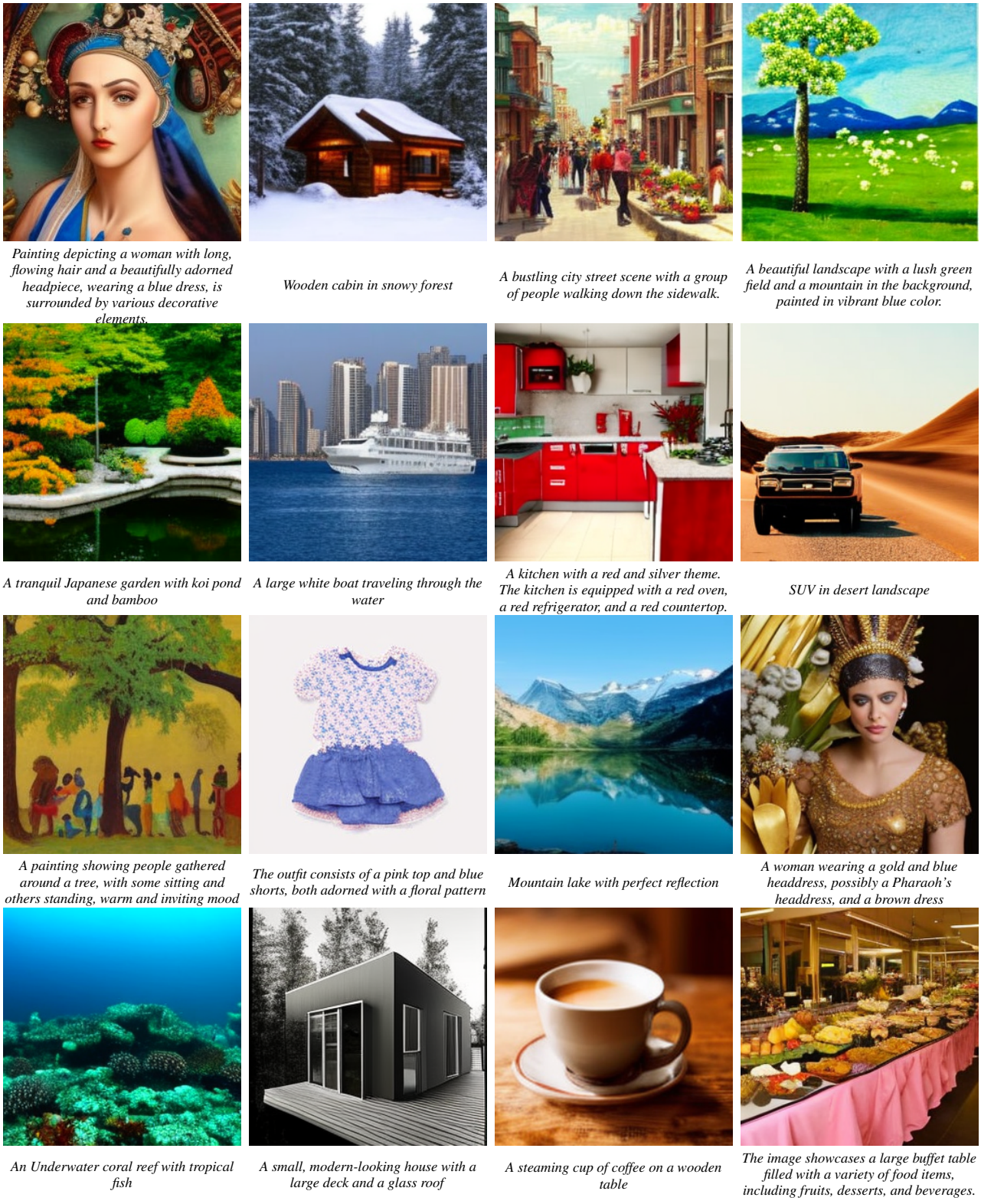
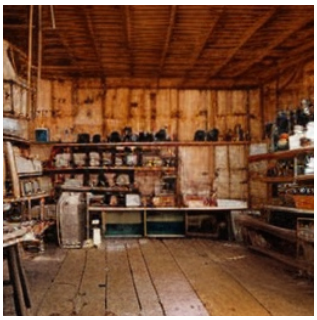


Figure 4. **Diverse Generation Examples (Set 2).** Additional samples showcasing consistent generation quality across various prompts and content types.



A rustic kitchen with wooden walls and a dirt floor.

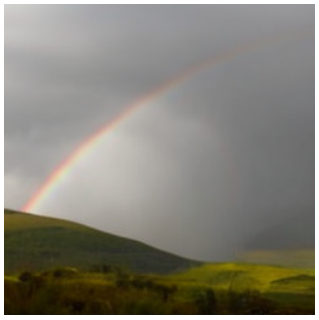
A snowy owl perched on a frost-covered branch

A serene scene of a grassy field with a hill in the background, filled with a variety of flowers, impressionistic art style

A serene scene of a tree-lined pathway overlooking the ocean, with predominantly green, blue, and white color, creating a calm and peaceful atmosphere

Wolf standing in the darkness.

A beautifully decorated Christmas tree in a living room and the tree is adorned with numerous red ornaments



A rainbow appearing after rainfall over hills

A cozy and colorful living space, featuring a small kitchen and a living area.

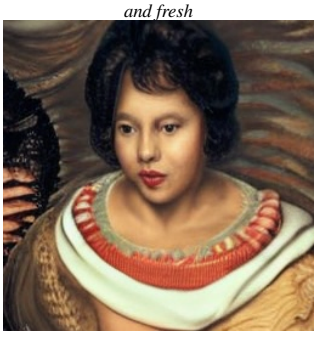
A small red house with white trim, surrounded by greenery. The house has a white door and a window, and it is adorned with a variety of potted plants, flowers, and vases.

