Heterogeneous Decentralized Diffusion Models

Zhiying Jiang Raihan Seraj Marcos Villagra Bidhan Roy

Bagel Labs

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

Abstract

Training frontier-scale diffusion models often requires substantial computational resources concentrated in tightly-coupled clusters, limiting participation to well-resourced institutions. While Decentralized Diffusion Models (DDM) enable training multiple experts in isolation, existing approaches require 1176 GPU-days and homogeneous training objectives across all experts. We present an efficient framework that dramatically reduces resource requirements while supporting heterogeneous training objectives. Our approach combines three key contributions: (1) a heterogeneous decentralized training paradigm that allows experts to use different objectives (DDPM and Flow Matching), unified at inference time without any retraining; (2) pretrained checkpoint conversion from ImageNet-DDPM to Flow Matching objectives, accelerating convergence and enabling initialization without objective-specific pretraining; and (3) PixArt-α's efficient AdaLN-Single architecture, reducing parameters while maintaining quality. Experiments on LAION-Aesthetics show that, relative to the training scale reported for prior DDM work, our approach reduces the compute by 16$\times$ and data by 14$\times$. Under aligned inference settings, our heterogeneous configuration achieves better FID and higher intra-prompt diversity than the homogeneous baseline. By eliminating synchronization requirements and enabling mixed DDPM/FM objectives, our framework makes decentralized generative model training accessible to contributors with single GPUs requiring only 24–48GB VRAM.

1. Introduction

Training frontier-scale diffusion models [2, 25, 27, 28] often requires hundreds of GPU-days on tightly-coupled clusters [23], concentrating capability within well-resourced institutions. This infrastructure barrier can limit broader participation in foundational model development.

Recent work on decentralized diffusion models (DDM) [21] offers a promising direction by demonstrating that multiple expert models can be trained in complete isolation on disjoint data partitions and later combined for high-quality generation. However, this framework assumes homogeneous training objectives across all experts, requiring coordination that may be impractical in truly decentralized settings where contributors operate independently with different resources, preferences, and technical constraints. Moreover, the computational requirements remain prohibitive, with the original DDM requiring 1176 A100-days for training on 158M images [21].

We present a heterogeneous decentralized diffusion framework that embraces the diversity inherent in distributed AI development. Our key insight is that different diffusion objectives, DDPM's ϵ-prediction [10] and Flow Matching's velocity-prediction [18, 19], induce complementary specialization patterns. By deliberately training experts with different objectives in complete isolation, we achieve greater generation diversity than homogeneous alternatives while maintaining semantic coherence.

To address the computational barrier, we introduce an efficient checkpoint conversion strategy that leverages pretrained ImageNet diffusion models [25]. We demonstrate that visual features learned under DDPM objectives [10] transfer effectively to Flow Matching formulations [18, 19], enabling faster convergence without requiring objective-specific pretraining. Combined with architectural optimizations from PixArt-α [1], specifically AdaLN-Single conditioning that reduces parameters by 30% while maintaining quality, our approach achieves strong generation results with dramatically reduced resource requirements.

Contributions. We make three primary contributions that advance decentralized diffusion model training:

- **Heterogeneous Decentralized Training:** We extend the DDM framework [21] to support mixed diffusion objectives, specifically DDPM [10] and Flow Matching [18, 19], across fully isolated experts. Building on parameterization analyses relating ϵ- and velocity-prediction [13, 29], we derive a schedule-aware deterministic conversion at inference time *without any retraining*, enabling seamless integration of heterogeneous experts.

