Compositional Generative Modeling from Decentralized Data

Mashrur M. Morshed [1] **Vishnu Naresh Boddeti** [1]

Abstract

Learning the compositional nature of the physical world requires joint observation of interacting factors. However, because practical data is often decentralized, these factors are fragmented across isolated silos. Existing decentralized generative approaches focus only on modeling the *union* of siloed data, overlooking *novel combinations* implied by the collective whole. To bridge this gap, we introduce Decentralized Compositional Flow Matching (DCFM), a framework that enforces structural constraints across the global set of generative factors, without exchanging any raw data. DCFM enables novel combinations to emerge through peer interactions, even when no single data source can independently support the composition. Empirically, DCFM substantially outperforms federated learning and mixture-of-experts baselines across conditional image generation, robotic spatial planning, and medical attribute co-occurrence modeling.

1. Introduction

Consider learning a generative model for outdoor robot navigation from data collected by multiple robots deployed under different environmental conditions. Due to communication constraints or operational isolation, each robot stores experience locally and cannot share raw trajectories. One robot observes navigation patterns exclusively in rain, another only in windy conditions, and a third only across varied terrain types (flat roads, off-road) (Figure 1). Although each robot's data captures valid aspects of the navigation task, many operationally critical scenarios are never observed by any single robot during training: navigating flat terrain during combined wind and rain, handling off-road conditions in rain, managing wind on uneven ground, or simultaneously contending with all three factors. Yet these multi-factor compositions are precisely the conditions that deployed robots must handle reliably in the real-world.

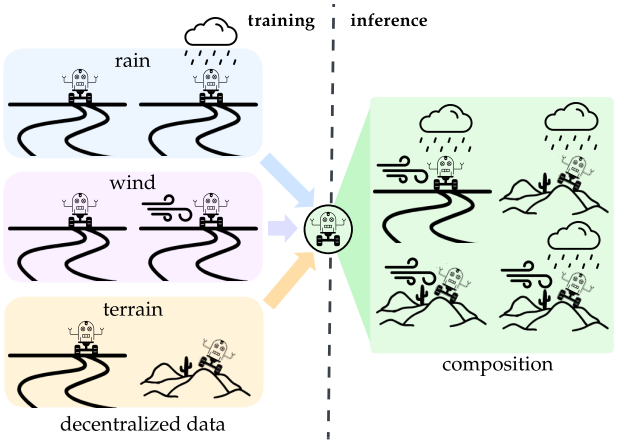


Figure 1. Illustration of compositional generalization in decentralized settings. At training (left), robots observe navigation under isolated environmental factors: rain, wind, and terrain variation. Compositional generalization requires handling unseen factor combinations (right): wind+rain on flat terrain, rain on off-road terrain, wind on off-road terrain, and wind+rain+off-road. Standard federated learning cannot generalize to these unobserved compositions.

Multiple solutions can be employed to achieve this goal. For example, federated learning (McMahan et al., 2017) can be employed to learn a global generative model without sharing data (Augenstein et al., 2020; Tun et al., 2023), although they are not explicitly designed for compositional generalization. Alternatively, mixture-of-experts (McAllister et al., 2025; Hahn & Lee, 2025) or product-of-experts (Zhang et al., 2025) can be used to compose locally trained independent generative models at inference time, leveraging the strengths of each expert. When the data distributions are heterogeneous, the latter strategy can, in theory, be effective and yield strong empirical performance.

However, as we demonstrate in this paper, when the factors required for composition are distributed across isolated data sources, the above solutions break down. Federated learning implicitly assumes that averaging parameters or gradients produces a coherent global model, an assumption violated when local datasets contain disjoint or highly skewed combinations of generative factors. Expert-based approaches, on the other hand, lack structural guarantees (e.g. a shared latent space) for how independently trained models should interact. As a result, novel compositions, such as trajectory patterns that require combining knowledge held by differ-

[1]Department of Computer Science and Engineering, Michigan State University, East Lansing, MI, USA. Correspondence to: Mashrur M. Morshed <morshedm@msu.edu>.

Preprint. June 3, 2026.

ent robots, remain inaccessible, even though all required components are present across the population.

This raises a central research question. *How can generative models recover compositional structure when no single data source contains sufficient information to support composition on its own, and raw data cannot be shared?*

We present Decentralized Compositional Flow Matching (DCFM) to learn generative models that are explicitly designed for compositional generalization from decentralized data without sharing raw data. The key idea is to satisfy conditional independence (CI) globally across data silos when the total coverage of attribute combinations is limited. We introduce two variations of DCFM. The first, DCFM-A, optimizes the global CI objective directly on real local datasets. It preserves a set of decentralized "experts" that maintain high empirical fidelity, though it remains computationally expensive during sample generation. The second, DCFM-B, is an efficient alternative that distills the collective knowledge of the local experts into a monolithic student model through synthetic data replay. By training on composite paths generated by the local flows, the student learns the global CI relations, enabling efficient sampling of novel attribute combinations never seen by any individual expert.

Summary of contributions.

1. We demonstrate that standard flow models trained independently suffer from (1) spurious correlation of independent attributes in the local models, and (2) incompatible velocity fields for composition, collectively limiting compositional generalization across isolated data silos. (§ 4)
2. We present the first framework to enable compositional generalization from decentralized data, introducing a training objective that enforces cross-expert compatibility via a shared velocity field and global conditional independence across the data silos. (§ 5)
3. We adopt this objective to learn both mixture-of-experts and a monolithic model. We demonstrate the effectiveness and versatility of DCFM for compositional generalization across conditional image generation and robotic path planning tasks. (§ 6)

2. Related Work

Federated Learning for Diffusion Models. Federated learning (FL) offers a promising framework for training generative models in privacy-sensitive domains by avoiding centralized data aggregation. Recent works have adapted diffusion training to this setting through federated averaging (Tun et al., 2023) or decentralized sampling protocols (Hahn & Lee, 2025). Despite this progress, practical adoption is hindered by the high computational and communication overheads inherent to the iterative training of large-scale diffusion models (Vora et al., 2024; de Goede et al., 2024). Furthermore, existing FL methods exhibit limited robustness to non-IID data distributions, often necessitating privacy-weakening workarounds such as partial data sharing (Jothiraj & Mashhadi, 2024) or frequent local retraining (Peng et al., 2025). Crucially, these approaches typically aim to approximate a single global distribution; they are not designed to exploit the specialized expertise within individual data silos, which limits both generation quality and compositional flexibility.

Mixture of Experts and Compositional Generalization. Prior work on compositional generalization relies on two critical, often unmet assumptions. In monolithic frameworks that combine conditional scores (Du et al., 2020; Liu et al., 2022a; Luo et al., 2025), the validity of composition hinges on the factors being statistically independent. Gaudi et al. (2025) demonstrated that standard training violates this condition and proposed CoInD to explicitly enforce independence. Conversely, mixture-of-experts approaches (McAllister et al., 2025; Zhang et al., 2025; Hahn & Lee, 2025) assume that independently trained models occupy compatible score spaces. While CoInD resolves the independence issue, it operates on centralized data and does not address the compatibility gap inherent to decentralized learning. When experts are trained on isolated data silos, their underlying score fields are not naturally aligned; this incompatibility renders standard composition techniques invalid, resulting in significantly degraded generative performance (e.g., FID) compared to centralized models trained on pooled data. Thus, achieving compositional generalization across users requires a framework that ensures both factor independence and cross-expert compatibility without centralized data access.

3. Problem Statement and Preliminaries

3.1. Problem Formulation

Settings. Let $\mathbf{x} \in \mathcal{X}$ denote a sample from the data space $\mathcal{X}$. Each $\mathbf{x}$ is associated with a composite attribute vector $\mathbf{y} = (y_1, y_2, \ldots, y_k)$, where each element y_i is drawn from a discrete attribute space $\mathcal{Y}_i = \{y_i^1, y_i^2, \ldots, y_i^{m_i}\}$ with $m_i = |\mathcal{Y}_i|$ outcomes or *classes*. The **total attribute space** is then defined as the Cartesian product $\mathcal{Y} = \mathcal{Y}_1 \times \mathcal{Y}_2 \times \cdots \times \mathcal{Y}_k$. We assume that these attributes are conditionally independent, such that their joint distribution given $\mathbf{x}$, factorizes as:

$$p(\mathbf{y} \mid \mathbf{x}) = \prod_{i=1}^{k} p(y_i \mid \mathbf{x}) \tag{1}$$

where (1) implies that any attribute y_i does not provide additional information about any other attribute y_j once the

sample $\mathbf{x}$ is observed.

