scMeanFlow: Fast, High-Quality, and Controllable scRNA-seq Synthesis via One-Step MeanFlow Matching

Anonymous authors
Paper under double-blind review

Abstract

Generative modeling of single-cell RNA sequencing (scRNA-seq) data shows great potential in alleviating sample scarcity and enhancing downstream analyses. However, existing deep generative models generally face critical limitations: they either operate on continuous approximations that ignore the inherently discrete and over-dispersed nature of raw counts, or rely on multi-step iterative sampling that severely restricts inference scalability. Furthermore, flexible condition injection mechanisms remain insufficiently explored. To address these challenges simultaneously, we propose **scMeanFlow**, an efficient two-stage generative framework. First, a Negative Binomial Autoencoder encodes discrete count vectors into a compact continuous latent space while explicitly preserving count-level statistical properties via a probabilistic decoder. Second, operating within this latent space, a MeanFlow network learns the interval-averaged velocity field for genuine one-step generation without ordinary differential equation (ODE) solvers, while Feature-wise Linear Modulation (FiLM) injects condition information through feature-wise affine modulation. Random condition dropout during training enables a single trained model to support unconditional, single-condition, and multi-condition synthesis. Extensive experiments on four real scRNA-seq datasets demonstrate that scMeanFlow achieves state-of-the-art generation quality. Notably, its inference speed is approximately 7x faster than the strongest flow-based baseline and over 45x faster than diffusion-based methods. Furthermore, scMeanFlow effectively improves rare cell type classification through data augmentation. Overall, scMeanFlow provides an efficient, count-aware, and controllable solution for single-cell data synthesis.

1 Introduction

Single-cell RNA sequencing (scRNA-seq) has revolutionized the characterization of cellular heterogeneity and tissue microenvironments by profiling gene expression at single-cell resolution Xu et al. (2026a); Luecken & Theis (2019). Facing critical challenges such as high wet-lab costs, clinical sample scarcity, and data privacy constraints Hwang et al. (2018); Xu et al. (2026b), scRNA-seq data generation has emerged as a vital computational paradigm aimed at synthesizing high-fidelity transcriptomic profiles via data-driven distribution modeling. Deep generative models can effectively enrich and supplement single-cell transcriptomic data, providing crucial support for key downstream tasks such as rare cell type data augmentation and cellular perturbation response prediction Rivero-Garcia et al. (2024); Hetzel et al. (2022); Xu et al. (2025).

Driven by the pursuit of higher biological fidelity and controllability, generative paradigms for scRNA-seq data have evolved significantly. Early **statistical simulation** methods, such as Splatter Zappia et al. (2017), parameterize marginal gene-wise distributions but fail to capture the complex joint expression patterns across cells. To model non-linear relationships, **VAE-based** approaches like scVI Lopez et al. (2018) were introduced; however, they inherently focus on representation learning and data reconstruction rather than direct synthesis from noise. To enhance realism, **GAN-based** models (e.g., scGAN Marouf et al. (2020)) employ adversarial training but frequently suffer from training instability and mode collapse, limiting their reliability. Recently, **diffusion-based** models (e.g., scDiffusion Luo et al. (2024), cfDiffusion Zhang et al. (2025),

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

scLDM Palla et al. (2026)) have achieved a leap in generative fidelity through Markovian denoising processes and gradient-guided condition mechanisms. Yet, they suffer from a severe speed–quality trade-off, requiring hundreds to thousands of iterative steps for inference. Most recently, continuous **flow-based** models like CFGen Palma et al. (2025) were proposed to explicitly model discrete, over-dispersed counts via negative binomial decoders, avoiding the noisy diffusion process. Nevertheless, they still depend on multi-step numerical integration via ordinary differential equation (ODE) solvers at inference time.

Despite advances in generative modeling, large-scale synthesis of scRNA-seq data faces three key limitations: (1) **Reliance on multi-step iterative sampling and lack of direct single-step generation.** Mainstream high-fidelity models, such as diffusion and continuous flow matching architectures, intrinsically depend on multi-step Markovian denoising or numerical integration of ODEs. This iterative sampling mechanics imposes substantial inference latency, fundamentally precluding rapid generation. (2) **Inability to reconcile generation efficiency with biological fidelity.** Conventional variational autoencoder frameworks such as scVI enable rapid sampling, but their reconstruction-oriented generative paradigm limits their ability to generate novel transcriptomic profiles. Conversely, diffusion and continuous flow models improve synthesis quality at the expense of computational speed. Consequently, current paradigms fail to simultaneously achieve single-step inference and count-aware high fidelity. (3) **Simplistic condition injection mechanisms.** Prevailing methods incorporate biological priors and experimental variables, such as cell types or batch effects, primarily through shallow feature concatenation Zhang et al. (2025) or element-wise addition Palma et al. (2025). Such linear schemes fail to capture the complex, non-linear interactions between conditional variables and the high-dimensional gene expression manifold.

To address these impediments, we propose **scMeanFlow**, a two-stage generative framework designed for fast, high-quality, and controllable **sc**RNA-seq synthesis via one-step **MeanFlow** Matching Geng et al. (2025). **In the first stage**, a Negative Binomial Autoencoder (NB-AE) constructs a count-aware latent space by: (1) encoding high-dimensional discrete counts into a compact continuous representation, and (2) explicitly parameterizing discrete and over-dispersed count statistics via an NB decoder to preserve biological fidelity. **In the second stage**, scMeanFlow performs rapid, controllable synthesis within this learned latent space by: (1) parameterizing an interval-averaged velocity field via a latent MeanFlow model, enabling one-step generation without ODE solvers or trajectory distillation ; (2) injecting biological and technical condition embeddings through Feature-wise Linear Modulation (FiLM) Perez et al. (2018), which applies learned scale-and-shift transformations to intermediate features to capture complex non-linear condition interactions; and (3) applying random condition dropout during training so one model handles condition subsets. By configuring the condition set at inference, a single trained scMeanFlow model supports unconditional, single-condition, and multi-condition synthesis. Our contributions are summarized as follows:

- We propose scMeanFlow, a two-stage framework that combines an NB-AE with a latent MeanFlow model, enabling count-aware one-step generation while preserving the discrete and over-dispersed characteristics of raw scRNA-seq data.
- We develop a flexible condition injection framework based on FiLM. It uses biological and technical covariate embeddings to apply learned affine transformations to intermediate representations. By independently dropping conditions during training and explicitly selecting the desired condition subset at inference, a single model supports unconditional, single-condition, and multi-condition synthesis.
- Extensive benchmarks confirm that scMeanFlow achieves state-of-the-art synthesis fidelity alongside unprecedented sampling efficiency, and effectively improves downstream rare cell type classification through targeted data augmentation.