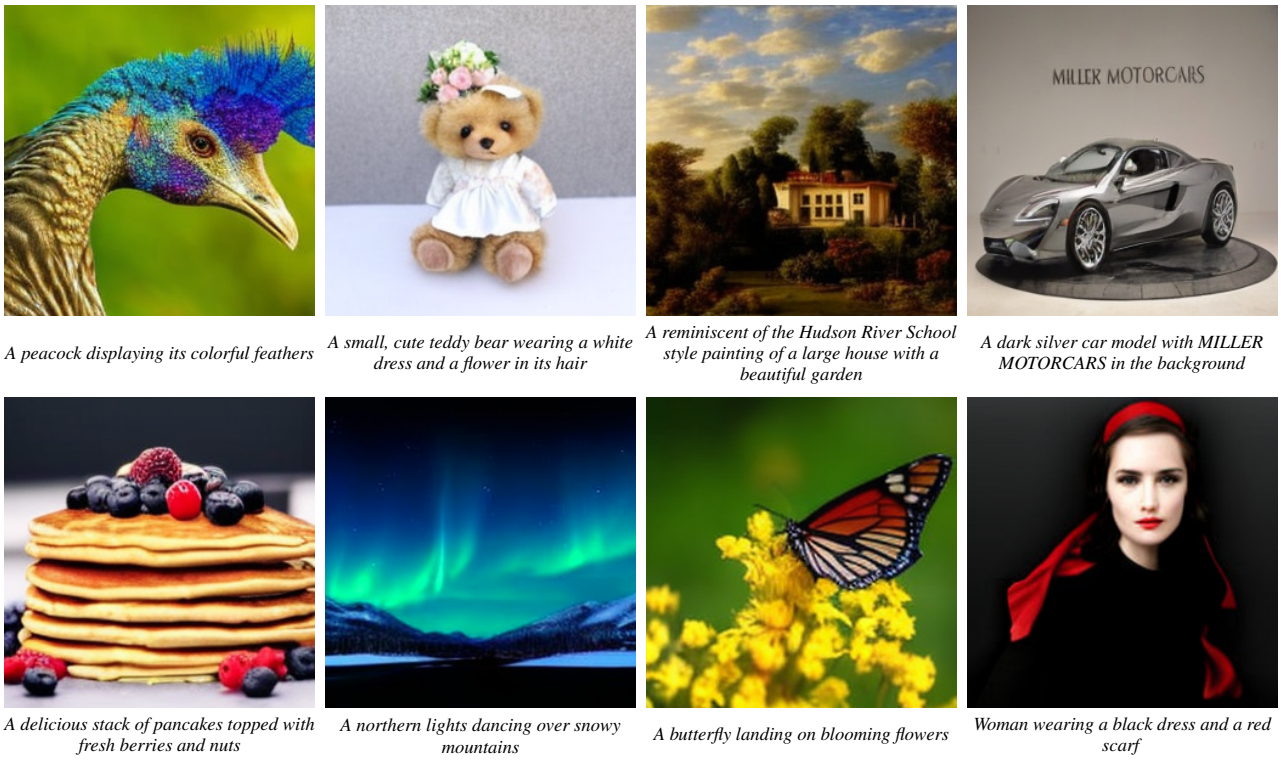


Figure 1. **Text-to-Image Generation with Heterogeneous Decentralized Diffusion.** Our framework combines multiple expert models trained with different objectives (DDPM and Flow Matching) in complete isolation to generate high-quality, diverse images from text prompts. All samples are generated at 256×256 resolution using 8 experts trained on LAION-Aesthetics with only 72 A100-equivalent GPU-days of compute.

- **Efficient Architecture with Checkpoint Initialization:** We adopt PixArt-α's AdaLN-Single conditioning [1] for each expert that achieves 30% parameter reduction while maintaining quality. We further demonstrate that pretrained ImageNet-DDPM checkpoints [25] can be effectively initialized for flow matching training [18] through architectural component transfer and layer reinitialization, achieving 1.2× faster loss reduction.
- **Scalable Decentralized Training:** Through heterogeneous objectives, architectural efficiency, and pretrained initialization, and relative to the training scale reported for prior DDM work [21], we reduce compute from 1176 to 72 A100-equivalent GPU-days (16×) and data from 158M to 11M images (14×), while each expert requires only 24–48GB VRAM for single-GPU deployment without specialized interconnects.

Our experimental evaluation on LAION-Aesthetics [30] demonstrates that decentralized training exceeds monolithic approaches. Using 8 DiT-B/2 [25] experts trained in complete isolation on LAION-Art, we achieve 23.7% FID improvement [8] over a centralized baseline at matched aggregate compute following McAllister et al. [21]. Under matched inference settings with 8 DiT-XL/2 experts, heterogeneous experts (2DDPM:6FM) improve both FID (11.88 vs. 12.45) and intra-prompt diversity (LPIPS [37] 0.631 vs. 0.617) relative to homogeneous experts (8FM). By eliminating synchronization requirements, our framework broadens the range of resources and training objectives that can participate in decentralized model development.

2. Related Work

Diffusion and Flow Matching. DDPM [10] introduced ϵ-prediction for iterative denoising, extended to latent space by LDMs [27] and scaled via DiT [25]. VDM [13] expresses the diffusion loss in terms of the signal-to-noise ratio, and Kingma & Gao [12] show that different objectives correspond to different noise-level weightings — an analysis we use in Section 3.3 to motivate complementary specialization. Flow Matching [16, 18, 19] learns velocity fields for continuous transport, enabling straighter trajectories; Diff2Flow [31] bridges DDPM and FM via timestep rescaling. Unlike these single-model methods, we enable *heterogeneous training* where experts use different objectives independently, and fuse their predictions through inference-time, schedule-aware alignment.

Decentralized and Efficient Training. Decentralized and

federated learning enable training across distributed nodes without centralized coordination [14, 20], with practical emphasis on communication efficiency [3, 33–35, 38]. Training efficiency for monolithic diffusion models has advanced through improved noise schedules [4, 5], parameter-efficient architectures [1], parallel score learning across time sub-intervals [6], distillation [22, 29], consistency models [7, 32], fast sampling [11, 36], and distributed parallel denoising [17]. DDM [21] trains isolated experts on clustered data and ensembles them via a router, eliminating high-bandwidth interconnect requirements but requiring homogeneous objectives across all experts. We extend DDM by (i) enabling *heterogeneous objectives*, so experts may train with DDPM or Flow Matching without coordination; and (ii) supporting *pretrained backbone transfer*, reusing existing checkpoints under a new objective.

3. Method

We present decentralized diffusion models with heterogeneous objectives that enable fully independent training of expert models without any gradient, parameter, or activation synchronization. Our framework builds upon decentralized flow matching theory [21], in which independently trained experts are combined via a learned router at inference time. We extend this foundation to support mixed diffusion objectives — experts may train with either DDPM or Flow Matching — by introducing a deterministic conversion that maps all expert predictions into a velocity space for unified sampling. To make decentralized training practical at scale, we further adopt an efficient architecture and a checkpoint initialization strategy that together reduce both parameter count and convergence time.

3.1. Decentralized Flow Matching

Following the decentralized Flow Matching formulation of McAllister et al. [21], we decompose the velocity field across K expert models trained on disjoint data partitions. The key theoretical foundation is that the marginal flow can be expressed as a weighted combination of conditional flows:

$$u_t(x_t) = \sum_{k=1}^{K} p_t(k|x_t) \cdot u_t^{(k)}(x_t), \tag{1}$$

where $u_t^{(k)}(x_t)$ is the velocity predicted by expert k trained only on cluster S_k, and $p_t(k|x_t)$ is the posterior probability that x_t belongs to cluster k.

We partition the dataset $\mathcal{D}$ into K semantic clusters $\{S_1, S_2, \ldots, S_K\}$ using DINOv2 [24] features, extracting 1024-dimensional representations and applying hierarchical k-means clustering. This produces semantically coherent partitions (e.g., portraits, landscapes, architecture) that enable meaningful expert specialization. Each expert θ_k then trains exclusively on its assigned cluster S_k without any communication with other experts, optimizing its assigned diffusion objective independently.

The router network ϕ learns to approximate p_t from noisy inputs:

$$p_\phi(k|x_t, t) = \text{softmax}(\text{Router}_\phi(x_t, t))_k, \tag{2}$$

trained with cross-entropy loss against ground-truth cluster assignments. At inference, the router dynamically selects and combines experts based on the noisy input and timestep.