Decentralization. We consider a decentralized setting where empirical observations of $(\mathbf{x}, \mathbf{y})$ pairs are distributed across n localized datasets, $\mathcal{D} = \{D_1, D_2, \dots D_n\}$. We also assume the following locality constraint over the data:

Constraint 1.

A sample $\mathbf{x} \in D_a$ is accessible only to client c_a. For any $a \neq b$, $D_a \cap D_b = \emptyset$ with respect to raw data exchange.

Note that, each client c_a possesses knowledge of the total attribute space $\mathcal{Y}$. Even if a specific $y_i^q \in \mathcal{Y}_i$ never occurs in D_a, the client is aware of y_i^q. Constraint 1 is thus defined over the raw data points $\mathbf{x}$ without involving the labels $\mathbf{y}$.

Coverage. Since individual clients c_a only observe a subset of the data space $\mathcal{X}$, we define two *coverage* metrics to quantify data availability. The **total coverage**, $\mathcal{C} : \mathcal{Y} \times \mathcal{D} \to [0, 1]$, denotes fraction of the combinations in $\mathcal{Y}$ that appear at least once in the aggregate dataset $\mathcal{D}$. Similarly, **marginal coverage**, $\mathcal{M}_i : \mathcal{Y}_i \times \mathcal{D} \to [0, 1]$, is the fraction of outcomes of a marginal attribute $\mathcal{Y}_i$ that are observed across the union of all localized datasets in $\mathcal{D}$. For the total attribute space $\mathcal{Y}$ to be feasible for learning, we require full global marginal coverage, $\mathcal{M}_i = 1 \; \forall i \in \{1, \dots, k\}$ (Wiedemer et al., 2023). This condition ensures that every attribute outcome is represented somewhere in the decentralized system, even if the combinatorial space $\mathcal{Y}$ is sparsely populated. To achieve this, a naive requirement would be to ensure we have at least as many samples as the largest attribute space i.e., $C > C_{\text{m}} = \max\{|\mathcal{Y}_i|\}_{i=1}^k / |\mathcal{Y}|$. However, C_m is insufficient to truly disentangle the attributes, as it does not guarantee that we can observe the variation of one attribute while others are held constant (or accounted for). We thus define a more robust **minimum feasible coverage**.

Definition 3.1 (Minimum Feasible Coverage)**.** To achieve full global marginal coverage ($\mathcal{M}_i = 1, \forall i$) in a manner that allows for attribute disentanglement, the total coverage $\mathcal{C}$ must satisfy the lower bound:

$$\mathcal{C} \geq C_{\text{min}} = \frac{1 + \sum_{i=1}^{k} (|\mathcal{Y}_i| - 1)}{|\mathcal{Y}|} \tag{2}$$

The numerator in (2) corresponds to the minimum number of unique observations required to satisfy all main-effect degrees of freedom in a compositional model. We assume that $\mathcal{D}$ satisfies $\mathcal{M}_i = 1$ globally with a total coverage $C \geq C_{min}$, which aligns with the coverage regimes explored in Wiedemer et al. (2023) and Gaudi et al. (2025).

Problem. Let $G_\theta : \mathcal{Z} \times \mathcal{Y} \to \mathcal{X}$ be a generative model that maps a latent noise variable $\mathbf{z} \in \mathcal{Z}$ and the composite attribute vector $\mathbf{y} \in \mathcal{Y}$ to the data space. Our goal is to learn the parameters θ over the decentralized data $\mathcal{D}$ such that G_θ can sample from $p(\mathbf{x} \mid \mathbf{y})$, even for attribute combinations $\mathbf{y} \in \mathcal{Y}$ that were unobserved during training (i.e., $\mathbf{y} \notin \mathcal{D}$). We formalize the inference objective as the simultaneous satisfaction of all marginal constraints:

Problem 1.

$$\max_{\theta} \mathop{\mathbb{E}}_{\mathbf{z} \in \mathcal{Z}, \mathbf{y} \in \mathcal{Y}} \left[\prod_{i=1}^{k} \mathbb{1} \left\{ G_\theta(\mathbf{z}, \mathbf{y}) \in \text{supp}(p(\mathbf{x} \mid y_i)) \right\} \right]$$

s.t. Constraint 1.

where the indicator function $\mathbb{1}\{\cdot\}$ enforces that for any combination $\mathbf{y}$, the generated sample $G_\theta(\mathbf{z}, \mathbf{y})$ must reside within the intersection of the supports of all k marginal conditional distributions $\{p(\mathbf{x}|y_i)\}_{i=1}^k$. This formulation implies that G_θ must achieve **compositional generalization** by correctly composing independent attribute representations learned from decentralized, sparse observations.

3.2. Flow Matching and Composition

We interpret G_θ as a flow matching model (Lipman et al., 2022; Liu et al., 2022b), a widely used family of state-of-the-art generative models that are strongly connected to diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020) and score-based generative models (Song & Ermon, 2019; Song et al., 2021). Flow matching learns a time-dependent velocity field $\mathbf{v}_\theta(\cdot, t)$ that generates a probability path to transform a source distribution $p_0(\mathbf{x})$ to a target distribution $p_1(\mathbf{x})$. We can define $G_\theta(\cdot)$ as an integration over $\mathbf{v}_\theta(\cdot, t)$ as follows:

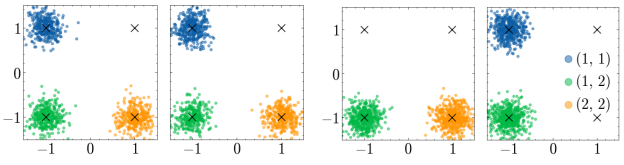
$$G_\theta(\mathbf{z}, \mathbf{y}) \coloneqq \mathbf{x}_0 + \int_0^1 \mathbf{v}_\theta(\mathbf{x}_t, t, \mathbf{y}) dt \tag{3}$$

where $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is equivalent to the latent noise variable $\mathbf{z}$. For brevity of notation, we choose to omit t and use $\mathbf{v}_\theta(\mathbf{x}_t)$ to imply $\mathbf{v}_\theta(\mathbf{x}_t, t)$. We also use the notation $\mathbf{v}_\theta^{(a)}$ to indicate a model local to some client c_a with distinct local parameters θ_a, and sole access to the dataset $D_a \in \mathcal{D}$.

Compositional sampling. Consider a novel condition vector $\mathbf{y} = (y_1, y_2, \dots, y_k) \notin \mathcal{D}$ but with known marginal attributes $y_i \in \mathcal{D} \; \forall i$. Previous works involving diffusion models, such as Liu et al. (2022a) and Ajay et al. (2023), sample from the product of known marginals as follows:

$$\hat{\epsilon}_\theta(\mathbf{x}_t, \mathbf{y}) = \epsilon_\theta(\mathbf{x}_t) + \sum_{i=1}^{k} w_i \left(\epsilon_\theta(\mathbf{x}_t, y_i) - \epsilon_\theta(\mathbf{x}_t) \right) \tag{4}$$

where w_i is analogous to the classifier-free guidance (Ho & Salimans, 2022) strength per attribute. Since the velocity field $\mathbf{v}_t$ in flow matching with Gaussian probability paths coincides with the score function $\nabla \log p_t(x)$ (up to a time-dependent scaling factor), the linear composition of scores



(a) IID partition. *(b)* Non-IID partition.

Figure 2. A set of two decentralized private datasets, $\{D_1, D_2\}$, where each point is associated with an attribute vector $\mathbf{y} \in \{1, 2\}^2$.

in (4) translates directly to the velocity space as follows:

$$\hat{\mathbf{v}}_\theta(\mathbf{x}_t, \mathbf{y}) = \mathbf{v}_\theta(\mathbf{x}_t) + \sum_{i=1}^{k} w_i \left(\mathbf{v}_\theta(\mathbf{x}_t, y_i) - \mathbf{v}_\theta(\mathbf{x}_t)\right) \quad (5)$$

We show the full justification behind Eq. (5) in § A.