A delicious dessert in a glass bowl, consisting of layers of cake and strawberries; the cake is white and creamy, while the strawberries are red and fresh

A picturesque scene of a tent pitched on a rocky hillside, overlooking a serene lake

A serene scene of a mountain lake surrounded by lush greenery, where the lake is filled with rocks

A bird showing off its beautiful feather

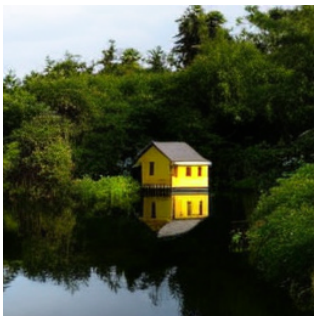


Realistic portrait of a young woman with soft, natural lighting, wearing a detailed red and white textured neckline, reminiscent of Renaissance or Baroque portraiture

A desert oasis with palm trees

A cozy, old-fashioned kitchen with a wooden table and chairs where the room is painted in a light green color and the table is set with bowls and spoons

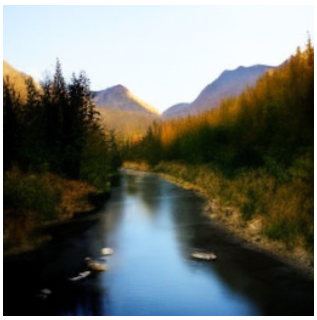
Figure 5. **Diverse Generation Examples (Set 3).** Further examples demonstrating the robustness and versatility of our heterogeneous framework.



A small yellow house situated on a hillside, overlooking a body of water

Sandwich with macaroni

A small wooden cabin with a shingled roof, situated in a forest. The cabin is surrounded by trees, creating a serene and peaceful atmosphere. The cabin has a porch with a bench

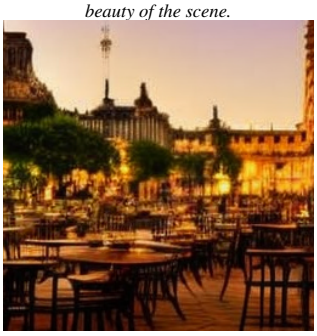


A serene landscape with a river flowing through a valley, surrounded by trees and mountains. The river is the main subject, with its calm waters reflecting the natural beauty of the scene.

A watercolor style painting illustrating a charming cottage with a garden setting, surrounded by a variety of flowers, including pink and purple ones.

A cozy library with tall bookshelves and warm lighting

A blue sports car parked in a dark parking lot

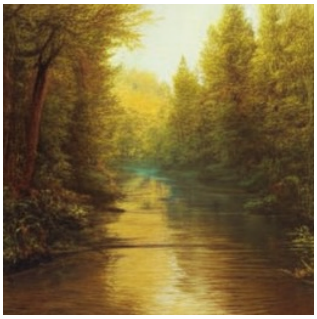


A cafe terrace in Paris at evening

A lively scene of numerous hot air balloons floating in the sky and are spread across the entire sky and a crowd of people gathered on the ground, watching the spectacle and enjoying the event

An impressionistic painting describing a bustling city scene with a large crowd of people walking. The cityscape features a mix of architectural styles, including a prominent cathedral and a castle-like structure

The impressionistic painting depicts a serene scene of a forest with a mix of trees and bushes. The trees are predominantly orange, creating a warm and vibrant atmosphere.



A painting depicting serene scene of a small stream flowing through a forest, with natural and peaceful mood

A rare blue poppy blooming among rocks

A brown teddy bear wearing a brown bow tie

a white teepee tent with a lacey canopy, set up in a grassy area, adorned with a white flower, adding a touch of elegance to the scene

An off-road rally car splashing through muddy terrain

Figure 6. **Diverse Generation Examples (Set 4).** Additional samples showcasing the framework's ability to generate high-quality images across various prompts and configurations.