3.2. Heterogeneous Objectives and Conversion

We extend the decentralized framework to support heterogeneous training objectives: n experts train with Flow Matching and m experts with DDPM, exploiting their complementary strengths. At inference, all predictions are mapped into a shared velocity space via deterministic conversion (Figure 2), enabling seamless ensemble without retraining. We assign experts to either ϵ-prediction or velocity-prediction objectives:

DDPM Experts predict the noise ϵ added during the forward process:

$$\mathcal{L}_{\text{DDPM}}^{(k)} = \mathbb{E}_{x_0 \in S_k, \epsilon, t}\left[\|\epsilon_{\theta_k}(\alpha_t x_0 + \sigma_t \epsilon, t) - \epsilon\|^2\right], \tag{3}$$

where α_t, σ_t follow a cosine schedule for stable training. The forward process corrupts the clean data x_0 by progressively adding Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$ according to the noise schedule, producing noisy observations $x_t = \alpha_t x_0 + \sigma_t \epsilon$ at timestep t. The model ϵ_{θ_k} is trained to predict the noise component ϵ from x_t, which can then be used to estimate the clean signal by inverting the linear forward map.

Flow Matching Experts directly predict velocity fields:

$$\mathcal{L}_{\text{FM}}^{(k)} = \mathbb{E}_{x_0 \in S_k, \epsilon, t}\left[\|v_{\theta_k}(x_t, t) - (\epsilon - x_0)\|^2\right], \tag{4}$$

where $x_t = (1-t)x_0 + t\epsilon$ represents the linear interpolation between clean data x_0 and Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$. Following the rectified flow framework [19], we parameterize the probability path with $t \in [0, 1]$, where $t = 0$ corresponds to the data distribution and $t = 1$ to the noise distribution. The model v_{θ_k} learns to predict the velocity field $v(x_t, t) = \frac{dx_t}{dt}$, which for our linear interpolation yields the target velocity $\epsilon - x_0$.

At inference, we unify predictions through schedule-aware deterministic conversion. Starting from the DDPM forward process $x_t = \alpha_t x_0 + \sigma_t \epsilon$, we recover an estimate of the clean sample by inverting the linear map:

$$\hat{x}_0 = \frac{x_t - \sigma_t \epsilon_{\theta_k}(x_t, t)}{\alpha_t}. \tag{5}$$

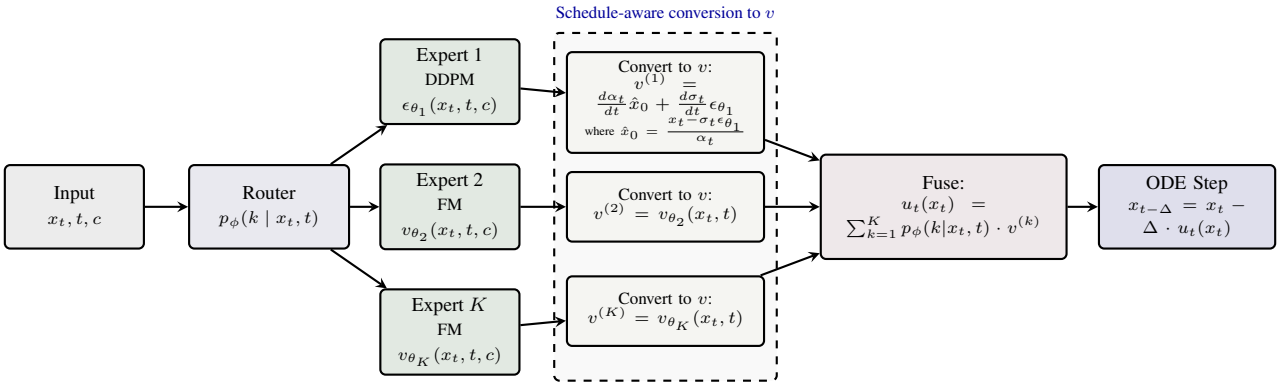


Figure 2. **Inference Pipeline for Heterogeneous Expert Fusion.** Given noisy input (x_t, t, c), the router predicts cluster probabilities $p_\phi(k|x_t, t)$ to weight expert contributions. DDPM experts output epsilon predictions while Flow Matching experts output velocity predictions. Schedule-aware conversion functions deterministically unify all predictions into a common velocity space $v^{(k)}$ without retraining, enabling router-weighted fusion $u_t(x_t) = \sum_{k=1}^{K} p_\phi(k|x_t, t) \cdot v^{(k)}$ for ODE-based sampling.

For any choice of schedule functions α_t, σ_t, substituting $\hat{x}_0$ back defines a deterministic path through the model's current estimates:

$$\tilde{x}_t(\hat{x}_0, \epsilon_\theta) = \alpha_t \hat{x}_0 + \sigma_t \epsilon_\theta(x_t, t). \tag{6}$$

Differentiating with respect to t while treating $\hat{x}_0$ and ϵ_θ as fixed at their current-timestep values gives the velocity along this path:

$$v(x_t, t) \equiv \frac{d\tilde{x}_t}{dt} = \frac{d\alpha_t}{dt}\hat{x}_0 + \frac{d\sigma_t}{dt}\epsilon_\theta(x_t, t). \tag{7}$$

For the linear interpolation schedule $\alpha_t = 1 - t$, $\sigma_t = t$, we have $\frac{d\alpha_t}{dt} = -1$, $\frac{d\sigma_t}{dt} = 1$, so Eq. (7) simplifies to

$$v(x_t, t) = \epsilon_\theta(x_t, t) - \hat{x}_0. \tag{8}$$

This is the data-to-noise velocity matching the FM target $v = \epsilon - x_0$; during sampling we integrate from t=1 to t=0 via $x_{t-\Delta t} = x_t - v \cdot \Delta t$. To ensure numerical stability, we clamp predicted $\hat{x}_0$ to $[-20, 20]$ for VAE latents, use $\alpha_{\text{safe}} = \max(\alpha_t, 0.01)$ in Eq. (5), and apply adaptive velocity scaling that dampens converted predictions at elevated noise levels where schedule derivatives become large.