Enforcing Conditional Independence. Both (Liu et al., 2022a) and (Ajay et al., 2023) note that (4) requires the attributes to be conditionally independent (1). To enforce this structure when $\mathcal{C}$ is sparse, we adapt the penalty from Gaudi et al. (2025) to the flow matching objective:

$$\mathcal{L}_{\text{CI}}(\theta) = \mathbb{E}_{t, \mathbf{x}_t, \mathbf{y} \sim D_a} \left\| \mathbf{v}_\theta(\mathbf{x}_t, \mathbf{y}) - \hat{\mathbf{v}}_\theta(\mathbf{x}_t, \mathbf{y}) \right\|^2 \quad (6)$$

where $\hat{\mathbf{v}}_\theta$ is the compositional field defined in (5), with $w_i = 1, \forall i$. By minimizing the discrepancy between the joint velocity field and the sum of its marginal counterparts, we encourage the model to learn a disentangled representation that generalizes to unobserved compositions.

4. Are Decentralized Flows Compositional?

We elucidate Prob. 1 with a simple example consisting of a mixture of Gaussian distributions. Suppose we have a set of two decentralized private datasets, $\mathcal{D} = \{D_1, D_2\}$. Each data point $\mathbf{x}$ in $\mathcal{D}$ is associated with a horizontal attribute $\mathcal{Y}_1 \in \{\text{left}, \text{right}\}$ and a vertical attribute $\mathcal{Y}_2 \in \{\text{top}, \text{bottom}\}$. By assigning a numeric label to each class, the total attribute space becomes $\mathcal{Y} \in \{1, 2\}^2$. Importantly, the particular combination $\mathbf{y} = (2, 1)$ does not occur in $\mathcal{D}$.

Though $\mathcal{D}$ can be arbitrarily partitioned, we show two informative configurations: (i) an IID partition, where both nodes possess all three available modes, and (ii) a non-IID partition, where both D_1 and D_2 observe two modes each, as shown in Fig. 2. We adopt three decentralized baselines to learn a flow matching model on $\mathcal{D}$ (results in Fig. 3).

(i) Federated Flow. We train a standard flow matching model $\mathbf{v}_\theta$, with iterations of parameter averaging, following Tun et al. (2023). We find that, for non-IID partition, the federated flow shows poor convergence. Further, even under IID partition, the model cannot recover the missing mode.

(ii) DDM (McAllister et al., 2025) trains an independent flow matching model on each node a, $v_\theta^{(a)}$, with no communication with any other node $b \neq a$ during training.

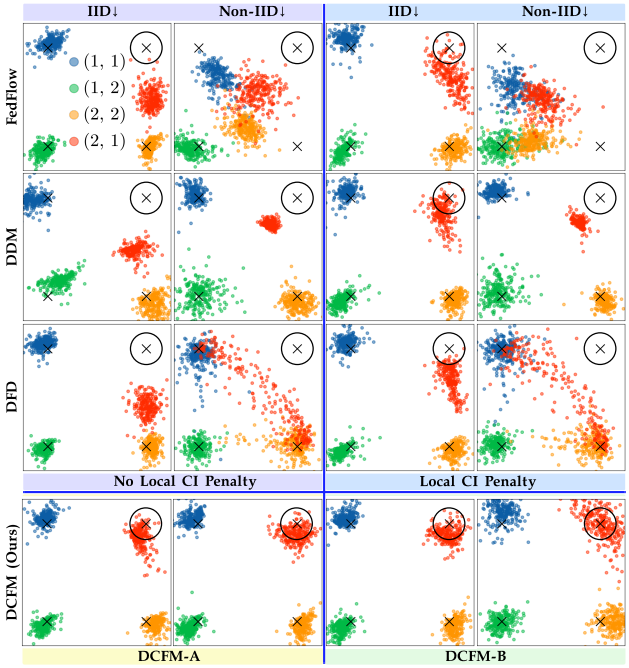


Figure 3. **Left two columns:** Prior methods fail to recover the unobserved $\mathbf{y} = (2, 1)$ mode (circled). **Right two columns:** Local conditional independence does not help in non-IID case. **Bottom:** DCFM recovers the missing mode in both IID and non-IID cases.

Although McAllister et al. (2025) perform a data clustering step before training, we avoid this step as it involves breaking the locality assumption in Constraint 1. After training, the models are shared to some central server (or transmitted to each other) for inference. We construct a lightweight router for the set of models using marginal label statistics. From Fig. 3, we observe that that DDM shows good recovery of collective decentralized data, in either partitioning. However, it struggles to generate the missing mode.

(iii) DFD (Hahn & Lee, 2025) involves a similar process of training local experts as DDM. However, instead of learning a router, DFD estimates the energy (unnormalized densities) to approximate a weight factor for each expert. Similar to DDM, DFD shows good coverage of the known regions in $\mathcal{D}$. However, under non-IID settings, the energy-based routing system fails for novel attribute composition.

Why these models fail. These models primarily fail to recover the missing mode not due to decentralization, but rather since the assumption of conditional independence (1) necessary for compositional generalization does not hold.

Can local conditional independence lead to global compositional generalization?

If the compositional generalization failure of the models stems from the lack of conditional independence, one may naturally pose the question: would enforcing the conditional independence penalty (6) following Gaudi et al. (2025) during local training help recover the missing mode?

The answer: not necessarily. Even if every local $p_a(\mathbf{y} \mid \mathbf{x})$ are conditionally independent, we do not have guarantees that the mixture $p^*(\mathbf{y} \mid \mathbf{x}) = \sum_{a=1}^{n} w_a p_a(\mathbf{y} \mid \mathbf{x})$ satisfies (1). We observe in Fig. 3 that adding a CI penalty *can be helpful* under IID conditions, but still results in failure when the data are non-IID.

Our approach, DCFM, is primarily concerned with how to learn *global* conditional independence, over the collective attribute space of all the clients. As shown in Fig. 3, DCFM enables mode recovery under both IID and non-IID settings.

5. Decentralized Compositional Flow Models

We now present two variants of DCFM corresponding to an idealized mixture-of-experts and a more practically efficient monolithic model. Both variants share a first stage of learning local models trained on local data. An overview of DCFM is shown in Fig. 4.

5.1. Stage I: Local Matching

Similar to mixture-of-expert methods like DDM and DFD, we first optimize local models $\mathbf{v}_\theta^{(a)}$ in order to sufficiently learn the distribution of locally available data. This phase is similar to standard FM training with two minor modifications: (1) marginal label learning, and (2) local CI penalty.

Marginal labels. We observe from (4) and (5) that inference utilizes marginal labels, y_i. However, during standard training, flow matching models usually have access to full joint labels $\mathbf{y}$ and unconditional labels $\{\emptyset\}^k$. Thus, with a small probability p_{marg}, we find it helpful to introduce marginal labels $(\emptyset, \ldots, y_i, \ldots, \emptyset)$ to the model.

Let $\mathbf{m} \in \{0,1\}^k$ be a random binary masking vector, drawn from some distribution $p(\mathbf{m})$. We then construct a masked attribute vector $\mathbf{y_m} = \mathbf{y} \odot \mathbf{m}$, where an attribute y_i is replaced by a "null" token $\emptyset$ if $m_i = 0$. We define $p(\mathbf{m})$ as:

$$p(\mathbf{m}) = \begin{cases} \pi_{\text{full}} & \text{if } \mathbf{m} = \mathbf{1} \\ \pi_{\text{marg}} \cdot \frac{1}{k} & \text{if } \mathbf{m} = \mathbf{e}_i, \text{ for } i \in \{1, \ldots, k\} \\ \pi_{\text{uncond}} & \text{if } \mathbf{m} = \mathbf{0} \end{cases} \tag{7}$$

$\pi_{\text{full}}, \pi_{\text{marg}}, \pi_{\text{uncond}} \in [0,1]$ are mixture weights that sum to 1. The local training objective becomes:

$$\mathcal{L}_{\text{FM}}^{(a)}(\theta) = \mathop{\mathbb{E}}_{\substack{t,\mathbf{y},\mathbf{m} \\ \mathbf{x}_t \sim p_t}} \left[\left\| \mathbf{v}_\theta^{(a)}(\mathbf{x}_t, t, \mathbf{y} \odot \mathbf{m}) - \mathbf{u}_t(\mathbf{x}_t | \mathbf{x}_1) \right\|^2 \right] \tag{8}$$

where $t \sim \mathcal{U}[0,1]$ is the timestep, and $\mathbf{u}_t(\mathbf{x}_t \mid \mathbf{x}_1)$ is the ground-truth conditional velocity field, which for a linear path is simply $\mathbf{x}_1 - \mathbf{x}_0$ (where $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\mathbf{x}_1 \in D_a$).

