Decentralized Autoregressive Generation

Anonymous authors
Paper under double-blind review

Abstract

The decentralization of autoregressive generation has attracted considerable attention in recent years as a solution to scaling bottlenecks. However, despite promising empirical results, this paradigm currently lacks rigorous theoretical justification. In this work, we formally establish the theoretical equivalence between decentralized and centralized training. To achieve this, we adapt the Discrete Flow Matching framework for autoregressive generation, leveraging its inherent properties to demonstrate that global models naturally decompose into independent experts. Finally, we conduct extensive experiments across diverse multimodal benchmarks, empirically validating that decentralized training maintains competitive parity with standard centralized architectures.

1 Introduction

Autoregressive generation has established itself as the dominant paradigm for a diverse range of machine learning tasks, encompassing natural language modeling, multimodal reasoning, robotics, and synthetic media generation (Comanici et al., 2025; Brohan et al., 2023; Zhou et al., 2025). The efficacy of these models is characterized by scaling laws (Henighan et al., 2020; Kaplan et al., 2020), which demonstrate that increasing the parameter count of a model and the amount of training data predictably improves its performance.

However, this necessary increase in parameter counts leads to significant infrastructural challenges. The standard approach relies on centralized training, which involves training a single large model across a vast cluster of GPU nodes. To function effectively, these thousands of GPUs have to continuously communicate with one another to synchronize gradients. This massive hardware requirement creates a severe accessibility barrier, making it extremely difficult for academic institutions and smaller organizations to participate in foundational model training. Furthermore, this tightly coupled setup is highly vulnerable to individual GPU node failures. In a centralized system, if even a single GPU fails, it disrupts the global training process; as noted by Grattafiori et al. (2024), this often requires a restart of the entire training job from the last saved checkpoint, wasting significant time and computational resources.

A promising solution to these hardware bottlenecks is decentralized training. Instead of training one model on all the data, this approach trains a collection of independent "expert" models. The process works in two stages. First, the massive global dataset is divided into separate, non-overlapping subsets. Each expert model is then trained exclusively on one of these subsets. Because the experts never need to communicate during training, they can be trained on completely separate, smaller computer systems. Second, during inference, a router evaluates the input and activates only one or a few the most relevant experts. While this router is often trained as a separate auxiliary model, it can also be derived directly from the mechanisms used to partition the data. Their outputs are then combined to generate the final response. This approach removes the need for a vast centralized computation and data clusters and makes collaborative training feasible for researchers with limited hardware.

Recent studies have shown that decentralized training is highly effective for text generation models in certain situations (Gururangan et al., 2023; Filippova et al., 2025; Li et al., 2022). However, these successes are largely based on practical experiments rather than strict theoretical foundations. Presently, it remains unknown whether the decentralized and centralized training paradigms are mathematically equivalent. Consequently,

decentralized training is often treated as a heuristic "black box". Without a theoretical foundation researchers lack any a priori expectations of model convergence or capability. They can only assess the success of a decentralized approach retrospectively, after the training is complete. This theoretical uncertainty highlights a significant gap in the literature and motivates a critical question: can we theoretically guarantee that a decentralized collection of experts is equivalent to a standard, centrally trained model?

To answer this question, we seek to establish a rigorous theoretical foundation for decentralized training. During exploration, we find a critical insight from recent advancements in continuous generative modeling: within the standard Flow Matching framework, the centralized and decentralized generating flows have already been shown to be mathematically equivalent (McAllister et al., 2025). This established equivalence presents a promising pathway. If autoregressive generation can be reformulated through the lens of Flow Matching, it could naturally inherit these same theoretical guarantees. To achieve this, we turn to the theory of Discrete Flow Matching (Campbell et al., 2024; Gat et al., 2024). While adapting the Discrete Flow Matching framework to the discrete-time mechanics of autoregressive generation is non-trivial, it enables us to express the generative process using the "probability generating velocity", the discrete analogue of a continuous generative flow. We demonstrate that this global velocity can be structurally decomposed and represented as a strict linear combination of local velocities produced by independent data clusters. By proving this mathematical decomposition, we move beyond empirical heuristics and formally establish the theoretical equivalence of the centralized and decentralized generating objectives for autoregressive models.

To realize this reformulation, it is necessary and sufficient to show that autoregressive generation is a special case of Discrete Flow Matching. However, establishing this link requires addressing a fundamental difference in how they operate. Standard flow matching assumes that probability distributions change smoothly over continuous time, whereas autoregressive generation relies on discrete, step-wise changes. Because of these non-smooth transitions, the core mathematical guarantee of flow matching, specifically, the link between the Continuity Equation and the generated probability path does not hold in an arbitrary discrete-time case. To resolve this, we demonstrate that imposing a 1-sparse constraint on the probability generating velocity (i.e., restricting probability changes to a single token position at any given timestep) restores the theoretical validity of the framework. By explicitly enforcing this property, we prove that autoregressive generation perfectly satisfies the Discrete Flow Matching paradigm. We provide a comprehensive theoretical analysis of this formulation in Section 4.

Our primary contributions are as follows:

- For the first time, we establish a rigorous theoretical equivalence between decentralized and centralized training paradigms for autoregressive generation. By proving that the global probability generating velocity can be exactly expressed as a linear combination of independent expert velocities, we provide the theoretical analysis that justifies decentralized training.
- To achieve this, we extend the Discrete Flow Matching framework into the discrete-time domain and formally align it with autoregressive generation. We introduce the necessary constraints, such as the 1-sparse property, to make this alignment possible. Beyond enabling our primary proof, this formulation serves as a novel, standalone mathematical bridge between continuous flow-based methods and discrete autoregressive models.
- We provide extensive empirical evidence demonstrating that this decentralized paradigm remains highly effective beyond standard text generation. We successfully apply decentralized training to Multimodal Large Language Models (MLLMs), conducting extensive experiments across multiple question-answering benchmarks utilizing the LLaVA (Liu et al., 2024a) and InternVL (Chen et al., 2025) architectures.