3.3. Implicit Timestep Weighting Across Objectives

To analyze why mixed objectives can be complementary, we compare the effective timestep weighting induced by ϵ-prediction and velocity prediction under the same variance-preserving (VP) perturbation family $x_t = \alpha_t x_0 + \sigma_t \epsilon$ with $\alpha_t^2 + \sigma_t^2 = 1$ (following the parameterization analysis of Kingma et al. [13]). Although our FM experts in Eq. (4) use linear interpolation, this calculation isolates the objective-induced weighting effect; we show in the Remark below that the conclusion holds for linear interpolation as well.

Notation. In this subsection, $v = \alpha_t \epsilon - \sigma_t x_0$ denotes the diffusion v-parameterization of Salimans and Ho [29], not the ODE velocity field $v(x_t, t)$ used elsewhere in the paper for sampling.

Proposition 1. *Let $\mathcal{L}_\epsilon(t)$ and $\mathcal{L}_v(t)$ denote per-timestep MSE losses for ϵ-prediction and velocity prediction, respectively. Writing both losses in terms of clean-sample estimation error yields*

$$\mathcal{L}_\epsilon(t) = \mathbb{E}\left[w_\epsilon(t)\, \|\hat{x}_0^{(\epsilon)} - x_0\|_2^2\right], \quad w_\epsilon(t) = \frac{\alpha_t^2}{\sigma_t^2}, \tag{9}$$

$$\mathcal{L}_v(t) = \mathbb{E}\left[w_v(t)\, \|\hat{x}_0^{(v)} - x_0\|_2^2\right], \quad w_v(t) = \frac{1}{\sigma_t^2}. \tag{10}$$

Hence

$$\frac{w_v(t)}{w_\epsilon(t)} = \frac{1}{\alpha_t^2}. \tag{11}$$

Proof. For ϵ-prediction, inverting the forward process gives $\hat{x}_0^{(\epsilon)} = (x_t - \sigma_t \epsilon_\theta)/\alpha_t$ (Eq. (5)), so

$$\begin{aligned} \epsilon_\theta - \epsilon &= \frac{\alpha_t}{\sigma_t}(x_0 - \hat{x}_0^{(\epsilon)}) \\ \implies \quad \|\epsilon_\theta - \epsilon\|_2^2 &= \frac{\alpha_t^2}{\sigma_t^2}\|\hat{x}_0^{(\epsilon)} - x_0\|_2^2, \end{aligned} \tag{12}$$

which proves Eq. (9).
For velocity prediction, the target is $v = \alpha_t \epsilon - \sigma_t x_0$ [29]. Under the VP constraint $\alpha_t^2 + \sigma_t^2 = 1$, the clean sample is recovered via $\hat{x}_0^{(v)} = \alpha_t x_t - \sigma_t v_\theta$, since $\alpha_t x_t - \sigma_t v = (\alpha_t^2 + \sigma_t^2)x_0 = x_0$. Then

$$\begin{aligned} v_\theta - v &= \frac{x_0 - \hat{x}_0^{(v)}}{\sigma_t} \\ \implies \quad \|v_\theta - v\|_2^2 &= \frac{1}{\sigma_t^2}\|\hat{x}_0^{(v)} - x_0\|_2^2, \end{aligned} \tag{13}$$

which proves Eq. (10). Dividing gives Eq. (11). □

Remark. Since $\alpha_t \leq 1$, the ratio $w_v/w_\epsilon = 1/\alpha_t^2 \geq 1$, with equality only at $t = 0$ and diverging as $\alpha_t \to 0$ (high noise). Velocity-prediction experts therefore receive relatively stronger gradients at high-noise timesteps, creating natural complementary specialization with ϵ-prediction experts that are relatively upweighted at low noise.

Moreover, this ratio depends only on α_t, not on the specific schedule: under linear interpolation ($\alpha_t = 1 - t$, $\sigma_t = t$) one obtains $w_v/w_\epsilon = 1/(1-t)^2$, recovering the same $1/\alpha_t^2$ structure. Thus the complementary weighting applies directly to our FM experts, not only to the VP family analyzed above.

3.4. Efficient Expert Architecture

Each expert employs a Diffusion Transformer (DiT) [25] adapted with PixArt-α optimizations [1]. The model processes $32 \times 32 \times 4$ VAE latents [27] using 2×2 patch embedding to create 256-token sequences.

AdaLN-Single Conditioning. This module [1] computes all layer-wise adaptive modulation parameters through a single global computation rather than per-block MLPs. Given timestep embedding $\tau(t) \in \mathbb{R}^d$, the global modulation is computed as:

$$\mathbf{c} = \text{MLP}_{\text{global}}(\tau(t)) \in \mathbb{R}^{6d}, \tag{14}$$

producing a single shared vector that is broadcast to every transformer block. Per-block differentiation comes from a learned embedding $\mathbf{E}_b \in \mathbb{R}^{6\times d}$; the six modulation vectors for block b are

$$[\beta_b^{\text{msa}}, \gamma_b^{\text{msa}}, \alpha_b^{\text{msa}}, \beta_b^{\text{mlp}}, \gamma_b^{\text{mlp}}, \alpha_b^{\text{mlp}}] = \mathbf{c}_{[6\times d]} + \mathbf{E}_b, \tag{15}$$

where $\mathbf{c}_{[6\times d]}$ denotes $\mathbf{c}$ reshaped to $6 \times d$. This reduces parameters by approximately 30% for text-conditioned DiT-XL/2 while maintaining quality.

