ABSORB: ABSORBING-STATE DISCRETE DIFFUSION FOR SINGLE-CELL GENE EXPRESSION IMPUTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Single-cell RNA sequencing produces matrices that are highly sparse, yet a large fraction of zero entries are missing: either they are silent in the gene expression or are lost during transcript capture. This ambiguity makes imputation a selective missing-data problem in contrast to a generic denoising problem. We introduce ABSORB, an absorbing-state discrete diffusion process that represent gene expressions using a dedicated absorbing [MASK] state. Our proposed framework tokenizes the expression into per-gene equal-mass quantile bins, and learns to recover masked observed entries from the remaining expression profile. During inference, measured entries remain fixed while candidate dropouts are replaced with a [MASK] and reconstructed by a role-conditioned Transformer. The model predicts categorical distributions for each gene rather than a conditional mean, ensuring non-negativity is held by construction. To distinguish genuine recovery of missing expression from the sparse input, we evaluate only against known truth and external labels. Across a four-level simulated dataset spanning 50–72% dropout, ABSORB achieves Spearman correlations of 0.90–0.94 and log-scale RMSE of 0.55–0.61, outperforming nine baselines. Additionally, we evaluate cell-type clustering as a label-assisted downstream analysis, while separating it from the independent evidence provided by held-out gene expression recovery.

1 INTRODUCTION

Single-cell RNA sequencing (scRNA-seq) measures transcript abundance at cell-level precision, which enables the study of heterogeneous cell populations, developmental programmes, and disease-associated states. However, the observed count matrix is only a stochastic measurement of an underlying expression (Sarkar & Stephens, 2021). This distinction is considered significant at zero entries: a zero may reflect genuinely silent transcripts, low expression that was not sampled, or technical loss during capture and sequencing. Moreover, droplet-based UMI(Unique Molecular Identifier) data can often be explained without an additional zero-inflation component beyond ordinary count sampling (Svensson, 2020). Therefore, imputation has to be selective: it should recover plausibly missing expression without indiscriminately replacing biological zeros.

Existing methods address this problem through smoothing, probabilistic count models, low-rank reconstruction, and neural generative models. Although these approaches have advanced scRNA-seq analysis, their estimates reflect different assumptions about neighbourhood structure, count distributions, or latent geometry. Diffusion models treat imputation as an iterative conditional generation and have recently been applied to time-series, spatiotemporal, and spatial-transcriptomic completion (Islam et al., 2025; Liu et al., 2025; Chen et al., 2025). For scRNA-seq, scIDPMs reconstructs candidate dropout entries via a continuous DDPM process that gradually maps real-valued expression towards Gaussian noise (Zhang & Liu, 2025; Ho et al., 2020). This is a valid generative mechanism, but it does not represent the information state of an unobserved entry: numerical corruption alters a value, whereas missing values remove the knowledge of it.

We, instead, adopt the principle that an unknown expression entry should begin from an explicit unknown state. We introduce ABSORB, an absorbing-state discrete diffusion framework for selective, entry-wise scRNA-seq imputation. Expression counts are encoded using a per-gene quantile vocabulary with a reserved zero token, while a dedicated `[MASK]` represents erased information. During self-supervised training, observed entries are randomly absorbed to create prediction targets;

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.

candidate dropout entries remain masked and are excluded from the loss since their true values are unavailable. At inference, measured entries remain fixed, and only candidate dropouts are initialised as `[MASK]` and reconstructed.

A role-conditioned Transformer predicts a categorical distribution over expression tokens for each masked gene. This formulation has three practical consequences. First, the model distinguishes immutable observed context from evolving targets directly in its input representation. Second, all intermediate and final states remain inside a bounded, non-negative expression vocabulary. Third, the absorbing structure supports confidence-first decoding, committing the most certain predictions first so they can become context for the rest; a complete imputation therefore takes a small number of parallel passes rather than a full generational path.

Evaluation is equally important because similarity to the sparse input need not indicate recovery of missing expression. We therefore separate numerical recovery evaluation from downstream structural analysis. Numerical recovery is measured only at positions where the withheld expression values are known: both on simulated data and masked observed entries of real data. We also examine whether the imputed matrices remain compatible with the known cell type structure. However, since the available cell-type partitions are used in construction of the candidate-dropout mask as well, the clustering results are label-assisted and are not treated as independent evidence of biological structure recovery. Across the simulated data, ABSORB achieves Spearman correlations of 0.90–0.94 and the lowest log-scale RMSE (0.55–0.61) among the evaluated methods. Across three real datasets, it improves cell-type clustering over the unimputed matrix while producing no invalid expression values.

Our contributions are:

- We introduce an absorbing-state self-supervised corruption framework for selective scRNA-seq imputation, in which observed expression provides immutable context and unknown candidate positions are expressed by a dedicated `[MASK]` state.
- We introduce the combination of per-gene equal-mass tokenization with a role-conditioned Transformer to model a categorical distribution over bounded expression states.
- We develop confidence-first parallel decoding, that progressively converts masked query positions into additional context over a few number of parallel passes.
- We evaluate the recovery of the complete framework using known-truth simulated datasets and artificially masked observed entries on real datasets.

2 Related Work

2.1 Single-Cell Expression Recovery

The large number of zeros in scRNA-seq matrices has motivated a broad range of recovery methods, but as noted above, not every zero is missing data (Sarkar & Stephens, 2021; Svensson, 2020), so selective imputation is preferable to treating every zero as a missing value.

Early methods recover expression by sharing information across cells or genes. MAGIC propagates expression over a cell-similarity manifold via graph diffusion (smoothing, not a generative process); because this smoothing is linear, it can output negative values, which are not a valid property of gene expression counts (van Dijk et al., 2018). scImpute estimates entry-wise dropout probabilities and modifies only likely dropout values (Li & Li, 2018), whereas SAVER uses an empirical-Bayes Poisson–Gamma model to infer posterior expression distributions (Huang et al., 2018). ALRA applies low-rank reconstruction with adaptive thresholding to preserve a subset of biological zeros (Linderman et al., 2022). These approaches introduce useful inductive biases, but may depend strongly on neighbourhood, mixture, regression, or low-rank assumptions.

Neural methods learn nonlinear expression structure. DCA uses a count-aware autoencoder (Eraslan et al., 2019), scVI combines deep latent variables with a negative-binomial observation model (Lopez et al., 2018), and scIGANs uses generated cells to guide adversarial imputation (Xu et al., 2020). Such methods can capture complex dependencies, but they generally reconstruct expression in one step and do not define a corruption process that explicitly represents the loss of information

at a candidate dropout entry. Recent benchmarks identify long-tailed expression and evaluation design as central challenges in single-cell modelling, while showing that numerical and downstream performance should be assessed separately (Qi et al., 2025; Iwashita et al., 2026).

2.2 Diffusion Models for Transcriptomic Data

DDPMs learn to reverse a gradual Gaussian corruption process (Ho et al., 2020). scIDPMs adapts this construction to selective scRNA-seq imputation by conditioning an attention-based denoiser on observed expression (Zhang & Liu, 2025). It is the closest continuous-diffusion predecessor to our work. However, scIDPMs operates in a continuous normalized space and initializes unknown entries from Gaussian noise. ABSORB instead begins from an explicit unknown state and predicts expression categories, making the reverse process a sequence of conditional unmasking decisions.

Beyond scRNA-seq imputation, diffusion has been used for complete-cell generation and spatial gene-expression completion (Luo et al., 2024; Li et al., 2024; Chen et al., 2025). These methods address whole-profile generation or spatial completion rather than selective entry-wise imputation in an observed scRNA-seq matrix.

2.3 Discrete and Count-Valued Diffusion

Discrete diffusion replaces additive Gaussian noise with a Markov chain over a discrete state space: D3PM introduced absorbing transition kernels that progressively mask tokens (Austin et al., 2021), and Blackout Diffusion formulated discrete-state death processes generating samples from an empty state (Santos et al., 2023); together, these works provide the foundations for erasure-based generative modelling.

Recent single-cell models adopt discrete diffusion for broader modelling goals. Discrete Cell Models operate directly on gene-expression states for unconditional generation and perturbation-conditioned prediction (Bhattacharya et al., 2026). ScDiVa uses masked discrete diffusion for large-scale single-cell representation learning and generation, explicitly relating masking to dropout-like corruption (Wang et al., 2026). Neither method is formulated as selective entry-wise imputation conditioned on an observed expression profile.

CountsDiff extends Blackout Diffusion to the natural numbers and evaluates the framework on general count generation and imputation, including scRNA-seq (Soatto et al., 2026). Its birth/death-style formulation directly preserves natural-number structure and supports non-monotone reverse dynamics. ABSORB addresses a complementary, task-specific objective. Rather than diffusing the entire count profile through a general count process, it maps only candidate missing entries to a dedicated semantic `[MASK]` state, keeps measured entries immutable, and learns conditional recovery in a bounded expression-token space. The contribution of ABSORB is therefore not discrete diffusion alone, but the coupling of *selective missingness detection*, *absorbing erasure*, and *context-clamped reverse sampling* for scRNA-seq imputation.

3 Methodology

ABSORB imputes candidate dropout entries via absorbing-state discrete diffusion: technical dropout is treated as erasure of a discrete measurement, not additive Gaussian noise. A reserved `[MASK]` token combines self-supervised training masks with inference-time dropout. Expression is quantized into a bounded per-gene vocabulary; observed context stays fixed while a role-conditioned Transformer predicts categorical distributions over masked targets.

3.1 Problem Formulation

Let $C \in \mathbb{N}^{N \times G}$ be a raw count matrix for N cells and G genes. A zero $C_{ig} = 0$ is ambiguous: the gene may be silent, or expressed but uncaptured. We encode this with an estimated observation mask $M \in \{0, 1\}^{N \times G}$, where $M_{ig} = 1$ marks an entry retained as observed context and $M_{ig} = 0$ marks a candidate dropout to be imputed. Imputation models plausible expression values at candidate dropout positions at $\{(i, g) : M_{ig} = 0\}$ while leaving observed entries unchanged. Therefore, imputation is considered selective, not a reconstruction of the whole matrix.

We consider one cell profile and represent it as a sequence of G ordinal tokens $\boldsymbol{x} = (x^1, \ldots, x^G) \in \mathcal{V}^G$ (cell index dropped throughout). Writing $\mathcal{O} = \{g : M_g = 1\}$ for the evidence set, imputation is conditional generation: learn the conditional distribution of the unknown tokens at $g \notin \mathcal{O}$ from $p_\theta(\boldsymbol{x}^{\notin \mathcal{O}} \mid \boldsymbol{x}^{\mathcal{O}})$.

3.2 Candidate Dropout Mask Estimation

Since a zero's meaning depends on cellular context, we resolve it against cells of the same type. Cells are partitioned into clusters using externally provided labels when available and otherwise Leiden community detection on a k-nearest-neighbour graph (Traag et al., 2019). Within a cluster, if a gene is non-zero in at least one cell, its zeros in that cluster are flagged as candidate dropouts ($M_{ig} = 0$); if it is zero throughout the cluster, those zeros are kept as biological zeros ($M_{ig} = 1$). This encodes the standard assumption that a gene expressed by a cell type but absent in a few of its members has most likely dropped out, whereas a gene absent across the whole type is genuinely silent. Only entries with $M_{ig} = 0$ are modified.

Clustering is used for the sole purpose of estimating which zeroes can be imputed through the observation mask M: it does not provide the missing expression values. Across real data, available reference cell type structures are used for this, which makes candidate-dropout identification label-assisted. These labels are not provided to the Transformer as prediction targets or model-conditioning variables in the proposed framework. Consequently, numerical held-out recovery remains evaluated against values unavailable to the model, while downstream clustering against the same cell type structures is not an independent validation as interpreted in Section 4.3.

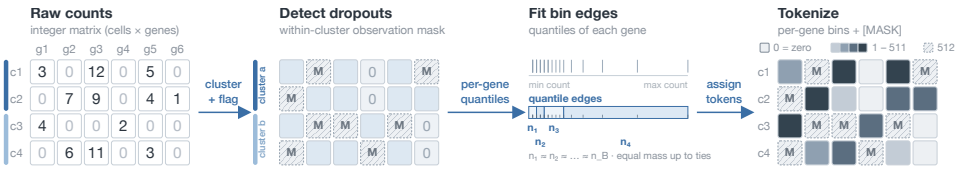


Figure 1: Tokenization. Zeros are resolved *within* a cell cluster into candidate dropouts versus biological zeros (mask M); observed counts are then binned per gene into $B-1$ equal-mass quantile bins , and candidate dropouts carry the shared `[MASK]` symbol, the same absorbing state used by the forward process. ***Right:*** equal-mass binning uses the full vocabulary where uniform binning on heavy-tailed counts would not.

3.3 Per-Gene Quantile Tokenization

Discretization must not waste vocabulary. UMI counts are heavy-tailed and gene dynamic ranges differ by orders of magnitude, so dividing by a per-gene maximum and binning uniformly, which is the natural first choice, collapses almost all mass into the lowest few bins and leaves the rest of the vocabulary unused. We instead fit bin edges *per gene* at equally spaced quantiles of that gene's non-zero counts, and give the exact zero its own token, since a raw zero is the quantity of interest rather than merely a small value. For gene g, let $e_{g,1} \le \cdots \le e_{g,B-2}$ be the empirical quantiles of its non-zero counts at levels $j/(B-1)$, estimated on the *training* cells only so that held-out entries never leak into the binning. A count c is encoded as

$$\tau_g(c) = \begin{cases} 0, & c = 0, \\ 1 + \min\big(|\{j : e_{g,j} \le c\}|,\, B-2\big), & c > 0, \end{cases} \tag{1}$$

giving B value tokens that preserve expression order and encode rank within each gene rather than raw count arithmetic. Adding the absorbing symbol yields the vocabulary

$$\mathcal{V} = \{0, 1, \ldots, B-1\} \cup \{\texttt{[MASK]}\}, \qquad |\mathcal{V}| = B + 1. \tag{2}$$

Each value token b is paired with a representative count $r_{g,b}$, the median of the training counts assigned to it ($r_{g,0} = 0$), which is how tokens are mapped back to the count scale. Equal-mass bins put resolution where the data lies; we verify in the supplementary material, §2.2, that the resulting quantization ceiling is not the binding constraint on accuracy.

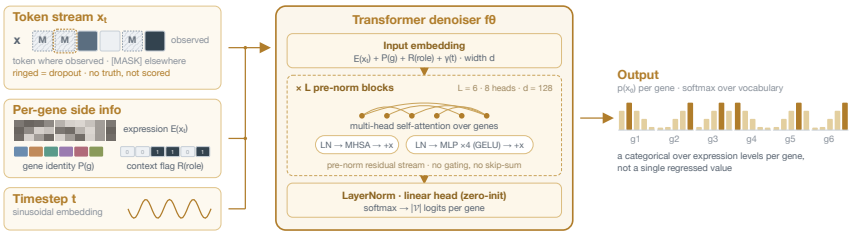


Figure 2: The denoiser f_θ. Each gene is embedded as its value token plus gene identity, role (context versus target), and timestep (Eq. 5); pre-norm Transformer blocks mix genes via self-attention so observed genes inform masked ones, and the head emits a full categorical distribution per gene (Eq. 6). Cross-entropy is taken only at masked-evidence positions. Ringed tokens mark candidate dropouts, which carry no ground truth and are excluded from the loss.

3.4 Selective Absorbing Corruption

Corruption acts independently per gene through an absorbing Markov chain Austin et al. (2021): a token either persists or jumps irrevocably to `[MASK]`. With per-step absorption probabilities $\{\beta_t\}_{t=1}^T$ and cumulative retention $\bar{\alpha}_t = \prod_{s\le t}(1-\beta_s)$, the t-step marginal of the standard chain is $q(x_t \mid x_0) = \bar{\alpha}_t\,\delta_{x_t,x_0} + (1-\bar{\alpha}_t)\,\delta_{x_t,\texttt{[MASK]}}$: each token independently survives with probability $\bar{\alpha}_t$. Unlike the Gaussian case, $\bar{\alpha}_t$ here is a genuine survival probability, so sampling x_t costs one Bernoulli draw per gene. We use the cosine $\bar{\alpha}_t$ of Nichol & Dhariwal (2021), which spends more steps at intermediate mask rates than a linear schedule.

Imputation, however, is selective, so we corrupt *only* the evidence set and keep candidate dropouts permanently absorbed:

$$q(x_t^g \mid x_0^g) = \begin{cases} \bar{\alpha}_t\,\delta_{x_t^g,\,x_0^g} + (1-\bar{\alpha}_t)\,\delta_{x_t^g,\,\texttt{[MASK]}}, & g \in \mathcal{O}, \\ \delta_{x_t^g,\,\texttt{[MASK]}}, & g \notin \mathcal{O}. \end{cases} \tag{3}$$

Entries outside $\mathcal{O}$ carry no ground truth and remain permanently absorbed. Corruption partitions the evidence at every step into a conditioning set and a target set,

$$\mathcal{C}_t = \{g \in \mathcal{O} : x_t^g \neq \texttt{[MASK]}\}, \qquad \mathcal{T}_t = \mathcal{O} \setminus \mathcal{C}_t, \tag{4}$$

so the schedule sweeps the model across all context densities. Since an absorbed token can only revert to its clean value, the reverse posterior $q(x_{t-1} \mid x_t, x_0)$ is a two-point distribution over $\{x_0, \texttt{[MASK]}\}$ (proof in the supplementary material, §1.2); our sampler exploits this by committing predicted tokens directly.