2 Related Work

Autoregressive Generation Autoregressive generation has become a state-of-the-art paradigm for multiple ML tasks, such as language modeling, multimodal reasoning, robotics, image generation, audio synthesis and others (Vaswani et al., 2023; Grattafiori et al., 2024; Comanici et al., 2025; Henighan et al., 2020; Kaplan

et al., 2020; Guo et al., 2025; Brohan et al., 2023; Tian et al., 2024; Zhou et al., 2025; Liu et al., 2024a; Chen et al., 2025). For example, Brohan et al. (2023) finetuned a pretrained visual language model on a mix of original training data and robot data, where robot actions were represented as tokens, resulting in significant generalization improvements. Another work, Tian et al. (2024), used an autoregressive approach to the image generation, using image scales of increasing resolutions as autoregressive units. Each next scale is predicted using only its prefix of previous scales. This approach provided a superior image generation performance to the previous work. Besides that, autoregressive generation showed its efficiency in visual question answering. As shown in Liu et al. (2024a), finetuning pretrained language model on a relatively small VQA dataset, where image features are projected into token space through Multilayer Perceptron, can lead to SOTA results on various VQA tasks. Chen et al. (2025) further demonstrated that this approach is capable of delivering a performance comparable to native multimodal models. The scaling laws (Kaplan et al., 2020; Henighan et al., 2020) allow us to predict model performance increase through the growth of model parameters and training dataset size. However, the maintenance of large amounts of data and training large centralized models across multiple GPUs presents significant infrastructural challenges (Grattafiori et al., 2024). Because standard centralized training requires highly synchronous communication, the system is vulnerable to GPU node failures, which can halt the entire training process. Furthermore, this paradigm imposes severe engineering demands for storing and managing colossal, centralized datasets across high-bandwidth networks.

Decentralized Training Decentralized training has emerged in the recent studies as an effective way to drastically reduce communication overhead between training nodes (Gururangan et al., 2023; Filippova et al., 2025; Ma et al., 2024; McAllister et al., 2025; Wang et al., 2025; Li et al., 2022). For example, Ma et al. (2024) trained a mixture of CLIP model experts independently on different data clusters, reducing the amount of false negative examples in each cluster. Another notable example is Decentralized Diffusion Models framework (McAllister et al., 2025), which uses a mixture of flow matching expert models for image generation to outperform dense baseline model. Another approach to decentralized flow matching is Federated Flow Matching (Wang et al., 2025), using semi-dual formulation of optimal transport to approximate global optimal coupling, leading to straighter probability paths and thereby superior performance compared to vanilla federated gradient averaging. While previous work empirically verifies the effectiveness of decentralized training and inference in certain scenarios, our work for the first time suggests a theoretical analysis of the decentralized autoregressive generation in unified settings. In addition, prior work focuses either on pure language modeling, image generation or contrastive learning, while our work conducts experiments for multimodal large language models for visual question answering.

Discrete Flow Matching Discrete Flow Matching extends Flow Matching theory (Lipman et al., 2023) to discrete probability spaces. It gained attention in text modeling as an alternative to autoregressive generation, protein co-design, graph generation and others (Campbell et al., 2024; Qin et al., 2025; Gat et al., 2024). Notably, Gat et al. (2024) considered probability paths that are convex sums of elementary probabilities and provided the closed formula for probability generating velocity. Davis et al. (2024) introduced an alternative theoretical foundation, by mapping probability manifold with Fisher-Rao metric to a sphere and considering probability paths as geodesics on sphere. While prior work provides substantial background for continuous time probability paths, the discrete time probability paths case is not investigated. In this work, we extend Discrete Flow Matching to the discrete time case, as required for autoregressive scenario, reformulate the equations and investigate the constraints under which probability generating velocities for discrete time paths are correctly defined.

3 Background

As discussed, our central objective is to establish a rigorous theoretical foundation for decentralized autoregressive generation. We begin by observing that autoregressive generation is, at its core, a specific instance of a probability path moving between distributions in a discrete token space. Therefore, mathematically formalizing and ultimately decentralizing this process requires a framework explicitly designed for discrete probability dynamics. This section briefly reviews the standard Discrete Flow Matching framework (Gat et al., 2024). We specifically focus on defining probability paths, probability generating velocities, and the

Continuity Equation. We highlight these exact concepts because the probability path formalizes the sequence's trajectory from source to target, the generating velocity dictates the transition rules at each step, and the Continuity Equation provides the mathematical guarantee that following this velocity yields the correct target distribution. Formally reintroducing these standard properties is an essential prerequisite before we can adapt them to the discrete-time mechanics of autoregressive models in our subsequent theoretical analysis.

3.1 Discrete Flow Matching (Continuous Time)

Discrete Flow Matching (Gat et al., 2024) is a theoretical framework that describes discrete token generation processes in terms similar to Continuous Flow Matching (Lipman et al., 2023). In this subsection we introduce foundational concepts from this framework that are necessary for our theoretical analysis in Section 4. In the following paragraphs, we will define the probability path, the probability generating velocity, and the Continuity Equation.

Probability paths Given vocabulary of tokens $[d]$, source distribution X_0 with probability mass function $p(x)$ of token sequences $x = (x^1, \dots, x^N) \in [d]^N$ and target distribution X_1 with probability mass function $q(x)$, we define the **probability path** $p_t(x), t \in [0,1]$ s.t. $p_0(x) = p(x)$, $p_1(x) = q(x)$ as follows:

$$p_t(x) = \sum_{(x_0,x_1)\in[d]^N\times[d]^N} p_t(x|x_0,x_1)\pi(x_0,x_1) \tag{1}$$

$$p_t(x|x_0,x_1) = \prod_{i=1,\dots,N} p_t(x^i|x_0,x_1) \tag{2}$$

where $\pi(x_0,x_1)$ is a joint probability mass function of (X_0,X_1), and $p_t(x^i|x_0,x_1)$ is a conditional marginal probability path for the token position i.

The $p_t(x|x_0,x_1)$ should satisfy the following conditions:

$$p_0(x|x_0,x_1) = \delta_{x_0}(x) = \begin{cases} 1, & x = x_0 \\ 0, & x \neq x_0 \end{cases} \tag{3}$$

$$p_1(x|x_0,x_1) = \delta_{x_1}(x) = \begin{cases} 1, & x = x_1 \\ 0, & x \neq x_1 \end{cases} \tag{4}$$

In the framework, in order for probability paths to be tractable, we consider only the probability paths that are represented by a convex sum of conditional probabilities $w^j(x^i|x_0,x_1)$ with coefficients provided by some scheduler $\kappa_t^{i,j}$:

$$p_t(x^i|x_0,x_1) = \sum_j \kappa_t^{i,j} w^j(x^i|x_0,x_1) \tag{5}$$

$$\sum_j \kappa_t^{i,j} = 1, \quad \kappa_t^{i,j} \geq 0 \tag{6}$$

Probability generating velocities We also define **probability generating velocity** $u_t^i(x,z), i \in [N]$. We sample our token sequences by the following rule (for each token position independently):

$$X_{t+h}^i \sim \delta_{X_t^i}(\cdot) + h u_t^i(\cdot, X_t), \quad h > 0 \tag{7}$$

We say that u_t generates the probability path $p_t(x)$ if X_{t+h}, sampled by the rule 7, satisfies:

$$X_{t+h} \sim p_{t+h} + o(h) \tag{8}$$

Probability generating velocity u_t can be constructed through conditional probability generating velocities $u_t^i(x|x_0,x_1)$. In fact, the following theorem holds:

Theorem 1 *Probability generating velocity $u_t^i(x^i, z)$ defined as follows:*

$$u_t^i(x^i, z) = \sum_{(x_0,x_1)\in[d]^N\times[d]^N} u_t^i(x^i, z|x_0, x_1)\frac{p_t(z|x_0, x_1)\pi(x_0, x_1)}{p_t(z)} \tag{9}$$

generates the probability path $p_t(x)$.