Transformer Block Architecture. Each block implements adaptive layer normalization with gated residual connections:

$$\mathbf{h}_1 = \mathbf{h} + \alpha_b^{\text{msa}} \cdot \text{MSA}(\text{LN}(\mathbf{h}) \odot (1 + \gamma_b^{\text{msa}}) + \beta_b^{\text{msa}}), \tag{16}$$

$$\mathbf{h}_2 = \mathbf{h}_1 + \text{CrossAttn}(\text{LN}(\mathbf{h}_1), \mathbf{e}_{\text{text}}), \tag{17}$$

$$\mathbf{h}' = \mathbf{h}_2 + \alpha_b^{\text{mlp}} \cdot \text{FFN}(\text{LN}(\mathbf{h}_2) \odot (1 + \gamma_b^{\text{mlp}}) + \beta_b^{\text{mlp}}), \tag{18}$$

where LN denotes layer normalization without learnable affine parameters, MSA is multi-head self-attention, and α_b parameters act as learnable gates controlling each sublayer's contribution.

For classifier-free guidance during inference, we randomly drop conditioning with probability $p_{\text{cfg}} = 0.1$ during training, using null embeddings obtained by encoding the empty string through the frozen CLIP text encoder for unconditional generation.

Initialization Strategy. Following Chen et al. [1], the per-block embeddings $\mathbf{E}_b$ are initialized with $\mathcal{N}(0, 1/\sqrt{d})$ to maintain gradient scale, while the $\text{MLP}_{\text{global}}$ linear layer uses $\mathcal{N}(0, 0.02)$ weights with zero bias, ensuring initial forward passes approximate an identity function. Cross-attention output projections are zero-initialized to stabilize early training when incorporating text conditioning.

3.5. Efficient Checkpoint Conversion for Experts

A critical challenge in scaling DDM is computational cost. We address this by converting pretrained ImageNet DiT checkpoints [25] to Flow Matching for accelerated convergence. This leverages the insight that low-level visual features learned under DDPM objectives remain valuable for alternative formulations like Flow Matching, despite different training targets.

Our conversion methodology transfers all core architectural components while reinitializing objective-specific layers:

$$\theta_{\text{expert}}^{(l)} = \begin{cases} \theta_{\text{DiT}}^{(l)} & \text{if } l \in \{\text{patch\_embed, blocks,} \\ & \quad \text{final\_layer.linear}\} \\ \text{sin-cos(grid)} & \text{if } l = \text{pos\_embed} \\ \mathcal{N}(0, 0.02) & \text{if } l = \text{text\_proj} \\ \varnothing & \text{if } l = \text{class\_embed} \end{cases} \tag{19}$$

Patch embeddings, transformer blocks, and the final layer's linear projection are fully transferred to preserve learned spatial and temporal dynamics. Positional embeddings are reinitialized with fixed sinusoidal-cosine encoding following PixArt-α's loading convention. Text projection is newly initialized, and class embeddings are removed.

A key technical consideration is timestep compatibility between objectives. DiT models expect timesteps in $[0, 999]$ while Flow Matching uses $t \in [0, 1]$. Rather than modifying pretrained weights, we implement runtime scaling that preserves the learned sinusoidal timestep encoding and subsequent MLP:

$$t_{\text{DiT}} = \begin{cases} 999 \cdot t & \text{if } t \in [0, 1] \text{ (Flow Matching experts),} \\ t & \text{if } t \in [0, 999] \text{ (DDPM experts).} \end{cases} \tag{20}$$

Since DiT's timestep module uses sinusoidal positional encoding followed by a learned MLP (rather than a discrete embedding table), it naturally handles continuous-valued inputs. This approach maintains temporal reasoning capabilities acquired during pretraining while adapting seamlessly to different noise schedules used by heterogeneous objectives.

Inference Strategy	FID-50K ↓
Monolithic (single model)	29.64
Top-1	30.60
Top-2	**22.60**
Full Ensemble (all experts)	47.89
Improvement vs. Monolithic	7.04

Table 1. FID-50K (lower is better) comparing monolithic training and decentralized multi-expert training with different inference strategies on LAION-Art. All models use the **DiT-B/2** architecture (129M parameters per expert).

4. Experiments

We evaluate our framework on LAION-Aesthetics [15] through three sets of experiments: (1) efficiency of decentralized multi-expert training compared to monolithic training under iso-FLOP conditions; (2) effectiveness of pretrained checkpoint conversion for accelerating convergence; and (3) comparison of heterogeneous versus homogeneous objectives under aligned inference settings. We evaluate generation quality using FID-50K on a held-out 50K test set following [21]. Unless otherwise noted, comparisons use matched inference settings (CFG$=$ 7.5, 50 steps) on this same holdout split.