3.5 Role-Conditioned Denoiser

Absorb learns the reverse process through an x_0-parameterization: a network f_θ predicts a categorical distribution over $\mathcal{V}$ at every gene. Two properties of the imputation setting shape it. First, observed genes must guide missing ones, which is exactly self-attention along the *gene* axis. Second, observed positions never change during sampling, so a separate copy of the observed profile carries no information beyond a flag saying which positions are fixed. We therefore encode a single token stream with an explicit role, embedding gene g as

$$h^g = E\big(x_t^g\big) + P(g) + R\big(\mathbf{1}[g \in \mathcal{C}_t]\big) + \gamma(t) \;\in\; \mathbb{R}^d, \tag{5}$$

where E embeds the value token, P is a learned gene-identity embedding (without which two genes at the same expression level would be indistinguishable), R is a two-entry role embedding separating immutable context from targets, and $\gamma(t)$ is a sinusoidal timestep embedding passed through an MLP and broadcast across genes. The backbone is a stack of L pre-norm transformer encoder blocks Vaswani et al. (2017) with multi-head self-attention over the G genes, followed by a layer norm and a linear head,

$$p_\theta\big(x_0^g \mid \boldsymbol{x}_t, \mathcal{C}_t, t\big) = \mathrm{softmax}\big(f_\theta(\boldsymbol{x}_t, \mathcal{C}_t, t)^g\big), \tag{6}$$

with $f_\theta(\cdot) \in \mathbb{R}^{G\times|\mathcal{V}|}$. The head is zero-initialised so training begins from a near-uniform distribution, and predicting logits over the vocabulary, rather than regressing a scalar, lets the model express multimodal uncertainty while making training a classification objective.

3.6 TRAINING

Training is self-supervised on reliably observed entries, whose values are known and can serve as targets. For each cell we draw $t \sim \mathcal{U}\{1, \dots, T\}$, corrupt by Eq. equation 3, read off the split of Eq. equation 4, and minimise the cross-entropy of the predicted distribution against the true tokens at masked-evidence positions only:

$$\mathcal{L}(\theta) = \mathbb{E}_{\boldsymbol{x}_0,\, t,\, \boldsymbol{x}_t}\left[\frac{1}{|\mathcal{T}_t|}\sum_{g\in\mathcal{T}_t} -\log p_\theta\big(x_0^g \mid \boldsymbol{x}_t, \mathcal{C}_t, t\big)\right]. \tag{7}$$

For the absorbing chain this x_0-parameterized objective coincides with the auxiliary reconstruction term of the D3PM variational bound Austin et al. (2021) (derived in the supplementary material, §1.1), and reads as masked language modelling over expression tokens Devlin et al. (2019). Restricting the sum to $\mathcal{T}_t$, evaluates the model only where it must genuinely predict: conditioning genes are given as input, and candidate dropouts have no ground truth. The `[MASK]` column of the head is never a target and is dropped at decoding time.

3.7 IMPUTATION BY CONFIDENCE DECODING

At inference all reliable measurements act as evidence, $\mathcal{C} = \mathcal{O}$, and the candidate dropouts $\mathcal{T} = \{g : M_g = 0\}$ are initialised to `[MASK]`, the maximally corrupted state $q(\boldsymbol{x}_T \mid \boldsymbol{x}_0)$. An ancestral chain would spend one network evaluation per timestep to unmask whichever genes the schedule happens to select; absorbing chains admit a far cheaper decoder that instead unmasks the genes the model is most confident about Chang et al. (2022). We use $S \ll T$ passes. At pass s, with progress $r = s/S$, we evaluate f_θ at timestep index $t_s = \lfloor (1-r)(T-1) \rceil$, take the softmax over value tokens only, and record per gene the most likely token $\hat{b}^g$ and its confidence $\kappa^g = \max_b p_\theta(b \mid \cdot)$. A cosine schedule prescribes how many targets may remain masked after the pass,

$$n_s = \big\lfloor \cos\!\big(\tfrac{\pi}{2} r\big)\, |\mathcal{T}| \big\rfloor, \tag{8}$$

and we commit $\hat{b}^g$ at the highest-confidence still-masked targets until that many remain, all of them at $s = S$. Committed tokens are held fixed and become context for the genes decoded after them, so each pass conditions on strictly more information than the last; evidence positions are outside $\mathcal{T}$ and are therefore never rewritten. Because $\mathcal{V}$ is a bounded expression vocabulary and the chain never leaves it, non-negativity and boundedness of the output hold by construction.

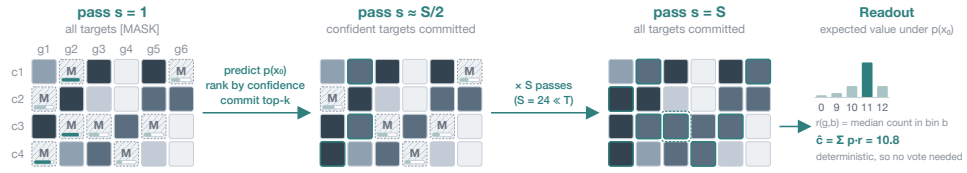


Figure 3: Confidence decoding. Observed genes stay fixed; each of S passes commits the most confident masked predictions as new context for the rest (Eq. 8). The final count-scale estimate is the categorical's expectation (Eq. 9).

Two properties distinguish this from count-valued diffusion conditioning. First, evidence enters as its *measured* token and stays there, since missingness is already encoded in the forward process (Eq. 3); RePaint-style samplers, by contrast, re-corrupt observations to the current noise level at each step (Lugmayr et al., 2022). Second, decoding is monotone: a committed token is never revisited, unlike birth/death reverse steps that adjust every site at every step (Soatto et al., 2026). Monotonicity makes the confidence ordering meaningful, since committing the most certain genes first is only useful if commitment is permanent.

Tokens are mapped back to the count scale by the expected value under the predictive distribution of the final pass $s = S$,

$$\hat{c}^g = \sum_{b=0}^{B-1} p_\theta\big(b \mid \cdot\big)\, r_{g,b}, \tag{9}$$

using the whole categorical rather than a hard mode. At that pass, targets committed earlier are already present in the input as their committed tokens, so for them Eq. equation 9 is read at a

position that holds its own committed value (supplementary material, §1.1.13). Imputed values are written only at $\mathcal{T}$; observed entries are returned unchanged. Both procedures are summarised as pseudocode in the supplementary material, §1.3.

3.8 Implementation

We use $B = 256$ quantile bins ($|\mathcal{V}| = 257$), a cosine schedule with $T = 100$, and a denoiser with $L = 6$ pre-norm blocks ($d = 128$, 8 heads, MLP ratio 4). Training uses AdamW (lr $= 5 \times 10^{-4}$) for 15,000 steps with cosine decay and EMA (decay 0.999) for inference; decoding uses $S = 24$ passes. Training takes 15–20 minutes and imputing a 500×1000 matrix 1–2 minutes on a single GPU; full hyperparameters are in the supplementary material, §3.1.

4 Results and Discussion

4.1 Simulated Data: Held-Out Recovery Against Known Truth

We first evaluate on data for which the true expression at every dropout position is known. Following the scIMC benchmark (Dai et al., 2022) used by scIDPMs, we simulate one Splatter configuration (Zappia et al., 2017) (500 cells, 1000 genes, four equal groups, `de.prob`=0.5) at four dropout levels, varying the `dropout.mid`. This gives a severity sweep from D1 (hardest, 71.9% zeros) to D4 (easiest, 50.0% zeros); full simulation parameters are in the supplementary material, §4.1. Each method is scored only at the held-out set $\mathcal{H} = \{(i, j) : \text{raw}_{ij} = 0 \wedge \text{truth}_{ij} > 0\}$, the positions the simulator zeroed out, against the value held before masking. Since that reference is never revealed to ABSORB and is never the observed input, a method that simply predicts zero everywhere earns no credit, unlike metrics comparing an imputed matrix back to its own sparse input; this makes held-out recovery the strongest evidence in this study.

We compare ABSORB against nine baselines: two continuous or discrete diffusion models (scIDPMs (Zhang & Liu, 2025), CountsDiff (Soatto et al., 2026)), a graph-diffusion method (MAGIC (van Dijk et al., 2018)), an empirical-Bayes estimator (SAVER (Huang et al., 2018)), a statistical mixture model (scImpute (Li & Li, 2018)), a low-rank completion method (ALRA (Linderman et al., 2022)), a generative-adversarial method (scIGANs (Xu et al., 2020)), and two neural-network regressors (DeepImpute (Arisdakessian et al., 2018), DCA (Eraslan et al., 2019)).

Recovery at held-out positions. Figure 4 reports six views of recovery at $\mathcal{H}$. Three are pointwise: Spearman ρ, which isolates whether a method recovers the correct relative ordering; Pearson r on $\log_{1p}$ values, the linear agreement on the scale downstream analysis uses; and RMSE on $\log_{1p}$ values, which catches a method that ranks entries correctly but is wrong in magnitude. Figures 4 and 5 report both correlations as magnitudes, $|\rho|$ and $|r|$. Three are distributional, asking whether the recovered *distribution* matches the truth rather than individual entries: energy distance (ED), maximum mean discrepancy (MMD, RBF kernel), and sliced Wasserstein distance (SWD).

	D1 (71.9% dropout)						D2 (64.4% dropout)						D3 (56.9% dropout)						D4 (50.0% dropout)					
	ρ ↑	r ↑	RMSE ↓	ED ↓	MMD ↓	SWD ↓	ρ ↑	r ↑	RMSE ↓	ED ↓	MMD ↓	SWD ↓	ρ ↑	r ↑	RMSE ↓	ED ↓	MMD ↓	SWD ↓	ρ ↑	r ↑	RMSE ↓	ED ↓	MMD ↓	SWD ↓
ABSORB	**0.902**	**0.920**	**0.609**	**362.1**	**0.144**	**29.66**	**0.928**	**0.923**	**0.595**	**267.9**	**0.110**	**26.38**	**0.931**	**0.928**	**0.580**	**220.4**	0.083	**23.51**	**0.942**	**0.936**	**0.549**	**153.1**	0.055	**19.98**
scIDPMs	0.882	0.893	0.994	3493.5	0.624	90.73	0.918	0.921	0.948	2572.2	0.495	84.03	0.919	0.916	0.947	2206.4	0.411	73.26	0.924	0.921	1.057	3330.5	0.463	105.6
countsdiff	0.890	0.877	1.151	3493.2	0.635	92.58	0.907	0.899	0.828	2291.6	0.476	72.79	0.916	0.903	0.695	1390.2	0.328	56.29	0.919	0.908	0.652	886.1	0.226	44.81
MAGIC	0.673	0.649	3.014	5120.3	0.764	119.1	0.808	0.772	3.008	4452.4	0.679	111.4	0.894	0.857	3.012	3643.2	0.568	101.8	0.923	0.895	3.032	2892.2	0.461	91.16
SAVER	-0.732	-0.713	3.750	5142.6	0.765	119.5	-0.661	-0.636	3.787	4476.9	0.681	111.9	-0.589	-0.555	3.800	3668.6	0.570	102.3	-0.471	-0.431	3.805	2916.2	0.463	91.76
scImpute	-0.489	-0.442	3.092	5088.6	0.760	119.0	-0.438	-0.406	2.935	4422.4	0.675	111.2	-0.315	-0.286	2.648	3604.6	0.564	100.9	-0.158	-0.120	2.340	2836.4	0.456	89.72
ALRA	0.876	0.874	0.793	549.9	0.179	38.29	0.860	0.846	0.885	445.2	0.128	34.07	0.884	0.876	0.785	327.1	0.110	30.46	0.891	0.881	0.771	251.5	0.082	26.99
scIGANs	0.016	0.0080	1.650	4595.6	0.715	112.5	0.013	0.016	1.665	4051.4	0.645	105.1	0.035	0.033	1.643	3351.3	0.545	95.99	-0.0020	-0.0050	1.927	2735.6	0.447	87.55
DeepImpute	0.883	0.886	0.643	713.8	0.158	42.38	0.914	0.919	0.637	437.4	0.118	37.42	0.921	0.920	0.624	338.1	**0.079**	29.29	0.936	0.930	0.579	180.5	**0.035**	23.95
DCA	0.863	0.854	1.525	4381.8	0.704	107.3	0.880	0.895	1.233	3448.1	0.587	94.13	0.907	0.919	0.949	2292.6	0.420	76.19	0.911	0.913	0.772	1488.8	0.285	60.80

Figure 4: **Held-out dropout recovery at $\mathcal{H}$** across the D1 (hardest, 71.9% dropout) to D4 (easiest, 50.0%) severity sweep, on the six pointwise and distributional metrics (abbreviations as in sec. Simulated Data: Held-Out Recovery Against Known Truth). Arrows give the direction of improvement, and the best value in each column is outlined in gold.

ABSORB takes 22 of the 24 (dataset, metric) cells, sweeping all six metrics on D1 and D2 and losing only MMD on D3 and D4, both to DeepImpute. Its accuracy improves monotonically as dropout eases on every pointwise axis (ρ $0.902 \rightarrow 0.942$, r $0.920 \rightarrow 0.936$, log-RMSE $0.609 \rightarrow 0.549$).

The distributional metrics separate the field more sharply than the pointwise ones, and they locate where ABSORB's advantage lies. Only ABSORB and DeepImpute are within an order of magnitude of each other; every other baseline is one to two orders behind on every dataset. Crucially, the margin is widest where dropout is worst: ABSORB's energy distance is $1.97\times$ lower than DeepImpute's at 71.9% dropout but only $1.18\times$ lower at 50.0%, and $9.6\times$ lower than countsdiff's at D1 against $5.8\times$ at D4, so the advantage concentrates where imputation is required. The clustering metric is substantially distinct than the direct recovery metrics: ABSORB, scIDPMs, and CountsDiff all reach the metric ceiling despite different held-out reconstruction errors. We therefore treat clustering here as a structural diagnostic rather than evidence of accurate value recovery.

4.2 Real Data: Held-Out Recovery via Masking

On real data the truth at a genuine dropout position is never observed, so we construct a second, independent held-out test: a fraction of the *already-observed*, non-zero entries in a real count matrix is hidden and scored as in Section 4.1, but against genuine biological dropout rather than a simulator's model of it. Two masking protocols are used per dataset: MCAR50 hides 50% of non-zero entries uniformly at random, and MNAR25 hides 25%, biased toward low-count entries, closer to how real dropout behaves. We report this on three real datasets: VISp (Allen Brain Atlas mouse cortex) (Tasic et al., 2018), PBMC68k (10x Genomics human blood) (Zheng et al., 2017), and Bone Marrow Mesenchyme (Han et al. mouse bone-marrow stroma) (Han et al., 2018); dataset sizes and preprocessing are given in the supplementary material, §4.2. Five of the six metrics of Section 4.1 are used: Spearman $|\rho|$, RMSE on $\log_{1p}$ values, energy distance, MMD, and sliced Wasserstein distance.

VISp — MCAR 50%

	|ρ| ↑	RMSE ↓	ED ↓	MMD ↓	SW ↓
ABSORB	0.6154	**2.3764**	1.40e+03	**0.0147**	**152.7058**
scIDPMs	0.3500	4.9548	*excl.*	0.0721	*excl.*
countsdiff	0.2779	4.0692	9.60e+03	0.0932	376.3684
MAGIC	0.6695	4.4834	1.03e+04	0.0996	385.7766
SAVER	0.5979	5.3876	1.04e+04	0.0999	386.2897
scImpute	0.6117	2.4003	1.43e+03	0.0162	169.8783
ALRA	0.5645	2.4331	6.83e+03	0.0530	391.2590
scIGANs	0.0855	2.7695	9.80e+03	0.0957	379.7630
DeepImpute	0.6136	2.3948	**1.21e+03**	0.0153	176.3238
DCA	**0.6908**	2.5437	3.26e+03	0.0351	237.1281

PBMC68k — MCAR 50%

	|ρ| ↑	RMSE ↓	ED ↓	MMD ↓	SW ↓
ABSORB	0.1969	**0.4778**	**0.1253**	**0.0026**	**0.0727**
scIDPMs	0.0514	0.7529	1.8616	0.0497	0.2725
countsdiff	0.0139	0.5671	0.7359	0.0211	0.1796
MAGIC	0.0597	0.7116	1.8440	0.0515	0.2731
SAVER	**0.2078**	0.8145	2.1193	0.0564	0.2869
scImpute	0.1123	0.8650	1.0513	0.0193	0.1870
ALRA	0.1243	0.8542	1.6343	0.0397	0.2480
scIGANs	0.0018	0.7502	1.7750	0.0475	0.2630
DeepImpute	0.1765	0.4966	0.3685	0.0096	0.1258
DCA	0.0479	0.7297	1.8188	0.0500	0.2700

Bone Marrow Mesenchyme — MCAR 50%

	|ρ| ↑	RMSE ↓	ED ↓	MMD ↓	SW ↓
ABSORB	0.3918	**0.7033**	**3.3658**	**0.0209**	0.5697
scIDPMs	0.3322	1.2402	84.2303	0.2758	4.0794
countsdiff	0.3854	2.0621	262.5920	0.4553	9.5819
MAGIC	0.4296	1.1098	39.7386	0.1266	3.0166
SAVER	0.3092	1.0703	8.1897	0.0470	0.9069
scImpute	0.3922	0.9240	3.8747	0.0246	0.7392
ALRA	0.3529	0.8063	5.0577	0.0239	0.7588
scIGANs	0.2638	1.1925	8.0048	0.0455	0.9086
DeepImpute	0.4190	0.8385	3.5273	0.0452	**0.4246**
DCA	**0.4437**	0.7106	3.7826	0.0239	0.6617

VISp — MNAR 25%

	|ρ| ↑	RMSE ↓	ED ↓	MMD ↓	SW ↓
ABSORB	0.6140	**2.2714**	**106.0348**	8.00e-04	**44.5574**
scIDPMs	0.4452	4.2456	*excl.*	0.0016	*excl.*
countsdiff	0.2494	3.5744	277.1786	0.0024	58.8932
MAGIC	0.6775	4.2220	452.9717	0.0040	71.2867
SAVER	0.5409	5.1197	461.4732	0.0041	71.9560
scImpute	0.5959	2.3613	356.3223	0.0033	85.0658
ALRA	0.5358	2.9338	173.6718	5.00e-04	46.2093
scIGANs	0.0517	2.5695	388.5547	0.0035	68.0193
DeepImpute	0.6043	2.4318	134.9720	**2.00e-04**	79.1692
DCA	**0.6789**	2.4375	140.4607	3.00e-04	56.5329

PBMC68k — MNAR 25%

	|ρ| ↑	RMSE ↓	ED ↓	MMD ↓	SW ↓
ABSORB	**0.3335**	**0.4120**	**0.0517**	**7.00e-04**	**0.0486**
scIDPMs	0.1209	0.5435	0.1399	0.0025	0.0767
countsdiff	0.0265	0.4612	0.0968	0.0016	0.0631
MAGIC	0.0946	0.4636	0.1146	0.0022	0.0672
SAVER	0.1158	0.5762	0.1634	0.0031	0.0787
scImpute	0.0732	0.7835	0.5088	0.0107	0.1461
ALRA	0.2147	0.5905	0.0782	0.0012	0.0573
scIGANs	0.1433	0.4318	0.1033	0.0020	0.0636
DeepImpute	0.3285	0.4278	0.0573	8.00e-04	0.0487
DCA	0.1516	0.4979	0.1193	0.0022	0.0685

Bone Marrow Mesenchyme — MNAR 25%

	|ρ| ↑	RMSE ↓	ED ↓	MMD ↓	SW ↓
ABSORB	0.4025	**0.6343**	0.5413	0.0032	0.2237
scIDPMs	0.3356	1.6150	53.2379	0.2072	3.5710
countsdiff	0.3298	0.9595	6.9807	0.0420	0.9023
MAGIC	0.4984	1.3908	37.4487	0.1480	3.5925
SAVER	0.4610	0.7878	0.2611	0.0013	0.1657
scImpute	**0.5104**	0.8258	1.2887	0.0070	0.4101
ALRA	0.4386	0.6879	0.1139	6.00e-04	0.1320
scIGANs	0.1776	0.6613	0.2456	0.0014	0.1600
DeepImpute	0.4657	0.6928	**0.0363**	**2.00e-04**	**0.0604**
DCA	0.5062	0.6593	0.0804	4.00e-04	0.0907

Figure 5: **Held-out recovery via masking on real data.** Each panel is one dataset under one masking protocol (MCAR50 top row, MNAR25 bottom row); rows are methods, columns the five metrics (abbreviations as in sec. Real Data: Held-Out Recovery via Masking), and cell colour encodes value within a column. scIDPMs numerically diverges on VISp under both protocols, so its ED and SW entries are marked *excl.*

The pattern is more mixed than on simulated data. On PBMC68k, ABSORB wins all five metrics under both protocols. On VISp, DCA leads Spearman ρ under both protocols, but ABSORB wins RMSE and distributional metrics. On Bone Marrow Mesenchyme, ABSORB wins three of five metrics under MCAR50 but only one under the harder MNAR25 protocol, where scImpute leads rank correlation and DeepImpute leads all distributional metrics by wide margins (e.g. energy distance $15\times$ lower). We report this weakness rather than omit it.