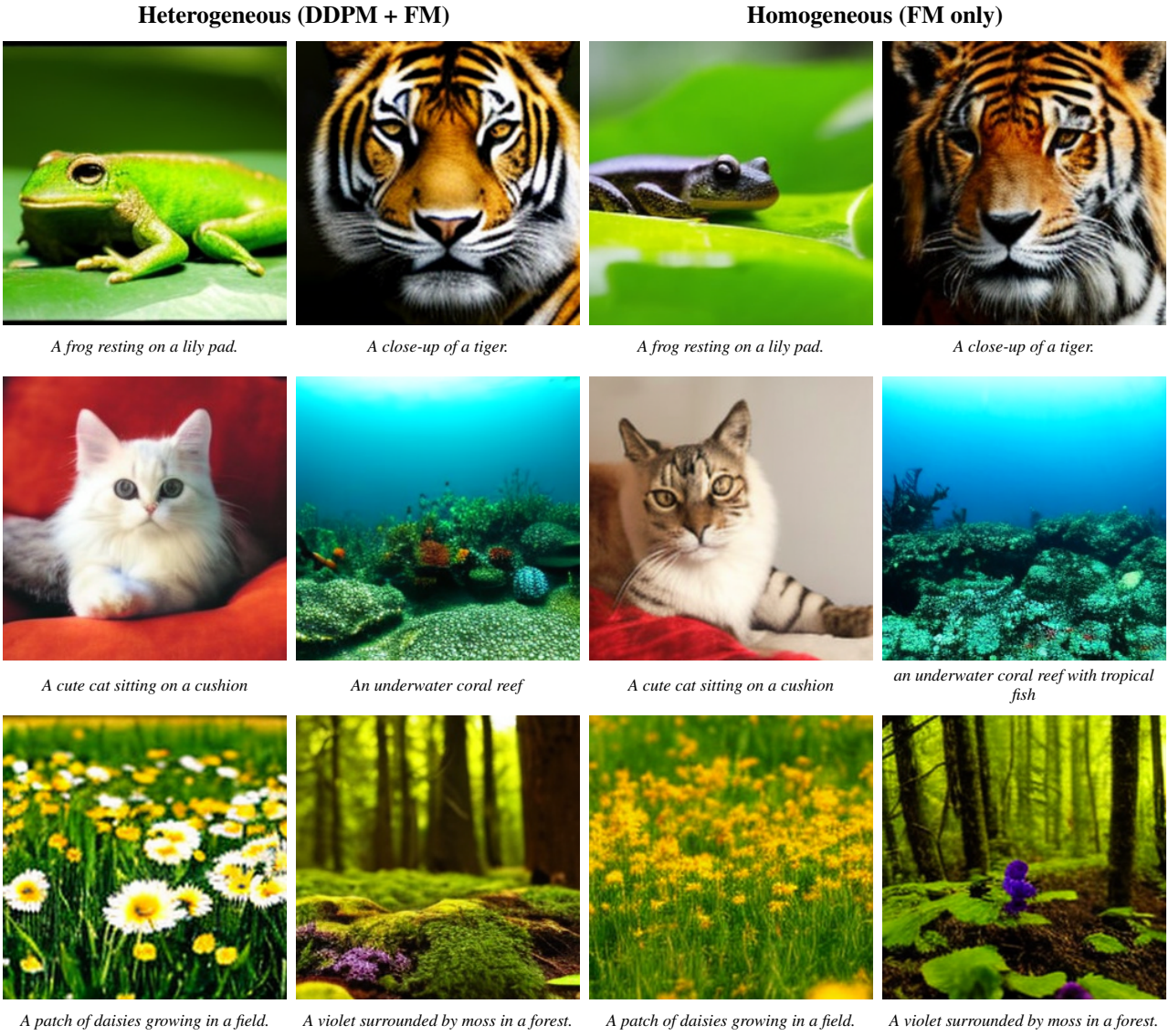


Figure 7. **Heterogeneous vs. Homogeneous Comparison (Part 1).** Direct side-by-side comparison showing that our heterogeneous approach (columns 1-2) maintains or improves visual quality compared to homogeneous baseline (columns 3-4) across diverse categories including animals and nature.

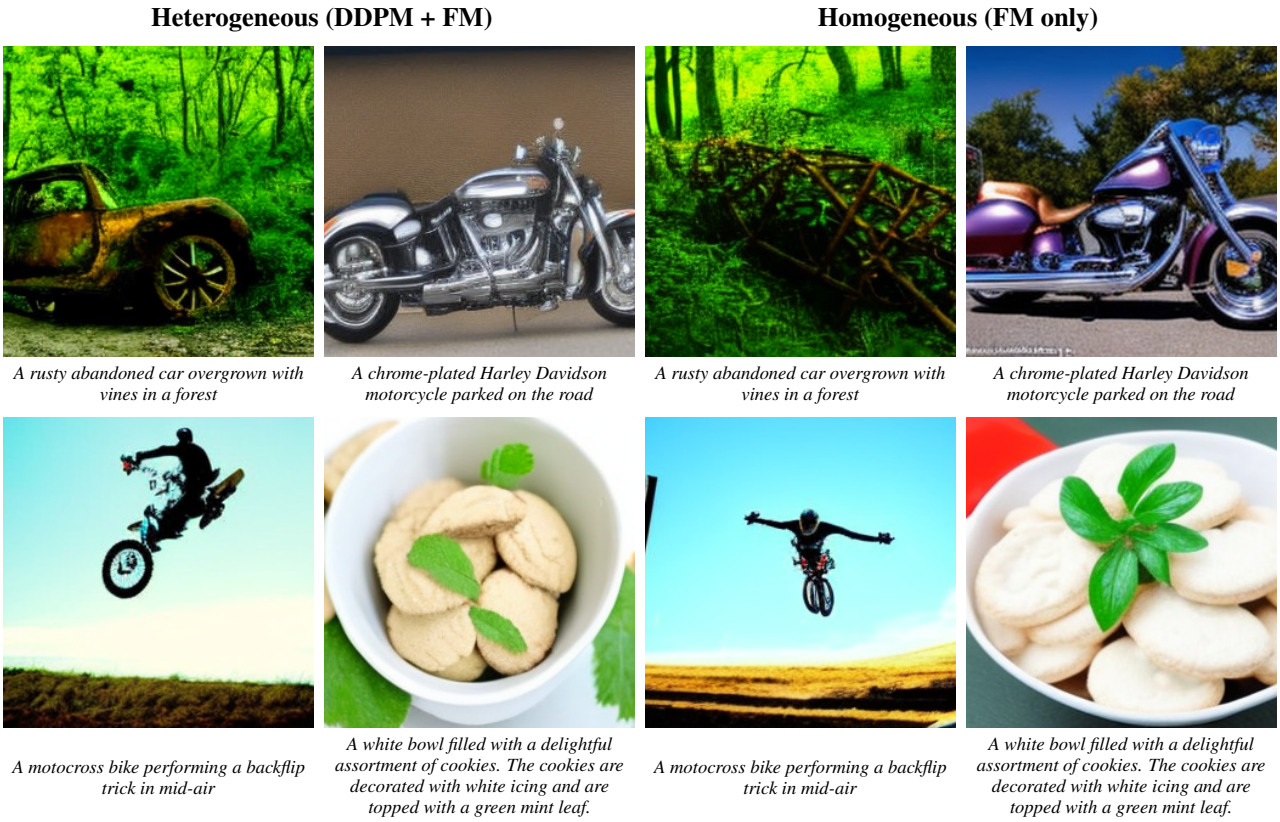


Figure 8. **Heterogeneous vs. Homogeneous Comparison (Part 2).** Continued comparison across vehicles and food categories, demonstrating consistent quality maintenance across different semantic domains.