Local CI. We adopt the local CI penalty (6) when the full labels are available. If $\hat{\mathbf{v}}$ denotes the composition of marginal velocities (5), we define the local penalty as:

$$\mathcal{L}_{\text{CI}}^{(a)}(\theta) = \mathop{\mathbb{E}}_{t,\mathbf{y},\mathbf{x}_t \sim p_t} \left[\mathbf{v}_\theta^{(a)}(\mathbf{x}_t, \mathbf{y}) - \hat{\mathbf{v}}_\theta^{(a)}(\mathbf{x}_t, \mathbf{y}) \right] \tag{9}$$

The total local loss becomes:

$$\mathcal{L}_{\text{total}}^{(a)}(\theta) = \mathcal{L}_{\text{FM}}^{(a)}(\theta) + \lambda \cdot \mathcal{L}_{\text{CI}}^{(a)}(\theta) \tag{10}$$

Model aggregation. After Stage I, each client shares their model with either a trusted server, or with other clients.

5.2. DCFM-A: Learning Local Experts With Cross-Peer Conditional Independence.

We consider that the set of clients perform an `allgather` operation over θ, and that each client now has access to a set of local experts, $\{\mathbf{v}_\theta^{(a)}\}_{a=1}^n$. Next, let $\mathbf{r} = (r_0, r_1, r_2, \ldots, r_k)$ be a routing vector, where any $r_i \in \{1, 2, \ldots, n\}$. For $i = 1 \ldots k$, r_i assigns a model for each of the k marginal attributes $\{y_i\}_{i=1}^k$, while r_0 is a model choice for the unconditional attribute $\emptyset$. The routing vector $\mathbf{r}$ allows us to redefine the conditional independence penalty (9) over an arbitrary combination of models as,

$$\mathbf{z}_\theta^{(r_0)}(\mathbf{x}_t) = \bar{\mathbf{v}}_\theta^{(r_0)}(\mathbf{x}_t) + w \sum_{i=1}^{k} \left(\bar{\mathbf{v}}_\theta^{(r_i)}(\mathbf{x}_t, y_i) - \bar{\mathbf{v}}_\theta^{(r_0)}(\mathbf{x}_t) \right)$$

$$\mathcal{L}_{\text{peerCI}}^{(a)}(\theta) = \mathop{\mathbb{E}}_{\mathbf{r},t,\mathbf{y},\mathbf{x}_t \sim p_t} \left[\mathbf{v}_\theta^{(a)}(\mathbf{x}_t, \mathbf{y}) - \mathbf{z}_\theta^{(r_0)}(\mathbf{x}_t) \right] \tag{11}$$

where $\bar{\mathbf{v}}_\theta^{(\cdot)}$ is a frozen model when $r \neq a$ and is defined as

$$\bar{\mathbf{v}}_\theta^{r} = \delta_{r,a} \mathbf{v}_\theta^{(a)} + (1 - \delta_{r,a}) \text{StopGrad}(\mathbf{v}_\theta^{(r)}). \tag{12}$$

where $\delta_{i,j}$ is the Kronecker delta. That is, we only update the parameters of the local model $\mathbf{v}_\theta^{(a)}$ while keeping the peer experts frozen. Further, any assignment $\mathbf{r}$ ensures that there is at least a single $r_i = a$. The total loss is now,

$$\mathcal{L}_{\text{DCFM-A}}^{(a)}(\theta) = \mathcal{L}_{\text{FM}}^{(a)}(\theta) + \lambda \cdot \mathcal{L}_{\text{peerCI}}^{(a)}(\theta) \tag{13}$$

Thus, (13) trains local experts $\mathbf{v}_\theta^{'(a)}$ that learn conditional independence both locally and across peers. As seen in Fig. 3 (bottom), sampling from a mixture of DCFM-A experts shows compositional generalization even when the local distributions are non-IID.

5.3. DCFM-B: Learning Globally Conditionally Independent Student.

One problem with DCFM-A is that, after a round of training, we obtain a new set of models $\{\mathbf{v}_\theta^{'(1)}, \mathbf{v}_\theta^{'(2)}, \ldots, \mathbf{v}_\theta^{'(k)}\}$. However, each updated model learns conditional independence with respect to the frozen models $\{\mathbf{v}_\theta^{(b)}\}_{b \neq a}$ that are obtained as a result of stage I, rather than the updated set $\{\mathbf{v}_\theta^{'(b)}\}_{b \neq a}$. This causes a potential mismatch between the

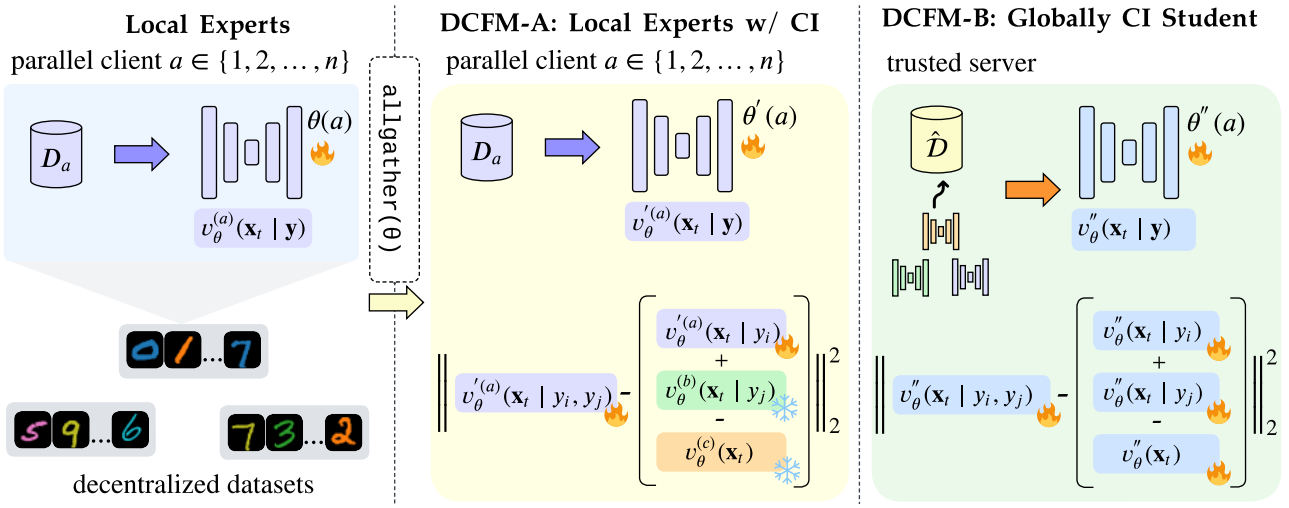


Figure 4. An overview of DCFM with 🔥 and ❄ indicating trainable and frozen parameters respectively. Stage 1 (left) trains local experts on local data. DCFM-A (middle) trains local experts with with cross-peer conditional independence constraints. DCFM-B (right) learns a globally conditionally independent monolithic student model.

models, solving which requires repeating the DCFM-A procedure several times until convergence. Although DCFM-A achieves convergence in a single iteration over the simple datasets shown in Fig. 2, this is not the case in more practical domains, such as images. Further, repeated training stages add to the computation and communication expense of training this group of local models.