2 Related Work

Generative Models for scRNA-seq. Generative modeling of scRNA-seq data has evolved from early probabilistic autoencoders to advanced iterative generative architectures. Variational autoencoders like scVI Lopez et al. (2018) model discrete expression using count-based likelihoods, while GAN-based approaches such as scGAN Marouf et al. (2020) and ACTIVA Heydari et al. (2022) incorporate adversarial training for cell population synthesis. More recently, diffusion models including scDiffusion Luo et al. (2024), cfDiffusion Zhang et al. (2025), and scLDM Palla et al.

(2026) achieve high fidelity through progressive denoising, yet suffer from severe inference latency due to iterative multi-step sampling. Continuous flow models, represented by CFGen Palma et al. (2025), attempt to improve sampling efficiency through probability transport, but remain bounded by numerical ODE solvers.

Diffusion and Flow-based Generative Models. Diffusion models generate samples through progressive denoising Ho et al. (2020), but suffer from high computational costs during inference. To reduce sampling latency, acceleration methods such as DDIM Song et al. (2020), consistency models Song et al. (2023), and progressive distillation Salimans & Ho (2022) enable generation with fewer evaluation steps. Flow-based generative models instead learn continuous probability transport between simple distributions and complex data. Recent frameworks, including Flow Matching Lipman et al. (2022) and Rectified Flow Liu et al. (2023), improve sampling efficiency by constructing straighter transport trajectories. MeanFlow Geng et al. (2025) further advances this paradigm by modeling interval-averaged velocity fields over time intervals, achieving genuine one-step generation without requiring numerical ODE solvers or trajectory distillation.

3 METHODOLOGY

3.1 FRAMEWORK OVERVIEW

The scMeanFlow framework, illustrated in Fig. 1, establishes an efficient two-stage generative framework that combines count-aware representation learning with one-step latent generation. To reconcile sampling efficiency with biological fidelity while expanding conditional flexibility, scMeanFlow couples an NB-AE with a latent MeanFlow generator equipped with FiLM-based modulation. In Stage 1 (**Count-Aware Latent Representation**), an NB-AE maps raw gene expression profiles into a smooth continuous space while explicitly preserving discrete and over-dispersed count statistics. In Stage 2 (**Latent MeanFlow Generation and Controllable Conditioning**), the framework achieves rapid and flexible synthesis through two core mechanisms: (1) **One-Step Latent Generation**, which models interval-averaged velocity fields to enable direct one-step generation without numerical ODE solvers; and (2) **Unified FiLM-Based Condition Injection**, which aggregates active condition embeddings and applies feature-wise affine modulation, with stochastic condition dropout during training enabling a shared model to support unconditional, single-condition, and multi-condition generation.

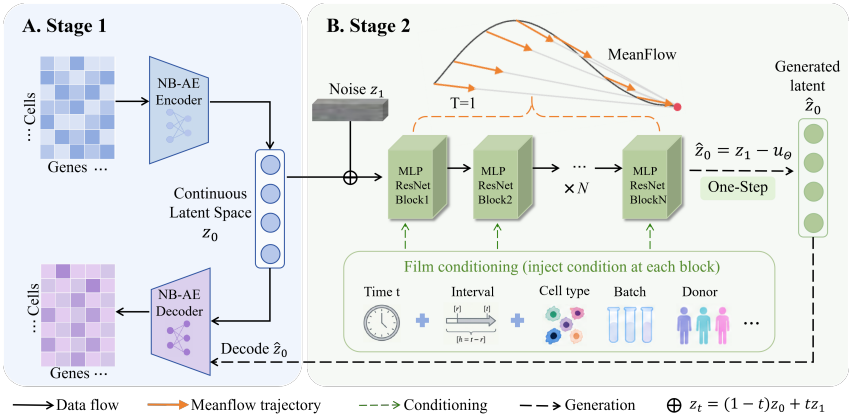

Figure 1: Framework Overview of the Proposed scMeanFlow.

3.2 COUNT-AWARE LATENT REPRESENTATION

To bridge discrete single-cell transcriptomic profiles with continuous generative modeling, we first establish a count-aware continuous latent space via a Negative Binomial Autoencoder (NB-AE). Let $\mathbf{x}_n = [x_{n1}, \ldots, x_{nG}]^\top \in \mathbb{N}_0^G$ denote the raw expression count vector of cell n across G genes, with library size $s_n = \sum_{g=1}^{G} x_{ng}$.

The deterministic encoder f_η maps the raw count vector $\mathbf{x}_n$ into a continuous latent code $\mathbf{z}_0 \in \mathbb{R}^d$:

$$\mathbf{z}_0 = f_\eta(\mathbf{x}_n). \tag{1}$$

To explicitly model the discrete and over-dispersed statistical properties of scRNA-seq data, the decoder h_ψ predicts normalized gene expression proportions, which are rescaled by the cell library size s_n to yield the mean parameter vector $\boldsymbol{\mu}_n \in \mathbb{R}_+^G$:

$$\boldsymbol{\mu}_n = s_n \cdot \mathrm{softmax}(h_\psi(\mathbf{z}_0)). \tag{2}$$

Each observed count x_{ng} is modeled by a Negative Binomial distribution, $x_{ng} \sim \mathcal{NB}(\mu_{ng}, \theta_g)$, where $\theta_g > 0$ denotes a learned gene-specific dispersion parameter. The NB-AE is optimized end-to-end by minimizing the Negative Binomial negative log-likelihood loss:

$$\mathcal{L}_{\mathrm{AE}} = -\frac{1}{N}\sum_{n=1}^{N}\sum_{g=1}^{G} \log p_{\mathcal{NB}}\left(x_{ng} \mid \mu_{ng}, \theta_g\right). \tag{3}$$

3.3 One-Step Latent Generation

To enable rapid, direct single-step synthesis without relying on numerical ODE solvers, scMeanFlow models latent trajectory transport via conditional MeanFlow matching. Given a latent real-cell vector $\mathbf{z}_0$ and a Gaussian noise prior $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, we define a linear probability path:

$$\mathbf{z}_t = (1-t)\mathbf{z}_0 + t\boldsymbol{\epsilon}, \qquad \mathbf{v}_t = \frac{d\mathbf{z}_t}{dt} = \boldsymbol{\epsilon} - \mathbf{z}_0. \tag{4}$$

Unlike standard Flow Matching, which fits an instantaneous velocity field $\mathbf{v}(\mathbf{z}_t, t)$ and requires iterative numerical integration during sampling, MeanFlow learns the average velocity field $\mathbf{u}(\mathbf{z}_t, r, t)$ over any time interval $[r, t]$:

$$\mathbf{u}(\mathbf{z}_t, r, t) \triangleq \frac{1}{t-r}\int_r^t \mathbf{v}(\mathbf{z}_\tau, \tau)\, d\tau. \tag{5}$$