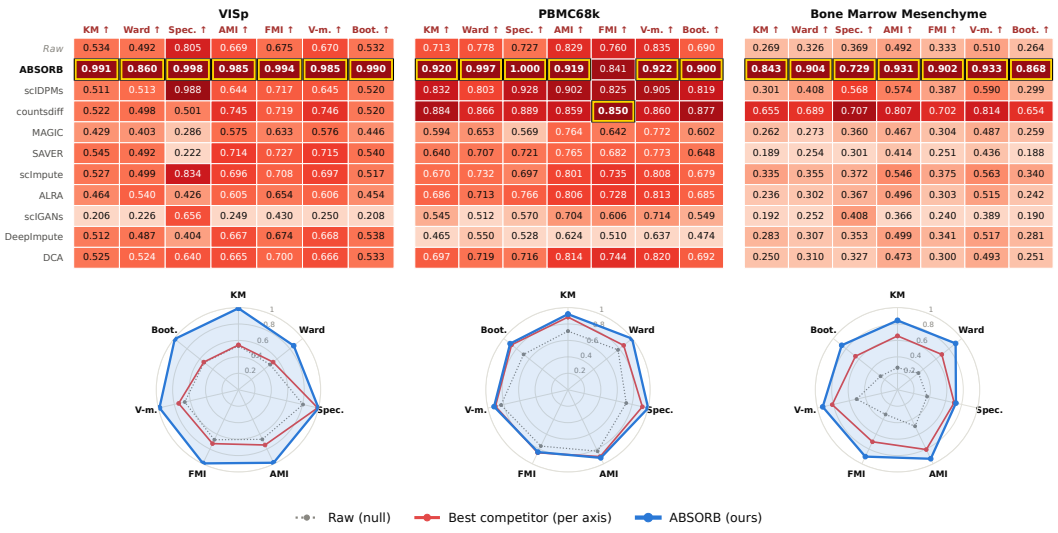


Figure 6: **Real-data clustering summary.** Seven clustering views per method (Raw on top, ABSORB second row; abbreviations as in sec. Real Data: Cell-Type Clustering Structure); darker cells are higher. Radar plots below each heatmap compare Raw (grey, dotted), the best competitor (red), and ABSORB (blue).

4.3 Real Data: Cell-Type Clustering Structure

Held-out recovery evaluates numerical value predictions at individual entries, while clustering provides a complementary view of how imputation alters population-level organization. Figure 6 summarises agreement between clusters derived from each imputed matrix and the available reference cell-type structures across three clustering algorithms (k-means, Ward, and spectral), three agreement scores on the k-means partition (AMI, FMI, and V-measure), and bootstrap stability.

As expected by the label-assisted evaluation, ABSORB shows strong agreement with the reference partitions across nearly all reported clustering metrics, including substantial improvements over the raw matrix. These results indicate that imputation remains compatible with the cell-type structure used during candidate-dropout identification.

These clustering results should not be interpreted as independent evidence that ABSORB recovers cell-type structure, since the same reference partitions influence candidate-dropout selection through M. Instead, they indicate that, given a label-assisted selective mask, the imputed matrix remains compatible with the corresponding cell-type structure.

5 Conclusion

We present ABSORB, an absorbing-state discrete diffusion framework for selective scRNA-seq imputation that imputes candidate dropout entries while preserving measured expression as immutable context. Across the controlled simulated data, ABSORB achieves strong known-truth recovery in both rank and log-scale error, while controlled real-data masking demonstrates competitive recovery across multiple missingness procedures. Label-assisted downstream clustering also shows high agreement with reference partitions, but these results are interpreted descriptively because those same partitions contribute to candidate-dropout mask construction and therefore do not constitute independent biological validation.

Our proposed framework presents three limitations. Real-data recovery is not uniform: under low-count-biased masking on Bone Marrow Mesenchyme, DeepImpute recovers the held-out distribution more accurately. The observation mask is estimated by a cluster partition and, real data uses available cell-type labels. Consequently, clustering against those structures is inherently label-assisted. Learning M independently or jointly with the denoiser is therefore an important next step.

AI USE STATEMENT

In this work, we used generative AI tools to evaluate the use of grammar, LaTeX code generation, and identification of relevant literature. We did not use generative AI tools for research purposes, including the development of theoretical models or conceptual frameworks, formulation of mathematical claims, writing of mathematical proofs, refinement of hypotheses, design of research methodology, implementation of methods. Other required-disclosure tasks, including synthetic dataset generation, translation, dataset cleaning or reformatting, and qualitative or thematic data analysis, were not applicable to this work. We take responsibility for the final content of this work, including text, claims or artefacts produced with the aid of generative AI.

ETHICS STATEMENT

We do not have any concerns.

REPRODUCIBILITY STATEMENT

We present the reproducibility details of our work in Section A.

REFERENCES

Cedric Arisdakessian, Olivier Poirion, Breck Yunits, Xun Zhu, and Lana X. Garmire. Deepimpute: an accurate, fast and scalable deep neural network method to impute single-cell rna-seq data. *bioRxiv*, 2018. doi: 10.1101/353607. URL https://www.biorxiv.org/content/early/2018/06/22/353607.

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems*, volume 34, pp. 17981–17993, 2021.

Sanjukta Bhattacharya, Christian Gensbigler, Shaamil Karim, and Jon Lees. Discrete diffusion for single-cell gene expression modeling. In *International Conference on Learning Representations*, pp. 2026–02. Cold Spring Harbor Laboratory, 2026.

Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T. Freeman. MaskGIT: Masked generative image transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11315–11325, 2022.

Tianyi Chen, Yunfei Zhang, Lianxin Xie, Wenjun Shen, Si Wu, and Hau-San Wong. SpotDiff: Spatial gene expression imputation diffusion with single-cell RNA sequencing data integration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 15848–15856, 2025. doi: 10.1609/aaai.v39i15.33740.

Chichi Dai, Yuxi Jiang, Cheng Yin, Ran Su, Xiangxiang Zeng, Quan Zou, Kenta Nakai, and Leyi Wei. scIMC: A platform for benchmarking comparison and visualization analysis of single-cell RNA-seq data imputation methods. *Nucleic Acids Research*, 50(9):4877–4899, 2022. doi: 10.1093/nar/gkac317.

Spyros Darmanis, Steven A. Sloan, Ye Zhang, Martin Enge, Christine Caneda, Lawrence M. Shuer, Melanie G. Hayden Gephart, Ben A. Barres, and Stephen R. Quake. A survey of human brain transcriptome diversity at the single cell level. *Proceedings of the National Academy of Sciences*, 112(23):7285–7290, 2015. doi: 10.1073/pnas.1507125112.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 4171–4186, 2019.

Gökcen Eraslan, Lukas M. Simon, Maria Mircea, Nikola S. Mueller, and Fabian J. Theis. Single-cell RNA-seq denoising using a deep count autoencoder. *Nature Communications*, 10:390, 2019. doi: 10.1038/s41467-018-07931-2.

Xiaoping Han, Renying Wang, Yincong Zhou, et al. Mapping the mouse cell atlas by microwell-seq. *Cell*, 172(5):1091–1107, 2018. doi: 10.1016/j.cell.2018.02.001.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851, 2020.

Hannah Hochgerner, Amit Zeisel, Peter Lönnerberg, and Sten Linnarsson. Conserved properties of dentate gyrus neurogenesis across postnatal development revealed by single-cell RNA sequencing. *Nature Neuroscience*, 21(2):290–299, 2018. doi: 10.1038/s41593-017-0056-2.

Mo Huang, Jingshu Wang, Eduardo Torre, Hannah Dueck, Sydney Shaffer, Roberto Bonasio, John I Murray, Arjun Raj, Mingyao Li, and Nancy R Zhang. Saver: gene expression recovery for single-cell rna sequencing. *Nature methods*, 15(7):539–542, 2018.

Mohammad Rafid Ul Islam, Prasad Tadepalli, and Alan Fern. Self-attention-based diffusion model for time-series imputation in partial blackout scenarios. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(17):17564–17572, 2025. doi: 10.1609/aaai.v39i17.33931.

Yuichiro Iwashita, Ahtisham Fazeel Abbasi, Muhammad Nabeel Asim, and Andreas Dengel. A large-scale comparative analysis of imputation methods for single-cell RNA sequencing data. *arXiv preprint arXiv:2603.24626*, 2026. URL `https://arxiv.org/abs/2603.24626`.

Kongming Li, Jiahao Li, Yuhao Tao, and Fei Wang. stDiff: A diffusion model for imputing spatial transcriptomics through single-cell transcriptomics. *Briefings in Bioinformatics*, 25(3):bbae171, 2024. doi: 10.1093/bib/bbae171.

Wei Vivian Li and Jingyi Jessica Li. An accurate and robust imputation method scimpute for single-cell rna-seq data. *Nature communications*, 9(1):997, 2018.

George C. Linderman, Jun Zhao, Manolis Roulis, et al. Zero-preserving imputation of single-cell RNA-seq data. *Nature Communications*, 13:192, 2022. doi: 10.1038/s41467-021-27729-z.

Zijin Liu, Xiang Zhao, and You Song. RDPI: A refine diffusion probability generation method for spatiotemporal data imputation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(12):12255–12263, 2025. doi: 10.1609/aaai.v39i12.33335.

Romain Lopez, Jeffrey Regier, Michael B. Cole, Michael I. Jordan, and Nir Yosef. Deep generative modeling for single-cell transcriptomics. *Nature Methods*, 15:1053–1058, 2018. doi: 10.1038/s41592-018-0229-2.

Andreas Lugmayr, Martin Danelljan, Andrés Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. RePaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11461–11471, 2022.

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.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8162–8171, 2021.

Jiaxin Qi, Yan Cui, Kailei Guo, Xiaomin Zhang, Jianqiang Huang, and Gaogang Xie. A simple and comprehensive benchmark for single-cell transcriptomics. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(1):676–684, 2025. doi: 10.1609/aaai.v39i1.32049.

Javier E. Santos, Zachary R. Fox, Nicholas Lubbers, and Yen Ting Lin. Blackout diffusion: Generative diffusion models in discrete-state spaces. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 9034–9059, 2023.

Abhishek Sarkar and Matthew Stephens. Separating measurement and expression models clarifies confusion in single-cell RNA sequencing analysis. *Nature Genetics*, 53:770–777, 2021. doi: 10.1038/s41588-021-00873-4.

Renzo G. Soatto, Anders Hoel, Greycen Ren, Shorna Alam, Stephen Bates, Nikolaos P. Daskalakis, Caroline Uhler, and Maria Skoularidou. CountsDiff: A diffusion model on the natural numbers for generation and imputation of count-based data. *arXiv preprint arXiv:2604.03779*, 2026. URL https://arxiv.org/abs/2604.03779.

Valentine Svensson. Droplet scRNA-seq is not zero-inflated. *Nature Biotechnology*, 38:147–150, 2020. doi: 10.1038/s41587-019-0379-5.

Bosiljka Tasic, Zizhen Yao, Lucas T. Graybuck, et al. Shared and distinct transcriptomic cell types across neocortical areas. *Nature*, 563:72–78, 2018. doi: 10.1038/s41586-018-0654-5.

Vincent A. Traag, Ludo Waltman, and Nees Jan van Eck. From Louvain to Leiden: Guaranteeing well-connected communities. *Scientific Reports*, 9(1):5233, 2019.

David van Dijk, Roshan Sharma, Juozas Nainys, et al. Recovering gene interactions from single-cell data using data diffusion. *Cell*, 174(3):716–729.e27, 2018. doi: 10.1016/j.cell.2018.05.061.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pp. 5998–6008, 2017.

Mingxuan Wang, Cheng Chen, Gaoyang Jiang, Zijia Ren, Chuangxin Zhao, Lu Shi, and Yanbiao Ma. ScDiVa: Masked discrete diffusion for joint modeling of single-cell identity and expression. *arXiv preprint arXiv:2602.03477*, 2026. URL https://arxiv.org/abs/2602.03477.

Yungang Xu, Zhigang Zhang, Lei You, Jiajia Liu, Zhiwei Fan, and Xiaobo Zhou. scIGANs: Single-cell RNA-seq imputation using generative adversarial networks. *Nucleic Acids Research*, 48(15): e85, 2020. doi: 10.1093/nar/gkaa506.

Luke Zappia, Belinda Phipson, and Alicia Oshlack. Splatter: Simulation of single-cell RNA sequencing data. *Genome Biology*, 18:174, 2017. doi: 10.1186/s13059-017-1305-0.

Zhiqiang Zhang and Lin Liu. scIDPMs: Single-cell RNA-Seq imputation using diffusion probabilistic models. *IEEE Journal of Biomedical and Health Informatics*, 29(4):3057–3068, 2025. doi: 10.1109/JBHI.2024.3430554. Published online in 2024.

Grace X. Y. Zheng, Jessica M. Terry, Phillip Belgrader, et al. Massively parallel digital transcriptional profiling of single cells. *Nature Communications*, 8:14049, 2017. doi: 10.1038/ncomms14049.

A Appendix

Notation. Throughout we fix a single cell and drop the cell index; all quantities are per-cell unless stated otherwise. Table 1 collects the symbols. We write $\mathbf{e}_v \in \{0,1\}^{|\mathcal{V}|}$ for the one-hot column vector of state v, $\mathbf{1}$ for the all-ones column vector, I for the identity, and $\delta_{u,v}$ for the Kronecker delta. Probability row-vectors multiply transition matrices on the left, so $[Q_t]_{uv} = q(x_t = v \mid x_{t-1} = u)$ and each row of Q_t sums to one. We write $\mathcal{B} = \{0, 1, \ldots, B-1\}$ for the value tokens, so that $\mathcal{V} = \mathcal{B} \cup \{\texttt{[MASK]}\}$, and $\mathcal{O}^{\mathsf{c}} = \{g : M_g = 0\}$ for the candidate dropouts. Inside displayed formulas we abbreviate the absorbing symbol `[MASK]` by m, and write $\mathbf{e}_{\mathsf{m}}$ for its one-hot vector; the algorithms and the surrounding prose keep the explicit `[MASK]`. Logarithms are natural, so divergences and entropies are in nats.

1. Additional Theory

1.1 Derivations

1.1.1 The forward kernel in matrix form. Corruption acts independently and identically across genes, so it suffices to describe one site. The absorbing kernel of (Austin et al., 2021) keeps the current token with probability $1 - \beta_t$ and sends it to `[MASK]` otherwise, and `[MASK]` is never left:

$$Q_t \;=\; (1-\beta_t)\, I \;+\; \beta_t\, \mathbf{1}\, \mathbf{e}_{\mathsf{m}}^{\top} \;\in\; \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}. \tag{10}$$

Symbol	Meaning
N, G	cells, genes
B	number of value tokens (bins)
$\mathcal{B}$	value tokens $\{0, \dots, B-1\}$
$\mathcal{V}$	$\mathcal{B} \cup \{\texttt{[MASK]}\}$, size $B+1$
M_g	observation mask, 1 = observed
$\mathcal{O}$	evidence set $\{g : M_g = 1\}$
$\mathcal{O}^{\mathsf{c}}$	candidate dropouts $\{g : M_g = 0\}$
T	diffusion steps
β_t	per-step absorption probability
$\overline{a}_t$	retention $\prod_{s \le t}(1-\beta_s)$, $\overline{a}_0$=1
η_t	unmasking hazard, Eq. equation 16
w	mask rate $1 - \overline{a}_t$
$\mathcal{C}_t, \mathcal{T}_t$	context / target split of $\mathcal{O}$ at step t
ℓ_t^g	per-position cross-entropy, Eq. equation 25
ς_g	step at which gene g is erased
τ_g	per-gene tokenizer, Eq. equation 1
$e_{g,j}$	j-th bin edge of gene g
$r_{g,b}$	representative count of token b
f_θ	denoiser, logits in $\mathbb{R}^{G \times \lvert\mathcal{V}\rvert}$
S	decoding passes, $S \ll T$
$r = s/S$	decoding progress at pass s
κ^g	confidence $\max_{b \in \mathcal{B}} p_\theta(b \mid \cdot)$
n_s	targets left masked after pass s
k_s	targets committed at pass s

Table 1: Notation used throughout the paper and this supplement.