We observe that, unlike discriminative models $F : \mathcal{X} \to \mathcal{Y}$ with intractable input space $\mathcal{X}$, the input space $\mathcal{Z}$ of a generative model G is tractable for every decentralized client. Previous works, such as rectified flows (Liu et al., 2022b) demonstrate that it is possible to optimize a flow model on its self-generated samples. We thus consider distilling a group of local experts into a single student. This distillation can be done on a trusted server node, or even any single client node after a single model exchange operation.

We define a peer matching objective as follows:

$$\mathcal{L}_{\text{student}}(\theta) = \mathop{\mathbb{E}}_{t,r,\hat{\mathbf{x}}_t} \left[\left\| \mathbf{v}_\theta(\hat{\mathbf{x}}_t, t, \mathbf{y} \odot \mathbf{m}) - \bar{\mathbf{v}}_\theta^{(r)}(\hat{\mathbf{x}}_t, \mathbf{y}) \right\|^2 \right] \tag{14}$$

where $\hat{\mathbf{x}}_t = \text{ODEInt}(\mathbf{x}_0, \bar{\mathbf{v}}_\theta^{(r)}, \mathbf{y}, 0, t)$ and $\bar{\mathbf{v}}$ is a frozen teacher. So, if $\hat{\mathbf{x}}_t$ belongs to the probability path of model $\bar{\mathbf{v}}_\theta^{(r)}$, we treat its response at $\hat{\mathbf{x}}_t$ as the ground truth velocity.

Distilling peer velocities following (14) allows us to generate samples that cover the union of $\mathcal{D}$. But, it does not sufficiently enable the recovery of missing combinations of $\mathbf{y}$. So, we again enforce conditional independence constraints between attributes during the peer distillation process. We add the following penalty to the training objective:

$$\mathbf{z}_\theta(\hat{\mathbf{x}}_t) = \left(\mathbf{v}_\theta(\hat{\mathbf{x}}_t) + \sum_{i=1}^{k} \left(\mathbf{v}_\theta(\hat{\mathbf{x}}_t, y_i) - \mathbf{v}_\theta(\hat{\mathbf{x}}_t) \right) \right)$$
$$\mathcal{L}_{\text{studentCI}}(\theta) = \mathbb{E}_{t,\mathbf{y},\hat{\mathbf{x}}_t} \left[\| \mathbf{v}_\theta(\hat{\mathbf{x}}_t, \mathbf{y}) - \mathbf{z}_\theta(\hat{\mathbf{x}}_t) \|_2^2 \right] \tag{15}$$

Note that, we do not use any frozen model in (15), unlike (11). While learning the collective velocity fields of each peer, we also want to learn global marginals that are conditionally independent. The overall DCFM-B objective is:

$$\mathcal{L}_{\text{DCFM-B}}(\theta) = \mathcal{L}_{\text{student}}(\theta) + \lambda \mathcal{L}_{\text{studentCI}}(\theta) \tag{16}$$

As seen in Fig. 3 (bottom), instead of simply aggregating non-IID experts into a single model, DCFM-B also successfully explores their implied unobserved combination.

6. Experiments

DCFM is designed to train generative models within isolated data silos while enabling compositional generalization to novel attribute combinations. We evaluate DCFM based on two primary objectives: (1) **Compositional Generalization:** Assessing the model's ability to generate valid samples for attribute combinations unobserved during decentralized training, and (2) **Variant Trade-offs:** Characterizing the performance of the two DCFM variants on both known and novel compositions. To demonstrate the framework's versatility, we consider three distinct benchmarks: image composition via Colored MNIST (§ 6.1), medical imaging on Chest X-rays (§ 6.3), and spatial composition for unconstrained offline robotic planning (§ 6.2).

6.1. Colored MNIST

Experiment Setting. The Colored MNIST dataset (LeCun, 1998) consists of images $\mathbf{x} \in \mathbb{R}^{28\times28\times3}$ with corresponding $k = 2$ discrete attribute spaces, $\mathcal{Y}_1$ and $\mathcal{Y}_2$, which represent a set of ten numbers and ten colors respectively. The total attribute space, $\mathcal{Y}$ has $|\mathcal{Y}_1||\mathcal{Y}_2| = 100$ combinations. We consider these images to be distributed across $n = 10$ decentralized datasets, $\mathcal{D} = \{D_a\}_{a=1}^{n}$ with a total coverage of $C = 1/2$; that is, half of $\mathcal{Y}$ remains unobserved. Further, we consider both IID and non-IID partition setups over $\mathcal{D}$. The non-IID partition utilizes the Dirichlet distribution $\text{Dir}(\alpha = 0.1)$.

Metrics. We utilize FID (Heusel et al., 2017), Precision (P), and Recall (R) (Kynkäänniemi et al., 2019) to measure the performance of G_θ. We further disentangle these metrics in terms of *known* (observed across all clients in $\mathcal{D}$) and *novel* (denoted by superscripts o and $*$ respectively).

Table 1. Performance on MNIST under IID and Non-IID settings.

	IID		Non-IID	
Method	FID^o ↓	FID* ↓	FID^o ↓	FID* ↓
FedFlow	9.41	20.83	15.02	20.02
FedFlow+L	12.37	18.65	15.81	17.12
DDM	**8.19**	25.19	7.46	17.98
DDM+L	9.03	16.38	7.99	18.84
DFD	8.81	29.71	**7.02**	38.27
DFD+L	9.58	19.13	8.17	31.36
DCFM-A (Ours)	8.53	**11.41**	7.33	9.29
DCFM-B (Ours)	9.32	12.24	8.49	**9.15**

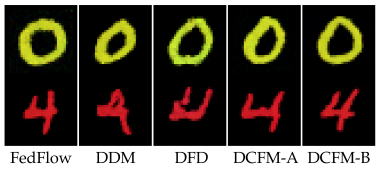


Figure 5. Novel compositions generated by DCFM vs. baselines.

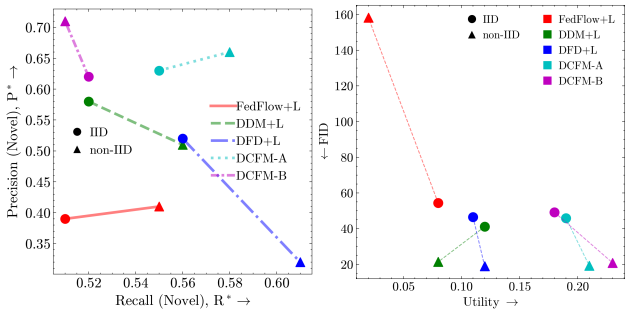


Figure 6. (Left) Precision vs. Recall on novel Colored MNIST compositions. (Right) FID vs. Utility on NIH Chest X-ray 14.

Results and Analysis. In Tab. 1, we compare DCFM with other decentralized approaches, where a +L indicates a model trained with a *local* CI penalty (1). We find that DFD (Hahn & Lee, 2025) shows the best result in recovering the known data, particularly under challenging non-IID conditions. However, they show poor generalization under novel attribute compositions, demonstrating a strong tendency to steer the samples towards known data regions. In contrast, both DCFM-A and DCFM-B show good performance in terms of FID^o and FID*, thus narrowing the gap between known and novel. We show some qualitative examples of novel combination generation in Fig. 5.

Fig. 6 (left) shows the novel Precision (P*) vs. novel Recall (R*), using combinations that are unobserved in the decentralized $\mathcal{D}$. We observe that DCFM shows a notably improved P* compared to baselines, suggesting it is better able to capture the important components of the missing data. We find that DCFM-B shows a reduced recall score compared to DCFM-A, which indicates some loss in diversity due to learning from synthetic data.

6.2. Unconstrained Offline Robotic Planning

Experiment Setting. The `cube-single-play` dataset from the OGBench benchmark (Park et al., 2025) contains trajectory data of a robot arm that moves a cube around within a confined 3D space. We characterize a complete trajectory plan by $k = 2$ attributes $(\text{Src}, \text{Dest})$, which indicate the location where the cube is picked up, and the destination where it is dropped off by the robot arm respectively. To make the attributes categorical, we divide the horizontal XY plane into 4 quadrants. We thus have two attributes, $\mathcal{Y}_1, \mathcal{Y}_2 \in \{\text{TL}, \text{BL}, \text{BR}, \text{TR}\}$, and the total attribute space $\mathcal{Y}$ has a cardinality of $|\mathcal{Y}_1||\mathcal{Y}_2| = 16$.