Continuity Equation Theorem 1 is proven by using the Continuity Equation:

$$\dot{p_t}(x) + div_x(p_t u_t) = 0 \tag{10}$$

The discrete divergence of $p_t u_t$ is defined in the following way:

$$div_x(v) = \sum_{z\in[d]^N} (v(z, x) - v(x, z)) \tag{11}$$

where $v(z, x) = p_t(x)u_t^i(z^i, x)$, $v(x, z) = p_t(z)u_t^i(x^i, z)$ if z differs from x only in the token position i, $v(x, x) = p_t(x)\sum_{i=1}^N u_t^i(x^i, x)$ and $v(z, x) = v(x, z) = 0$ otherwise.

The divergence of $p_t u_t$ can be shown to be equal to:

$$div_x(p_t u_t) = -\sum_{z\in[d]^N} p_t(z)\sum_{i=1}^N \delta_z(x^{\bar{i}})u_t^i(x^i, z) \tag{12}$$

where $\delta_z(x^{\bar{i}}) = \prod_{j\neq i}\delta_z(x^j)$.

As shown in Gat et al. (2024), if the continuity equation holds, then the probability generating velocity generates a probability path as defined in (7-8).

4 Theoretical Analysis

The primary objective of this section is to mathematically prove that the training objective for autoregressive sampling can be completely decentralized. To achieve this, we formulate autoregressive generation as a special case of Discrete Flow Matching. By establishing this theoretical connection, we can demonstrate that the Discrete Flow Matching objective is inherently decentralizable, which subsequently allows us to apply a decentralized formulation directly to autoregressive generation. To build this mathematical proof, we proceed as follows. First, in Subsection 4.1, because autoregressive generation operates in discrete time steps, we extend the Discrete Flow Matching framework into the discrete time domain. Next, in Subsection 4.2, we formalize the probability path and the probability generating velocity specifically for autoregressive sampling. Since verifying that the probability generating velocity correctly yields the probability path reduces to checking the Continuity Equation, we establish a sufficient 1-sparse condition that the discrete velocity must satisfy. Finally, in Subsection 4.3, we define the Discrete Decentralized Flow Matching objective, which naturally applies to autoregressive generation, thereby completing our proof of theoretical equivalence between the decentralized and centralized training settings.

4.1 Discrete Flow Matching (Discrete Time)

In this subsection, we formally extend the Discrete Flow Matching framework into the discrete time domain. We sequentially introduce the discrete sampling rule, the necessary conditions for the probability generating velocity, and the adapted Continuity Equation. Once these foundational elements are established, we will be able to specifically define the probability paths and the probability generating velocities, and ultimately confirm that the velocity correctly generates the path by verifying the Continuity Equation.

We begin by transitioning to a discrete-time domain, $t \in \{0, 1, \ldots, n\}$, and correspondingly redefining the terms. In this setting, the sampling rule 7 becomes:

$$X_{t+1}^i \sim \delta_{X_t^i}(\cdot) + u_t^i(\cdot, X_t), \quad t \in \{0, 1, \ldots, n-1\} \tag{13}$$

where probability generating velocity u_t is now defined in discrete timesteps. u_t now generates $p_t(x)$ if X_{t+1} sampled by the rule 13 satisfies the following:

$$X_{t+1} \sim p_{t+1} \tag{14}$$

Here, in order to define proper probability mass function, it is necessary and sufficient that probability velocity u_t satisfies the following conditions:

$$\sum_{x^i \in [d]} u_t^i(x^i, z) = 0 \tag{15}$$

$$-1 \leq u_t^i(z^i, z) \leq 0, \quad 0 \leq u_t^i(x^i, z) \leq 1 \quad \text{for } x^i \neq z^i \tag{16}$$

The continuity equation now takes the form:

$$p_{t+1}(x) - p_t(x) + div_x(p_t u_t) = 0 \tag{17}$$

It remains to confirm that satisfying 17 implies generation. This does not hold for an arbitrary velocity. The sampling rule 13 admits simultaneous transitions at several token positions, contributing cross-terms that are products of multiple velocity entries. In continuous time these are $O(h^2)$ and vanish under the $o(h)$ rule 7; at the discrete step $h = 1$ they contribute at first order, and the Continuity Equation no longer implies generation. We therefore impose a 1-sparse condition (Subsection 4.2), restricting transitions to a single position and eliminating these cross-terms. Under it, verifying generation reduces to verifying the Continuity Equation.

4.2 Autoregressive sampling

In this subsection, we formulate autoregressive sampling as a special instance of the discrete time Discrete Flow Matching framework established in Subsection 4.1. Following this theoretical foundation, we sequentially introduce the source and target distributions, the conditional probability paths, and the corresponding schedulers. Next, we establish the conditional probability generating velocity and verify that the Continuity Equation holds under these conditions. Finally, we demonstrate that when the probability generating velocity satisfies the 1-sparse property, the Continuity Equation becomes strictly equivalent to the generating probability path. Through this sequence of proofs, we formally validate that autoregressive sampling is a correct instance of Discrete Flow Matching.

To define autoregressive sampling, we define the source and target distribution as a special case of C-coupling from Gat et al. (2024):

$$(X_0, X_1) \sim (\mathbb{I} \odot X_1 + (1 - \mathbb{I}) \odot (\mathbf{m}, \mathbf{m}, \ldots, \mathbf{m}), X_1) \tag{18}$$

where $X_1 \sim q$, $\mathbf{m}$ is a mask token, $\mathbb{I} \in \{0, 1\}^N$ is an indicator vector of the form $\mathbb{I} = (1, 1, \ldots, 1, 0, \ldots, 0)$ which has all ones before some token position and all zeros after.