Each row sums to one, since I and $\mathbf{1}\mathbf{e}_{\mathsf{m}}^\top$ are both row-stochastic. Row [MASK] of Eq. equation 10 is $(1-\beta_t)\mathbf{e}_{\mathsf{m}}^\top + \beta_t \mathbf{e}_{\mathsf{m}}^\top = \mathbf{e}_{\mathsf{m}}^\top$, so [MASK] is absorbing, which is the defining property we exploit below. The matrix $\Pi := \mathbf{1}\mathbf{e}_{\mathsf{m}}^\top$ is a rank-one projector, $\Pi^2 = \mathbf{1}(\mathbf{e}_{\mathsf{m}}^\top \mathbf{1})\mathbf{e}_{\mathsf{m}}^\top = \Pi$, because $\mathbf{e}_{\mathsf{m}}^\top \mathbf{1} = 1$; this single identity drives every algebraic simplification that follows.

An absorbing kernel is not the only discrete corruption available: D3PM also considers uniform and Gaussian-banded kernels over the token space. We use absorption because it is the only kernel whose corrupted state is *informationally* rather than *numerically* degraded: a [MASK] says "unknown", whereas a token perturbed to a neighbouring bin says "a slightly different value was measured", which is a false statement about a dropout entry. Everything below the two-point posterior, exact clamping, monotone decoding; is a consequence of absorption specifically, and none of it survives a banded kernel.

1.1.2 The t-step marginal. The following derivation establishes the closed-form marginal stated in the main paper's section "Selective Absorbing Corruption" and used in Eq. equation 3 of the main paper.

Lemma 1 (t-step marginal). *Let $\overline{a}_t = \prod_{s \le t}(1-\beta_s)$ with $\overline{a}_0 = 1$. Then*

$$\bar{Q}_t \;:=\; \prod_{s=1}^{t} Q_s \;=\; \overline{a}_t\, I \;+\; (1-\overline{a}_t)\,\Pi, \tag{11}$$

and consequently $q(x_t \mid x_0) = \overline{a}_t\, \delta_{x_t, x_0} + (1-\overline{a}_t)\, \delta_{x_t, \mathsf{m}}$.

Proof. By induction. The case $t = 0$ is $\bar{Q}_0 = I$, matching Eq. equation 11 with $\overline{a}_0 = 1$. Assume the claim at $t-1$. Then, using $I\Pi = \Pi I = \Pi$ and $\Pi^2 = \Pi$,

$$\begin{aligned}
\bar{Q}_t &= \big[\overline{a}_{t-1} I + (1-\overline{a}_{t-1})\Pi\big]\big[(1-\beta_t)I + \beta_t \Pi\big] \\
&= \overline{a}_{t-1}(1-\beta_t)\, I \;+\; \big[\overline{a}_{t-1}\beta_t + (1-\overline{a}_{t-1})\big]\Pi \\
&= \overline{a}_t\, I + \big[1 - \overline{a}_{t-1}(1-\beta_t)\big]\Pi \;=\; \overline{a}_t I + (1-\overline{a}_t)\Pi.
\end{aligned} \tag{12}$$

Reading off row x_0 gives the stated marginal.

1.1.3 The selective forward process. Imputation is selective, so the chain is run only on the evidence set. We define the forward process conditionally on the mask M:

$$q(\boldsymbol{x}_{1:T} \mid \boldsymbol{x}_0, M) = \prod_{t=1}^{T} \Big[\prod_{g \in \mathcal{O}} q(x_t^g \mid x_{t-1}^g) \Big] \Big[\prod_{g \in \mathcal{O}^{\mathsf{c}}} \delta_{x_t^g, \mathsf{m}} \Big]. \tag{13}$$

Combining Eq. equation 13 with Lemma 1 yields the selective marginal given in Eq. equation 3 of the main paper's section "Selective Absorbing Corruption." Because the $\mathcal{O}^{\mathsf{c}}$ factors are deterministic and identical under the forward and reverse processes, they contribute no free parameters and no divergence; everything derived below is about $\log p_\theta(\boldsymbol{x}_0^{\mathcal{O}} \mid M)$, the only quantity for which ground truth exists.

1.1.4 The reverse posterior. This derivation proves the two-point reverse-posterior statement made after Eq. equation 4 in the main paper's section "Selective Absorbing Corruption."

For $g \in \mathcal{O}$ the true token satisfies $x_0^g \in \mathcal{B}$, i.e. it is never `[MASK]`. Bayes' rule applied to a single site gives the D3PM posterior

$$\begin{aligned} q(x_{t-1} \mid x_t, x_0) &= \frac{q(x_t \mid x_{t-1})\, q(x_{t-1} \mid x_0)}{q(x_t \mid x_0)} \\ &= \frac{\big(Q_t \mathbf{e}_{x_t}\big) \odot \big(\bar{Q}_{t-1}^{\top} \mathbf{e}_{x_0}\big)}{\mathbf{e}_{x_0}^{\top} \bar{Q}_t \mathbf{e}_{x_t}}. \end{aligned} \tag{14}$$

Substituting Eqs. equation 10 and equation 11 collapses Eq. equation 14 to a two-point law (Property P3 of §1.2, where the case analysis is carried out):

$$q(x_{t-1} \mid x_t, x_0) = \begin{cases} \delta_{x_{t-1}, x_t}, & x_t \neq \mathsf{m}, \\ \eta_t\, \delta_{x_{t-1}, x_0} + (1 - \eta_t)\, \delta_{x_{t-1}, \mathsf{m}}, & x_t = \mathsf{m}, \end{cases} \tag{15}$$

where we name the coefficient the *unmasking hazard*

$$\eta_t \;:=\; \frac{\overline{\alpha}_{t-1} - \overline{\alpha}_t}{1 - \overline{\alpha}_t} \;=\; \frac{\beta_t\, \overline{\alpha}_{t-1}}{1 - \overline{\alpha}_t} \;\in\; [0, 1], \qquad \eta_1 = 1. \tag{16}$$

Let $\varsigma_g = \min\{t : x_t^g = \mathsf{m}\}$ be the step at which site g is erased. Its law is $\Pr[\varsigma_g = t] = \overline{\alpha}_{t-1} - \overline{\alpha}_t$ and its survival function is $\Pr[\varsigma_g > t] = \overline{\alpha}_t$, so $\eta_t = \Pr[\varsigma_g = t \mid \varsigma_g \geq t]$ is the discrete hazard rate of erasure, and, read backwards in time, the probability that a still-masked site is the one the reverse step restores. We use η rather than γ because $\gamma(t)$ already denotes the timestep embedding in Eq. equation 5 of the main paper.

1.1.5 The x_0-parameterized reverse model. ABSORB does not parameterize $p_\theta(x_{t-1} \mid \boldsymbol{x}_t)$ directly. It predicts a distribution over clean tokens and pushes it through the exact posterior,

$$p_\theta(x_{t-1}^g \mid \boldsymbol{x}_t) = \sum_{\tilde{x}_0 \in \mathcal{B}} q\big(x_{t-1}^g \mid x_t^g, \tilde{x}_0\big)\, p_\theta\big(\tilde{x}_0 \mid \boldsymbol{x}_t, \mathcal{C}_t, t\big), \tag{17}$$

with $p_\theta(\cdot \mid \boldsymbol{x}_t, \mathcal{C}_t, t)$ the softmax head defined in Eq. equation 6 of the main paper. Restricting the sum to $\mathcal{B}$ is exact, not an approximation: the true x_0^g is a value token by construction, so the posterior places zero mass on `[MASK]`, and dropping the `[MASK]` logit at decoding time as described in the main paper's section "Training". Writing Eq. equation 17 out with Eq. equation 15,

$$p_\theta(x_{t-1}^g \mid \boldsymbol{x}_t) = \begin{cases} \delta_{x_{t-1}^g, x_t^g}, & x_t^g \neq \mathsf{m}, \\ (1 - \eta_t)\, \delta_{x_{t-1}^g, \mathsf{m}} & \\ \quad + \eta_t\, p_\theta(x_{t-1}^g \mid \boldsymbol{x}_t, \mathcal{C}_t, t), & x_t^g = \mathsf{m}. \end{cases} \tag{18}$$

1.1.6 The variational bound Let $p(\boldsymbol{x}_T) = \prod_g \delta_{x_T^g, \mathsf{m}}$ be the prior and $p_\theta(\boldsymbol{x}_{0:T}) = p(\boldsymbol{x}_T) \prod_{t=1}^{T} p_\theta(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)$ the reverse model. We write the bound, step by step since the selective construction of §1.1.3 differs from the standard setting.

Step 1 (importance weight and Jensen). Multiplying and dividing by the forward joint and applying Jensen's inequality to the convex function $-\log$, writing $q_\star := q(\boldsymbol{x}_{1:T} \mid \boldsymbol{x}_0, M)$ for conciseness,

$$\begin{aligned} -\log p_\theta(\boldsymbol{x}_0^{\mathcal{O}} \mid M) &= -\log \sum_{\boldsymbol{x}_{1:T}} q_\star \, \frac{p_\theta(\boldsymbol{x}_{0:T})}{q_\star} \\ &\le \mathbb{E}_q\Big[-\log \frac{p_\theta(\boldsymbol{x}_{0:T})}{q_\star} \Big] \;=:\; \mathcal{L}_{\mathrm{vb}}. \end{aligned} \tag{19}$$

Step 2 (telescoping). Expanding both products,

$$\begin{aligned} \mathcal{L}_{\mathrm{vb}} = \mathbb{E}_q\Big[&-\log p(\boldsymbol{x}_T) - \textstyle\sum_t \log p_\theta(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t) \\ &+ \textstyle\sum_t \log q(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}) \Big], \end{aligned} \tag{20}$$

both sums running over $t = 1, \dots, T$. For $t \ge 2$ use the Markov property, $q(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}) = q(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}, \boldsymbol{x}_0)$, and Bayes' rule in the form $q(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}, \boldsymbol{x}_0) = q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0)\, q(\boldsymbol{x}_t \mid \boldsymbol{x}_0) / q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_0)$. The ratio $q(\boldsymbol{x}_t \mid \boldsymbol{x}_0)/q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_0)$ telescopes over $t = 2, \dots, T$ to $q(\boldsymbol{x}_T \mid \boldsymbol{x}_0)/q(\boldsymbol{x}_1 \mid \boldsymbol{x}_0)$, and the surviving $\log q(\boldsymbol{x}_1 \mid \boldsymbol{x}_0)$ from $t = 1$ cancels the denominator. Collecting terms,

$$\mathcal{L}_{\mathrm{vb}} = L_T + \sum_{t=2}^{T} L_{t-1} + L_0, \tag{21}$$

$$\begin{aligned} L_T &= D_{\mathrm{KL}}\big(q(\boldsymbol{x}_T \mid \boldsymbol{x}_0) \,\big\|\, p(\boldsymbol{x}_T)\big), \\ L_{t-1} &= \mathbb{E}_{q(\boldsymbol{x}_t \mid \boldsymbol{x}_0)} D_{\mathrm{KL}}\big(q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0) \,\big\|\, p_\theta(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)\big), \\ L_0 &= \mathbb{E}_{q(\boldsymbol{x}_1 \mid \boldsymbol{x}_0)}\big[-\log p_\theta(\boldsymbol{x}_0 \mid \boldsymbol{x}_1)\big]. \end{aligned} \tag{22}$$

Step 3 (the gap). Jensen's inequality in Eq. equation 19 is precisely when the importance weight is constant, and the deficit is the standard one,

$$\begin{aligned} &\mathcal{L}_{\mathrm{vb}} + \log p_\theta(\boldsymbol{x}_0^{\mathcal{O}} \mid M) \\ &\qquad = D_{\mathrm{KL}}\big(q(\boldsymbol{x}_{1:T} \mid \boldsymbol{x}_0, M) \,\big\|\, p_\theta(\boldsymbol{x}_{1:T} \mid \boldsymbol{x}_0)\big) \;\ge\; 0. \end{aligned} \tag{23}$$

Lemma 2 (KL of product measures). *If $q(\mathbf{a}) = \prod_i q_i(a_i)$ and $p(\mathbf{a}) = \prod_i p_i(a_i)$ then $D_{\mathrm{KL}}(q\|p) = \sum_i D_{\mathrm{KL}}(q_i\|p_i)$.*

Proof. $D_{\mathrm{KL}}(q\|p) = \mathbb{E}_q \sum_i \log \frac{q_i(a_i)}{p_i(a_i)}$, and each summand depends on one coordinate only, whose marginal under q is q_i.

Both arguments of L_{t-1} factorize the forward posterior by Eq. equation 13 and the reverse model by construction, so Lemma 2 applies and the divergence decomposes over genes. We now evaluate all three groups of terms.

The prior term vanishes. We set $\beta_T = 1$, so $\overline{\alpha}_T = 0$ and $q(\boldsymbol{x}_T \mid \boldsymbol{x}_0) = p(\boldsymbol{x}_T)$ exactly, giving $L_T = 0$. This is not cosmetic: if $\overline{\alpha}_T > 0$ the forward marginal retains mass on x_0 where the point-mass prior has none, and $L_T = \infty$ (P7, §1.2). Any implementation that clips β_t at 0.999 for numerical reasons exempts the final step.

The KL terms are weighted cross-entropies. By Lemma 2 it suffices to treat one gene. For $g \in \mathcal{O}^{\mathsf{c}}$ both laws are δ_{m} and the term is zero. For $g \in \mathcal{O}$ with $x_t^g \neq \mathsf{m}$ both laws are $\delta_{x_t^g}$ by Eqs. equation 15 and equation 18, and the term is again zero. The only surviving case is $g \in \mathcal{T}_t$, where both laws are supported on $\{x_0^g, \mathsf{m}\}$ and, assign the same mass $1 - \eta_t$ to m. The mask coordinates cancel and

$$\begin{aligned} D_{\mathrm{KL}} &= \eta_t \log \frac{\eta_t}{\eta_t \, p_\theta(x_0^g \mid \boldsymbol{x}_t, \mathcal{C}_t, t)} + (1 - \eta_t) \log \frac{1 - \eta_t}{1 - \eta_t} \\ &= -\,\eta_t \log p_\theta\big(x_0^g \mid \boldsymbol{x}_t, \mathcal{C}_t, t\big). \end{aligned} \tag{24}$$

So the per-step KL is a cross-entropy at masked-evidence positions, scaled by a schedule constant. No approximation has been made, and in particular no λ-weighted surrogate has been substituted. From here on we abbreviate the per-position cross-entropy as

$$\ell_t^g \;:=\; -\log p_\theta\big(x_0^g \mid \boldsymbol{x}_t, \mathcal{C}_t, t\big). \tag{25}$$

The reconstruction term is the $t = 1$ case. Since $\overline{\alpha}_0 = 1$ we have $\eta_1 = 1$, and $L_0 = \sum_{g\in\mathcal{T}_1} \ell_1^g$ is exactly Eq. equation 24 evaluated at $t = 1$. The three groups therefore merge into one sum.

Theorem 1 (exact bound for the selective absorbing chain). With $\beta_T = 1$ and the parameterization of Eq. equation 17,

$$\mathcal{L}_{\mathrm{vb}} = \mathbb{E}_{\boldsymbol{x}_0} \sum_{t=1}^{T} \eta_t \, \mathbb{E}_{\boldsymbol{x}_t|\boldsymbol{x}_0}\Big[\sum_{g\in\mathcal{T}_t} \ell_t^g\Big]. \tag{26}$$

Proof. Substitute $L_T = 0$, Eq. equation 24 for $t = 2, \ldots, T$, and $\eta_1 = 1$ for L_0 into Eq. equation 21.

Corollary 1 (one cross-entropy per gene). The contribution of a single gene $g \in \mathcal{O}$ to Eq. equation 26 is

$$\sum_{t=1}^{T} \Pr[\varsigma_g = t] \; \mathbb{E}\big[\, \ell_t^g \;\big|\; x_t^g = \mathsf{m} \,\big], \tag{27}$$

a proper expectation over the erasure time ς_g.

Proof. Fix g and take the expectation inside: $\mathbb{E}_{\boldsymbol{x}_t}[\mathbf{1}[x_t^g{=}\mathsf{m}]\, f] = (1-\overline{\alpha}_t)\,\mathbb{E}[f \mid x_t^g{=}\mathsf{m}]$ by Lemma 1. Multiplying by η_t and using Eq. equation 16 gives $\eta_t(1-\overline{\alpha}_t) = \overline{\alpha}_{t-1} - \overline{\alpha}_t = \Pr[\varsigma_g = t]$. These weights sum to $\overline{\alpha}_0 - \overline{\alpha}_T = 1$ by P2.

1.1.7 From the bound to the training objective. We now relate the exact variational objective to the practical self-supervised loss given in Eq. equation 7 of the main paper. Eq. equation 7 of the main paper draws $t \sim \mathcal{U}\{1, \ldots, T\}$ and averages the cross-entropy over $\mathcal{T}_t$:

$$\mathcal{L}(\theta) = \mathbb{E}_{\boldsymbol{x}_0} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}_{\boldsymbol{x}_t|\boldsymbol{x}_0}\Big[\frac{1}{|\mathcal{T}_t|} \sum_{g\in\mathcal{T}_t} \ell_t^g\Big]. \tag{28}$$

Both Eq. equation 26 and Eq. equation 28 are members of the weighted family

$$\mathcal{L}_w(\theta) = \mathbb{E}_{\boldsymbol{x}_0} \sum_{t=1}^{T} \mathbb{E}_{\boldsymbol{x}_t|\boldsymbol{x}_0}\Big[w_t(\boldsymbol{x}_t) \sum_{g\in\mathcal{T}_t} \ell_t^g\Big], \tag{29}$$

with $w_t^{\mathrm{vb}} = \eta_t$ and $w_t^{\mathrm{ours}}(\boldsymbol{x}_t) = 1/(T\,|\mathcal{T}_t|)$. Both weights are strictly positive, are independent of θ, and, according to §1.1.3(i), depend on $\boldsymbol{x}_0$ only through $\boldsymbol{x}_t$.