Table 2. Success rate (SR) of DCFM vs. baselines on `cube-single-play`, averaged over 3 groups.

Method	**Partition**	**SR^o**	**SR***
FedFlow+L	IID	$\mathbf{58.33 \pm 2.62}$	39.67 ± 4.11
(Tun et al., 2023)	Non-IID	27.67 ± 7.76	9.0 ± 3.74
DDM+L	IID	53.33 ± 2.05	38.33 ± 5.73
(McAllister et al., 2025)	Non-IID	67.67 ± 2.87	29.67 ± 2.87
DFD+L	IID	58.0 ± 2.16	40.0 ± 4.9
(Hahn & Lee, 2025)	Non-IID	$\mathbf{68.67 \pm 1.25}$	18.33 ± 5.44
DCFM-A	IID	57.67 ± 2.62	$\mathbf{56.33 \pm 5.31}$
(Ours)	Non-IID	68.33 ± 2.87	53.0 ± 2.94
DCFM-B	IID	54.0 ± 3.74	53.33 ± 2.36
(Ours)	Non-IID	65.67 ± 2.49	$\mathbf{54.67 \pm 2.49}$

Following (Janner et al., 2022), we define a trajectory $\mathbf{x}$ as a tuple $[\mathbf{s}, \mathbf{a}]$, where $\mathbf{s} = (\mathbf{s}^0, \mathbf{s}^1, \ldots, \mathbf{s}^{H-1})$ is a sequence of *observations*, $\mathbf{a} = (\mathbf{a}^0, \mathbf{a}^1, \ldots, \mathbf{a}^{H-1})$ is a corresponding sequence of *actions*, and H is the planning horizon. We want to learn a planning model, $\mathbf{v}_\theta([\mathbf{s}_t, \mathbf{a}_t], t, \mathbf{y})$, where $\mathbf{s}_t$ and $\mathbf{a}_t$ indicate a noisy trajectory at timestep t.

The trajectory data is split across $n = 2$ decentralized datasets $\mathcal{D} = \{D_1, D_2\}$ for both IID and non-IID parti-

tions. We set the total coverage to $3/4$, omitting trajectory data with the four attribute combinations (TL, BR), (TR, BL), (BR, TL), and (BL, TR) respectively. That is, the robot may have data of moving between horizontal quadrants (like TL↔TR) or vertical quadrants (TL↔BL). But it has no observed information about moving between diagonal quadrants, like (BL↔TR). In order to learn diagonal movement, the robot planner needs to understand that *the destination of a trajectory is independent of its source*. We provide a visual overview of the problem in Fig. 8.

Goal. We rely on the planner to suggest a feasible goal. Given a condition vector $\mathbf{y}$, the flow planner $\mathbf{v}_\theta$ generates a trajectory $\mathbf{x} = [\mathbf{s}, \mathbf{a}]$. We take the first and final observations within a trajectory, $\mathbf{s}_0$ and $\mathbf{s}_{H-1}$, and extract the cube positions c^0 and c^{H-1}, respectively. We consider a plan *feasible* if: (i) the cube is on the floor at both c^0 and c^{H-1}, (ii) c^0 is located at the given source quadrant y_1, and (iii) c^{H-1} is at the given destination quadrant y_2. Only if the initial plan defines a feasible goal c^{H-1}, we execute the plan with the inpainting-based method proposed by Janner et al. (2022).

Evaluation Metrics. We report Success Rate (SR), defined as the percentage of trials where: (a) the generated plan's start and end states match the queried quadrants, and (b) the executed trajectory places the cube within a threshold distance of the target.

Results and Analysis. From Tab. 2, we observe that DCFM significantly outperforms all baselines in planning under novel attribute compositions, while retaining comparable performance for the compositions known in the dataset.

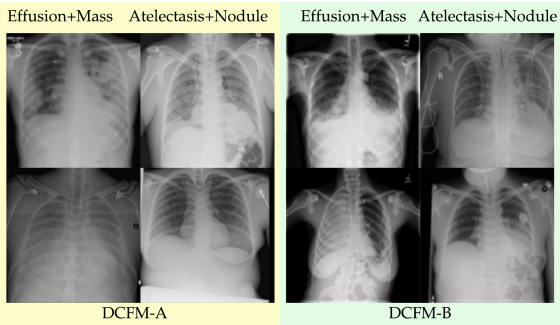


Figure 7. Disease combinations generated by DCFM.

6.3. Disease Co-Occurrence in Chest X-Rays

Experiment Setting. The NIH Chest X-ray 14 dataset (Wang et al., 2017) contains over 100K chest X-ray (CXR) images labeled with 14 disease attributes. We use images of size 256×256, that are distributed across $n = 4$ decentralized datasets $\mathcal{D}$ with IID and non-IID partitions.

Attribute Sparsity. Given $k = 14$ binary attributes, $|\mathcal{Y}|$ grows steeply to 2^{14}. However, a large portion of the product space is infeasible. For example, it is impossible to see all 14 diseases occur simultaneously, and at most 9 diseases ever co-occur together in the training data (with a count of 2). We find that ∼99% of the training data contains 3 or fewer co-occurring diseases. Instead of using the total attribute space $\mathcal{Y}$, we define a feasible attribute space $\mathcal{Y}_\mathcal{F}$ as the set of all $\mathbf{y} \in \mathcal{Y}$ with a frequency of at least 0.1% in $\mathcal{D}$. This leaves us with a feasible space $\mathcal{Y}_\mathcal{F}$ with 54 combinations. **Goal.** Our objective is to correctly synthesize the disease combinations in $\mathcal{Y}_\mathcal{F}$, by learning a decentralized model that disentangles the conditional dependencies between diseases.

Evaluation Metrics. We use FID to measure the generative performance of the trained decentralized flows. We do not separate the FID into known vs. novel, as we lack sufficient ground truth data for the rare combinations in the dataset. To quantify the usefulness of the synthetic samples, we measure **utility (U)**, as the *compositional* recall (sensitivity) score on the real test set of a downstream classifier trained on purely synthetic data. Suppose $\mathcal{A}$ is a set of co-occurring diseases. We define a compositional recall as,

$$R(\mathcal{A}) = \frac{\{\mathbf{y}_{\text{pred}} = \mathbb{1}_\mathcal{A}\} \wedge \{\mathbf{y}_{\text{true}} = \mathbb{1}_\mathcal{A}\}}{\text{count}(\mathbf{y}_{\text{true}} = \mathbb{1}_\mathcal{A})}, \qquad (17)$$

where $\mathbb{1}_\mathcal{A} \in \{0, 1\}^k$ is a binary attribute vector that is 1 at the indices $\mathcal{A}$. We define U as the macro-average of Eq. (17) over the distinct combinations in $\mathcal{Y}_\mathcal{F}$. **Models.** Across all baselines, we train a latent flow matching model (Rombach et al., 2022; Esser et al., 2024) at 256×256 resolution, utilizing the latent space of a pretrained autoencoder. We use a ResNet-18 model as the downstream classifier.

Results and Analysis. From Fig. 6 (right), we observe that, though DCFM achieves similar quality as other decentralized generative approaches, a classifier trained on DCFM shows significantly improved sensitivity to the joint occurrence of chest diseases, suggesting that DCFM achieves better attribute disentanglement. Fig. 7 shows some qualitative examples of images generated by DCFM.

6.4. Comparing DCFM-A and DCFM-B

Computation. Compositional inference (5) requires $O(k)$ evaluations over the marginal attributes, and for each attribute, DCFM-A uses a mixture of n models. DCFM-A thus has an inference complexity of $O(k \times n)$, compared to $O(k)$ for DCFM-B. Further, in terms of GPU-hour utilization during training, we empirically observe that DCFM-A is at least $2\times$ as expensive as DCFM-B. **Communication Cost.** DCFM-A and DCFM-B require $n - 1$ and one `(send, recv)` operation, respectively, per client. We briefly discuss costs related to baselines in § C.