2.3. Effects of Expert Selection and Router Thresholds

Routing threshold sweep. We examine how the routing threshold t affects the transition point between DDPM and FM experts, evaluating on 1,000 held-out samples using converted DDPM and native FM experts under the same cosine schedule. The threshold t determines the transition: for timesteps $t' \leq t$ the DDPM expert is used, while for $t' > t$ the FM expert is used.

Figure 9 reveals a clear quality–diversity trade-off. Threshold 0.2 achieves optimal FID (38.28) with FM-dominated denoising but lower LPIPS diversity. Threshold 0.5 produces the highest LPIPS diversity with balanced workload but elevated FID. Lower values (0.2–0.3) favor quality, while mid-range values (0.4–0.5) favor diversity.

Expert ordering and threshold interaction. When combining heterogeneous experts in a 2-expert configuration (1 converted DDPM + 1 Flow Matching), both the order of expert application and the router confidence threshold significantly impact generation quality. Figure 10 presents a comparison on identical prompts (sunset scenes) under a unified schedule, varying: (1) expert ordering (DDPM→FM vs. FM→DDPM), and (2) router confidence threshold $\tau \in \{0.3, 0.5, 0.7\}$.

The results reveal a striking asymmetry between the two orderings. The FM→DDPM configuration (bottom row) produces cleaner, more coherent images with smooth gradients and well-defined structures across all threshold values. In contrast, DDPM→FM ordering (top row) exhibits visible quality degradation, particularly at higher thresholds ($\tau = 0.7$), where blocky artifacts and oversaturation appear in the sky regions. At lower thresholds ($\tau = 0.3$), DDPM→FM recovers somewhat by allowing earlier transition to the native FM expert, though still showing less refinement than the FM→DDPM counterpart.

This asymmetry highlights a practical limitation of epsilon-to-velocity conversion at different noise levels. When DDPM operates first (handling high noise levels where $\alpha_t \to 0$), the conversion formula $\hat{x}_0 = (x_t - \sigma_t \epsilon_\theta)/\alpha_t$ becomes numerically unstable. Our clamping and scaling safeguards (Section 3) introduce systematic biases that manifest as blocky artifacts and color distortions. Critically, these errors occur early in the reverse diffusion process and become "baked into" the emerging image structure, which the subsequent low-noise FM expert cannot fully correct. Conversely, when native FM handles the high-noise phase first, it establishes a clean structural foundation without conversion artifacts. The converted DDPM expert then refines this foundation at low noise levels ($\alpha_t \approx 1$), where the conversion is numerically stable and introduces minimal bias.

The threshold parameter τ controls when the router switches between experts during the denoising trajectory. Lower thresholds ($\tau = 0.3$) favor earlier transitions, allowing the second expert to contribute more to the generation. Higher thresholds ($\tau = 0.7$) maintain the first expert's influence longer. For DDPM→FM, lower thresholds mitigate conversion artifacts by transitioning to native FM earlier, explaining the quality improvement at $\tau = 0.3$. For FM→DDPM, the ordering is already optimal, so threshold variation has minimal impact on overall quality.

These qualitative findings suggest that **DDPM-to-velocity conversion may be best restricted to low-noise regimes** ($t < 0.5$) for this simple conversion method, with high-noise generation handled by native Flow Matching experts or unconverted DDPM experts operating in their original parameterization.

2.4. Effects of Noise Schedules

The choice of noise schedule fundamentally shapes how diffusion models learn to denoise at different noise levels, affecting both training dynamics and generation quality. To investigate the impact of schedule heterogeneity in our framework, we conducted controlled experiments using a 2-expert configuration (1 DDPM expert + 1 Flow Matching expert, both trained on the same data cluster). Figure 11 compares two training strategies: (1) **Different schedules**, DDPM with cosine schedule and Flow Matching with linear interpolation, versus (2) **Same schedule**, where both experts are constrained to train with the cosine schedule.

The qualitative examples suggest that allowing each expert to train with its preferred schedule can improve some visual attributes. Images from the different-schedules configuration (left column) exhibit better visual coherence, sharper details, and more natural color gradients compared to the same-schedule baseline (right column). Vehicle samples demonstrate improved material rendering and structural clarity when experts use their native schedules.

This performance difference stems from fundamental differences in how each objective interacts with its noise schedule. Our FM experts use the linear-path formulation $x_t = (1-t)x_0 + t\epsilon$ [5], for which the target velocity $v_t = \epsilon - x_0$ is simple and time-independent. When forced to train with a cosine schedule instead, the FM objective must learn a more complex, time-varying velocity field that accounts for the nonlinear signal-to-noise ratio progression. This additional complexity can hinder optimization and lead to suboptimal learned representations, particularly at intermediate noise levels where the cosine schedule's curvature is most pronounced. (More broadly, Flow Matching [4] supports a general family of Gaussian probability paths, including diffusion-style paths; our discussion here concerns the specific linear-path variant used in our experiments.)