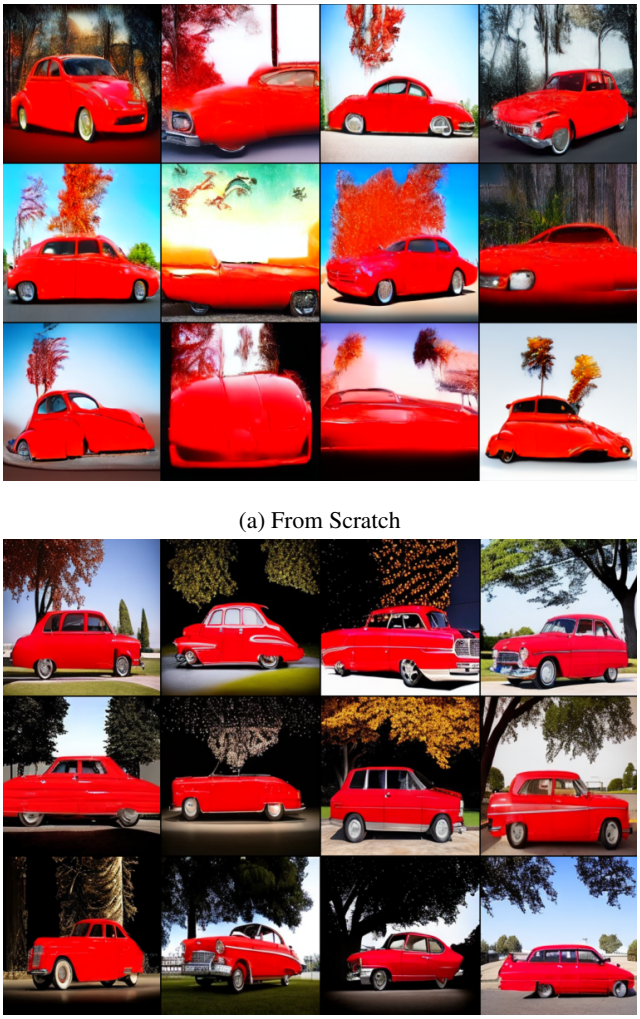
4.1. Experimental Setup

We train on a subset of 11M LAION-Aesthetics images. Each expert trains on semantically clustered data partitions obtained via DINOv2 [24] features. We train at two scales: **DiT-B/2** (129M parameters per expert) for the monolithic comparison in Section 4.2, and **DiT-XL/2** (605M parameters per expert) with PixArt-α's AdaLN-Single conditioning for all other experiments.

For the XL/2 experiments, we evaluate two configurations: a **homogeneous** baseline where all $K{=}8$ experts use Flow Matching objectives (8FM), and a **heterogeneous** configuration where experts 0 and 3 are assigned DDPM objectives and the remaining six use Flow Matching (2DDPM:6FM). All experts train in complete isolation without gradient, parameter, or activation synchronization. At inference, a learned router selects experts per timestep via *Top-1*, *Top-K*, or *Full*. Preprocessing and training hyperparameters are provided in the supplement.

4.2. Monolithic versus DDM

We compare our decentralized multi-expert approach against a monolithic baseline on LAION-Art (3.9M images). The monolithic model trains a single DiT-B/2 on the entire dataset, while our approach distributes training across 8 independent experts on semantic clusters ($<$500K images per expert). To ensure fair comparison, we match the aggregate computational budget following [21]: the monolithic batch size of 256 becomes a per-expert batch size of 32, ensuring equivalent total FLOPs. Both train from scratch without pretrained checkpoints. For this evaluation, all experts are trained with Flow Matching objectives.



(a) From Scratch

(b) Pretrained Initialization

Figure 3. **Impact of Pretrained Checkpoint Conversion.** Comparison of generated samples after 75K optimization steps using the text prompt "a red car in front of the tree".

At inference, we evaluate different expert combination strategies. *Top-K* uses a learned router to select the K most confident experts per step, combining predictions via weighted averaging. *Full Ensemble* combines all 8 experts with router-weighted contributions, incurring higher computational overhead.

Results show Top-2 achieves FID 22.60, outperforming the monolithic baseline by 7.04 points (23.7% improvement), demonstrating that strategic expert selection leverages specialized knowledge more effectively. Full Ensem-

Method	Data	Compute	FID@50K↓
DDM [21]†	158M	1176 A100 days	5.5–10.5
Ours (Homo)	11M	72 A100 days	12.45
Ours (Hetero)	11M	72 A100 days	11.88

Table 2. **Baseline Comparison.** Resource comparison against DDM. Both of our rows use the same aligned inference settings (CFG= 7.5, 50 steps). Compute is in A100-equivalent GPU-days; our experiments use A40 48GB GPUs (120 A40 GPU-days total, normalized via measured training throughput). †The FID range 5.5–10.5 is estimated from results at varying training FLOPs in McAllister et al. [21]

ble underperforms (FID 47.89), suggesting that indiscriminate combination introduces prediction conflicts. Selective expert activation proves crucial for superior performance at this scale.

4.2.1. Resource Efficiency

Table 2 places our results in the context of the model scale reported for prior DDM work [21]. Relative to that reported scale, our approach reduces compute from 1176 to 72 A100-equivalent GPU-days (16×) and data from 158M to 11M images (14×). Note that the DDM FID range of 5.5–10.5 is achieved at substantially larger training scale and data; our numbers are therefore not directly comparable in absolute FID terms, but illustrate that competitive generation quality is attainable at a fraction of the resources. Under our homogeneous baseline, we achieve 12.45 FID. Introducing heterogeneous objectives further improves FID to 11.88, demonstrating that objective diversity provides an additional quality gain at no extra training cost.

4.2.2. Impact of Pretrained Checkpoint Initialization

We validate checkpoint conversion effectiveness by comparing models at 75K optimization steps. Figure 3 shows that models initialized with converted ImageNet checkpoints generate substantially higher-quality samples than those trained from scratch. Validation loss confirms 1.2× faster loss reduction (see supplement), demonstrating effective transfer of visual priors across objectives. Pretrained models produce sharper details and better semantic alignment, while scratch-trained models exhibit artifacts and lower fidelity.

4.3. DDPM→FM without Training

We investigate converting DDPM experts to FM objectives without retraining, enabling flexible expert combination at inference time for heterogeneous DDM deployment.

4.3.1. Conversion Experiment Configuration

We use two inference settings in Section 4. Conversion-focused analyses in this subsection use CFG [9] scale 6 with 75 sampling steps on 5,000 held-out samples. Within this conversion setting, we evaluate five sampling configurations using experts trained on the same data cluster to isolate objective conversion effects from data distribution differences. Both DDPM and FM experts use the DiT-XL/2 architecture with identical hyperparameters. For combined experts, a deterministic router switches between experts at a native-time threshold $t = 0.5$, allocating high-noise timesteps ($t > 0.5$) to FM experts and low-noise timesteps to DDPM experts.

Sampling Method	LPIPS↑	FID↓	CLIP↑
Native DDPM	0.787	27.04	$0.316_{\pm 0.030}$
FM	0.752	20.23	$0.324_{\pm 0.034}$
DDPM→FM	0.761	25.61	$0.319_{\pm 0.032}$
Combined (same schedule)	0.782	32.67	$0.312_{\pm 0.035}$
Combined (diff. schedules)	0.777	33.29	$0.312_{\pm 0.034}$

Table 3. **Sampling Quality Comparison.** DDPM→FM conversion improves over native DDPM and enables mixed-objective sampling. Combined experts achieve higher diversity (LPIPS) than single FM, though at a FID cost, reflecting the quality–diversity trade-off of heterogeneous fusion.