Next, the conditional probability path for a pair x_0, x_1 s.t. x_0 has P tokens revealed at timestep 0 will take the form:

$$p_t(x^i | x_0, x_1) = \kappa_t^i \delta_{x_1}(x^i) + (1 - \kappa_t^i)\delta_{x_0}(x^i) \tag{19}$$

$$\kappa_t^i = \begin{cases} 0, & t < i - P \\ 1, & t \geq i - P \end{cases} \tag{20}$$

This definition of conditional probability path and scheduler simply means that at timestep t exactly $t + P$ tokens are revealed.

Note that at any time step t, $p_t(x|x_0, x_1)$ is a degenerate distribution with a single result x_t:

$$p_t(x|x_0, x_1) = \delta_{x_t}(x) \tag{21}$$

Now, define the conditional generating velocity $u_t^i(x^i, z|x_0, x_1)$ as:

$$u_t^i(x^i, z|x_0, x_1) = \begin{cases} \delta_{x_{t+1}}(x^i) - \delta_{x_t}(x^i), & z = x_t \\ 0, & z \neq x_t \end{cases} \tag{22}$$

From the above formula it follows that for a fixed timestep t, $u_t(x^i, x_t|x_0, x_1) = 0$ for any token position except possibly $i = P + t + 1$. This property is sufficient to show that the continuity equation implies the generation of probability path by probability generation velocity in the discrete time domain.

Next, in order to show that defined conditional probability generating velocity generates conditional probability path, we need to 1) Show that continuity equation holds 2) Show that if the continuity equation holds, then generation follows.

First, check that the continuity equation holds:

$$p_{t+1}(x|x_0, x_1) - p_t(x|x_0, x_1) + div_x(p_t u_t) = 0 \tag{23}$$

Since $p_t(x|x_0, x_1)$ has only one outcome x_t and $u_t^i = 0$ for any $i \neq P + t + 1$, the divergence formula 12 simplifies to the following:

$$div_x(p_t u_t) = -\delta_{x_t}(x^{\overline{P+t+1}}) u_t(x^{P+t+1}, x_t|x_0, x_1) \tag{24}$$

Substituting 22, we get:

$$\begin{aligned} & p_{t+1}(x|x_0, x_1) - p_t(x|x_0, x_1) + div_x(p_t u_t) \\ & = \delta_{x_{t+1}}(x) - \delta_{x_t}(x) - \delta_{x_t}(x^{\overline{P+t+1}}) u_t(x^{P+t+1}, x_t|x_0, x_1) \\ & = \delta_{x_{t+1}}(x) - \delta_{x_t}(x) - \delta_{x_t}(x^{\overline{P+t+1}})(\delta_{x_{t+1}}(x^{P+t+1}) - \delta_{x_t}(x^{P+t+1})) \\ & = \delta_{x_{t+1}}(x) - \delta_{x_t}(x) - \delta_{x_{t+1}}(x) + \delta_{x_t}(x) \\ & = 0 \end{aligned}$$

Now, once we verified that the continuity equation holds, we show that satisfying the continuity equation implies generation. For that, we require a 1-sparsity property of generating probability velocity: for a fixed timestep t $u_t^i = 0$ for any token position i except possibly one, denoted by $j = j(t)$ depending only on a timestep t. As discussed above, velocity defined in (22) satisfies it. Then the PMF of X_{t+1} sampled by 13 is:

$$\begin{aligned} p_{X_{t+1}}(x) &= \sum_{z \in [d]^N} p_t(z) \prod_{i=1}^{N} \big(\delta_z(x^i) + u_t^i(x^i, z)\big) \\ &= \sum_{z \in [d]^N} p_t(z) \Big(\delta_z(x) + \delta_z(x^{\bar{j}})\, u_t^j(x^j, z)\Big) \\ &= p_t(x) + \sum_{z \in [d]^N} p_t(z)\, \delta_z(x^{\bar{j}})\, u_t^j(x^j, z) \\ &= p_t(x) - \mathrm{div}_x\big(p_t u_t\big) = p_t(x) - (p_t(x) - p_{t+1}(x)) = p_{t+1}(x) \end{aligned}$$

We defined autoregressive sampling as an instance of discrete flow matching in a discrete time domain. We proved that for probability generating velocities that are non-zero at only one token position for any fixed timestep satisfying continuity equation implies generation. We showed that the autoregressive probability path defined in 19 and the velocity defined in 22 satisfy the continuity equation and therefore define a valid probability path and generating velocity.

4.3 Decentralization

In this subsection, we demonstrate that the Discrete Flow Matching training objective can be decentralized, and, by extension, that autoregressive sampling is also decentralizable. Specifically, we formally represent

the global probability generating velocity as a linear combination of independent expert velocities. Because autoregressive sampling was proven to be an instance of Discrete Flow Matching in Subsection 4.2, this mathematical decomposition automatically establishes that the autoregressive sampling objective is fully decentralizable. This finalizes our theoretical framework, completing our overarching proof that the decentralized training objective is mathematically equivalent to the centralized baseline.

The formalization from Subsection 4.2 automatically makes Theorem 1 true for autoregressive sampling. Now, similar to McAllister et al. (2025), if we define a target distribution X_1 as a union of disjoint clusters $S_k, \quad k \in \{1, 2, \ldots, K\}$, we can rewrite Equation 9 as follows:

$$u_t^i(x^i, z) = \sum_{(x_0,x_1)\in[d]^N\times[d]^N} u_t^i(x^i, z|x_0, x_1)\frac{p_t(z|x_0, x_1)\pi(x_0, x_1)}{p_t(z)} \tag{25}$$

$$= \sum_{k=1}^{K} \frac{1}{p_t(z)} \sum_{(x_0,x_1):\, x_1\in S_k} u_t^i(x^i, z|x_0, x_1)p_t(z|x_0, x_1)\pi(x_0, x_1|S_k)p(S_k) \tag{26}$$

$$= \sum_{k=1}^{K} \frac{p_t(z|S_k)p(S_k)}{p_t(z)} \sum_{(x_0,x_1):\, x_1\in S_k} u_t^i(x^i, z|x_0, x_1)\frac{p_t(z|x_0, x_1)\pi(x_0, x_1|S_k)}{p_t(z|S_k)} \tag{27}$$

The inner sum is called an expert velocity, and the weighting coefficient is called a router. The equation above simply means that theoretically, the overall generating probability velocity is represented as a weighted sum of expert velocities.

5 Method

Since the effectiveness of decentralized training for pure language models has already been established in previous work (Gururangan et al., 2023; Filippova et al., 2025), our empirical evaluation focuses on Vision Language Models as a more complex instance of autoregressive sampling. The purpose of this section is to provide a comprehensive overview of our experimental methodology. At a high level, our approach involves training individual expert models entirely independently on disjoint data subsets, and subsequently combining their predictions using a routing mechanism during the inference phase. In the following subsections, we will detail our complete decentralized pipeline: first describing the training process, which includes the data partitioning strategy, the routing mechanism, and the independent training of expert models, and finally outlining our inference strategy for utilizing these experts during generation.