Conversely, DDPM with cosine scheduling benefits from a carefully calibrated noise distribution that allocates more

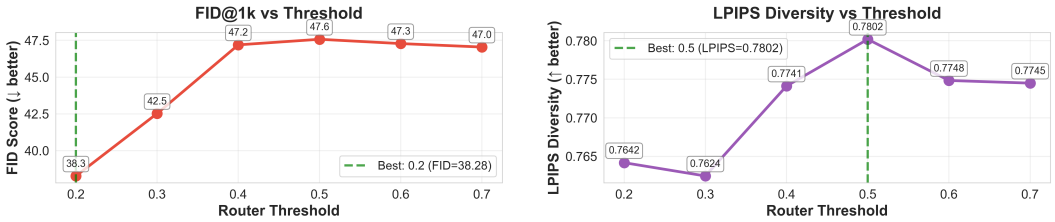


Figure 9. **Impact of Router Threshold on Generation Quality.** Different thresholds affect quality-diversity trade-offs.

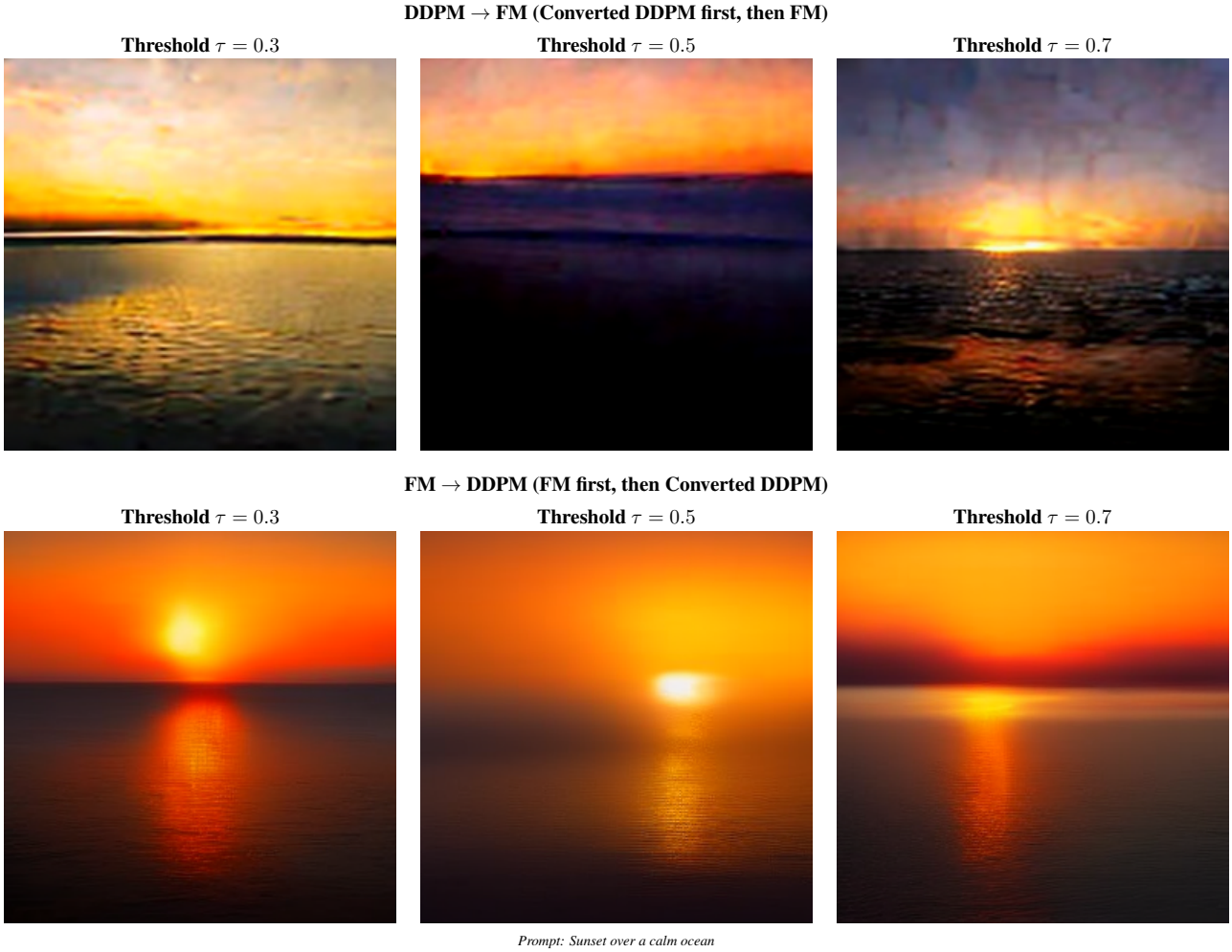


Figure 10. **Expert Ordering and Router Threshold Effects.** Comparison of 2-expert heterogeneous configurations showing the impact of expert ordering (DDPM→FM vs. FM→DDPM) and router threshold values. All experiments use the same unified schedule. FM→DDPM ordering produces more stable, coherent results, while DDPM→FM shows higher sensitivity to threshold selection.

training emphasis to perceptually important noise levels. The cosine schedule's design, maintaining higher signal-to-noise ratios for longer before rapid transition to pure noise, aligns well with DDPM's epsilon-prediction objective, enabling more stable gradient flow during training. This natural alignment between objective and schedule cannot be fully recovered when forcing disparate objectives to share a common schedule.

Our findings suggest that schedule-aware training can improve some qualitative visual traits in this 2-expert setting, even though the same-schedule setting achieves slightly better FID in the main-text comparison. The improved quality from schedule-aware training may justify the additional engineering complexity of supporting multiple schedules during both training and inference.