Performance. From observing Tab. 1, Tab. 2, and Fig. 6, we consistently find that DCFM-B underperforms w.r.t. DCFM-A on the *known* attributes. Further, Fig. 6 (left) shows that DCFM-B has reduced recall, suggesting reduced diversity

of the samples. We hypothesize this to be caused by the synthetic distillation process, as the experts obtained from Stage I are not necessarily perfect approximators of the real data. Nevertheless, DCFM-B is able to achieve comparable or better performance on the unobserved or novel attributes while requiring less computation and communication.

7. Concluding Remarks

We introduced Decentralized Compositional Flow Matching (DCFM), a framework that enables generative models to achieve compositional generalization across isolated data silos without exchanging raw data. Unlike prior decentralized approaches, which often suffer from factor entanglement or incompatible latent spaces, DCFM structurally enforces conditional independence through peer-to-peer knowledge distillation in a shared flow velocity space. This design allows novel attribute combinations to emerge from decentralized sources, even when no individual silo possesses the information required to support such compositions. Across diverse benchmarks, including Colored MNIST, chest X-rays, and robotic spatial planning, DCFM significantly outperforms federated learning and mixture-of-experts baselines. This work establishes a foundation for scaling generative systems in locality-sensitive and resource-constrained environments, where the ability to synthesize global knowledge from fragmented, localized observations is essential.

Acknowledgements: This work was supported by the National Science Foundation (award #2500983). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of NSF.

Impact Statement

We can characterize the present state of machine learning as follows: the model architectures and training algorithms are known; the model weights may even publicly available to download (for instance, Stable Diffusion or FLUX models). However, the data always remains private. With open architectures and training methods, curated data often ends up as the deciding factor in the performance of generative models. It is likely that people will be significantly less inclined to share their private, local data in the near future, as data would become something like an *asset*. Under these circumstances, it is important to investigate along the lines of our paper: how should we learn good decentralized generative models, that do not violate the locality of data?

We also envision a promising avenue of research at the intersection of compositional generalization and decentralized learning. Despite their apparent differences, they are structurally isomorphic: both are concerned with constructing a 'whole' from isolated components. We posit that, under specific conditions, the problem of decentralized learning can be effectively reframed as a problem of compositional generalization.

References

Ajay, A., Du, Y., Gupta, A., Tenenbaum, J., Jaakkola, T., and Agrawal, P. Is conditional generative modeling all you need for decision-making? In *International Conference on Learning Representations*, 2023.

Augenstein, S., McMahan, H. B., Ramage, D., Ramaswamy, S., Kairouz, P., Chen, M., Mathews, R., et al. Generative models for effective ml on private, decentralized datasets. *International Conference on Learning Representations*, 2020.

de Goede, M., Cox, B., and Decouchant, J. Training diffusion models with federated learning. *arXiv preprint arXiv:2406.12575*, 2024.

Du, Y., Li, S., and Mordatch, I. Compositional visual generation with energy based models. *Advances in Neural Information Processing Systems*, 2020.

Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

Gaudi, S., Sreekumar, G., and Boddeti, V. Coind: Enabling logical compositions in diffusion models. In *International Conference on Learning Representations*, 2025.

Hahn, S.-J. and Lee, J. Diffusion federated dataset. In *Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=1GCWcrZTX8.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

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

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Janner, M., Du, Y., Tenenbaum, J. B., and Levine, S. Planning with diffusion for flexible behavior synthesis. In *International Conference on Machine Learning*, 2022.

Jothiraj, F. V. S. and Mashhadi, A. Phoenix: A federated generative diffusion model. In *Companion Proceedings of the ACM on Web Conference 2024*, pp. 1568–1577, 2024.

Kynkäänniemi, T., Karras, T., Laine, S., Lehtinen, J., and Aila, T. Improved precision and recall metric for assessing generative models. *Advances in neural information processing systems*, 32, 2019.

LeCun, Y. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

Lipman, Y., Chen, R. T., Ben-Hamu, H., Nickel, M., and Le, M. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Liu, N., Li, S., Du, Y., Torralba, A., and Tenenbaum, J. B. Compositional visual generation with composable diffusion models. In *European conference on computer vision*, 2022a.

Liu, X., Gong, C., and Liu, Q. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022b.

Luo, Y., Mishra, U. A., Du, Y., and Xu, D. Generative trajectory stitching through diffusion composition. In *Neural Information Processing Systems*, 2025.

McAllister, D., Tancik, M., Song, J., and Kanazawa, A. Decentralized diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 23323–23333, 2025.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pp. 1273–1282. PMLR, 2017.

Park, S., Frans, K., Eysenbach, B., and Levine, S. Ogbench: Benchmarking offline goal-conditioned rl. In *International Conference on Learning Representations (ICLR)*, 2025.

Peng, Z., Wang, X., Chen, S., Rao, H., Shen, C., and Jiang, J. Federated learning for diffusion models. *IEEE Transactions on Cognitive Communications and Networking*, 2025.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. pmlr, 2015.

Song, Y. and Ermon, S. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

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

Tun, Y. L., Thwal, C. M., Yoon, J. S., Kang, S. M., Zhang, C., and Hong, C. S. Federated learning with diffusion models for privacy-sensitive vision tasks. In *2023 International Conference on Advanced Technologies for Communications (ATC)*, pp. 305–310. IEEE, 2023.

Vora, J., Bouacida, N., Krishnan, A., and Mohapatra, P. Feddm: enhancing communication efficiency and handling data heterogeneity in federated diffusion models. *arXiv preprint arXiv:2407.14730*, 2024.

Wang, X., Peng, Y., Lu, L., Lu, Z., Bagheri, M., and Summers, R. M. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

Wiedemer, T., Mayilvahanan, P., Bethge, M., and Brendel, W. Compositional generalization from first principles. *Advances in Neural Information Processing Systems*, 36: 6941–6960, 2023.

Zhang, Y., Murtuza-Lanier, C., Li, Z., Du, Y., and Wu, J. Product of experts for visual generation. *arXiv preprint arXiv:2506.08894*, 2025.

Zheng, Q., Le, M., Shaul, N., Lipman, Y., Grover, A., and Chen, R. T. Guided flows for generative modeling and decision making. *arXiv preprint arXiv:2311.13443*, 2023.

A. Compositional Sampling in Flows

A.1. Score-based Composition

Given a joint condition $\mathbf{y} = (y_1, y_2, \ldots, y_k)$, we can break down the joint conditional score $\nabla \log p_t(\mathbf{x}_t \mid \mathbf{y})$ by the Bayes' Theorem as follows:

$$\begin{aligned} \nabla \log p_t(\mathbf{x}_t \mid y_1, y_2, \ldots, y_k) &= \nabla \log \frac{p_t(\mathbf{x}_t) p_t(y_1, y_2, \ldots, y_k \mid \mathbf{x}_t)}{p_t(\mathbf{y})} \\ &= \nabla \log p_t(\mathbf{x}_t) + \nabla \log p_t(y_1, y_2, \ldots, y_k \mid \mathbf{x}_t) \end{aligned} \tag{18}$$

By assuming the attributes y_i are **conditionally independent** and plugging in Eq. (1), we get:

$$\begin{aligned} \nabla \log p_t(\mathbf{x}_t \mid y_1, y_2, \ldots, y_k) &= \nabla \log p_t(\mathbf{x}_t) + \nabla \log \left(\prod_{i=1}^{k} p_t(y_i \mid \mathbf{x}_t) \right) \\ &= \nabla \log p_t(\mathbf{x}_t) + \sum_{i=1}^{k} \nabla \log p_t(y_i \mid \mathbf{x}) \\ &= \nabla \log p_t(\mathbf{x}_t) + \sum_{i=1}^{k} (\nabla \log p_t(\mathbf{x}_t \mid y_i) - \nabla \log p_t(\mathbf{x}_t)) \quad \text{[Applying Bayes' Theorem]} \end{aligned} \tag{19}$$

The factorization in Eq. (19) enables compositional score sampling with classifier-free guidance,

$$\nabla \log \hat{p}_t(\mathbf{x}_t \mid \mathbf{y}) = \nabla \log p_t(\mathbf{x}_t) + \sum_{i=1}^{k} (\nabla \log p_t(\mathbf{x}_t \mid y_i) - \nabla \log p_t(\mathbf{x}_t)) \tag{20}$$

Where $\hat{p}_t(\mathbf{x}_t \mid \mathbf{y}) \propto p_t(\mathbf{x}_t)^{(1-\sum_i w_i)} \prod_{i=1}^{k} p_t(\mathbf{x}_t \mid y_i)^{w_i}$ is a geometric average of the unconditional distribution and the marginal conditional distributions.