5.1 Training

Data partitioning Following established practices in multimodal representation learning (Radford et al., 2021), we use a pretrained vision encoder (CLIP) to extract image features from the unique images in the dataset. CLIP is widely adopted for this purpose because it provides robust and semantically rich visual representations. We train a spherical balanced k-means algorithm to cluster these features into K clusters of equal size (see Figure 1). Specifically, we employ cosine distance between the feature vectors, as CLIP representations are naturally optimized for this metric. The centroid vectors are L_2-normalized to ensure they reside on a unit sphere. This normalization effectively handles the high dimensionality of CLIP features. All samples are evenly distributed among the clusters based on their distance to the centroids, ensuring each expert receives the same number of unique training images.

Router We establish the routing mechanism without introducing any additional trainable parameters or auxiliary neural networks. Instead, the K cluster centroids derived from the spherical balanced k-means algorithm serve directly as the router. For any given input sample, we extract its image features using the same pretrained CLIP vision encoder utilized during the partitioning phase. The router then computes the cosine similarity between the extracted feature vector and each fixed cluster centroid. The sample is subsequently assigned to the corresponding cluster based on these similarity scores. This approach ensures a highly efficient routing mechanism that perfectly mirrors the initial data distribution strategy.

Experts training Following the data partitioning phase, we initialize K distinct expert models. Each expert is trained completely independently on its exclusively assigned data cluster. Because these experts do not require any communication or gradient synchronization with one another during the optimization process, they can be trained simultaneously across completely isolated computing nodes.

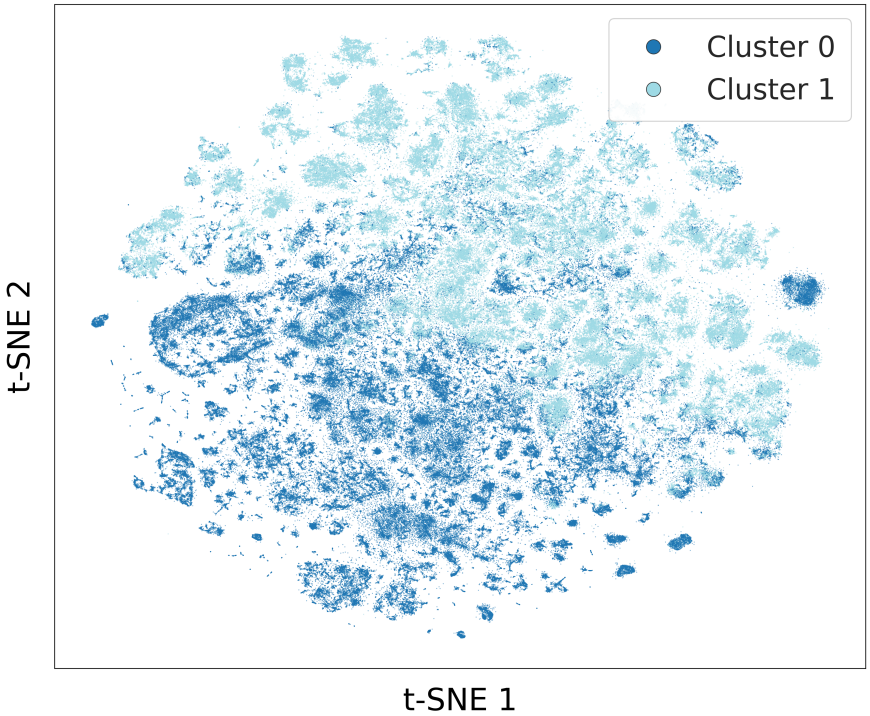

Figure 1: t-SNE 2D projection of balanced spherical k-means clustering with K=2 applied to the training samples of the InternVL models. Distinct image features from the mixed dataset are extracted using the vision encoder (CLIP-ViT-B/16) and then clustered. Each main cluster is composed of multiple smaller clusters, corresponding to groups of similar objects.

5.2 Inference

During inference time, for a sample with image I we set cluster probabilities as:

$$p_t(S_k|x) = \frac{\exp(\tau \cdot \cos(x, c_k))}{\sum_{j=1}^{K} \exp(\tau \cdot \cos(x, c_j))} \tag{28}$$

where $x = f_{\text{CLIP}}(I)$ – CLIP features vector, $\cos(\cdot, \cdot)$ is the cosine similarity, f_{CLIP} is a feature extractor, c_j are cluster centroids, τ is the temperature parameter. This implies that routing is time independent and agnostic of current token sequence state.

To balance theoretical fidelity with computational efficiency, we employ a top-k strategy. The final output probabilities are top-k filtered and renormalized. For the main experiments, we take $k = 1$ to ensure that the ensemble inference computation budget (excluding routing cost) is the same as that of the dense model.

6 Experiments

In this section, we provide a comprehensive empirical evaluation of our proposed decentralized training paradigm. We first describe the core implementation details, including the specific training dataset, the vision encoder utilized for feature extraction, and the evaluation benchmarks. Subsequently, we present the

extensive experimental results for both the LLaVA and InternVL models to demonstrate the practical parity and robustness of our decentralized approach.

6.1 LLaVA

Training dataset For the finetuning of our decentralized expert models, we utilize the standard LLaVA visual instruction tuning dataset (Liu et al., 2024a). This widely adopted corpus aggregates diverse multi-modal instruction following data, including visual conversations, complex reasoning scenarios, and detailed image description tasks. By training our decentralized experts on this exact standard corpus, we ensure a rigorous foundation for evaluating our training approach.

Implementation details During the data partitioning stage, text-only samples are randomly and equally distributed between the clusters.

We train $K = 2$ experts. As an initialization checkpoint for experts, we employ the LLaVA-1.5-7B model after the vision language alignment stage. We effectively perform the visual instruction tuning stage of the experts.

Each expert is trained on the same number of GPU devices as the dense baseline. However, we halve the per-device batch size to ensure that the total number of training steps remains consistent with the original model training schedule. All other hyperparameters remain unchanged.

We utilize CLIP-ViT-L/14@336px as the vision encoder for our routing mechanism. We select this specific version because the LLaVA-1.5 architecture already requires it for visual processing. Consequently, with a shared encoder implementation, the feature extraction step for routing incurs almost zero additional computational overhead during inference.