Directly learning $\mathbf{u}$ via this integral definition is computationally intractable. Instead, MeanFlow enforces an exact differential identity derived by taking the total time derivative of $(t-r)\mathbf{u}(\mathbf{z}_t, r, t) = \int_r^t \mathbf{v}(\mathbf{z}_\tau, \tau)\, d\tau$ with respect to t:

$$\mathbf{u}(\mathbf{z}_t, r, t) = \mathbf{v}(\mathbf{z}_t, t) - (t-r)\left[(\partial_{\mathbf{z}}\mathbf{u})\,\mathbf{v}_t + \partial_t \mathbf{u}\right]. \tag{6}$$

In scMeanFlow, we parameterize the average velocity field as $\mathbf{u}_\theta(\mathbf{z}_t, r, t \mid \mathbf{e}_\mathcal{C})$, where $\mathbf{e}_\mathcal{C}$ denotes the combined embedding of the active conditions. The model is trained end-to-end by minimizing a regression loss with a stop-gradient operator $\mathrm{sg}(\cdot)$:

$$\mathcal{L}_{\mathrm{MF}}(\theta) = \mathbb{E}_{\mathbf{z}_0, \boldsymbol{\epsilon}, r, t, \mathcal{C}} \left\|\mathbf{u}_\theta(\mathbf{z}_t, r, t \mid \mathbf{e}_\mathcal{C}) - \mathrm{sg}(\mathbf{u}_{\mathrm{tgt}})\right\|_2^2, \tag{7}$$

where the regression target is defined as:

$$\mathbf{u}_{\mathrm{tgt}} = \mathbf{v}_t - (t-r)\left[(\partial_{\mathbf{z}}\mathbf{u}_\theta)\,\mathbf{v}_t + \partial_t \mathbf{u}_\theta\right]. \tag{8}$$

During inference, this learned interval-averaged formulation naturally achieves direct one-step generation. By evaluating the network across the entire time interval ($r = 0, t = 1$), a target cell latent vector $\widehat{\mathbf{z}}_0$ is directly synthesized from a noise sample $\mathbf{z}_1 = \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ conditioned on $\mathbf{e}_\mathcal{C}$ in a single forward pass:

$$\widehat{\mathbf{z}}_0 = \mathbf{z}_1 - \mathbf{u}_\theta(\mathbf{z}_1, 0, 1 \mid \mathbf{e}_\mathcal{C}). \tag{9}$$

The generated latent vector $\hat{z}_0$ is decoded by the frozen NB decoder h_ψ. Given a library size s, the decoder first produces normalized gene-expression proportions, from which the Negative Binomial mean parameters and the final count samples are obtained:

$$\hat{\rho} = \mathrm{softmax}\left(h_\psi(\hat{z}_0)\right), \qquad \hat{\mu} = s\hat{\rho}, \qquad \hat{x}_g \sim \mathrm{NB}\left(\hat{\mu}_g, \theta_g\right),\ g = 1, \ldots, G, \tag{10}$$

where $\hat{x} \in \mathbb{N}_0^G$ is the final synthetic scRNA-seq count vector.

3.4 UNIFIED FiLM-BASED CONDITION INJECTION

Most existing single-cell generative models inject conditions by directly concatenating or adding condition embeddings to intermediate hidden features. However, such static injection mechanisms merely introduce coarse additive shifts, failing to dynamically adapt feature channels based on nonlinear condition interactions. To address this limitation, we integrate a Feature-wise Linear Modulation (FiLM) mechanism into the MeanFlow generator, enabling more flexible and highly expressive conditioning.

Let $\mathcal{Y}$ denote the available categorical condition types (e.g., cell type, batch, or donor). Each condition type $c \in \mathcal{Y}$, with categorical value y_c, is mapped via an independent learnable embedding $\mathbf{e}_c(y_c)$. For an active condition-type set $\mathcal{C} \subseteq \mathcal{Y}$, the joint condition vector $\mathbf{e}_\mathcal{C}$ incorporates these condition embeddings alongside temporal representations of the current time t and interval length $(t-r)$:

$$\mathbf{e}_\mathcal{C} = \mathbf{e}_t(t) + \mathbf{e}_h(t-r) + \sum_{c\in\mathcal{C}} \mathbf{e}_c(y_c), \tag{11}$$

where $\mathbf{e}_t$ and $\mathbf{e}_h$ encode the time and interval projections, respectively.

In each residual block l, condition injection is operationalized through a Feature-wise Linear Modulation (FiLM) layer. Specifically, a projection sub-network $\Psi^{(l)}$ maps the joint condition vector $\mathbf{e}_\mathcal{C}$ into channel-wise scale ($\boldsymbol{\gamma}^{(l)}$) and shift ($\boldsymbol{\beta}^{(l)}$) parameters:

$$\mathbf{a}^{(l)} = f_1^{(l)}(\mathbf{h}^{(l-1)}), \qquad [\boldsymbol{\gamma}^{(l)}, \boldsymbol{\beta}^{(l)}] = \Psi^{(l)}(\mathbf{e}_\mathcal{C}), \tag{12}$$

$$\mathbf{h}^{(l)} = \mathbf{h}^{(l-1)} + f_2^{(l)}\Big(\mathbf{a}^{(l)} \odot (\mathbf{1} + \boldsymbol{\gamma}^{(l)}) + \boldsymbol{\beta}^{(l)}\Big), \tag{13}$$

where $\odot$ denotes the element-wise Hadamard product, $\mathbf{1}$ is an all-ones vector, and $f_1^{(l)}, f_2^{(l)}$ represent block-level transformations. Through this FiLM affine modulation, biological and technical conditions dynamically rescale and shift intermediate hidden features at the channel level, enabling fine-grained control over the generative process.

3.5 OPTIMIZATION

scMeanFlow is optimized in two stages. First, the NB-AE is trained by minimizing $\mathcal{L}_{\mathrm{AE}}$ in Eq. equation 3. With the encoder f_η and decoder h_ψ frozen, the latent MeanFlow generator is optimized with $\mathcal{L}_{\mathrm{MF}}$ in Eq. equation 7. During MeanFlow training, we apply random condition dropout. Each condition type $c \in \mathcal{Y}$ is independently kept with probability $p_{\mathrm{cond}} = 0.8$ and otherwise masked to zero. The active set is

$$\mathcal{C}_{\mathrm{train}} = \{\, c \in \mathcal{Y} : M_c = 1 \,\}, \qquad M_c \sim \mathrm{Bernoulli}(p_{\mathrm{cond}}). \tag{14}$$

This scheme allows all $2^{|\mathcal{Y}|}$ condition subsets to be sampled during training, from $\mathcal{C}_{\mathrm{train}} = \varnothing$ to $\mathcal{C}_{\mathrm{train}} = \mathcal{Y}$. At inference no dropout is applied. The active set is chosen explicitly as $\varnothing$ for unconditional generation, a single condition for single-condition generation, or several conditions for multi-condition generation. The retained embeddings are summed with the time and interval embeddings via Eq. equation 11 and injected through FiLM. One model therefore supports all three modes without retraining.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Dataset. Model performance is benchmarked on four diverse scRNA-seq datasets: PBMC 68k Zheng et al. (2017), Tabula Muris Schaum et al. (2018), PBMC 3k 10x Genomics (2016), and Dentate Gyrus La Manno et al. (2018). We use the full gene space without highly variable gene filtering, condition generation on cell type, and apply a random 80/20 train/test split to each dataset. Multi-condition generation is additionally evaluated on the NeurIPS 2021 bone marrow dataset Luecken et al. (2021), which contains cell-type and site covariates.