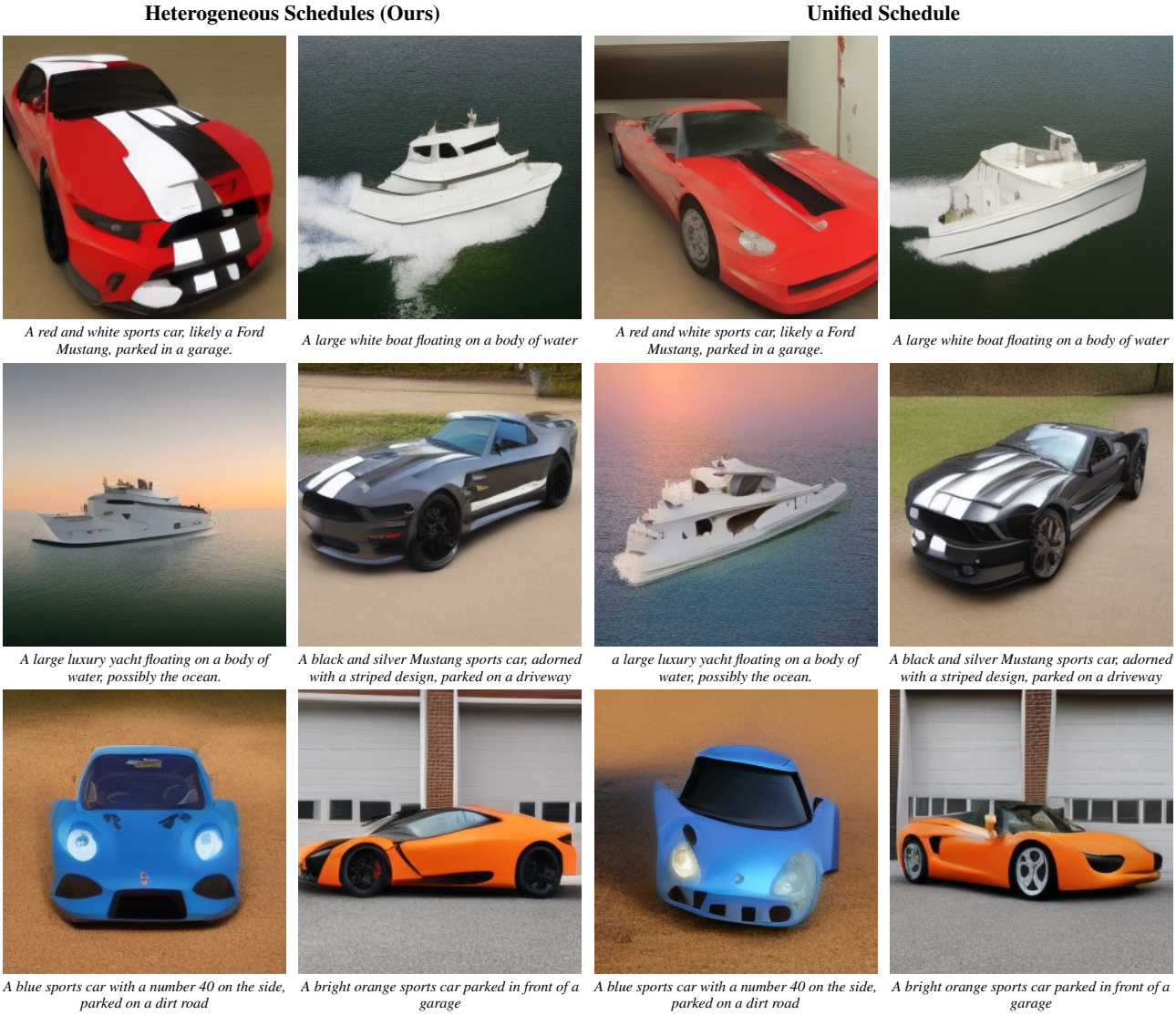


Figure 11. **Noise Schedule Comparison.** Controlled comparison using 2-expert configuration with different training strategies. Columns 1-2: Different schedules — DDPM expert trained with cosine schedule and Flow Matching expert trained with linear interpolation schedule (their natural configurations). Columns 3-4: Same schedule baseline — both DDPM and Flow Matching experts trained with cosine schedule. The different-schedules approach shows qualitatively sharper details in these examples, although the same-schedule setting achieves slightly better FID in Table 3 of the main paper.

2.5. Effects of Sampling Strategies

Figure 12 compares three key strategies: (1) **Top-1 selection**, which queries only the single highest-confidence expert at each timestep, (2) **Top-K sampling**, which averages predictions from the K most confident experts, and (3) **Full ensemble**, which performs weighted averaging across all K=8 experts.

While the full ensemble strategy theoretically implements the complete router posterior $p(k|x_t, t)$ and should minimize prediction variance, our empirical results reveal a more nuanced picture. Interestingly, the Top-2 strategy achieves the best FID score, outperforming both Top-1 and the full K=8 ensemble. This non-monotonic relationship between ensemble size and generation quality suggests that expert prediction diversity introduces both complementary information and conflicting guidance that must be carefully balanced.

The Top-1 strategy produces perceptually sharp samples by committing to a single expert's denoising trajectory at each timestep, maintaining strong sample coherence but sacrificing the benefits of multi-expert collaboration. As shown in Figure 12, Top-1 samples exhibit clear details

and consistent styles, though they may miss refinements that alternative experts could provide. The Top-2 strategy strikes an optimal balance: it leverages complementary information from two specialized experts while avoiding the over-smoothing that occurs when averaging many potentially conflicting velocity predictions. This finding is consistent with practical observations in mixture-of-experts systems, where top-K routing with small K is standard practice [3, 10], likely because it avoids averaging over conflicting specializations.