4.3.2. Results and Analysis

Table 3 presents our quantitative evaluation. In addition to FID, we include CLIP [26] for text-image alignment and mean pairwise LPIPS [37] across generated samples as a diversity metric. Because LPIPS measures perceptual distance between image pairs, higher mean pairwise LPIPS indicates greater output diversity (LPIPS↑).

Three key findings emerge from our analysis:

(1) Effective inference-time alignment: The DDPM→FM conversion improves generation quality compared to native DDPM (FID 25.61 vs. 27.04) and preserves semantic coherence (CLIP score 0.319 vs. 0.316), enabling interoperability with FM experts. Native FM remains the strongest single-expert baseline (FID 20.23), indicating that the conversion is most valuable as a compatibility mechanism rather than a lossless objective replacement.

(2) Enhanced diversity through combination: Combined expert sampling achieves higher output diversity (mean pairwise LPIPS), approaching native DDPM levels (0.787) while surpassing a single FM expert (0.752). This demonstrates that heterogeneous objectives create complementary generation patterns, producing more varied outputs than a single-objective expert.

(3) Schedule impact on combination: Interestingly, using the same cosine schedule for both objectives yields marginally better results than different schedules (FID 32.67 vs. 33.29), suggesting that schedule alignment facilitates smoother expert transitions. However, both combinations exhibit similar diversity gains, indicating that objective heterogeneity drives the primary benefits rather than schedule diversity.

Model	CFG	Steps	FID-50K↓
Homogeneous (8FM)	7.5	50	12.45
Heterogeneous (1DDPM:7FM)	6.0	75	19.75
Heterogeneous (2DDPM:6FM)	6.0	75	15.09
Heterogeneous (2DDPM:6FM)	7.5	50	**11.88**

Table 4. **Homogeneous vs Heterogeneous Comparison.** The first and last rows are directly comparable (same CFG and steps).

The increased FID for combined methods compared to single experts reflects the challenge of seamless expert switching during sampling. However, this trade-off is acceptable given the substantial diversity gains and the practical benefits of heterogeneous training, where experts can leverage different computational resources and training strategies while maintaining compatible inference.

A routing-threshold sweep (see supplement) confirms a quality–diversity trade-off: lower thresholds (0.2–0.3) favor FID while mid-range values (0.4–0.5) favor diversity.

4.4. Homogeneous versus Heterogeneous

To isolate the effect of objective heterogeneity, we evaluate homogeneous and heterogeneous 8-expert models under aligned inference settings (CFG$=$ 7.5, 50 steps) on the same held-out 50K split.

Quantitative results. Under aligned settings (CFG$=$ 7.5, 50 steps), the heterogeneous 2DDPM:6FM model achieves 11.88 FID, outperforming homogeneous 8FM (12.45). Under the conversion setting (CFG$=$ 6, 75 steps), increasing DDPM experts from 1 to 2 improves FID from 19.75 to 15.09. Intra-prompt diversity (mean pairwise LPIPS [37] over 10 images per prompt, 100 prompts) is also higher for heterogeneous experts (0.631 ± 0.078 vs. 0.617 ± 0.074), confirming more varied outputs for identical prompts.

Qualitative results. Figure 4 compares homogeneous (FM-only) and heterogeneous (FM+DDPM) models on identical prompts and seeds. Heterogeneous models tend to preserve sharper local structure and richer textures, consistent with the FID improvement in Table 4.

Figure 4. **Qualitative comparison: Homogeneous vs. Heterogeneous models.** Images generated from identical prompts and random seeds. Homogeneous models (left, trained with Flow Matching only) often appear smoother in texture. Heterogeneous models (right, combining FM and DDPM experts) often preserve sharper local details and richer texture variation.

5. Conclusion

We presented a decentralized diffusion framework in which experts train independently with heterogeneous objectives and are combined at inference time through schedule-aware alignment without retraining. By pairing this with efficient architecture choices and pretrained checkpoint transfer, the framework achieves an order-of-magnitude reduction in compute and data relative to prior DDM-scale training, while heterogeneous experts improve both FID and generation diversity over homogeneous baselines under aligned settings. These results suggest that embracing objective diversity across independently trained experts is a practical path toward more accessible decentralized generative model development.

Limitations. Our experiments evaluate only a narrow set of DDPM-to-FM ratios (2:6); the optimal allocation likely depends on the data distribution and downstream requirements. The inference-time algebraic conversion relies on hand-tuned numerical safeguards and is empirically strongest when converted DDPM experts operate in the low-noise regime; a more robust conversion that generalizes across arbitrary schedules without manual stabilization remains open. Finally, the framework currently supports only ϵ- and velocity-prediction; extending to additional objectives such as x_0-prediction or consistency targets [32] would require generalizing both the conversion and routing mechanisms.

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, 3, 5

[2] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021. 1

[3] Akash Dhasade, Anne-Marie Kermarrec, Rafael Pires, Rishi Sharma, and Milos Vujasinovic. Decentralized learning made easy with decentralizepy. In *Proceedings of the 3rd Workshop on Machine Learning and Systems*, pages 34–41, 2023. 3

[4] Tiankai Hang, Shuyang Gu, Chen Li, Jianmin Bao, Dong Chen, Han Hu, Xin Geng, and Baining Guo. Efficient diffusion training via min-snr weighting strategy. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7441–7451, 2023. 3

[5] Tiankai Hang, Shuyang Gu, Jianmin Bao, Fangyun Wei, Dong Chen, Xin Geng, and Baining Guo. Improved noise schedule for diffusion training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4796–4806, 2025. 3

[6] Etrit Haxholli and Marco Lorenzi. Faster training of diffusion models and improved density estimation via parallel score matching. In *Advances in Neural Information Processing Systems*, 2023. 3