Baseline Methods. We compare scMeanFlow against six representative baselines across three categories: (1) VAE-based Generative Models: These models learn latent representations through variational inference and reconstruct gene expression profiles using probabilistic decoders, including scVI Lopez et al. (2018) and ACTIVA Heydari et al. (2022). (2) Diffusion-based Generative Models: These approaches formulate scRNA-seq synthesis as a progressive denoising process, modeling data distributions via iterative noise removal, including scDiffusion Luo et al. (2024), cfDiffusion Zhang et al. (2025), and scLDM Palla et al. (2026). (3) Flow-based Generative Models: These methods learn continuous transport trajectories between simple priors and biological data distributions, represented by CFGen Palma et al. (2025).

Evaluation Metrics. Generation quality is evaluated in a shared 50-dimensional PCA space using Maximum Mean Discrepancy (MMD, ↓) for global alignment, 2-Wasserstein Distance (WD, ↓) for manifold fidelity, and kNN F1-score (KNNc, ↑) for local cell-type structure preservation.

4.2 Quantitative Results

Table 1: Conditional generation quality on four scRNA-seq datasets over three random seeds. Each value is reported as mean±std. **Bold** and underline denote best/runner-up.

Dataset	Metric	scVI	ACTIVA	scDiffusion	cfDiffusion	CFGen	scLDM	Ours
PBMC 68k	MMD↓	0.86±0.00	1.24±0.00	0.84±0.01	0.75±0.01	0.27±0.00	0.27±0.00	**0.25±0.00**
	WD↓	20.78±0.03	26.67±0.09	19.60±0.10	19.25±0.18	12.55±0.05	12.30±0.19	**11.97±0.02**
	KNNc↑	0.51±0.01	0.36±0.00	0.18±0.00	0.18±0.00	0.58±0.01	0.59±0.01	**0.60±0.00**
Tabula Muris	MMD↓	0.63±0.00	0.78±0.00	0.31±0.00	0.47±0.00	0.26±0.00	0.32±0.00	**0.22±0.00**
	WD↓	23.31±0.06	30.68±0.04	12.95±0.00	12.57±0.07	12.93±0.03	11.56±0.08	**11.48±0.03**
	KNNc↑	0.86±0.01	0.75±0.00	0.91±0.00	0.94±0.00	0.93±0.00	0.88±0.00	**0.95±0.00**
PBMC 3k	MMD↓	1.00±0.01	0.98±0.01	1.01±0.02	1.09±0.02	0.72±0.00	0.66±0.01	**0.66±0.00**
	WD↓	24.10±0.43	31.80±0.61	25.68±0.42	25.94±1.04	22.51±0.10	22.50±0.74	**21.07±0.43**
	KNNc↑	0.48±0.01	0.23±0.02	0.32±0.01	0.34±0.01	0.46±0.01	0.39±0.01	**0.58±0.00**
Dentate Gyrus	MMD↓	0.91±0.00	0.93±0.00	0.40±0.00	0.49±0.00	0.33±0.00	0.37±0.00	**0.28±0.00**
	WD↓	25.63±0.07	29.27±0.12	**10.94±0.04**	12.30±0.04	13.41±0.04	12.11±0.06	12.22±0.03
	KNNc↑	0.87±0.00	0.27±0.02	**0.95±0.00**	0.92±0.00	0.86±0.00	0.82±0.01	0.92±0.00

Table 1 reports the conditional generation performance across four benchmark datasets. scMeanFlow achieves the best performance in 10 out of 12 dataset–metric pairs, ranking first across all metrics on PBMC 68k, Tabula Muris, and PBMC 3k. On Dentate Gyrus, scMeanFlow obtains the superior MMD (0.28) and runner-up KNNc (0.92), with scDiffusion leading on WD. Notably, scMeanFlow achieves these fidelity gains in a single inference step, bypassing the expensive numerical integration required by existing flow and diffusion models.

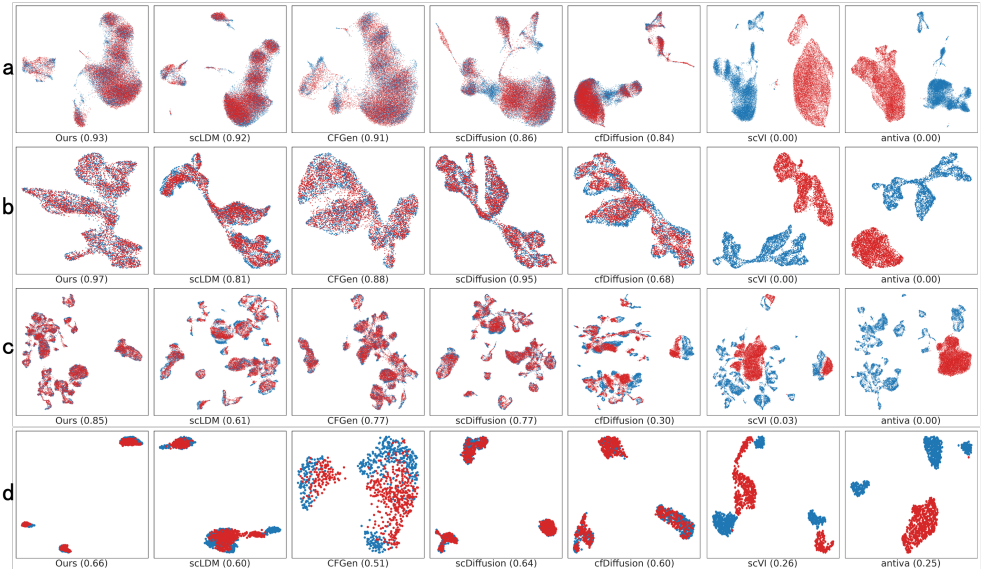

Figure 2: UMAP visualizations of real and scMeanFlow-generated cells on four scRNA-seq benchmark datasets: (a) PBMC 68k, (b) Dentate Gyrus, (c) Tabula Muris, and (d) PBMC 3k.Values in parentheses are the 15-NN overlap scores for each method.

4.3 BIOLOGICAL FIDELITY

As qualitative UMAP projections in Fig. 2 illustrate, scMeanFlow-generated cells (red) seamlessly overlap with reference real cells (blue) across all four benchmark datasets. Quantitatively, scMeanFlow consistently achieves the highest 15-nearest-neighbor (15-NN) overlap score on every dataset: 0.93 on PBMC 68k, 0.97 on Dentate Gyrus, 0.85 on Tabula Muris, and 0.66 on PBMC 3k. These metrics substantially outperform state-of-the-art flow and diffusion models as well as traditional frameworks (e.g., scVI and ACTIVA, which suffer from severe manifold separation).