The full K=8 ensemble, while theoretically sound, suffers from averaging artifacts when combining predictions from experts with divergent specializations. When DDPM and Flow Matching experts disagree on fine-grained details, their weighted average can blur sharp features or introduce color inconsistencies, as visible in the full ensemble samples. This suggests that choosing the most relevant subset rather than averaging all available predictions may be more effective for heterogeneous diffusion ensembles. From a computational perspective, Top-K strategies also offer a significant speedup over the full ensemble.

3. DDPM to Flow Matching Conversion: Implementation Details

The conversion of DDPM expert outputs to Flow Matching velocity predictions is critical for heterogeneous ensemble inference. While theoretically straightforward, the practical implementation requires careful handling of numerical instabilities, schedule-dependent derivatives, and multi-expert coordination. This section provides comprehensive details on our conversion methodology.

3.1. Theoretical Foundation

3.1.1. General Conversion Formula

For any noise schedule parameterized by α_t and σ_t, the conversion from epsilon prediction $\epsilon_\theta(x_t, t)$ to velocity prediction $v(x_t, t)$ follows from the time derivative of the forward process. Given the forward diffusion:

$$x_t = \alpha_t x_0 + \sigma_t \epsilon, \tag{1}$$

we first recover the clean sample estimate by inverting the forward process:

$$\hat{x}_0 = \frac{x_t - \sigma_t \epsilon_\theta(x_t, t)}{\alpha_t}. \tag{2}$$

Treating $\hat{x}_0$ and ϵ_θ as fixed at their current-timestep values, the velocity field is computed as the time derivative of the deterministic path $\tilde{x}_t = \alpha_t \hat{x}_0 + \sigma_t \epsilon_\theta$:

$$v(x_t, t) = \frac{d\alpha_t}{dt} \hat{x}_0 + \frac{d\sigma_t}{dt} \epsilon_\theta(x_t, t). \tag{3}$$

This is the data-to-noise (forward) velocity. During sampling we integrate from t=1 to t=0, updating $x_{t-\Delta t} = x_t - v \cdot \Delta t$.

3.1.2. Schedule-Specific Formulations

For the linear interpolation schedule used in standard Flow Matching ($\alpha_t = 1 - t$, $\sigma_t = t$), the derivatives simplify to $\frac{d\alpha_t}{dt} = -1$ and $\frac{d\sigma_t}{dt} = 1$, yielding:

$$v(x_t, t) = \epsilon_\theta(x_t, t) - \hat{x}_0. \tag{4}$$

This matches the Flow Matching objective for the forward (data-to-noise) path.

For the cosine schedule commonly used in DDPM training:

$$\alpha_t = \cos\left(\frac{\pi t}{2}\right), \quad \sigma_t = \sin\left(\frac{\pi t}{2}\right), \tag{5}$$

$$\frac{d\alpha_t}{dt} = -\frac{\pi}{2}\sin\left(\frac{\pi t}{2}\right), \quad \frac{d\sigma_t}{dt} = \frac{\pi}{2}\cos\left(\frac{\pi t}{2}\right), \tag{6}$$

resulting in a more complex velocity computation that varies significantly with timestep.

3.2. Numerical Stability Challenges

3.2.1. Division by Small α_t

The primary numerical challenge arises when $\alpha_t \to 0$ at high noise levels ($t \to 1$), causing the clean sample recovery $\hat{x}_0 = \frac{x_t - \sigma_t \epsilon_\theta}{\alpha_t}$ to become unstable. For cosine schedules, α_t approaches zero rapidly near $t = 1$, amplifying any prediction errors in ϵ_θ.

3.2.2. Large Schedule Derivatives

The derivatives $\frac{d\alpha_t}{dt}$ and $\frac{d\sigma_t}{dt}$ can become large, particularly for cosine schedules. At $t \approx 0$, we have $|\frac{d\sigma_t}{dt}| = \frac{\pi}{2}|\cos(\frac{\pi t}{2})| \approx \frac{\pi}{2}$, while at $t \approx 1$, we have $|\frac{d\alpha_t}{dt}| = \frac{\pi}{2}|\sin(\frac{\pi t}{2})| \approx \frac{\pi}{2}$. These large derivatives can amplify velocity magnitudes and cause integration instability during sampling.

3.2.3. Accumulation of Conversion Errors

In multi-expert ensembles where some experts use DDPM objectives, conversion errors accumulate across the sampling trajectory. Small biases in individual conversions compound through the iterative sampling process, potentially causing divergence or color shifts in generated images.

3.3. Implementation Solutions

3.3.1. Adaptive Clamping

We implement data-type-aware clamping to prevent $\hat{x}_0$ from reaching unrealistic values:

$$\hat{x}_0^{\text{clamp}} = \text{clamp}(\hat{x}_0, -r, r), \quad r = \begin{cases} 20.0 & \text{for VAE latents} \\ 5.0 & \text{for pixel space} \end{cases} \tag{7}$$