Evaluation benchmarks and metrics For our empirical evaluation, we utilize a comprehensive suite of visual reasoning and question answering benchmarks. We specifically selected these datasets because they were prominently featured in the original LLaVA paper (Liu et al., 2024a), ensuring that our baseline comparisons are directly aligned with established literature. To assess general visual question answering, text reading, and scientific reasoning, we evaluate on VQAv2 (Goyal et al., 2017), GQA (Hudson & Manning, 2019), VizWiz (Gurari et al., 2018), TextVQA (Singh et al., 2019), and the image subset of ScienceQA (Lu et al., 2022). For these benchmarks, performance is measured using standard evaluation accuracy, following official consensus based scoring methods. Furthermore, we measure the susceptibility of our models to object hallucination using the POPE benchmark (Li et al., 2023), calculating the F1 score across its adversarial, random, and popular splits. To evaluate broader perception and cognition capabilities, we incorporate the MME benchmark (Fu et al., 2025), reporting its standard cumulative score. Finally, for the MMBench benchmark (Liu et al., 2024b), we test performance on both the English and Chinese subsets, reporting the overall multiple choice accuracy.

Results We evaluate our LLaVA models across a total of 8 benchmarks used by Liu et al. (2024a), comparing our performance directly against the official baseline statistics reported in their work. Overall, the expert ensemble achieves near-parity with the compute-matched dense baseline (see Tables 1 and 2) with small fluctuations. We observe small gains in visual-question answering (VQA) tasks and minor trade-offs in OCR tasks. The decentralized training preserves the baseline performance.

Method	VQAv2	GQA	VizWiz	SciQA-IMG	TextVQA
Dense baseline	78.50	**62.00**	**50.00**	66.80	**58.20**
2 experts	**79.99**	61.97	45.53	**67.03**	56.67

Table 1: LLaVA experts results on academic-task-oriented datasets

Method	POPE			MME	MMBench	
	adv	rand	pop		en	zh
Dense baseline	85.9	**87.3**	86.1	**1510.7**	64.3	**58.3**
2 experts	**87.1**	85.7	**87.3**	1477.38	**65.03**	56.87

Table 2: LLaVA experts results on benchmarks for instruction-following LLMs

6.2 InternVL

Training dataset In the second part of our experiments, we utilize a strategically curated subset of the InternVL 2.5 (Chen et al., 2025) Stage 2 finetuning data mixture. To ensure that our models develop a comprehensive set of multimodal capabilities without requiring the entire massive dataset, we constructed this mixture by deliberately sampling one or two representative datasets from every primary task category. As shown in Table 3, this methodology guarantees broad exposure across distinct domains. Specifically, our training mixture includes high quality data for image captioning, general visual question answering, chart comprehension, optical character recognition, knowledge based reasoning, document understanding, visual grounding, scientific reasoning, and complex conversational interactions. By intentionally selecting prominent datasets for each specific competency, we maintain a diverse training distribution that robustly evaluates our decentralized methodology across a wide spectrum of visual reasoning challenges.

Implementation details Consistent with the first experimental phase 6.1, we train $K = 2$ experts. We initialize the experts using the InternVL-2.5-1B-Pretrained checkpoint (post-Stage 1.5 ViT+MLP pretraining) and perform full Stage 2 finetuning. For dataset partitioning, we apply the same balanced k-means algorithm described previously.

To ensure a fair comparison, both the dense baseline and the experts are trained on an identical number of GPU devices with a per-device batch size of 1. We halve the number of gradient accumulation steps for the expert runs. This ensures that the total number of optimization steps remains comparable to the dense model baseline.

Unlike the standard InternVL training pipeline, we disable dynamic resolution to accelerate training. Additionally, the number of training devices is set to 2, and the context length is capped at 8192 tokens. All other hyperparameters remain unchanged.

We utilize CLIP-ViT-B/16 as a vision encoder for dataset partitioning and routing. This results an additional 5.4% computational overhead during inference compared to dense baseline.

We selected InternVL as an alternative to LLaVA to verify the effectiveness of our expert training strategy across different architectures and training setups. Specifically, this allows us to contrast two distinct paradigms: (1) LLaVA-1.5-7B, which uses a fixed CLIP vision encoder and only finetunes the MLP connector and LLM; and (2) InternVL 2.5-1B, which uses an Intern-ViT-300M encoder and performs full-parameter finetuning (ViT+MLP+LLM) during the instruction tuning stage.

Evaluation benchmarks and metrics For our InternVL setting, we evaluate the models across a diverse suite of 13 benchmarks used in the original InternVL-2.5 technical report (Chen et al., 2025), encompassing general question answering, OCR, and other multimodal tasks. We specifically selected this comprehensive array of datasets to thoroughly assess capabilities across general question answering, optical character recognition, and complex visual grounding. To assess general visual question answering and scientific reasoning, we measure standard accuracy on VQAv2 (Goyal et al., 2017), GQA (Hudson & Manning, 2019), AI2D (Kembhavi et al., 2016), and the image subset of ScienceQA (Lu et al., 2022). For text heavy tasks requiring robust optical character recognition, we evaluate performance on DocVQA (Mathew et al., 2021b) and InfoVQA (Mathew et al., 2021a) using the Average Normalized Levenshtein Similarity to account for minor text extraction variations, whereas for TextVQA (Singh et al., 2019) we report the standard VQA accuracy. Additionally, we measure chart comprehension on ChartQA (Masry et al., 2022) using a relaxed accuracy metric. To quantify the susceptibility of our models to object hallucination, we report the average score across the POPE (Li et al., 2023) benchmark, and we capture broader perception capabilities using

Task type	Datasets
Captioning	TextCaps (Sidorov et al., 2020), ShareGPT4o (Cui et al., 2024)
General QA	VQAv2(Goyal et al., 2017), GQA(Hudson & Manning, 2019)
Chart	ChartQA(Masry et al., 2022)
OCR	InfoVQA(Mathew et al., 2021a), TextVQA(Singh et al., 2019)
Knowledge	KVQA(Shah et al., 2019), A-OKVQA(Schwenk et al., 2022)
Document	DocVQA(Mathew et al., 2021b)
Grounding	RefCOCO/+/g (Mao et al., 2016; Yu et al., 2016)
Science	AI2D(Kembhavi et al., 2016), ScienceQA(Lu et al., 2022)
Conversation	ALLaVA(Chen et al., 2024)

Table 3: InternVL-type models finetuning data mixture

the cumulative score on MME (Fu et al., 2025). Finally, to evaluate precise spatial understanding, we test performance on the RefCOCO, RefCOCO+, and RefCOCOg (Yu et al., 2016) visual grounding benchmarks, reporting bounding box accuracy based on a standard Intersection over Union threshold.