Proposition 1 (reweighting preserves the optimum). Let $w_t(\boldsymbol{x}_t) > 0$ be any weight measurable with respect to $(\boldsymbol{x}_t, M, t)$. Over all conditional distributions p_θ, the unique minimiser of $\mathcal{L}_w$ is the true conditional

$$p^\star(b \mid \boldsymbol{x}_t, \mathcal{C}_t, t) = q(x_0^g = b \mid \boldsymbol{x}_t, \mathcal{C}_t, t) \tag{30}$$

for every t, every $\boldsymbol{x}_t$ of positive probability, and every $g \in \mathcal{T}_t$. In particular Eqs. equation 26 and equation 28 have the same population optimum.

Proof. Since w_t does not depend on x_0, we may write

$$\mathcal{L}_w = \sum_t \sum_{\boldsymbol{x}_t} \Pr[\boldsymbol{x}_t]\, w_t(\boldsymbol{x}_t) \sum_{g\in\mathcal{T}_t} \mathbb{E}_{x_0^g|\boldsymbol{x}_t}\big[\ell_t^g\big].$$

The inner expectation equals $H(q^g) + D_{\mathrm{KL}}(q^g \| p_\theta^g)$, where $q^g = q(x_0^g \mid \boldsymbol{x}_t, \mathcal{C}_t, t)$. The entropy is a constant of the data, and by Gibbs' inequality the divergence is uniquely minimised at $p_\theta^g = q^g$. The objective decouples across the triples $(t, \boldsymbol{x}_t, g)$, and each is multiplied by a strictly positive constant, so the joint minimiser is attained coordinate-wise.

Proposition 1 explains the reweighting used in Eq. equation 7 of the main paper.

1.1.8 Relation to the D3PM auxiliary loss and to masked language modelling. (Austin et al., 2021) train with $\mathcal{L}_\lambda = \mathcal{L}_{\mathrm{vb}} + \lambda\,\mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)}[-\log p_\theta(\boldsymbol{x}_0 \mid \boldsymbol{x}_t)]$, adding an unweighted x_0 reconstruction term to stabilise the bound. Eq. equation 28 is that auxiliary term, with two task-specific restrictions. Initially, the sum runs over $\mathcal{T}_t$ rather than all G positions: at $g \in \mathcal{C}_t$ the clean token is an input coordinate, so ℓ_t^g is minimised by copying and supplies no learning signal, while at $g \in \mathcal{O}^{\mathrm{c}}$ there is no ground truth at all. Second, the average is taken per cell, with the consequence noted in §1.1.7(iii). Since $\mathcal{L}_{\mathrm{vb}}$ is itself a positively weighted sum of the same per-position cross-entropies (Theorem 1), $\mathcal{L}_\lambda$ collapses to a single reweighting inside the family of Eq. equation 29, with $w_t = \eta_t + \lambda/T$, and Proposition 1 can be applied to it. The two terms of $\mathcal{L}_\lambda$ are therefore not in tension: for the absorbing kernel specifically, the "auxiliary" loss is the same as the bound, differently weighted. That is a property of absorption, not of D3PM; for a uniform kernel the t-th KL does not reduce to a cross-entropy and the two terms genuinely differ.

1.1.9 The continuous-time limit and schedule invariance. The relationship between Eqs. equation 26 and equation 28 is sharpest in the limit $T \to \infty$, where it becomes an identity rather than a proportionality. Let $u = t/T \in (0, 1]$ and let $\alpha(u)$ be the smooth limit of $\overline{\alpha}_{\lceil uT \rceil}$, with $\alpha(0) = 1$, $\alpha(1) = 0$, and α strictly decreasing. Assign $w = 1 - \alpha(u)$ for the mask rate and

$$\bar{\ell}(w) \;:=\; \mathbb{E}\Big[\tfrac{1}{|\mathcal{T}|}\sum_{g\in\mathcal{T}} \ell^g \;\Big|\; \text{mask rate } w\Big] \tag{31}$$

for the average cross-entropy per masked token at that rate. Since $\eta_t \to -\alpha'(u)\,du/(1-\alpha(u))$ and $|\mathcal{T}_t| \to (1-\alpha(u))|\mathcal{O}|$, Theorem 1 becomes

$$\mathcal{L}_{\mathrm{vb}} \;\to\; |\mathcal{O}| \int_0^1 \frac{-\alpha'(u)}{1-\alpha(u)}\,\big(1-\alpha(u)\big)\,\bar{\ell}\big(w(u)\big)\,du, \tag{32}$$

and substituting $w = 1 - \alpha(u)$, $dw = -\alpha'(u)\,du$,

$$\boxed{\mathcal{L}_{\mathrm{vb}} = |\mathcal{O}| \int_0^1 \bar{\ell}(w)\,dw} \tag{33}$$

Eq. equation 33 is schedule-invariant: in continuous time the bound depends on the model only through $\bar{\ell}(\cdot)$ as a function of mask rate, integrated uniformly, and not on how the schedule traverses those rates. The same substitution applied to Eq. equation 28 gives

$$\mathcal{L}(\theta) \;\to\; \int_0^1 \bar{\ell}\big(w(u)\big)\,du \;=\; \int_0^1 \bar{\ell}(w)\,\frac{dw}{-\alpha'\big(u(w)\big)}. \tag{34}$$

Corollary 2 (when the objective is the bound). In the continuous-time limit, $\mathcal{L}(\theta)$ is an average of $\bar{\ell}(w)$ under the density $\varphi(w) = 1/(-\alpha'(u(w)))$ on $[0, 1]$, while $\mathcal{L}_{\mathrm{vb}}/|\mathcal{O}|$ is the average under the uniform density. The two coincide, exactly, if and only if α is affine, i.e. $\overline{\alpha}_t = 1 - t/T$, which is the schedule D3PM's absorbing kernel produces with $\beta_t = 1/(T - t + 1)$. For the cosine schedule, $w(u) = \sin^2(\pi u/2)$ and

$$\varphi(w) = \frac{1}{\pi\sqrt{w(1-w)}}, \tag{35}$$

the arcsine density.

Proof. The first claim is Eqs. equation 33–equation 34. Affinity: $\varphi \equiv 1$ iff $-\alpha'$ is constant. Cosine: $w = \sin^2(\pi u/2)$ gives $dw/du = (\pi/2)\sin(\pi u) = \pi\sqrt{w(1-w)}$, using $\sin(\pi u) = 2\sin(\frac{\pi u}{2})\cos(\frac{\pi u}{2})$; invert. Both densities integrate to 1, so the comparison is between two averages of the same function.

Corollary 2 also reproduces Eq. equation **??**: the ratio of the two weightings is $\varphi(w) = 2/(\pi\sin(\pi u))$, matching the discrete calculation.

1.1.10 The cosine schedule and the induced mask-rate law. This analysis clarifies the cosine retention schedule introduced in the main paper's section "Selective Absorbing Corruption" and the choice $T = 100$ reported in its "Implementation" section.

We use the cosine retention of Nichol & Dhariwal (2021), transplanted from the Gaussian signal rate to the absorbing survival probability:

$$\overline{a}_t = \frac{f(t)}{f(0)}, \quad f(t) = \cos^2\Big(\frac{t/T + \varepsilon_{\mathrm{o}}}{1 + \varepsilon_{\mathrm{o}}} \cdot \frac{\pi}{2}\Big), \tag{36}$$

with offset $\varepsilon_{\mathrm{o}} = 0.008$, $\beta_t = 1 - \overline{a}_t/\overline{a}_{t-1}$ clipped to $[0, 0.999]$ for $t < T$, and $\beta_T = 1$ as required by §1.1.6. Note $f(T) = \cos^2(\pi/2) = 0$, so the schedule already terminates at $\overline{a}_T = 0$.

By Corollary 2 the schedule is in the continuous-time limit, a choice of importance density over mask rates rather than a change to the objective, and Eq. equation 35 identifies that density as the arcsine law. Three comparisons follow, and they resolve an ambiguity in how the cosine-schedule statement in the main paper's section "Selective Absorbing Corruption" is to be interpreted.

Against the linear β schedule of Ho et al. (2020), where $\overline{a}_t \approx \exp(-\sum_{s \le t} \beta_s)$ falls quasi-exponentially; with the usual β range and $T = 1000$, $\overline{a}_{T/2} \approx 0.08$, so $w \approx 0.92$ already at the midpoint and the whole second half of the chain sits at near-total masking, the cosine schedule explores further steps at intermediate mask rates. This is the comparison Nichol & Dhariwal (2021) draw and the sense in which the cosine-schedule statement in the main paper holds.

Against the linear $\overline{a}$ schedule $\overline{a}_t = 1 - t/T$, which Austin et al. (2021) obtain for the absorbing kernel with $\beta_t = 1/(T - t + 1)$ and which makes w exactly uniform, the cosine schedule instead oversamples both extremes, by the factor $\varphi(w) = 1/(\pi\sqrt{w(1-w)})$. We state this plainly because Eq. equation 35 makes it unambiguous, and because the linear-$\overline{a}$ schedule is the more natural point of comparison for an absorbing chain.

We keep the cosine schedule because the two extremes are the operationally important parts here. The regime $w \to 0$ is where the decoder spends its late passes and, more fundamentally, is the only regime in which the entire evidence set is present: the condition that holds at every decoding pass (§1.1.13). The regime $w \to 1$ is the first pass, where nearly every target is still masked. The arcsine emphasis is thus aligned with the decoder rather than with likelihood estimation, which is consistent with the reweighting argument of §1.1.7(ii).

1.1.11 The count-scale readout. Decoding returns, for each imputed gene, a categorical p^g over $\mathcal{B}$ and the per-gene representative table $r_{g,\cdot}$. These induce a predictive distribution over counts: $\hat{C}^g = r_{g,b}$ with $b \sim p^g$. The point estimate to report depends on the loss.

Proposition 2 (the expectation is the MMSE readout). $\arg\min_{c \in \mathbb{R}} \ \mathbb{E}_{b \sim p^g}\big[(c - r_{g,b})^2\big] = \sum_{b \in \mathcal{B}} p_b^g \, r_{g,b}$, *which is Eq. equation 9.*

Proof. The objective is strictly convex in c with derivative $2\big(c - \sum_b p_b^g r_{g,b}\big)$; setting it to zero gives the mean, and the second derivative $2 > 0$ confirms the minimum.

For completeness: the 0–1 loss optimum is the mode $r_{g,\arg\max_b p_b^g}$, the absolute-error optimum is the weighted median of $\{r_{g,b}\}$ under p^g, and the optimum for squared error on the $\log_{1p}$ scale used by our RMSE metric is $\exp\big(\sum_b p_b^g \log(1 + r_{g,b})\big) - 1$, which by Jensen's inequality is no larger than Eq. equation 9 of the main paper. We report the raw-scale mean because the imputed matrix is handed to downstream tools on the count scale: the gap is the difference between an arithmetic and a geometric mean of $1 + r_{g,b}$ under p^g, so it is second order in the spread of the predictive and vanishes as the model becomes confident.

Two structural guarantees follow immediately, and they are the formal content of the non-negativity claim in the main text. Because $r_{g,0} = 0$ and $r_{g,b} \ge 0$ for all b, and because p^g is a probability vector, Eq. equation 9 of the main paper gives

$$0 \ \le \ \hat{c}^g \ \le \ \max_b r_{g,b} = r_{g,B-1}, \tag{37}$$

so no imputed value can be negative, as linear smoothing can produce van Dijk et al. (2018), and none can exceed the largest representative of that gene. The upper bound is a genuine limitation as well as a guarantee: the top bin $[e_{g,B-2}, \infty)$ is unbounded, its representative is a median, and therefore counts in the extreme right tail are systematically shrunk toward $r_{g,B-1}$. Using a high quantile of the top bin instead of its median would trade this shrinkage against variance.

1.1.12 Properties of the quantile tokenizer. Let Z_g denote the distribution of gene g's non-zero counts over the training cells, and let $e_{g,j}$ be its $j/(B-1)$ quantile for $j = 1, \dots, B-2$. Eq. equation 1 assigns token $1 + j$ to the interval $[e_{g,j}, e_{g,j+1})$, with $e_{g,0} = 0^+$ and $e_{g,B-1} = \infty$.

(i) Equal mass. If Z_g is continuous then $\Pr[Z_g \in [e_{g,j}, e_{g,j+1})] = \frac{j+1}{B-1} - \frac{j}{B-1} = \frac{1}{B-1}$ for every interior j, and the two end intervals carry $\Pr[Z_g < e_{g,1}] = \frac{1}{B-1}$ and $\Pr[Z_g \geq e_{g,B-2}] = 1 - \frac{B-2}{B-1} = \frac{1}{B-1}$. All $B-1$ non-zero tokens therefore receive equal mass, and together with the reserved zero token they exhaust the B value tokens of Eq. equation 2.

(ii) Order preservation. τ_g is non-decreasing: $c \leq c'$ implies $|\{j : e_{g,j} \leq c\}| \leq |\{j : e_{g,j} \leq c'\}|$ and the $\min$ with $B-2$ is monotone. Hence the token sequence carries the same ordering as the counts, and any rank statistic computed on tokens agrees with the same statistic on counts up to within-bin ties.

(iii) Ties and the effective vocabulary. UMI counts are small integers with large atoms at $1, 2, 3$. If $\Pr[Z_g = v] > 1/(B-1)$ then several consecutive quantiles coincide at v, the intervening intervals are empty, and the realised number of occupied tokens obeys

$$B_g^{\text{eff}} \;\leq\; \min\big(B-1,\; |\operatorname{supp}(Z_g)|\big). \tag{38}$$

Eq. equation 38 is the mechanism behind the ablation of §2.2: past the point where $B-1$ exceeds the number of distinct counts a gene actually takes, additional bins are unreachable and raising B cannot reduce quantization error. Conversely, every value v that is the unique support point of its bin is encoded exactly, because the bin median is then v itself. Since the low integers carry most of the mass, quantization error is confined to the upper tail, where bins are wide and the $\log_{1p}$ transform used by our metrics compresses the residual.

(iv) Representatives. Setting $r_{g,b}$ to the median of the training counts in bin b minimises the expected within-bin absolute error and is robust to the unbounded top bin; the mean would minimise squared error but is dragged upward by outliers. Empty bins are assigned $r_{g,b} = r_{g,b-1}$, which keeps $r_{g,\cdot}$ non-decreasing: required for the bound in Eq. equation 37 and for the monotonicity in (ii) to survive the decoding step. Unreachable tokens are harmless: they receive no training targets, so the head's logits for them are driven down by the softmax normalisation and they are effectively pruned.

(v) The quantization ceiling. Define the token oracle as the imputer that predicts the correct token at every held-out position and reads out its representative. Its error on the metric of §Simulated Data is

$$\big(\mathrm{RMSE}^\star\big)^2 = \frac{1}{|\mathcal{H}|} \sum_{(i,g)\in\mathcal{H}} \big(\ell_{ig} - \hat{\ell}_{ig}\big)^2, \tag{39}$$

where $\ell_{ig} = \log(1 + C_{ig})$ is the true value on the metric scale and $\hat{\ell}_{ig} = \log\big(1 + r_{g,\tau_g(C_{ig})}\big)$ is its tokenize-to-detokenize round trip. It is computed with edges and representatives fitted on the training cells only, so Eq. equation 39 is measurable without running any model, and it isolates the error attributable purely to discretization.

(vi) Leakage. Edges and representatives are estimated on training cells only. Held-out entries in $\mathcal{H}$ never enter a quantile or a median, so the tokenizer cannot transmit the answer. See §1.5 for the corresponding statement about the observation mask, which is the weaker of the two.

1.1.13 Confidence decoding: what is exact and what is approximate. An exact ancestral sampler for the reverse chain draws $\boldsymbol{x}_{t-1} \sim p_\theta(\cdot \mid \boldsymbol{x}_t)$ for $t = T, \dots, 1$ using Eq. equation 18. Confidence decoding replaces it by $S \ll T$ passes. The replacement is the composition of three distinct approximations.

(a) Step skipping is exact. Since Eq. equation 18 factorizes across genes and leaves unmasked sites untouched, a reverse step at which no site unmasks leaves $\boldsymbol{x}_{t-1} = \boldsymbol{x}_t$ and therefore leaves the

network output unchanged; evaluating f_θ at such a step is pure waste. The probability that a given site stays masked from step t down to step t' telescopes,

$$\prod_{u=t'+1}^{t} \big(1-\eta_u\big) = \prod_{u=t'+1}^{t} \frac{1-\overline{a}_{u-1}}{1-\overline{a}_u} = \frac{1-\overline{a}_{t'}}{1-\overline{a}_t}, \tag{40}$$

so each target's unmasking time can be drawn up front from Eq. equation 40 and evaluate the network only at the distinct times drawn. This is an exact reformulation, and it caps the cost at $\min\big(T, |\mathcal{T}|\big)$ evaluations. In our regime $|\mathcal{T}| \gg T$, the expected number of distinct unmasking times is $\sum_t \big[1-(1-\Pr[\varsigma{=}t])^{|\mathcal{T}|}\big] \approx T$, so the cap is not binding and (a) alone saves nothing. All of the saving comes from (b).

(b) Deterministic grouping. We replace the random unmasking times by a deterministic schedule: after pass s, with $r = s/S$, exactly $n_s = \lfloor \cos(\frac{\pi}{2} r)|\mathcal{T}| \rfloor$ targets remain masked (Eq. equation 8), so pass s commits

$$k_s = n_{s-1} - n_s, \qquad \sum_{s=1}^{S} k_s = |\mathcal{T}|, \tag{41}$$

with $n_0 = |\mathcal{T}|$ and $n_S = 0$. The endpoint $n_S = 0$ holds because $\cos(\pi/2) = 0$, so every target is committed by the last pass and no post-hoc completion is required. Since $\cos$ is strictly decreasing on $[0, \pi/2]$, $k_s \geq 0$ always, but the floor can make $k_s = 0$ when $|\mathcal{T}|$ is small relative to S; we then force $k_s \geq 1$ so that no pass is wasted.