Table 2: Marker gene quantitative comparison on PBMC 68k.

Method	Scope	Pearson $r\uparrow$	RMSE$\downarrow$	MAE$\downarrow$
Ours	Expression	**0.9860**	**0.1484**	**0.0846**
	Fraction	**0.9722**	**5.7033**	**3.2581**
scLDM	Expression	0.9827	0.1660	0.0923
	Fraction	0.9593	7.2004	3.9441
CFGen	Expression	0.9751	0.1998	0.1208
	Fraction	0.9442	8.0171	4.8151
scDiffusion	Expression	0.0431	1.2154	0.6108
	Fraction	0.1844	59.1038	54.4847

At the gene-specific level, Fig. 3 shows that scMeanFlow faithfully reproduces cell-type marker gene profiles in both average expression levels and detection fractions. This is further substantiated by Tab. 2, where scMeanFlow yields the highest Pearson correlation alongside the lowest RMSE and MAE across all marker metrics. Collectively, these evaluations confirm that scMeanFlow captures both global cellular landscapes and fine-grained biological patterns.

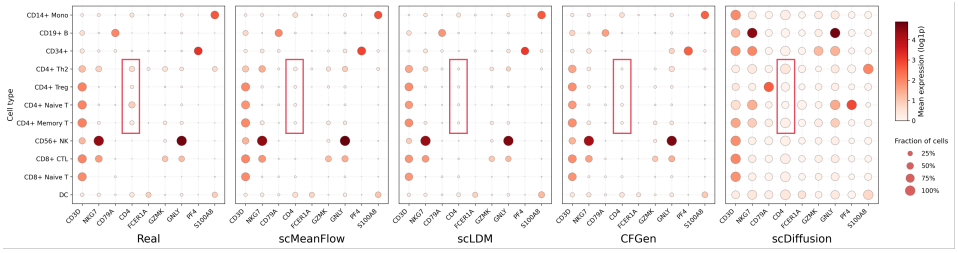

Figure 3: Marker gene dot plots on PBMC 68k. Dot color: average expression; dot size: detection fraction.

4.4 SAMPLING EFFICIENCY

To assess whether one-step generative transport achieves practical inference acceleration, we measure the sampling time on a single NVIDIA V100 GPU: each method generates 10,000 cells (10 batches of 1,000 cells) using its default number of generation steps, and the timing covers the full pipeline from Gaussian noise to final gene counts, including NB decoding. As shown in Fig. 4 and Fig. 5, scMeanFlow completes generation in just 0.72s, outperforming the fastest baseline CFGen (5.15s) by 7.1$\times$ and the fastest diffusion model cfDiffusion (33.28s) by 45$\times$. These results confirm that scMeanFlow effectively resolves the primary inference bottleneck by bypassing iterative numerical integration, thereby enabling scalable real-time single-cell synthesis.

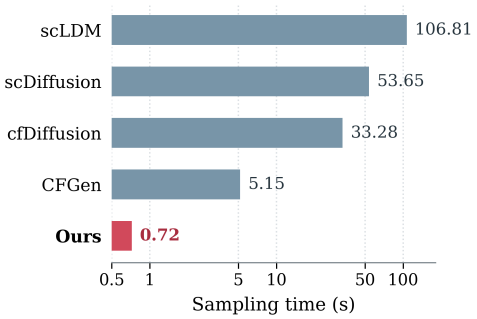

Figure 4: Sampling efficiency for generating 10,000 cells.

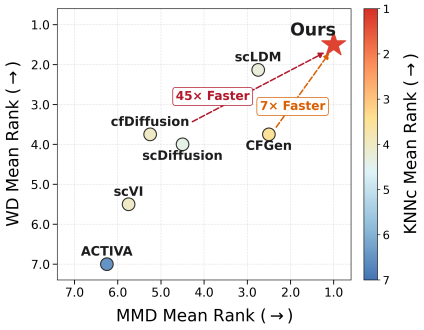

Figure 5: Overall method ranking across evaluation metrics.

4.5 Multi-condition Generation

We evaluate whether a single scMeanFlow model can control biological and technical covariates at once. On the NeurIPS 2021 bone marrow dataset, we test three regimes: a technical attribute (Site1), a biological attribute (CD14+ Mono), and their joint combination (CD14+ Mono and Site1). As shown in Fig. 6, generated cells (red) match the manifold position of the corresponding real cells (blue) against the full cell background (grey). Under joint conditioning, the model isolates the target subpopulation at the target site without distributional drift. Unconditional generation, which uses the empty condition set, is demonstrated on the four benchmark datasets in Fig. 12. These results confirm that one model supports unconditional, single-condition, and multi-condition generation without retraining per combination.

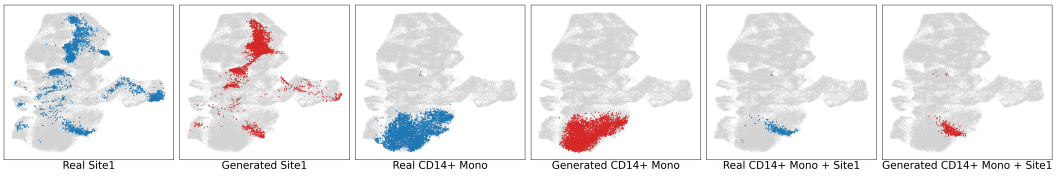

Figure 6: UMAP visualizations of controllable multi-condition generation on the NeurIPS 2021 bone marrow benchmark across single-condition and joint-condition settings.

4.6 Rare Cell Annotation Enhancement

To evaluate whether synthetic cell generation improves downstream cell-type annotation beyond global distribution alignment, we benchmark scMeanFlow on the imbalanced PBMC 68k dataset (80%/20% train-test split). We construct an augmented training set by oversampling low-frequency cell types via scMeanFlow and evaluate it against an equal-sized bootstrap resampling baseline. Both setups share a 50-dimensional PCA space derived from 2,000 highly variable genes, employing kNN classifiers ($k = 10$) to measure per-cell-type F1 score improvements on held-out test data.

As Fig. 7 illustrates, scMeanFlow demonstrates a pronounced inverse correlation between F1 gains and relative cell-type frequency ($r = -0.89$). For ultra-rare cell populations with frequencies near 10^{-3}, scMeanFlow yields substantial accuracy improvements approaching 0.50 (blue points), whereas bootstrap resampling gains remain capped near 0.20 with a weaker correlation ($r = -0.78$). Furthermore, the markedly steeper regression trajectory of scMeanFlow indicates that direct generative mapping enables more precise augmentation of rare cell types, rather than merely repeating existing sparse samples.