Results Consistent with our LLaVA findings, the decentralized expert ensemble achieves near-parity with the compute-matched dense baseline (see Tables 4–6). We observe the experts yield slight gains on science question answering and diagram-based tasks, balanced by modest trade-offs on OCR tasks and broad multimodal evaluations. These results confirm that decentralized training preserves core abilities across different model architectures, with minor fluctuations reflecting the natural specialization of the experts.

Method	AI2D	ChartQA	TextVQA	DocVQA	InfoVQA
Dense baseline	61.66	61.96	**55.47**	48.49	**25.02**
2 experts	**62.50**	**62.24**	50.91	48.49	24.00

Table 4: InternVL experts results on OCR, chart, and document understanding

Method	VQAv2	GQA	SciQA-IMG	POPE (avg)	MME
Dense baseline	**72.5**	56.35	83.64	**86.17**	**1531.7**
2 experts	72.19	**57.36**	**83.89**	85.55	1475.78

Table 5: InternVL experts results on general QA, multimodal understanding and hallucination performance

	RefCOCO			RefCOCO+			RefCOCOg	
Method	val	testA	testB	val	testA	testB	val	test
Dense baseline	**77.05**	**82.98**	71.05	**64.82**	**73.47**	55.61	**72.18**	**71.34**
2 experts	75.47	80.20	**71.36**	62.46	71.29	**56.43**	71.94	70.45

Table 6: InternVL experts results on visual grounding

6.3 Ablation Study and other results

In this section, we conduct ablation studies under the InternVL setting over 3 factors: number of experts; the choice of vision encoder for image feature extraction for dataset partitioning and routing; and clustering algorithm.

The impact of number of experts For the main InternVL experiments, we train $K = 2$ experts. To test the impact of the number of experts, we also train $K = 4, 6$ experts using spherical balanced k-means

on CLIP ViT-B/16 features. As shown in Table 7, performance degrades in a controlled manner as K grows, with the largest drops on MME and TextVQA. The further increase in the number of experts to $K = 6$ leads to only marginal changes. The regression compared to the $K = 2$ setting is explained by data fragmentation, where the further partitioning reduces sample density per expert.

Method	VQAv2	TextVQA	MME	RefCOCO		
				val	testA	testB
2 experts	**72.19**	**50.91**	**1475.78**	**75.47**	**80.20**	**71.36**
4 experts	70.96	47.59	1330.41	72.05	78.47	66.79
6 experts	70.45	46.16	1353.1	70.92	76.21	67.30

Table 7: The impact of number of experts

The impact of vision encoder In the main InternVL experiments, we use CLIP ViT-B/16 as a feature extractor for k-means input for dataset partitioning and routing. To investigate the impact of the vision encoder, we also train $K = 2$ experts with balanced k-means algorithm using CLIP ViT-L/14@336px and CLIP RN50. As shown in Table 8, using a larger vision encoder leads to a marginal increase in QA (+0.4 VQAv2) and OCR (+0.81 TextVQA) performance. At the same time, using a smaller encoder leads to performance decrease on visual question answering (-4.14 VQAv2) and marginal improvement in visual grounding (+0.59 RefCOCO val).

Method	VQAv2	TextVQA	MME	RefCOCO		
				val	testA	testB
ViT-B/16	72.19	50.91	**1475.78**	75.47	80.20	**71.36**
ViT-L/14@336px	**72.56**	**51.72**	1450.04	74.74	80.75	71.01
RN50	68.05	47.5	1382.9	**76.06**	**81.74**	70.95

Table 8: The impact of vision encoder

The impact of clustering algorithm In the main InternVL experiments, we use spherical balanced k-means clustering algorithm for dataset partitioning and routing. To test the impact of the clustering algorithm, we also train $K = 2$ experts using 2-stage balanced spherical k-means algorithm inspired by McAllister et al. (2025). Namely, the first stage is fine clustering into $k = 1024$ clusters in an unbalanced manner. The second stage is coarse clustering of fine centroids into $K = 2$ coarse clusters in a balanced manner. As shown in Table 9, the 2-stage approach yields comparable performance with minor trade-offs on standard visual question-answering tasks. While we observe a wider performance variance on multi-task benchmarks, the overall competitive results across the majority of metrics demonstrate that the decentralized training paradigm is generally robust to the choice of clustering algorithm, though the simpler single-stage approach provides greater stability.

Method	VQAv2	TextVQA	MME	RefCOCO		
				val	testA	testB
balanced k-means	**72.19**	50.91	**1475.78**	**75.47**	80.20	**71.36**
2-stage balanced k-means	71.82	**51.69**	1421.01	74.76	**80.99**	69.99

Table 9: The impact of clustering algorithm

7 Conclusion

In this work, for the first time, we establish a rigorous theoretical foundation for the decentralized training of autoregressive models. We formally establish the theoretical equivalence between decentralized and

centralized training objectives. To demonstrate this, we prove that the global probability generating velocity can be exactly formulated as a linear combination of independent expert velocities. To achieve this mathematical decomposition, we extend the Discrete Flow Matching framework into the discrete time domain, introducing necessary constraints such as the 1-sparse property to provide a novel alignment with autoregressive sampling. Finally, we validate our theoretical findings with extensive empirical evidence. By successfully applying our decentralized training paradigm to Multimodal Large Language Models, including the LLaVA and InternVL architectures, we demonstrate its practical effectiveness and robust performance across a diverse set of question answering benchmarks.

References

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alexander Herzog, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control, 2023. URL https://arxiv.org/abs/2307.15818.

Andrew Campbell, Jason Yim, Regina Barzilay, Tom Rainforth, and Tommi Jaakkola. Generative flows on discrete state-spaces: Enabling multimodal flows with applications to protein co-design, 2024. URL https://arxiv.org/abs/2402.04997.

Guiming Hardy Chen, Shunian Chen, Ruifei Zhang, Junying Chen, Xiangbo Wu, Zhiyi Zhang, Zhihong Chen, Jianquan Li, Xiang Wan, and Benyou Wang. Allava: Harnessing gpt4v-synthesized data for lite vision-language models, 2024. URL https://arxiv.org/abs/2402.11684.

Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, Lixin Gu, Xuehui Wang, Qingyun Li, Yiming Ren, Zixuan Chen, Jiapeng Luo, Jiahao Wang, Tan Jiang, Bo Wang, Conghui He, Botian Shi, Xingcheng Zhang, Han Lv, Yi Wang, Wenqi Shao, Pei Chu, Zhongying Tu, Tong He, Zhiyong Wu, Huipeng Deng, Jiaye Ge, Kai Chen, Kaipeng Zhang, Limin Wang, Min Dou, Lewei Lu, Xizhou Zhu, Tong Lu, Dahua Lin, Yu Qiao, Jifeng Dai, and Wenhai Wang. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling, 2025. URL https://arxiv.org/abs/2412.05271.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities, 2025. URL https://arxiv.org/abs/2507.06261.