A.2. Flow-based Composition

Eq. (20) has been utilized previously by Liu et al. (2022a); Ajay et al. (2023); Gaudi et al. (2025), and admits Eq. (4) for diffusion models. However, it is not well studied whether this compositional procedure extends to flow matching models. In the following, we show that Eq. (5) in the main body of the paper is equivalent to the compositional score (20) and follows the same underlying probability distribution.

Let $p_t(\mathbf{x}_t \mid \mathbf{x}_1)$ denote a probability path between a source distribution p_0 and the data distribution p_1. If p_0 is the noise distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$, we may define p_t as:

$$p_t(\mathbf{x}_t \mid \mathbf{x}_1) = \mathcal{N}(\mathbf{x}_t \mid \alpha_t \mathbf{x}_1, \sigma_t^2 \mathbf{I}) \tag{21}$$

Where the pair (α_t, σ_t) follows the boundary conditions $(\alpha_0, \sigma_0) = (0, 1)$ and $(\alpha_1, \sigma_1) = (1, 0)$. Thus at $t = 0$ and $t = 1$, p_t results in the noise and data distribution respectively. Next, consider a conditional velocity field $\mathbf{v}(\mathbf{x}_t, t \mid y)$, which we denote as $\mathbf{v}_t(\mathbf{x}_t \mid \mathbf{y})$ for brevity. We propose that the joint condition $\mathbf{y} = (y_1, \ldots, y_k)$ factorizes as follows:

$$\hat{\mathbf{v}}_t(\mathbf{x}_t \mid y_1, y_2, \ldots y_k) = \mathbf{v}_t(\mathbf{x}_t) + \sum_{i=1}^{k} w_i \left(\mathbf{v}_t(\mathbf{x}_t \mid y_i) - \mathbf{v}_t(\mathbf{x}_t) \right) \tag{22}$$

By **Lemma 1** from Zheng et al. (2023), if p_t follows Eq. (21), the velocity $\mathbf{v}_t$ can be related to the score function $\nabla \log p_t(\mathbf{x}_t \mid \mathbf{y})$ as follows:

$$\mathbf{v}_t(\mathbf{x}_t \mid \mathbf{y}) = a_t \mathbf{x} + b_t \log p_t(\mathbf{x}_t \mid \mathbf{y}), \tag{23}$$

where,

$$a_t = \frac{\dot{\alpha}_t}{\alpha_t}, \quad b_t = (\dot{\alpha}_t \sigma_t - \alpha_t \dot{\sigma}_t) \frac{\sigma_t}{\alpha_t} \tag{24}$$

Plugging Eq. (23) to the RHS of Eq. (22) gives us:

$$\begin{aligned}
\hat{\mathbf{v}}_t(\mathbf{x}_t \mid y_1, y_2, \dots y_k) &= a_t \mathbf{x} + b_t \nabla \log p_t(\mathbf{x}_t) + \sum_{i=1}^{k} w_i \left(a_t \mathbf{x} + b_t \nabla \log p_t(\mathbf{x}_t \mid y_i) - a_t \mathbf{x} - b_t \nabla \log p_t(\mathbf{x}_t) \right) \\
&= a_t \mathbf{x} + b_t \nabla \log p_t(\mathbf{x}_t) + b_t \sum_{i=1}^{k} w_i \left(\nabla \log p_t(\mathbf{x}_t \mid y_i) - \nabla \log p_t(\mathbf{x}_t) \right) \\
&= a_t \mathbf{x} + b_t \left(\nabla \log p_t(\mathbf{x}_t) + \sum_{i=1}^{k} w_i \left(\nabla \log p_t(\mathbf{x}_t \mid y_i) - \nabla \log p_t(\mathbf{x}_t) \right) \right) \\
&= a_t \mathbf{x} + b_t \nabla \log \hat{p}_t(\mathbf{x}_t \mid y_1, y_2, \dots, y_k)
\end{aligned} \tag{25}$$

Thus, we show by (25) that the compositional velocity $\hat{\mathbf{v}}_t(\mathbf{x} \mid \mathbf{y})$ defined in Eq. (22) coincides with the compositional score $\nabla \log \hat{p}_t(\mathbf{x} \mid \mathbf{y})$ (20).

B. Decentralized Compositional Spatial Planning

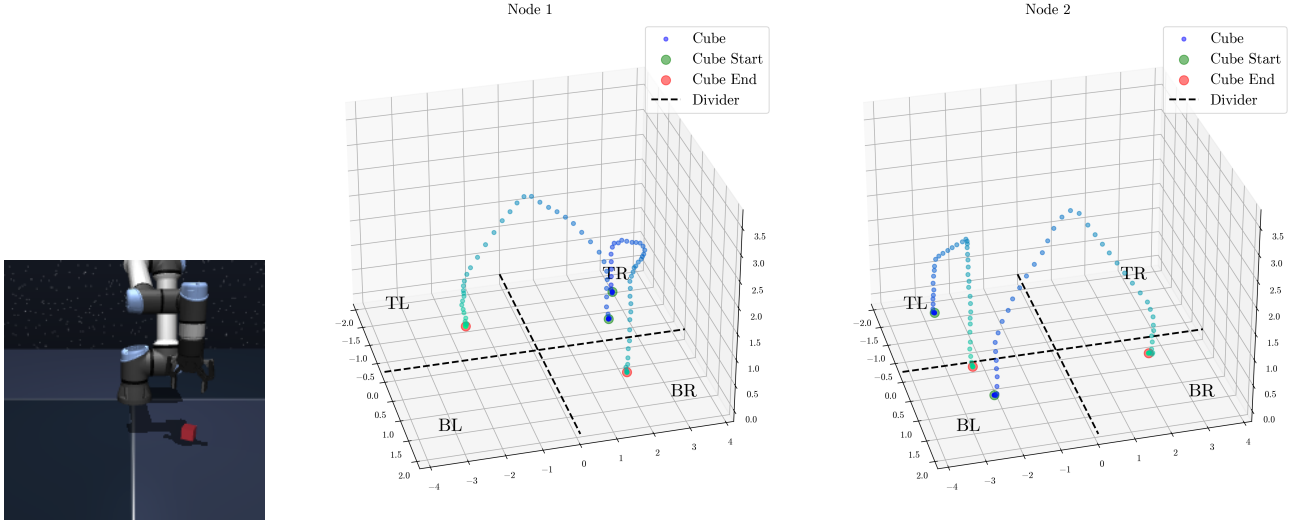


Figure 8. Non-IID setup for robotic planning experiment. The trajectory data is distributed across $n = 2$ decentralized nodes, where D_1 only contains TL↔TR and TR↔BR data, and D_2 only contains TL↔BL and BL↔BR data respectively. Our objective is to see if a decentralized generative planner can learn to move diagonally between TR↔BL.

C. Communication Cost of Baselines

Federated Learning. In an FL round, every client c_a locally optimizes their model via a local flow matching loss. At the end of the round, each client transmits the local parameters θ_a to a central server, and subsequently receives an averaged parameter $\bar{\theta}$. For a T round training process, each client has a communication cost of $2 \times T$. The total volume of θ transmissions amount to $2 \times n \times T$.

Decentralized Diffusion Models. No communication is required during training, as models are trained independently. But in order to be capable of inference, the models need to be transmitted to a single server (N transmissions), or broadcasted to each peer ($N \times (N-1)$ transmissions). DDM thus has a minimum cost of N.

Diffusion Federated Dataset. DFD actually does not require communicating the models during inference, as it transmits the velocity field $\mathbf{v}_t$ instead. While this approach is useful for privacy, the number of communications actually become unbound (as there is no limit on how many times a model may be used to perform inference). We thus consider DFD to also transmit their model to a central server, leading to a cost of N.