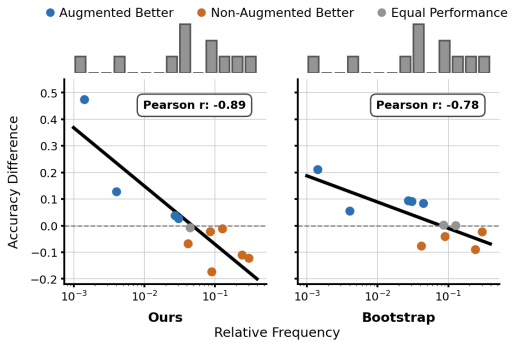

Figure 7: F1 score gains by cell-type frequency on PBMC 68k (scMeanFlow vs. Bootstrap).

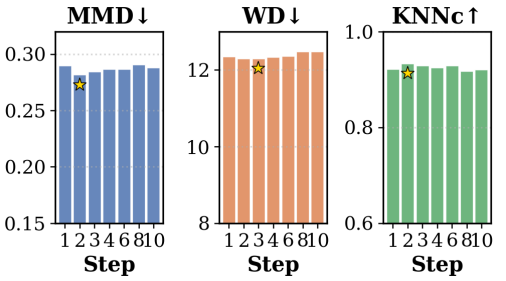

Figure 8: Generation quality across sampling steps on the Dentate Gyrus benchmark. Stars mark optimal step counts.

4.7 Generation Step Analysis

To investigate the effect of generation steps, we evaluate scMeanFlow on the Dentate Gyrus dataset. As shown in Fig. 8, one-step generation already achieves competitive generation quality. Increasing the number of steps to 2–3 further improves some metrics, but beyond three steps, only limited and inconsistent performance gains are observed. These results indicate that scMeanFlow supports

flexible generation strategies: one-step generation can serve as an efficient default setting, while additional steps can be adopted when higher generation fidelity is desired.

4.8 Ablation Study

To evaluate the contributions of FiLM and the NB-AE, we conduct two ablation studies.

FiLM ablation. We replace FiLM with additive conditioning while keeping the remaining architecture and training procedure unchanged. As shown in Fig. 9, the full model achieves lower MMD and WD across all four datasets, with higher KNNc on three datasets. These consistent improvements indicate that feature-wise modulation adapts more effectively to condition information than uniform additive conditioning, leading to better global distribution alignment and preservation of local cell type structure.

NB-AE ablation. We replace the NB-AE with a standard autoencoder while keeping the MeanFlow generator unchanged. As shown in Fig. 10, the NB-AE better preserves the gene-wise mean–variance relationship and per-cell zero-count distribution of real data on PBMC 3k. These results demonstrate that the Negative Binomial observation model is important for retaining characteristic count-level variation and sparsity in the generated transcriptomes.

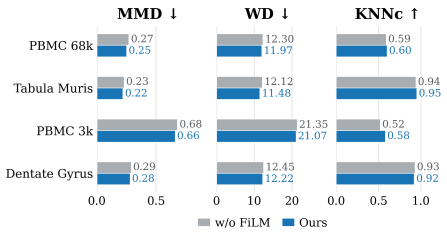

Figure 9: Ablation: FiLM vs. additive conditioning across four datasets.

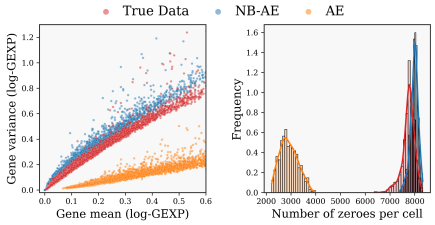

Figure 10: Comparison of gene-wise mean–variance trends and per-cell zero count distributions between real and generated datasets on PBMC 3k.

5 Conclusion

In this paper, we presented scMeanFlow, a two-stage framework for count-aware, efficient, and condition-aware scRNA-seq synthesis. By combining a Negative Binomial Autoencoder with a latent MeanFlow generator, scMeanFlow models the discrete and over-dispersed characteristics of gene expression while enabling one-step generation without numerical ODE solving or trajectory distillation. The NB-AE provides a count-aware latent representation, while MeanFlow models the interval-averaged transport field for efficient synthesis. FiLM with stochastic condition dropout further allows a shared generator to operate under different condition configurations, supporting unconditional, single-condition, and multi-condition generation without retraining. Experiments on four real scRNA-seq datasets demonstrate strong distributional fidelity and substantial gains in inference speed over flow and diffusion based baselines, while gene expression analyses show that generated cells preserve marker expression patterns and detection fractions. Ablation studies further reveal complementary benefits: FiLM improves the preservation of cellular structure under different conditions, whereas the NB-AE better retains count variation and sparsity. Moreover, augmenting the training data with scMeanFlow generated cells improves rare-cell annotation, with larger gains observed for low-frequency populations. Despite these results, systematic evaluation of generalization to condition combinations absent from training and broader biological discovery remains an important direction. Extending scMeanFlow to multimodal and spatial single-cell data, as well as perturbation response generation, may further broaden its applications in biological data synthesis and analysis.

AI USE STATEMENT

In this work, we sparingly used generative AI tools to assist with manuscript drafting, polishing, and summarizing experimental results. Generative AI was not used for deriving mathematical conclusions, designing research methodologies, or implementing algorithms. All AI-assisted content has been manually reviewed and verified, and we take full responsibility for the final content of this work.

ETHICS STATEMENT

This work uses publicly available single-cell RNA-seq datasets and does not involve human subjects, personal data, or potentially harmful applications. We adhere to the ICLR Code of Ethics and see no additional ethical concerns raised by this submission.

REPRODUCIBILITY STATEMENT

All datasets are publicly available, and the complete preprocessing procedures are described in Appendix A. Implementation details, including the NB-AE and MeanFlow training configurations, are provided in Appendix B, and the experimental settings (metrics, random seeds, evaluation protocol) are specified in Section 4.1. The anonymized source code will be released as supplementary material to ensure full reproducibility of all reported results.

REFERENCES

10x Genomics. 3k pbmcs from a healthy donor. https://www.10xgenomics.com/datasets/3-k-pbm-cs-from-a-healthy-donor-1-standard-1-0-0, 2016. Accessed: 2026-07-28.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 75460–75482. Curran Associates, Inc., 2025. doi: 10.52202/085713-2534. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/6d13e085b79d454da5910e4ca82a3d9d-Paper-Conference.pdf.

Leon Hetzel, Simon Boehm, Niki Kilbertus, Stephan Günnemann, Fabian Theis, et al. Predicting cellular responses to novel drug perturbations at a single-cell resolution. *Advances in Neural Information Processing Systems*, 35:26711–26722, 2022.

A Ali Heydari, Oscar A Davalos, Lihong Zhao, Katrina K Hoyer, and Suzanne S Sindi. Activa: realistic single-cell rna-seq generation with automatic cell-type identification using introspective variational autoencoders. *Bioinformatics*, 38(8):2194–2201, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Byungjin Hwang, Ji Hyun Lee, and Duhee Bang. Single-cell rna sequencing technologies and bioinformatics pipelines. *Experimental & molecular medicine*, 50(8):1–14, 2018.