Erfei Cui, Yinan He, Zheng Ma, Zhe Chen, Hao Tian, Weiyun Wang, Kunchang Li, Yi Wang, Wenhai Wang, Xizhou Zhu, Lewei Lu, Tong Lu, Yali Wang, Limin Wang, Yu Qiao, and Jifeng Dai. Sharegpt-4o: Comprehensive multimodal annotations with gpt-4o, 2024. URL https://sharegpt4o.github.io/.

Oscar Davis, Samuel Kessler, Mircea Petrache, İsmail İlkan Ceylan, Michael Bronstein, and Avishek Joey Bose. Fisher flow matching for generative modeling over discrete data, 2024. URL https://arxiv.org/abs/2405.14664.

Anastasiia Filippova, Angelos Katharopoulos, David Grangier, and Ronan Collobert. No need to talk: Asynchronous mixture of language models. In *ICLR*, 2025. URL https://arxiv.org/abs/2410.03529.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, Rongrong Ji, Caifeng Shan, and Ran He. Mme: A comprehensive evaluation benchmark for multimodal large language models, 2025. URL https://arxiv.org/abs/2306.13394.

Itai Gat, Tal Remez, Neta Shaul, Felix Kreuk, Ricky T. Q. Chen, Gabriel Synnaeve, Yossi Adi, and Yaron Lipman. Discrete flow matching. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=GTDKo3Sv9p.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, et al. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025. ISSN 1476-4687. doi: 10.1038/s41586-025-09422-z. URL http://dx.doi.org/10.1038/s41586-025-09422-z.

Danna Gurari, Qing Li, Abigale J. Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P. Bigham. Vizwiz grand challenge: Answering visual questions from blind people, 2018. URL https://arxiv.org/abs/1802.08218.

Suchin Gururangan, Margaret Li, Mike Lewis, Weijia Shi, Tim Althoff, Noah A. Smith, and Luke Zettlemoyer. Scaling expert language models with unsupervised domain discovery, 2023. URL https://arxiv.org/abs/2303.14177.

Tom Henighan, Jared Kaplan, Mor Katz, Mark Chen, Christopher Hesse, Jacob Jackson, Heewoo Jun, Tom B. Brown, Prafulla Dhariwal, Scott Gray, Chris Hallacy, Benjamin Mann, Alec Radford, Aditya Ramesh, Nick Ryder, Daniel M. Ziegler, John Schulman, Dario Amodei, and Sam McCandlish. Scaling laws for autoregressive generative modeling, 2020. URL https://arxiv.org/abs/2010.14701.

Drew A. Hudson and Christopher D. Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering, 2019. URL https://arxiv.org/abs/1902.09506.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models, 2020. URL https://arxiv.org/abs/2001.08361.

Aniruddha Kembhavi, Michael Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. *ArXiv*, abs/1603.07396, 2016. URL https://api.semanticscholar.org/CorpusID:2682274.

Margaret Li, Suchin Gururangan, Tim Dettmers, Mike Lewis, Tim Althoff, Noah A. Smith, and Luke Zettlemoyer. Branch-train-merge: Embarrassingly parallel training of expert language models, 2022. URL https://arxiv.org/abs/2208.03306.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models, 2023. URL https://arxiv.org/abs/2305.10355.

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*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning, 2024a. URL https://arxiv.org/abs/2310.03744.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, Kai Chen, and Dahua Lin. Mmbench: Is your multi-modal model an all-around player?, 2024b. URL https://arxiv.org/abs/2307.06281.

Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *The 36th Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

Jiawei Ma, Po-Yao Huang, Saining Xie, Shang-Wen Li, Luke Zettlemoyer, Shih-Fu Chang, Wen-Tau Yih, and Hu Xu. Mode: Clip data experts via clustering, 2024. URL https://arxiv.org/abs/2404.16030.

Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions, 2016. URL https://arxiv.org/abs/1511.02283.

Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. Chartqa: A benchmark for question answering about charts with visual and logical reasoning, 2022. URL https://arxiv.org/abs/2203.10244.

Minesh Mathew, Viraj Bagal, Rubèn Pérez Tito, Dimosthenis Karatzas, Ernest Valveny, and C. V Jawahar. Infographicvqa, 2021a. URL https://arxiv.org/abs/2104.12756.

Minesh Mathew, Dimosthenis Karatzas, and C. V. Jawahar. Docvqa: A dataset for vqa on document images, 2021b. URL https://arxiv.org/abs/2007.00398.

David McAllister, Matthew Tancik, Jiaming Song, and Angjoo Kanazawa. Decentralized diffusion models, 2025. URL https://arxiv.org/abs/2501.05450.

Yiming Qin, Manuel Madeira, Dorina Thanou, and Pascal Frossard. Defog: Discrete flow matching for graph generation. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=KPRIwWhqAZ.

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, 2021. URL https://arxiv.org/abs/2103.00020.

Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. A-okvqa: A benchmark for visual question answering using world knowledge, 2022. URL https://arxiv.org/abs/2206.01718.

Sanket Shah, Anand Mishra, Naganand Yadati, and Partha Pratim Talukdar. Kvqa: Knowledge-aware visual question answering. In *AAAI*, 2019.

Oleksii Sidorov, Ronghang Hu, Marcus Rohrbach, and Amanpreet Singh. Textcaps: a dataset for image captioning with reading comprehension, 2020. URL https://arxiv.org/abs/2003.12462.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read, 2019. URL https://arxiv.org/abs/1904.08920.

Keyu Tian, Yi Jiang, Zehuan Yuan, BINGYUE PENG, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=gojL67CfS8.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. URL https://arxiv.org/abs/1706.03762.

Zifan Wang, Anqi Dong, Mahmoud Selim, Michael M. Zavlanos, and Karl H. Johansson. Federated flow matching, 2025. URL https://arxiv.org/abs/2509.21250.

Licheng Yu, Patrick Poirson, Shan Yang, Alexander C. Berg, and Tamara L. Berg. Modeling context in referring expressions, 2016. URL https://arxiv.org/abs/1608.00272.

Siyi Zhou, Yiquan Zhou, Yi He, Xun Zhou, Jinchao Wang, Wei Deng, and Jingchen Shu. Indextts2: A breakthrough in emotionally expressive and duration-controlled auto-regressive zero-shot text-to-speech, 2025. URL https://arxiv.org/abs/2506.21619.