(c) Confidence ordering and hard commitment. Within a pass we choose *which* k_s sites to commit by confidence rather than at random, and we commit $\arg\max_b p_b^g$ rather than a sample.

Proposition 3 (any-order decoding is exact at $k = 1$). Suppose the model is exact, i.e. $p_\theta(x_0^g \mid \boldsymbol{x}, \mathcal{C}, t) = q(x_0^g \mid \boldsymbol{x}^{\mathcal{C}})$ for every revealed set $\mathcal{C}$. Run the decoder with $k_s = 1$ for all s, choosing the site to reveal by any rule that is a function of the already-revealed values (the confidence rule qualifies), and *sampling* the committed token from p^g. Then after $|\mathcal{T}|$ passes the revealed vector is an exact sample from $q(\boldsymbol{x}_0^{\mathcal{T}} \mid \boldsymbol{x}_0^{\mathcal{O}})$.

Proof. Let $g_1, \ldots, g_K$ be the revealed order, $K = |\mathcal{T}|$. Each g_k is measurable with respect to $\sigma(\boldsymbol{x}^{\mathcal{O}}, x^{g_1}, \ldots, x^{g_{k-1}})$, because the confidences that determine it are outputs of f_θ applied to already-revealed information. The law of the produced vector is therefore $\prod_{k=1}^{K} q\big(x^{g_k} \mid \boldsymbol{x}^{\mathcal{O}}, x^{g_1}, \ldots, x^{g_{k-1}}\big)$, which is the chain-rule factorisation of $q(\boldsymbol{x}^{\mathcal{T}} \mid \boldsymbol{x}^{\mathcal{O}})$ under the adapted order.

Proposition 4 (parallel commitment costs exactly the block's total correlation). Fix a pass with revealed context $\mathcal{R}$ and committed block A. Let p^A be the model's true joint over A given $\mathcal{R}$ and let p^g be its marginals. Committing A simultaneously samples from $\prod_{g\in A} p^g$, and

$$\begin{aligned} D_{\mathrm{KL}}\Big(p^A \,\Big\|\, \textstyle\prod_{g\in A} p^g\Big) &= \sum_{g\in A} H(p^g) - H(p^A) \\ &= \mathrm{TC}(A \mid \mathcal{R}), \end{aligned} \tag{42}$$

the total correlation (multi-information) of the block. It is zero iff the block is conditionally independent given $\mathcal{R}$, and it vanishes identically when $|A| = 1$, recovering Proposition 3.

Proof. Expand the divergence as

$$\sum_{\mathbf{a}} p^A(\mathbf{a}) \log p^A(\mathbf{a}) \;-\; \sum_{\mathbf{a}} p^A(\mathbf{a}) \sum_{g\in A} \log p^g(a_g).$$

The first term is $-H(p^A)$; in the second, each summand depends on one coordinate whose p^A-marginal is p^g, giving $\sum_g H(p^g)$. Non-negativity and the equality condition are Gibbs' inequality.

Eq. equation 42 is also the concrete reading of the ELBO gap Eq. equation 23 promised in §1.1.6: the reverse model factorizes over genes, so the divergence it incurs at each step is the dependence among the genes unmasked at that step. Training loss and decoding error are the same quantity, and

both shrink as fewer sites are resolved per step. This is the honest form of the statement that "more passes are better": the budget S buys a reduction in $\sum_s \mathrm{TC}(A_s)$, and $S = |\mathcal{T}|$ buys exactness.

Proposition 5 (confidence ordering minimises a bound on the cost). Let $\Phi(\kappa) := H_b(\kappa) + (1-\kappa)\log(B-1)$ with $H_b(\kappa) = -\kappa\log\kappa - (1-\kappa)\log(1-\kappa)$. Then for any block A drawn from the still-masked set U,

$$\mathrm{TC}(A \mid \mathcal{R}) \;\le\; \sum_{g\in A} H(p^g) \;\le\; \sum_{g\in A} \Phi(\kappa^g), \tag{43}$$

and Φ is non-increasing on $[1/B, 1]$, which contains every attainable κ. Consequently, among all $A \subseteq U$ with $|A| = k$, the top-k by confidence minimises the bound in Eq. equation 43.

Proof. The first inequality is Eq. equation 42 with $H(p^A) \ge 0$. For the second, among distributions on B outcomes with maximum probability κ, entropy is maximised by placing κ on one outcome and spreading $1-\kappa$ uniformly over the other $B-1$, giving $H \le -\kappa\log\kappa - (1-\kappa)\log\frac{1-\kappa}{B-1} = \Phi(\kappa)$; the configuration is feasible because $\kappa \ge 1/B$ always. Monotonicity: $\Phi'(\kappa) = \log\frac{1-\kappa}{(B-1)\kappa} \le 0$ exactly when $\kappa \ge 1/B$. The selection claim then follows because the bound is a sum of k terms drawn from the multiset $\{\Phi(\kappa^g)\}_{g\in U}$, minimised by taking the k smallest values of Φ, i.e. the k largest κ.

Proposition 5 is the justification for confidence-first decoding, and it is stronger than the intuitive one. The intuitive argument: that the top-k maximises $\prod_{g\in A}\kappa^g$, the model's belief that the whole block is correct, is true but is only a statement about marginals. Proposition 5 instead bounds the distributional error of the parallel step, the exact quantity identified by Proposition 4, and shows that the same greedy rule minimises it.

Hard commitment. Argmax rather than sampling makes the procedure mode-seeking: it does not sample from p_θ and Proposition 3 does not apply to it. Two things limit the damage. First, the reported value at gene g is the expectation under the cached p^g, not the committed token, so $\arg\max$ affects the output only through the *context* it supplies to later passes. Second, imputation is scored by pointwise accuracy and by distributional agreement of the recovered population, not by within-cell sample diversity, and for pointwise loss a mode-seeking context is the right bias. A temperature-T_s stochastic variant, committing $b \sim \mathrm{softmax}(\log p^g / T_\mathrm{s})$, is available and reduces to the deterministic rule as $T_\mathrm{s} \to 0$; we use $T_\mathrm{s} = 1$ for the reported distributions and $\arg\max$ for the committed tokens.

Cached readout. A subtlety follows from (c) that materially affects the implementation. A gene committed at pass s becomes context at pass $s+1$, so if the network were re-evaluated at the end it would simply read its own committed token back and return a near-degenerate distribution. The expectation of Eq. equation 9 must therefore be taken under the distribution p^g cached at the pass where g was committed, not under a final sweep. Algorithm 5 makes this explicit.

Schedule alignment. Pass s queries the network at $t_s = \lfloor (1-r)(T-1) \rceil$. At that index, training exposed a fraction $1 - \bar{\alpha}_{t_s} \approx \sin^2\big(\frac{\pi}{2}(1-r)\big) = \cos^2(\frac{\pi}{2}r)$ of the evidence as masked, whereas the decoder holds $n_s/|\mathcal{T}| = \cos(\frac{\pi}{2}r)$ of its targets masked. The two agree at the endpoints $r \in \{0,1\}$ and differ by a square in between, the decoder being the more conservative of the two: it keeps more sites masked than the training marginal at the same index, i.e. it commits more slowly than the forward chain erases. Exact matching is not required, because the timestep is a soft conditioning signal while the hard information about which positions are context is carried by the token stream itself (§1.1.15). Setting $n_s = \lfloor \cos^2(\frac{\pi}{2}r)|\mathcal{T}| \rfloor$ would align the two exactly, at the cost of committing more aggressively in the early passes, where by Proposition 5 the confidence is lowest and the total correlation cost is highest.

Index convention. Training draws $t \in \{1,\dots,T\}$ while $t_s = \lfloor (1-r)(T-1) \rceil$ ranges over $\{0,\dots,T-1\}$, so the final pass queries $t = 0$, one index below the training support. Since $\gamma(\cdot)$ is a sinusoidal embedding followed by an MLP, it extrapolates smoothly over one index out of $T = 100$: clamping $t_s \leftarrow \max(1, t_s)$ removes the discrepancy exactly and is what Algorithm 5 does.

1.1.14 Why no RePaint-style projection is needed. In Gaussian inpainting, the observed region must be re-noised to the current level and pasted in at every step, because the model was trained on fully corrupted inputs and $p_\theta(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)$ has no notion of a clamped coordinate: the pasted values are then inconsistent with the model's own transition, which is why Lugmayr et al. (2022) must

resample forward and backward to re-harmonise them. Here the situation is structurally different in two ways. First, missing information is inside the forward process: Eq. equation 13 defines a distinct chain for each mask M, and the network is conditioned on M through the input itself, so the state the decoder is in, is a state the network was trained on. Second, by Eq. equation 18 the reverse kernel at an unmasked site is the identity under the model itself. Clamping is therefore not a projection applied on top of the sampler; it is what the sampler already does, and no re-harmonisation step exists to be skipped. The same argument explains why ABSORB needs no extra hyperparameter in the role of RePaint's resampling count.

1.1.15 What the role and timestep embeddings actually contribute. A careful reader will notice that both conditioning signals in Eq. equation 5 are redundant. We state this rather than leave it to be discovered, because the correct architectural claim is slightly different and stronger than the one the redundancy might seem to undercut.

The role embedding is a reparameterisation, not added capacity. By definition $\mathcal{C}_t = \{g \in \mathcal{O} : x_t^g \neq \texttt{[MASK]}\}$, and every $g \in \mathcal{O}^{\mathsf{c}}$ carries `[MASK]` as well, so

$$\mathbf{1}[g \in \mathcal{C}_t] = \mathbf{1}\big[x_t^g \neq \texttt{[MASK]}\big], \tag{44}$$

a deterministic function of the value token at g. Adding $R(\mathbf{1}[g \in \mathcal{C}_t])$ to $E(x_t^g)$ is therefore equivalent to redefining $E(\texttt{[MASK]}) \leftarrow E(\texttt{[MASK]}) + R(0)$ and $E(b) \leftarrow E(b) + R(1)$ for $b \in \mathcal{B}$: the two parameterisations span the same function class. We keep R because separating "what value" from "is this position fixed" is the natural factorisation for the two variants that are not redundant: a three-way role that distinguishes candidate dropouts from artificially masked evidence, and a soft-clamping variant that permits evidence positions to be revised, and because it makes the computation graph legible. It buys no expressiveness in the configuration reported.

The timestep embedding is nearly redundant too. By Lemma 1, $|\mathcal{T}_t| \sim \mathrm{Binomial}(|\mathcal{O}|, 1-\overline{a}_t)$, so the network can estimate $1-\overline{a}_t$ from the input to within $O(|\mathcal{O}|^{-1/2})$ simply by counting `[MASK]` symbols, and t is then recoverable because $\overline{a}$ is strictly decreasing. What $\gamma(t)$ supplies is a noise-free, zero-cost version of a statistic the network would otherwise have to compute in its first attention layer. This is why the schedule-alignment mismatch of §1.1.13 is benign: the timestep is a convenience feature, and the load-bearing information; which positions are known and what they contain, is in the token stream. It also predicts that ABSORB should be insensitive to moderate misspecification of t_s at decoding time, which is consistent with $S = 24$ passes over a $T = 100$ schedule working at all.

The gene-identity embedding is not redundant. $P(g)$ is the only term that breaks permutation equivariance over the gene axis. Without it, two genes holding the same token at the same step would be exchangeable, and the model could not learn that gene g's token b means a different count than gene g''s token b. This is also why P is the one parameter block that grows with G (§1.4).

1.2 ABSORBING-CHAIN PROPERTIES

P1 (Closed-form marginal). $q(x_t \mid x_0) = \overline{a}_t \delta_{x_t, x_0} + (1-\overline{a}_t)\delta_{x_t, \mathsf{m}}$.

Proof. Lemma 1, §1.1.2.

P2 (Monotone absorption). The masked set $\mathcal{T}_t = \{g \in \mathcal{O} : x_t^g = \mathsf{m}\}$ is non-decreasing in t almost surely: $\mathcal{T}_{t-1} \subseteq \mathcal{T}_t$. Each site is erased exactly once, at the stopping time $\varsigma_g = \min\{t : x_t^g = \mathsf{m}\}$, whose law is $\Pr[\varsigma_g = t] = \overline{a}_{t-1} - \overline{a}_t$ with $\sum_{t=1}^{T} \Pr[\varsigma_g = t] = \overline{a}_0 - \overline{a}_T = 1$.

Proof. Row m of Q_t is $\mathbf{e}_{\mathsf{m}}^{\top}$ for every t, so once $x_t^g = \mathsf{m}$ the site remains m with probability one; hence the indicator $\mathbf{1}[x_t^g = \mathsf{m}]$ is non-decreasing in t and ς_g is well defined. By P1, $\Pr[\varsigma_g > t] = \Pr[x_t^g \neq \mathsf{m}] = \overline{a}_t$, and the difference gives the pmf. The total is 1 because $\overline{a}_T = 0$ (§1.1.6).

P3 (Two-point reverse posterior). For $x_0 \in \mathcal{B}$ the posterior $q(x_{t-1} \mid x_t, x_0)$ is supported on at most two states and is given by Eq. equation 15 with $\eta_t = (\overline{a}_{t-1} - \overline{a}_t)/(1-\overline{a}_t)$. This is the claim referenced from the main text.

Proof. Write $\bar{q}_{t-1}(\cdot \mid x_0)$ for the $(t-1)$-step marginal of P1. Two cases.

Case $x_t \neq \mathsf{m}$. By P2 the site was not yet erased at t, so $x_{t-1} \neq \mathsf{m}$; and since a non-mask state can only persist, $x_{t-1} = x_t = x_0$. Formally, $q(x_t \mid x_{t-1}) = 0$ unless $x_{t-1} = x_t$, so the numerator of Eq. equation 14 is supported on the single state x_t and the posterior is δ_{x_{t-1},x_t}, independent of the schedule.

Case $x_t = \mathsf{m}$. The predecessor is either the clean token, erased at this step, or already m. The two unnormalised weights are

$$\begin{aligned} x_{t-1} = x_0 :\ & q(\mathsf{m} \mid x_0)\ \bar{q}_{t-1}(x_0 \mid x_0) = \beta_t\, \overline{a}_{t-1}, \\ x_{t-1} = \mathsf{m} :\ & q(\mathsf{m} \mid \mathsf{m})\ \bar{q}_{t-1}(\mathsf{m} \mid x_0) = 1 - \overline{a}_{t-1}. \end{aligned} \tag{45}$$

No other predecessor has positive probability, because a state $v \notin \{x_0, \mathsf{m}\}$ has $\bar{q}_{t-1}(v \mid x_0) = 0$ by P1. The normaliser is

$$\beta_t \overline{a}_{t-1} + 1 - \overline{a}_{t-1} = (\overline{a}_{t-1} - \overline{a}_t) + 1 - \overline{a}_{t-1} = 1 - \overline{a}_t, \tag{46}$$

using $\overline{a}_t = \overline{a}_{t-1}(1 - \beta_t)$; it agrees with $q(x_t = \mathsf{m} \mid x_0) = 1 - \overline{a}_t$ from P1, as expected. Dividing gives η_t and $1 - \eta_t$.

P4 (Exact clamping). Under the x_0-parameterization of Eq. equation 17, if $x_t^g \neq \mathsf{m}$ then $p_\theta(x_{t-1}^g \mid \boldsymbol{x}_t) = \delta_{x_{t-1}^g, x_t^g}$ for every θ. Observed tokens are therefore fixed points of the reverse process, and the corresponding KL term in Eq. equation 22 is identically zero.

Proof. By P3 the mixture components $q(x_{t-1} \mid x_t, \tilde{x}_0)$ appearing in Eq. equation 17 all equal δ_{x_t} when $x_t \neq \mathsf{m}$; a convex combination of identical point masses is that point mass, and p_θ drops out. The KL against $q(x_{t-1} \mid x_t, x_0) = \delta_{x_t}$ is then zero.

P5 (Termination and one-shot unmasking). The reverse process restores each site exactly once, and after step $t = 1$ no evidence site is masked, since $\eta_1 = 1$ forces $x_0^g \neq \mathsf{m}$ with probability one.

Proof. By P4 an unmasked site never changes again. At $t = 1$, Eq. equation 15 with $\eta_1 = (\overline{a}_0 - \overline{a}_1)/(1 - \overline{a}_1) = 1$ places all mass on x_0.

P6 (Closed, bounded, non-negative support). Every state of both the forward and the reverse chain lies in $\mathcal{V}$, and every count produced by Eq. equation 9 lies in $[0, r_{g,B-1}]$.

Proof. The chains are Markov chains on the finite set $\mathcal{V}$, so the first claim is immediate: unlike a continuous process there is no state to leave. The second is Eq. equation 37: $\hat{c}^g$ is a convex combination of non-negative, non-decreasing representatives.

P7 (Vanishing prior term). $L_T = 0$ if and only if $\overline{a}_T = 0$: otherwise $L_T = \infty$ for the point-mass prior.

Proof. $q(\boldsymbol{x}_T \mid \boldsymbol{x}_0)$ places mass $\overline{a}_T$ on $x_0 \neq \mathsf{m}$ while $p(\boldsymbol{x}_T)$ places none, so the KL is finite only when $\overline{a}_T = 0$, in which case the two coincide.

P8 (Hazard identity). $\eta_t(1 - \overline{a}_t) = \overline{a}_{t-1} - \overline{a}_t$ and $\prod_{u=t'+1}^{t}(1 - \eta_u) = (1 - \overline{a}_{t'})/(1 - \overline{a}_t)$.

Proof. The first is Eq. equation 16 cleared of its denominator. For the second, $1 - \eta_u = (1 - \overline{a}_{u-1})/(1 - \overline{a}_u)$, and the product telescopes, which is Eq. equation 40.

P9 (Exchangeability of the corruption). Conditioned on $|\mathcal{T}_t| = k$, the masked set is uniformly distributed over the $\binom{|\mathcal{O}|}{k}$ subsets of $\mathcal{O}$ of size k.