Gioele La Manno, Ruslan Soldatov, Amit Zeisel, Emelie Braun, Hannah Hochgerner, Viktor Petukhov, Katja Lidschreiber, Maria E Kastriti, Peter Lönnerberg, Alessandro Furlan, et al. Rna velocity of single cells. *Nature*, 560(7719):494–498, 2018.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The eleventh international conference on learning representations*, 2022.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International conference on learning representations (ICLR)*, 2023.

Romain Lopez, Jeffrey Regier, Michael B Cole, Michael I Jordan, and Nir Yosef. Deep generative modeling for single-cell transcriptomics. *Nature methods*, 15(12):1053–1058, 2018.

Malte D Luecken and Fabian J Theis. Current best practices in single-cell rna-seq analysis: a tutorial. *Molecular systems biology*, 15(6):MSB188746, 2019.

Malte D Luecken, Daniel Bernard Burkhardt, Robrecht Cannoodt, Christopher Lance, Aditi Agrawal, Hananeh Aliee, Ann T Chen, Louise Deconinck, Angela M Detweiler, Alejandro A Granados, et al. A sandbox for prediction and integration of dna, rna, and proteins in single cells. In *Thirty-fifth conference on neural information processing systems datasets and benchmarks track (Round 2)*, 2021.

Erpai Luo, Minsheng Hao, Lei Wei, and Xuegong Zhang. scdiffusion: conditional generation of high-quality single-cell data using diffusion model. *Bioinformatics*, 40(9):btae518, 2024.

Matteo Marouf, Pierre Machart, Vikas Bansal, Christoph Kilian, Daniel S Magruder, Christian F Krebs, and Stefan Bonn. Realistic in silico generation and augmentation of single-cell rna-seq data using generative adversarial networks. *Nature communications*, 11(1):166, 2020.

Giovanni Palla, Sudarshan Babu, Payam Dibaeinia, James D Pearce, Donghui Li, Aly A Khan, Theofanis Karaletsos, and Jakub M. Tomczak. Scalable single-cell gene expression generation with latent diffusion models. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=IRwg87oeJo.

Alessandro Palma, Till Richter, Hanyi Zhang, Manuel Lubetzki, Alexander Tong, Andrea Dittadi, and Fabian J Theis. Multi-modal and multi-attribute generation of single cells with CFGen. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=3MnMGLctKb.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Inés Rivero-Garcia, Miguel Torres, and Fátima Sánchez-Cabo. Deep generative models in single-cell omics. *Computers in Biology and Medicine*, 176:108561, 2024.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022.

Nicholas Schaum, Jim Karkanias, Norma F Neff, Andrew P May, Stephen R Quake, Tony Wyss-Coray, Spyros Darmanis, Joshua Batson, Olga Botvinnik, Michelle B Chen, et al. Single-cell transcriptomics of 20 mouse organs creates a tabula muris: The tabula muris consortium. *Nature*, 562(7727):367, 2018.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. 2023.

Ping Xu, Zhiyuan Ning, Pengjiang Li, Wenhao Liu, Pengyang Wang, Jiaxu Cui, Yuanchun Zhou, and Pengfei Wang. scsiameseclu: A siamese clustering framework for interpreting single-cell rna sequencing data. *arXiv preprint arXiv:2505.12626*, 2025.

Ping Xu, Pengjiang Li, Tian Du, Zaitian Wang, Jiawei Gu, Ziyue Qiao, Pengfei Wang, and Yuanchun Zhou. scllm-dsc: Llm-knowledge enhanced cross-modal deep structural clustering for single-cell rna sequencing. *arXiv preprint arXiv:2606.13007*, 2026a.

Ping Xu, Zaitian Wang, Zhirui Wang, Pengjiang Li, Jiajia Wang, Ran Zhang, Pengfei Wang, and Yuanchun Zhou. scclubench: Comprehensive benchmarking of clustering algorithms for single-cell rna sequencing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 1364–1372, 2026b.

Luke Zappia, Belinda Phipson, and Alicia Oshlack. Splatter: simulation of single-cell rna sequencing data. *Genome biology*, 18(1):174, 2017.

Tianjiao Zhang, Zhongqian Zhao, Jixiang Ren, Ziheng Zhang, Hongfei Zhang, and Guohua Wang. cfdiffusion: diffusion-based efficient generation of high quality scrna-seq data with classifier-free guidance. *Briefings in Bioinformatics*, 26(1):bbaf071, 2025.

Grace XY Zheng, Jessica M Terry, Phillip Belgrader, Paul Ryvkin, Zachary W Bent, Ryan Wilson, Solongo B Ziraldo, Tobias D Wheeler, Geoff P McDermott, Junjie Zhu, et al. Massively parallel digital transcriptional profiling of single cells. *Nature communications*, 8(1):14049, 2017.

A Dataset Details

We evaluate scMeanFlow on four scRNA-seq benchmarks and an additional dataset for multi-condition generation. We use the full gene space without highly variable gene filtering.

Table 3: Statistics of datasets used in our experiments.

Dataset	Cells	Genes	Cell types
PBMC 68k	68,579	17,789	11
Tabula Muris	57,007	18,996	12
PBMC 3k	2,638	8,573	8
Dentate Gyrus	18,213	17,002	14
NeurIPS 2021	90,261	14,087	45

B Implementation Details

Models are implemented in PyTorch and trained on an NVIDIA V100 GPU. The Stage-1 NB-AE undergoes 300 epochs of training; its weights are then frozen while the Stage-2 MeanFlow generator is trained for 1,500 epochs.

Input representation. Let $\mathbf{X} \in \mathbb{N}_0^{N \times G}$ denote the raw gene-count matrix. The encoder input is the log-transformed count vector $\log(1+\mathbf{x})$ without normalization to 10^4 or division by the library size; the per-cell library size $s = \sum_g x_g$ is used by the decoder, and raw counts remain the reconstruction targets.

Autoencoder. The encoder is an MLP with hidden dimensions 512 and 256, with linear transformation, batch normalization, and ELU in each hidden layer: $G \to 512 \to 256 \to d_z$. We use $d_z = 50$ for PBMC 3k and $d_z = 100$ otherwise. The decoder is symmetric and uses a negative-binomial (NB) observation model. Decoder logits form $\boldsymbol{\mu} = \mathrm{softmax}(f_{\mathrm{dec}}(\mathbf{z}))s$, and counts are sampled as $\widehat{\mathbf{X}} \sim \mathrm{NB}(\boldsymbol{\mu}, \boldsymbol{\theta})$.