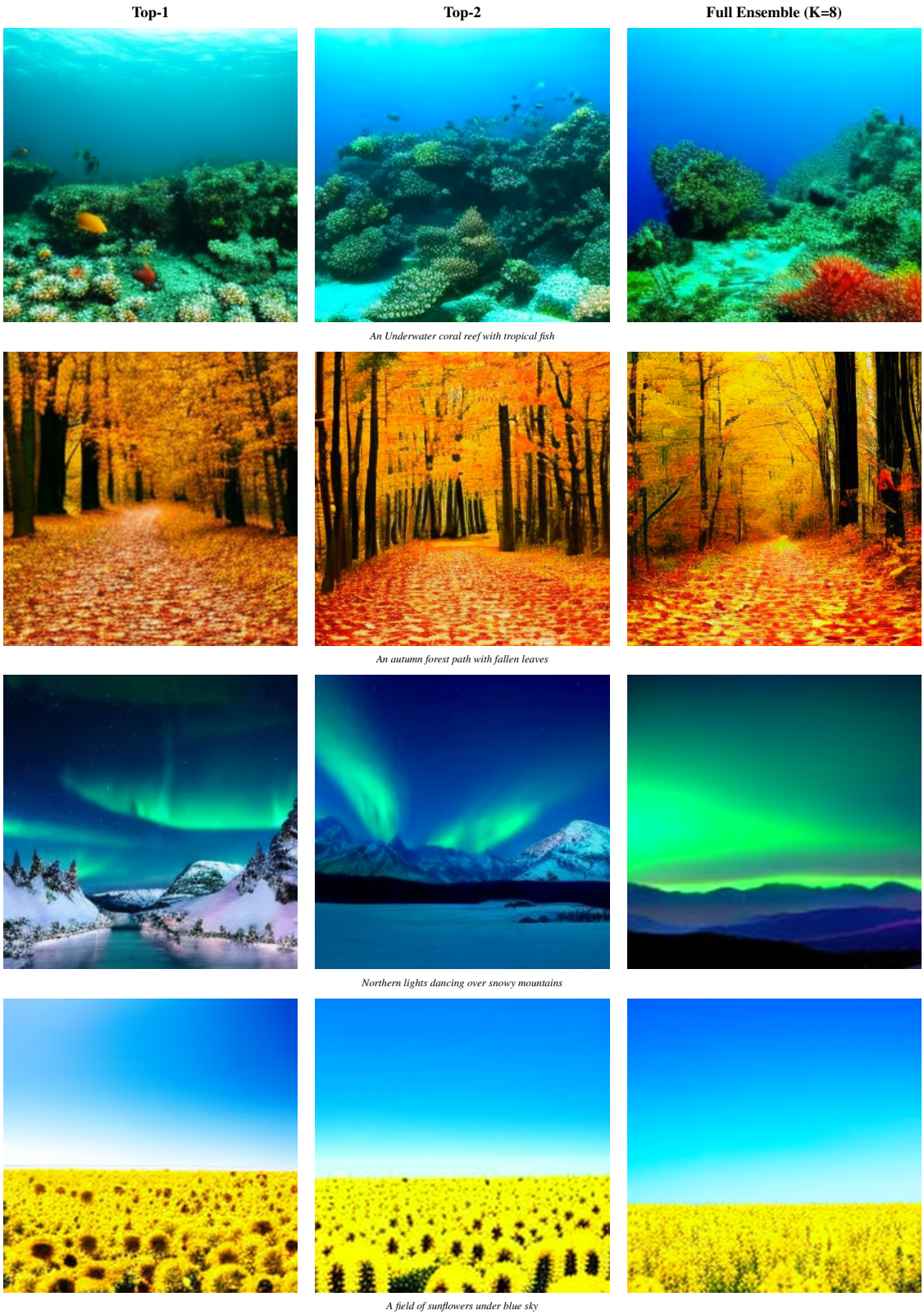


Figure 12. **Sampling Strategy Comparison.** Visual comparison of expert selection strategies on identical prompts. Top-1 commits to a single expert per timestep, producing sharp but potentially limited samples. Top-2 achieves optimal FID by balancing complementary information with prediction coherence. Full ensemble (K=8) can over-smooth details when averaging divergent expert predictions. All strategies use 75 Euler steps with CFG scale 6.

This range is empirically determined based on the typical distribution of clean samples in each representation space. VAE latents require a wider range due to their unbounded nature, while pixel-space values are typically normalized to either $[-1, 1]$ or $[0, 1]$ depending on the implementation.

3.3.2. Safe Division with Minimum Threshold

To handle small α_t values, we implement safe division:

$$\alpha_{\text{safe}} = \max(\alpha_t, 0.01), \tag{8}$$

ensuring numerical stability while minimizing bias. This threshold is chosen to balance stability against accuracy, as values below 0.01 occur only at extreme noise levels where exact recovery is inherently difficult.

3.3.3. Schedule Derivative Computation

For accurate velocity conversion, we compute finite-difference derivatives of the schedule coefficients:

$$\frac{d\alpha_t}{dt} \approx \frac{\alpha_{t+h} - \alpha_{t-h}}{2h}, \quad \frac{d\sigma_t}{dt} \approx \frac{\sigma_{t+h} - \sigma_{t-h}}{2h}, \tag{9}$$

where $h = 10^{-4}$ is the derivative epsilon. These derivatives are essential for computing the correct velocity under non-linear schedules.

3.3.4. Schedule-Aware Velocity Scaling

For cosine schedules, we apply adaptive dampening based on the noise level to control velocity magnitudes:

$$v_{\text{scaled}} = s(t) \cdot v(x_t, t), \quad s(t) = \begin{cases} 0.88 & \text{if } t > 0.85 \\ 0.93 & \text{if } 0.6 < t \leq 0.85 \\ 0.96 & \text{if } t \leq 0.6 \end{cases} \tag{10}$$

References

[1] Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis. *arXiv preprint arXiv:2310.00426*, 2023. 1

[2] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 2

[3] Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. Gshard: Scaling giant models with conditional computation and automatic sharding. *arXiv preprint arXiv:2006.16668*, 2021. 13

[4] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations (ICLR)*, 2023. 10

[5] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2023. 10

[6] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *Proceedings of the 38th International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021. 1

[7] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. 1

[8] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023. 1

[9] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 1

[10] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2017. 13