Proof. The Bernoulli thinning of Lemma 1 is i.i.d. across sites, so all subsets of a given size are equally likely.

Properties P2 and P4 combined makes confidence ordering meaningful: commitment is permanent, so the decision of which site to commit first is a real decision rather than a temporary assignment that a later step may overwrite. This is the structural difference from birth/death count diffusions, whose reverse dynamics adjust every site at every step (Soatto et al., 2026): in such a process the notion of "committing the most confident gene" has no operational meaning, because nothing is committed. P9 is what allows the training-time mask to be described entirely by its size, and it is also the precise sense in which the training corruption is missing-completely-at-random.

1.3 Algorithm Pseudocode

The pipeline has five stages: estimate the observation mask (Algorithm 1), fit the tokenizer (Algorithm 2), build the schedule (Algorithm 3), train the denoiser (Algorithm 4), and decode (Algorithm 5). Stages 1–3 are fitted on training cells only.

Algorithm 1 EstimateMask: candidate dropouts

Input: counts $C \in \mathbb{N}^{N\times G}$; optional cell-type labels y; neighbourhood size k; Leiden resolution ϱ
Output: observation mask $M \in \{0,1\}^{N\times G}$

if labels y are available **then**
  $\mathcal{K} \leftarrow$ partition of cells induced by y
else
  $\tilde{C} \leftarrow \log\big(1+\mathrm{CPM}(C)\big)$; $Z \leftarrow \mathrm{PCA}_{50}(\tilde{C})$
  build the k-NN graph on Z (Euclidean)
  $\mathcal{K} \leftarrow \mathrm{Leiden}(\text{graph}, \varrho)$ (Traag et al., 2019)
end if
$M \leftarrow \mathbf{1}^{N\times G}$ {default: every entry is trusted}
for each cluster $K \in \mathcal{K}$ **do**
  for each gene $g = 1, \dots, G$ **do**
    if $\max_{i\in K} C_{ig} > 0$ **then**
      $M_{ig} \leftarrow 0$ for every $i \in K$ with $C_{ig} = 0$ {candidate dropout}
    end if
    {else: silent across the cluster, keep $M_{ig} = 1$}
  end for
end for
return M

Algorithm 2 FitTokenizer: per-gene equal-mass bins

Input: training counts C^{tr}; bin count B
Output: edges $e_{g,1:B-2}$, representatives $r_{g,0:B-1}$

for each gene $g = 1, \dots, G$ **do**
  $Z_g \leftarrow \{C^{\mathrm{tr}}_{ig} : C^{\mathrm{tr}}_{ig} > 0\}$
  if $Z_g = \emptyset$ **then**
    $e_{g,\cdot} \leftarrow 0$; $r_{g,\cdot} \leftarrow 0$; **continue**
  end if
  for $j = 1, \dots, B-2$ **do**
    $e_{g,j} \leftarrow \mathrm{Quantile}\big(Z_g,\ j/(B-1)\big)$
  end for
  $r_{g,0} \leftarrow 0$ {the exact-zero token}
  for $b = 1, \dots, B-1$ **do**
    $S_{g,b} \leftarrow \{c \in Z_g : \tau_g(c) = b\}$ {Eq. equation 1}
    if $S_{g,b} \neq \emptyset$ **then**
      $r_{g,b} \leftarrow \mathrm{median}(S_{g,b})$
    else
      $r_{g,b} \leftarrow r_{g,b-1}$ {empty bin: keep $r_{g,\cdot}$ non-decreasing}
    end if
  end for
end for
return $\{e_{g,\cdot}\}, \{r_{g,\cdot}\}$

1.4 Computational Complexity

Parameters. With L pre-norm blocks of width d, h heads, MLP ratio 4, vocabulary $|\mathcal{V}| = B+1$, and G genes, the parameter count is

$$\underbrace{(B{+}1)d}_{E} + \underbrace{Gd}_{P} + \underbrace{2d}_{R} + \underbrace{8d^2}_{\gamma\text{-MLP}} + \underbrace{12Ld^2}_{\text{blocks}} + \underbrace{d(B{+}1)}_{\text{head}}, \tag{47}$$

Algorithm 3 COSINESCHEDULE

Input: steps T; offset $\varepsilon_{\mathrm{o}} = 0.008$
Output: $\{\overline{a}_t\}_{t=0}^{T}$, $\{\beta_t\}_{t=1}^{T}$, $\{\eta_t\}_{t=1}^{T}$

$f(t) \leftarrow \cos^2\big(\frac{\pi}{2} \cdot \frac{t/T+\varepsilon_{\mathrm{o}}}{1+\varepsilon_{\mathrm{o}}}\big)$ for $t = 0, \ldots, T$
$\overline{a}_t \leftarrow f(t)/f(0)$ $\{\overline{a}_0 = 1, \overline{a}_T = 0\}$
for $t = 1, \ldots, T-1$ **do**
    $\beta_t \leftarrow \min\big(1 - \overline{a}_t/\overline{a}_{t-1},\ 0.999\big)$
end for
$\beta_T \leftarrow 1$ {exempt from clipping so that $L_T = 0$; P7}
$\eta_t \leftarrow (\overline{a}_{t-1} - \overline{a}_t)/(1 - \overline{a}_t)$
return $\{\overline{a}_t\}, \{\beta_t\}, \{\eta_t\}$

Algorithm 4 ABSORB: training

Input: tokens $X_0 = \tau(C^{\mathrm{tr}})$; mask M; schedule $\{\overline{a}_t\}$; denoiser f_θ; iterations N_{it}; batch n_b; EMA rate μ
Output: EMA weights $\bar{\theta}$

initialise θ; zero-initialise the output head; $\bar{\theta} \leftarrow \theta$
for it $= 1, \ldots, N_{\mathrm{it}}$ **do**
    sample a minibatch $\mathcal{I}$ of n_b cells
    for each cell $i \in \mathcal{I}$ **do**
        $\mathcal{O}_i \leftarrow \{g : M_{ig} = 1\}$
        repeat
            $t_i \sim \mathcal{U}\{1, \ldots, T\}$
            $u^g \sim \mathrm{Bernoulli}(\overline{a}_{t_i})$ for each $g \in \mathcal{O}_i$
            $\mathcal{C}_i \leftarrow \{g \in \mathcal{O}_i : u^g = 1\}$; $\mathcal{T}_i \leftarrow \mathcal{O}_i \setminus \mathcal{C}_i$
        until $\mathcal{T}_i \neq \emptyset$ {needed for the $1/|\mathcal{T}_i|$ average}
        $x_t^g \leftarrow X_{0,ig}$ for $g \in \mathcal{C}_i$, else `[MASK]` {$g \notin \mathcal{O}_i$ stays `[MASK]`}
        $\rho^g \leftarrow \mathbf{1}[g \in \mathcal{C}_i]$ {role flags, Eq. equation 5}
    end for
    $Z \leftarrow f_\theta\big(\{x_t\}, \{\rho\}, \{t_i\}\big)$ {logits, $n_b \times G \times |\mathcal{V}|$}
    drop the `[MASK]` column of Z; $p \leftarrow \mathrm{softmax}(Z)$ over $\mathcal{B}$
    $\mathcal{L} \leftarrow \frac{1}{n_b} \sum_{i \in \mathcal{I}} \frac{1}{|\mathcal{T}_i|} \sum_{g \in \mathcal{T}_i} -\log p_{ig}\big(X_{0,ig}\big)$ {Eq. equation 7}
    backpropagate; clip $\|\nabla_\theta \mathcal{L}\|_2$ to 1.0
    AdamW step with warmup-then-cosine learning rate
    $\bar{\theta} \leftarrow \mu\,\bar{\theta} + (1-\mu)\,\theta$
end for
return $\bar{\theta}$

where each block contributes $4d^2$ for the attention projections and $8d^2$ for the MLP. At the default $B = 256$, $d = 128$, $L = 6$, $G = 1000$ this is ≈ 1.5M parameters; the gene-identity table P is the only term that grows with G, adding 0.38M at $G = 4000$. Note that the model is small by current standards, and deliberately so: each dataset is fitted from scratch on N cells with N in the hundreds to thousands, so capacity is bounded by data, not by compute.

Time. One forward pass over one cell costs

$$L\big(\ \underbrace{12Gd^2}_{\text{projections, MLP}} + \underbrace{2G^2d}_{\text{attention}}\ \big) \text{ multiply–accumulates,} \tag{48}$$

which at $G = 1000$, $d = 128$, $L = 6$ is ≈ 2.7 GMAC, of which the attention term is already the majority (1.5 of 2.7). Training performs $N_{\mathrm{it}}\, n_b$ such passes plus backward (roughly $2\times$ forward), so 15,000 steps at $n_b = 64$ amount to ≈ 7.8 PMAC, consistent with the 15–20 minutes reported in the main text on a single GPU. Decoding costs exactly S passes per cell: $S = 24$ gives ≈ 65 GMAC per cell, or ≈ 33 TMAC for a 500×1000 matrix, consistent with the reported 1–2 minutes. The comparison that matters is against a full ancestral chain: that would require $T = 100$ passes, so confidence decoding is $T/S \approx 4.2\times$ cheaper here, and the ratio grows with T since S need not

Algorithm 5 ABSORB: imputation by confidence decoding

Input: one cell's tokens $\boldsymbol{x}_0$ and counts c; mask M; weights $\bar{\theta}$; passes S; steps T; representatives r
Output: imputed counts $\hat{c} \in \mathbb{R}^G_{\geq 0}$

$\mathcal{O} \leftarrow \{g : M_g = 1\}$; $\mathcal{T} \leftarrow \{g : M_g = 0\}$; $U \leftarrow \mathcal{T}$
$x^g \leftarrow x_0^g$ for $g \in \mathcal{O}$, else `[MASK]`; $\rho^g \leftarrow \mathbf{1}[g \in \mathcal{O}]$
$P[g] \leftarrow \perp$ for all $g \in \mathcal{T}$ {cached commit-time distributions}
for $s = 1, \ldots, S$ **do**
  $r_s \leftarrow s/S$; $t_s \leftarrow \max\big(1,\ \lfloor (1 - r_s)(T - 1) \rceil\big)$
  $Z \leftarrow f_{\bar{\theta}}(x, \rho, t_s)$; drop the `[MASK]` column
  $p^g \leftarrow \mathrm{softmax}(Z^g)$ over $\mathcal{B}$, for $g \in U$
  $\hat{b}^g \leftarrow \arg\max_b p_b^g$; $\kappa^g \leftarrow \max_b p_b^g$
  $n_s \leftarrow \big\lfloor \cos(\frac{\pi}{2} r_s)\, |\mathcal{T}| \big\rfloor$ {Eq. equation 8}
  $k_s \leftarrow \max\big(|U| - n_s,\ 1\big)$; $k_s \leftarrow \min(k_s, |U|)$
  $A \leftarrow$ the k_s genes of U with largest κ^g {Proposition 5}
  for each $g \in A$ **do**
    $x^g \leftarrow \hat{b}^g$; $\rho^g \leftarrow 1$; $P[g] \leftarrow p^g$; $U \leftarrow U \setminus \{g\}$
  end for
end for
$\hat{c}^g \leftarrow \sum_{b \in \mathcal{B}} P[g]_b\, r_{g,b}$ for $g \in \mathcal{T}$ {Eq. equation 9}
$\hat{c}^g \leftarrow c^g$ for $g \in \mathcal{O}$ {measurements returned unchanged}
return $\hat{c}$

scale with it. Against the one-site-at-a-time alternative, which by Proposition 3 would be exact, the ratio is $|\mathcal{T}|/S \approx 700/24 \approx 29$ at $G = 1000$ and 70% dropout, and it grows linearly with panel size.

Memory. The binding constraint is the attention score tensor, which holds $n_b\, h\, G^2$ entries per block. At $G = 1000$, $h = 8$, $n_b = 64$ and half precision this is ≈ 1.0 GB per block if scores are materialised and retained for the backward pass; at $G = 4000$ the same expression gives ≈ 16 GB, which is why the batch size is halved to 32 on the 4000-gene datasets. Memory-efficient attention removes the G^2 term from the activation budget but not from the $O(G^2 d)$ time in Eq. equation 48, and the quadratic gene-axis cost remains the limit on panel size noted in the conclusion.

Preprocessing. Mask estimation costs $O(Nkd_{\mathrm{pca}})$ for the approximate k-NN graph plus near-linear Leiden refinement, and $O(|\mathcal{K}|G)$ for the per-cluster scan. Tokenizer fitting costs $O(G\, N \log N)$ with sorting, or $O(GN)$ with selection, and is performed once. Both are negligible next to training.

1.5 ASSUMPTIONS, AND WHERE EACH IS USED

The derivations above are exact, given a small number of assumptions. We list them below:

(A1) The observation mask is correct. Everything is conditional on M (§1.1.3). If the cluster heuristic marks a biological zero as a candidate dropout, ABSORB will impute a positive value there; if it marks a technical zero as biological, that entry is frozen at zero and used as evidence. The method has no mechanism to recover from either error, and the conclusion already names joint learning of M as the natural next step.

(A2) Training corruption is MCAR; real dropout is not. By P9 the training mask is an i.i.d. thinning of $\mathcal{O}$, whereas real dropout is biased toward low counts. The learned conditional is therefore matched to the wrong missing information mechanism at inference. This is a testable prediction, and the MNAR25 protocol of §Real Data: Held-Out Recovery via Masking is the test: it is the setting in which ABSORB's margin is smallest, and on Bone Marrow Mesenchyme it is where the method is beaten. A low-count-biased corruption during training is therefore the remedy.

(A3) The conditional transfers from evidence positions to dropout positions. Training never supplies a target at $g \in \mathcal{O}^{\mathrm{c}}$ (§1.1.3), so transfer rests on the assumption that $p(x^g \mid \boldsymbol{x}^{\mathcal{C}})$ is the same function at dropout positions as at artificially masked evidence positions. What makes the assumption repre-

sentable is that context and target are distinguished only by whether the token is `[MASK]` (§1.1.15), so an intermediate decoding state is structurally indistinguishable from a training state. Nothing about the decoder's trajectory is out of distribution; only the identity of the positions is.

(A4) Clustering for the mask is transductive. Algorithm 1 runs on the full count matrix, including the cells whose entries are later scored. No held-out value is used: the held-out truth in $\mathcal{H}$ is never an input, and entries hidden by the MCAR50/MNAR25 protocols become zeros that the same heuristic re-detects as candidate dropouts, similar to the way it treats genuine zeros, but the clustering does see the test cells' observed profiles. This is standard for imputation, where all cells are present at fit time by construction; we state it rather than leave it implicit. The tokenizer, by contrast, is fitted on training cells only (§1.1.12(vi)).

(A5) The reverse model factorizes over genes. This makes parallel decoding possible and is the source of both the ELBO gap (Eq. equation 23) and the decoding error (Eq. equation 42). It is an approximation, quantified exactly as the total correlation of each committed block, and it is removable at the cost of $|\mathcal{T}|$ network evaluations (Proposition 3).

(A6) The bound is reweighted, not optimised as stated. Eq. equation 7 is the family member of Eq. equation 29 with weight $1/(T|\mathcal{T}_t|)$, which shares the population optimum with the exact bound (Proposition 1) and equals it, in continuous time, for the linear-$\bar{\alpha}$ schedule (Corollary 2).

2. Additional Experimental Results

2.2 Sensitivity to the number of quantile bins

Table 2: Effect of quantile bin count B on held-out recovery (Sp. ρ, l-RMSE) and whole-matrix distributional metrics (ED, MMD, SW) on simulated D1 (71.9% dropout) and PBMC68k (MCAR 50%). This sweep was run separately from the main experiments and its distributional metrics are computed on the whole imputed matrix, so absolute values are not comparable to Figs. 4–5; compare trends across B within each panel only. Sp. ρ: Spearman correlation; l-RMSE: log-scale RMSE; ED: energy distance; MMD: maximum mean discrepancy; SW: sliced Wasserstein distance.

B	Sp. $\rho\uparrow$	l-RMSE $\downarrow$	ED $\downarrow$	MMD $\downarrow$	SW $\downarrow$
Simulated Dataset D1 (71.9% dropout)					
64	0.921	0.611	4378	0.1488	28.38
128	0.924	0.607	4421	0.1489	27.87
256	0.925	0.596	4239	0.1480	31.38
512	0.927	0.592	4339	0.1488	32.28
1024	0.927	0.591	4333	0.1485	31.26
PBMC68k Dataset (MCAR 50%)					
64	0.1950	0.4983	0.1264	0.0026	0.0776
128	0.1971	0.4870	0.1265	0.0027	0.0790
256	0.1970	0.4777	0.1255	0.0026	0.0730
512	0.1971	0.4775	0.1253	0.0028	0.0742
1024	0.1969	0.4774	0.1255	0.0028	0.0740

Table 2 varies the vocabulary size B over a 16-fold range. On D1, Spearman ρ rises from 0.921 at $B = 64$ to 0.927 at $B = 512$ and does not change at $B = 1024$; log-RMSE decreases slightly (0.611 to 0.591). The distributional metrics show no consistent trend with B: MMD varies by less than 1%, ED fluctuates without direction, and SW is lowest at small B. On PBMC68k, ρ stays within 0.195–0.197 and log-RMSE levels off from $B = 256$ onward. Our default $B = 256$ lies on this plateau: it is within 0.002 in ρ and 0.005 in log-RMSE of the best value on D1, within 0.0003 on both measures on PBMC68k, and gives the lowest ED and MMD on D1 and the lowest SW on PBMC68k. Accuracy is therefore largely insensitive to B, consistent with quantization not being the binding constraint. For count data, ties among small integers cap the effective vocabulary B_g^{eff} well below B (§1.1.12(iii)), so additional bins are largely unreachable. PBMC68k is available only in normalised form, where this tie argument does not apply; there the flat response reflects the low recoverable signal on this dataset.