MeanFlow. The frozen-autoencoder latent model is a dimension-preserving ResNet with projection $d_z \to 64$, three FiLM-conditioned residual blocks, and projection $64 \to d_z$. FiLM is $\mathrm{FiLM}(\mathbf{h}; \boldsymbol{\gamma}, \boldsymbol{\beta}) = \mathbf{h} \odot (1+\boldsymbol{\gamma}) + \boldsymbol{\beta}$. The current time t and integration interval $h = t - r$ use separate sinusoidal embeddings (maximum period 10^4) and SiLU MLPs of size $100 \to 100 \to 100$. Every categorical covariate has a learned 100-dimensional embedding. During training, random condition dropout keeps each covariate embedding with probability $p_{\mathrm{cond}} = 0.8$ and zeroes it otherwise. The active embeddings are summed with the time and interval embeddings as $\mathbf{e}_{\mathrm{cond}} = \mathbf{e}_t(t) + \mathbf{e}_h(t-r) + \sum_{c \in \mathcal{C}_{\mathrm{active}}} \mathbf{e}_c$ and injected into every residual block through FiLM. Time pairs (t, r) are drawn from a logit-normal distribution ($P_{\mathrm{mean}} = -0.4$, $P_{\mathrm{std}} = 1.0$) and ordered as $t \geq r$; for 75% of the pairs we set $r = t$ (zero-length interval), yielding the integration interval $h = t - r$ in the interval-averaged velocity objective.

Random condition dropout. Consider two conditions, cell type and site. At each training step the two embeddings are kept independently with probability $p_{\mathrm{cond}} = 0.8$ and dropped otherwise. The four outcomes occur with probabilities 0.64 (both kept), 0.16 (cell type only), 0.16 (site only), and 0.04 (neither kept). The kept embeddings are summed with the time and interval embeddings and injected through FiLM. Training thus covers all four subsets of the two conditions. At inference, the subset is chosen explicitly, so one model generates cells under no condition, one condition, or both conditions.

Optimization. The autoencoder uses AdamW for 300 epochs (learning rate 10^{-3}, weight decay 10^{-5}, batch size 256); MeanFlow uses AdamW for 1,500 epochs (learning rate 10^{-4}, weight decay 10^{-6}, batch size 64). Dropout is disabled and gradients are clipped at norm 1.0.

Table 4: Default model and training configuration.

Configuration	Autoencoder	MeanFlow
Hidden width	$[512, 256]$	32
Latent size	$50/100$	d_z
Residual blocks	–	3
Embedding size	–	20
Learning rate	10^{-3}	10^{-4}
Weight decay	10^{-5}	10^{-6}
Batch size	256	64
Training epochs	300	1,500

C Training and Evaluation Protocol

Each dataset is split into 80% training and 20% held-out cells. Generated cells match the held-out total and every cell-type count. Metrics are averaged over cell types: $\bar{q} = C^{-1} \sum_{c=1}^{C} q_c$. For evaluation only, real counts are normalized to 10^4, transformed by $\log(1 + x)$, and jointly projected with generated cells into 50-dimensional PCA space.

MMD. Using $X = \{x_i\}_{i=1}^m$, $Y = \{y_j\}_{j=1}^n$ and $k(u, v) = \sum_{\sigma \in \{1,2,4,8,16\}} \exp(-\|u - v\|_2^2/(2\sigma^2))$, we report

$$\widehat{\mathrm{MMD}}(X, Y) = \Bigg[\max \Bigg(0, \frac{1}{m^2} \sum_{i,i'} k(x_i, x_{i'}) + \frac{1}{n^2} \sum_{j,j'} k(y_j, y_{j'}) - \frac{2}{mn} \sum_{i,j} k(x_i, y_j) \Bigg) \Bigg]^{1/2}. \tag{15}$$

2-Wasserstein distance. With uniform weights $a_i = 1/m, b_j = 1/n$ and transport polytope $\Pi(a, b)$, the reported WD is

$$W_2(X, Y) = \left[\min_{T \in \Pi(a,b)} \sum_{i,j} T_{ij} \|x_i - y_j\|_2^2 \right]^{1/2}. \tag{16}$$

kNN classification. A 10-NN classifier with Euclidean distance is trained on 75% of the real PCA representations in the shared 50-dimensional PCA space; the held-out 25% of real cells are not used. For each cell type, the F1 score is computed on the generated cells of that cell type by comparing their annotated labels with the kNN predictions, and KNNc is the unweighted average over cell types:

$$\mathrm{KNNc} = \frac{1}{C} \sum_{c=1}^{C} \frac{2 P_c R_c}{P_c + R_c}. \tag{17}$$

D Additional Experimental Results

Table 5: Marker gene quantitative comparison on Dentate Gyrus.

Method	Scope	Pearson $r\uparrow$	RMSE$\downarrow$	MAE$\downarrow$
scMeanFlow	Expression	0.9883	0.1239	**0.0682**
	Fraction	**0.9866**	**5.0320**	**3.0758**
scLDM	Expression	0.8768	0.3881	0.2530
	Fraction	0.8665	15.2313	10.7004
CFGen	Expression	0.9897	0.1224	0.0835
	Fraction	0.9796	6.2277	4.3873
scDiffusion	Expression	**0.9951**	**0.0909**	0.0723
	Fraction	0.8596	50.2474	47.4173

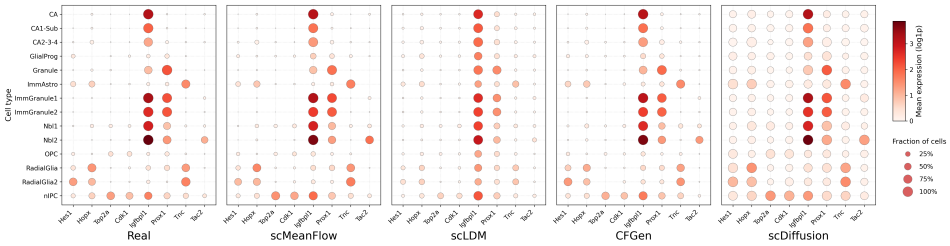

Figure 11: Marker gene dot plots on Dentate Gyrus. Dot color: average expression; dot size: detection fraction.

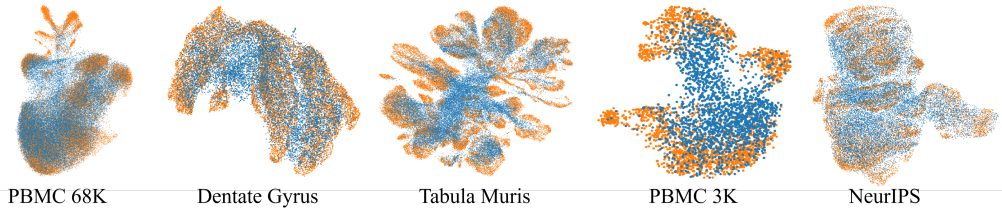

Figure 12: UMAP visualizations of unconditional generation on five scRNA-seq benchmark datasets, from left to right: PBMC 68k, Dentate Gyrus, Tabula Muris, PBMC 3k, and NeurIPS 2021 bone marrow.