[7] Jonathan Heek, Emiel Hoogeboom, and Tim Salimans. Multistep consistency models. *arXiv preprint arXiv:2403.06807*, 2024. 3

[8] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, 2017. 2

[9] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 7

[10] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, pages 6840–6851, 2020. 1, 2

[11] Hongxu Jiang, Muhammad Imran, Teng Zhang, Yuyin Zhou, Muxuan Liang, Kuang Gong, and Wei Shao. Fast-ddpm: Fast denoising diffusion probabilistic models for medical image-to-image generation. *IEEE Journal of Biomedical and Health Informatics*, 2025. 3

[12] Diederik P. Kingma and Ruiqi Gao. Understanding diffusion objectives as the ELBO with simple data augmentation. In *Advances in Neural Information Processing Systems*, 2023. 2

[13] Diederik P. Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. In *Advances in Neural Information Processing Systems*, pages 21696–21707, 2021. 1, 2, 4

[14] Anastasia Koloskova, Sebastian Stich, and Martin Jaggi. Decentralized stochastic optimization and gossip algorithms with compressed communication. In *International conference on machine learning*, pages 3478–3487. PMLR, 2019. 3

[15] LAION-AI. Laion-aesthetics v2: Aesthetic-filtered subset of laion-5b. https://laion.ai/blog/laion-aesthetics/, 2022. Subset of LAION-5B filtered for high predicted aesthetic scores. 6

[16] Sangyun Lee, Zinan Lin, and Giulia Fanti. Improving the training of rectified flows. *Advances in neural information processing systems*, 37:63082–63109, 2024. 2

[17] Muyang Li, Tianle Cai, Jiaxin Cao, Qinsheng Zhang, Han Cai, Junjie Bai, Yangqing Jia, Ming-Yu Liu, Kai Li, and Song Han. Distrifusion: Distributed parallel inference for high-resolution diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7183–7193, 2024. 3

[18] 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. 1, 2

[19] 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. 1, 2, 3

[20] Yucheng Lu and Christopher De Sa. Optimal complexity in decentralized training. In *International conference on machine learning*, pages 7111–7123. PMLR, 2021. 3

[21] David McAllister, Matthew Tancik, Jiaming Song, and Angjoo Kanazawa. Decentralized diffusion models. *arXiv preprint arXiv:2501.05450*, 2025. 1, 2, 3, 6, 7

[22] Chenlin Meng, Robin Rombach, Ruiqi Gao, Diederik Kingma, Stefano Ermon, Jonathan Ho, and Tim Salimans. On distillation of guided diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14297–14306, 2023. 3

[23] 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

[24] 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. 3, 6

[25] 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, 2, 5

[26] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry,

Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 7

[27] 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, 2, 5

[28] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022. 1

[29] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022. 1, 3, 4

[30] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. LAION-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022. 2

[31] Johannes Schusterbauer, Ming Gui, Frank Fundel, and Björn Ommer. Diff2flow: Training flow matching models via diffusion model alignment. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 28347–28357, 2025. 2

[32] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *International Conference on Machine Learning*, pages 32211–32252. PMLR, 2023. 3, 8

[33] Stefanie Warnat-Herresthal, Hartmut Schultze, Krishnaprasad Lingadahalli Shastry, Sathyanarayanan Manamohan, Saikat Mukherjee, Vishesh Garg, Ravi Sarveswara, Kristian Händler, Peter Pickkers, N Ahmad Aziz, Sofia Ktena, Florian Tran, Michael Bitzer, Stephan Ossowski, Nicolas Casadei, Christian Herr, Daniel Petersheim, Uta Behrends, Fabian Kern, Tobias Fehlmann, Philipp Schommers, Clara Lehmann, Max Augustin, Jan Rybniker, Janine Altmüller, Neha Mishra, Joana P Bernardes, Benjamin Krämer, Lorenzo Bonaguro, Jonas Schulte-Schrepping, Elena De Domenico, Christian Siever, Michael Kraut, Milind Desai, Bruno Monnet, Maria Saridaki, Charles Martin Siegel, Anna Drews, Melanie Nuesch-Germano, Heidi Theis, Jan Heyckendorf, Stefan Schreiber, Sarah Kim-Hellmuth, Jacob Nattermann, Dirk Skowasch, Ingo Kurth, Andreas Keller, Robert Bals, Peter Nürnberg, Olaf Rieß, Philip Rosenstiel, Mihai G Netea, Fabian Theis, Sach Mukherjee, Michael Backes, Anna C Aschenbrenner, Thomas Ulas, Monique M B Breteler, Evangelos J Giamarellos-Bourboulis, Matthijs Kox, Matthias Becker, Sorin Cheran, Michael S Woodacre, Eng Lim Goh, and Joachim L Schultze. Swarm learning for decentralized and confidential clinical machine learning. *Nature*, 594(7862): 265–270, 2021. 3

[34] Chuhan Wu, Fangzhao Wu, Lingjuan Lyu, Yongfeng Huang, and Xing Xie. Communication-efficient federated learning via knowledge distillation. *Nature communications*, 13(1): 2032, 2022.

[35] Anqi Zhang, Ping Zhao, Wenke Lu, and Guanglin Zhang. Decentralized federated learning towards communication efficiency, robustness, and personalization. *ACM Transactions on Sensor Networks*, 21(3):1–20, 2025. 3

[36] Kexun Zhang, Xianjun Yang, William Yang Wang, and Lei Li. Redi: efficient learning-free diffusion inference via trajectory retrieval. In *International Conference on Machine Learning*, pages 41770–41785. PMLR, 2023. 3

[37] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2, 7, 8

[38] Shenglong Zhou, Kaidi Xu, and Geoffrey Ye Li. Communication-efficient decentralized federated learning via one-bit compressive sensing. In *2024 IEEE 99th Vehicular Technology Conference (VTC2024-Spring)*, pages 1–5. IEEE, 2024. 3