2.3 Heldout Recovery for Dentate Gyrus and Human Brain Dataset

This section highlights the heldout recovery performance on the additional Human Brain dataset. For this, performance is evaluated under both Missing Completely At Random (MCAR) 50% and Missing Not At Random (MNAR) 25% mechanisms. The heatmaps evaluate the comparative performance of various imputation methods across multiple axes of recovery, including point-estimate errors (RMSE, MAE), rank preservation (Spearman correlation), and distributional fidelity (Wasserstein distance, MMD). The proposed method demonstrates robust and balanced recovery, preserving the underlying structural distribution and ranking order even against severe artificial dropout. (Note: Visualizations for the Dentate Gyrus dataset will be appended to this section once generated.)

MCAR 50%

	ρ ↑	RMSE ↓	ED ↓	MMD ↓	SW ↓
ABSORB	**0.5797**	**2.5430**	**366.7430**	0.0129	**107.3130**
CountsDiff	0.2419	3.7734	6.21e+03	0.1383	450.5950
MAGIC	0.5586	3.5176	1.88e+03	0.0593	212.9900
SAVER	0.4044	4.7551	1.93e+03	0.0603	215.4610
scImpute	0.5022	2.8859	1.36e+03	0.0327	232.2580
ALRA	0.3320	3.1152	1.24e+03	0.0291	265.7310
scIGANs	0.0840	3.4719	1.62e+03	0.0531	200.8940
DeepImpute	0.4383	2.7658	488.2690	**0.0039**	118.4370
DCA	0.5340	2.9830	589.9590	0.0226	128.0720

(a) Human Brain: MCAR 50%

MNAR 25%

	ρ ↑	RMSE ↓	ED ↓	MMD ↓	SW ↓
ABSORB	**0.5486**	**2.5467**	149.0430	0.0030	33.5252
CountsDiff	0.1872	3.5079	729.9250	0.0175	123.9660
MAGIC	0.5329	3.1360	179.9446	0.0016	45.7195
SAVER	0.4120	4.3922	185.3046	0.0017	47.3162
scImpute	0.4766	2.8407	386.2550	0.0081	99.3342
ALRA	0.4361	2.9367	153.7404	0.0010	53.0462
scIGANs	0.1175	3.3755	322.6950	0.0045	81.3325
DeepImpute	0.4251	2.5771	126.1038	**2.00e-04**	30.1818
DCA	0.5112	2.9225	**125.1901**	5.00e-04	**26.1733**

(b) Human Brain: MNAR 25%

Figure 7: Heldout recovery heatmaps for the Human Brain dataset under MCAR 50% and MNAR 25% artificial dropout. The heatmaps depict multifaceted imputation performance, highlighting the proposed method's robust recovery capabilities against severe missingness.

2.4 Clustering Summary Heatmaps

The following heatmaps provide an aggregated view of clustering performance across multiple evaluation criteria, including K-means ARI, Spectral ARI, AMI, FMI, and Bootstrap Stability. On the Human Brain dataset, ABSORB and countsdiff emerge as the leading methods, both significantly outperforming the raw data and other baselines across all metrics. Similarly, on the Dentate Gyrus dataset, ABSORB demonstrates robust performance: particularly excelling in Spectral ARI (0.902) and maintaining highly competitive scores across K-means, Ward hierarchical clustering, and information-theoretic metrics. Together, these heatmaps confirm that the proposed method consistently ranks at or near the top across diverse clustering topologies for both additional datasets.

Human Brain

	KM ↑	Ward ↑	Spec. ↑	AMI ↑	FMI ↑	V-m. ↑	Boot. ↑
Raw	0.678	0.757	0.514	0.768	0.764	0.777	0.673
ABSORB	**0.764**	**0.943**	0.926	**0.870**	**0.817**	**0.874**	0.739
scIDPMs	0.431	0.277	**0.967**	0.403	0.513	0.425	0.545
countsdiff	0.708	0.730	0.600	0.812	0.790	0.818	0.725
MAGIC	0.637	0.781	0.723	0.698	0.689	0.709	0.647
SAVER	0.681	0.732	0.485	0.743	0.754	0.752	0.673
scImpute	0.744	0.754	0.773	0.804	0.810	0.811	**0.741**
ALRA	0.673	0.829	0.710	0.781	0.766	0.789	0.675
scIGANs	0.476	0.594	0.529	0.547	0.612	0.562	0.476
DeepImpute	0.695	0.646	0.733	0.784	0.786	0.791	0.693
DCA	0.700	0.755	0.758	0.779	0.776	0.786	0.716

(a) Human Brain

Dentate Gyrus

	KM ↑	Ward ↑	Spec. ↑	AMI ↑	FMI ↑	V-m. ↑	Boot. ↑
Raw	0.465	0.436	0.630	0.658	0.501	0.662	0.451
ABSORB	**0.662**	**0.757**	**0.902**	**0.834**	**0.757**	**0.836**	**0.664**
countsdiff	0.635	0.682	0.888	0.810	0.730	0.811	0.624
MAGIC	0.465	0.460	0.633	0.659	0.505	0.663	0.457
SAVER	0.389	0.378	0.606	0.584	0.418	0.589	0.378
scImpute	0.555	0.560	0.524	0.723	0.572	0.726	0.562
ALRA	0.474	0.501	0.576	0.658	0.524	0.662	0.472
scIGANs	0.464	0.481	0.667	0.659	0.510	0.662	0.466
DeepImpute	0.438	0.425	0.609	0.641	0.501	0.645	0.435
DCA	0.476	0.477	0.637	0.681	0.547	0.684	0.471

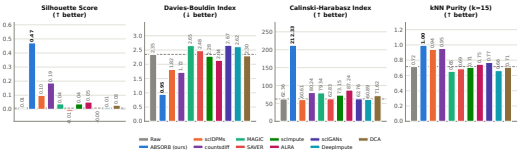
(b) Dentate Gyrus

Figure 8: Clustering summary heatmaps for (a) Human Brain and (b) Dentate Gyrus.

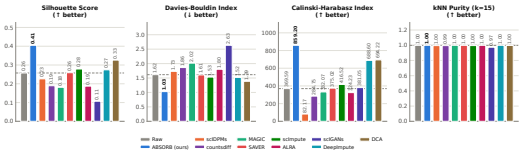
2.5 Comprehensive Clustering Evaluation

This section evaluates clustering performance and latent-space geometry across all five datasets (Bone Marrow, VISp, PBMC68k, Human Brain, and Dentate Gyrus) along seven complemen-

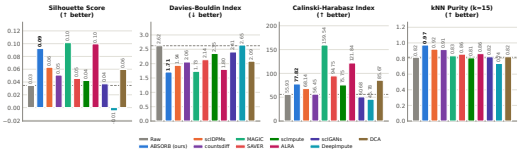
tary axes: cluster quality (Silhouette, Davies-Bouldin, Calinski-Harabasz, kNN purity), within-to-between cluster distance ratio, k-means stability across random initialisations, Leiden resolution sweeps, kNN purity sweeps over neighbourhood size, bootstrap stability under cell resampling, and within-cluster variance ratio. All methods, including the unimputed Raw baseline, are embedded identically before scoring, so differences reflect the imputation and not the downstream pipeline.



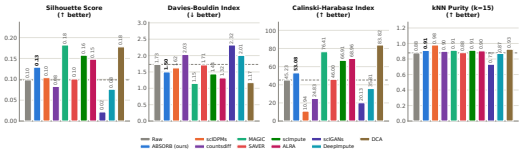
(a) Bone Marrow



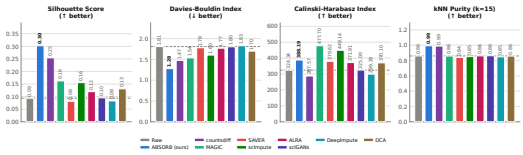
(b) VISp



(c) PBMC68k



(d) Human Brain



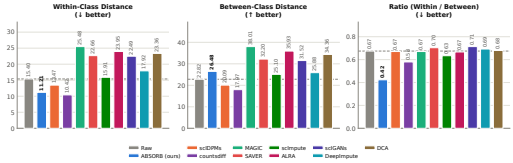
(e) Dentate Gyrus

Figure 9: **Cluster quality of the post-imputation embedding, all five datasets.** Every method, including the unimputed Raw baseline, passes through the identical pipeline of Section 4.2 (clip at 0, $\log_{1p}$, per-gene z-score, PCA to 50 components) and is scored against the reference labels of Table 4. Silhouette and kNN purity ($k = 15$) are higher-is-better; Davies-Bouldin is lower-is-better. ABSORB delivers the largest gains on the three datasets whose reference partition is fine-grained and dropout-limited. On **Bone Marrow** it reaches Silhouette 0.471 against 0.187 (countsdiff), 0.099 (scIDPMs) and 0.008 (Raw), while Davies-Bouldin falls from 2.35 to 0.95 and Calinski-Harabasz rises from 62 to 212; kNN purity is 0.995 against 0.953 (countsdiff) and 0.716 (Raw). On **Dentate Gyrus** it reaches Silhouette 0.302 against 0.254 (countsdiff) and 0.094 (Raw), with kNN purity 0.993 against 0.989 and 0.859. On **VISp** it reaches 0.405 against 0.326 (DCA) and 0.257 (Raw), with Calinski-Harabasz 859 against 694 and 370. The two exceptions are the two smallest matrices, where heavy kernel smoothing compresses within-cluster spread most effectively: on **PBMC68k** MAGIC takes Silhouette (0.102 against 0.092 for ABSORB, both above Raw at 0.035) although ABSORB still leads kNN purity at 0.973 against 0.915 (countsdiff) and 0.816 (Raw); and on **Human Brain** MAGIC (0.182) and DCA (0.178) lead Silhouette against 0.129 for ABSORB, and scIDPMs leads kNN purity at 0.980 against 0.910, with ABSORB fifth. Note that the VISp and PBMC68k panels are scored at coarse granularity (4 major classes and 11 Louvain clusters respectively); the finer 23-subclass view of VISp, where the ranking inverts, is reported in Figures 11, 12 and 13.

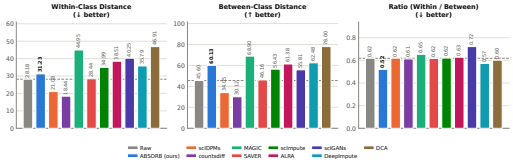
3. Reproducibility Details

3.1 Full Hyperparameters

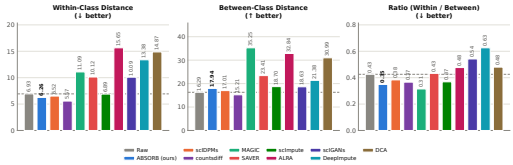
We use $B = 256$ quantile bins ($|\mathcal{V}| = 257$), a cosine schedule with $T = 100$, and $L = 6$ pre-norm Transformer encoder blocks of width $d = 128$ with 8 attention heads and MLP ratio 4. Training uses AdamW (learning rate 5×10^{-4}, weight decay 10^{-4}) for 15,000 steps at batch size 64 (32 on the



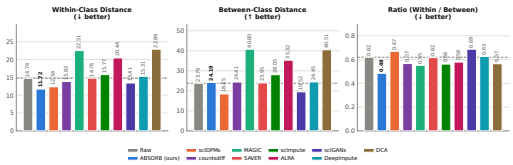
(a) Bone Marrow



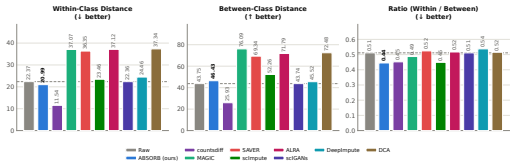
(b) VISp



(c) PBMC68k

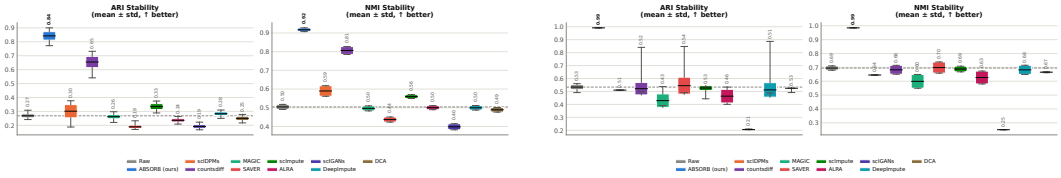


(d) Human Brain



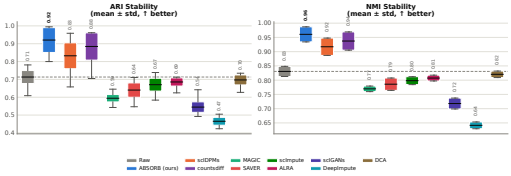
(e) Dentate Gyrus

Figure 10: **Within-to-between cluster distance ratio, all five datasets.** Each panel reports the mean pairwise distance between cells of the same reference label, the mean distance between all cells, and their ratio, computed in the shared 50-PC embedding. A lower ratio means clusters are tight relative to how far apart they sit. ABSORB attains the lowest ratio on four of the five datasets: Bone Marrow 0.423 (countsdiff 0.580, scImpute 0.634, Raw 0.675), VISp 0.519 (DeepImpute 0.573, DCA 0.601, Raw 0.618), Human Brain 0.484 (MAGIC 0.552, scImpute 0.562, Raw 0.620), and Dentate Gyrus 0.445 (scImpute 0.449, countsdiff 0.452, Raw 0.511). On PBMC68k, MAGIC is lowest at 0.315 against 0.349 for ABSORB, with both well below Raw at 0.426. The two underlying columns show that ABSORB and the smoothing methods reach a low ratio by opposite routes. On VISp, ABSORB *increases* the within-class distance from 28.2 to 31.2 while pushing the between-class distance from 45.6 to 60.1, so the gain comes from separating cell types rather than from collapsing them; MAGIC instead inflates both terms (45.0 and 68.9) and ends with a worse ratio than Raw. On Bone Marrow ABSORB improves both terms at once, tightening within-class distance from 15.4 to 11.2 while widening between-class distance from 22.8 to 26.5. This distinction matters because a method that mapped every cell to its cluster centroid would also score well here, and is the reason this figure should be read alongside the kNN purity sweep of Figure 13.

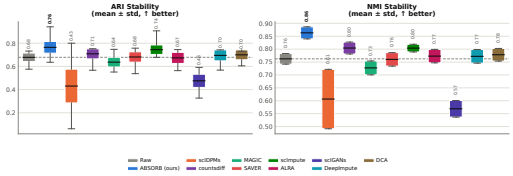


(a) Bone Marrow

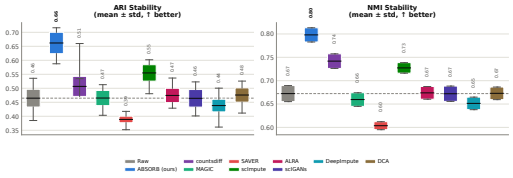
(b) VISp



(c) PBMC68k

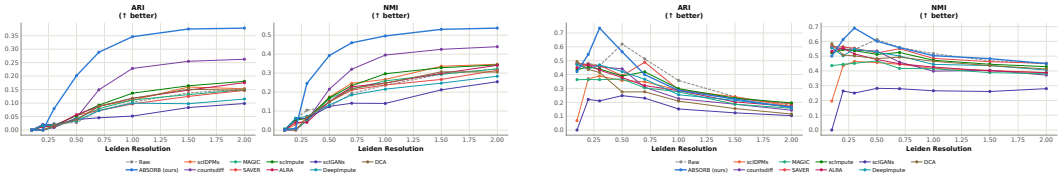


(d) Human Brain



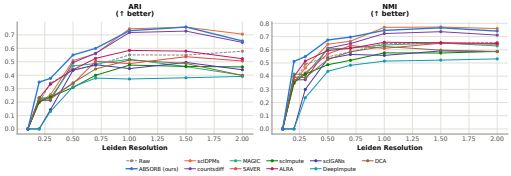
(e) Dentate Gyrus

Figure 11: **k-means stability across random initialisations, all five datasets.** k-means is run 50 times from different random seeds with k set to the number of reference labels; each box is the resulting ARI distribution, so the median measures accuracy and the box width measures how much the partition depends on initialisation. ABSORB is the most accurate and among the tightest on every dataset scored at its native granularity. **Bone Marrow** ($k = 19$): 0.843 ± 0.030 (range 0.772–0.900, NMI 0.917) against countsdiff 0.655 ± 0.041, scImpute 0.335, and Raw 0.269 ± 0.014; this is the largest margin over the strongest baseline anywhere in the study. **Dentate Gyrus** ($k = 14$): 0.662 ± 0.038 against scImpute 0.555, countsdiff 0.507, and Raw 0.465. **Human Brain** ($k = 8$): 0.764 ± 0.050 against scImpute 0.744, countsdiff 0.708, DCA 0.700, and Raw 0.678, while scIDPMs is markedly unstable at 0.431 ± 0.146. **PBMC68k**: against the 11-cluster Louvain reference ABSORB reaches 0.920 ± 0.069 versus countsdiff 0.884, scIDPMs 0.832, and Raw 0.713; against the coarser 10-way bulk-label annotation every method collapses into the band 0.34–0.49 and ABSORB (0.474) is indistinguishable from Raw (0.489). **VISp** shows the sharpest granularity effect and we report both levels rather than only the favourable one. At the 4-class level ABSORB attains ARI 0.991 with *zero* variance across all 50 seeds, against 0.534 for Raw, the single largest margin we observe; at the 23-subclass level it falls to 0.338 while Raw reaches 0.774 and SAVER 0.775. The embedding is therefore merging transcriptomically adjacent subclasses into their parent class: excellent when the question is which broad type a cell belongs to, actively harmful when the question is which subtype. The same caveat governs Figures 12 and 13, and any single-number clustering claim about VISp must state which granularity it refers to.

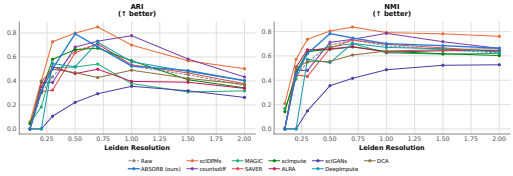


(a) Bone Marrow

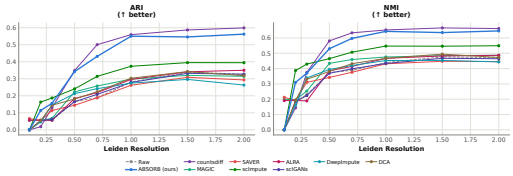
(b) VISp



(c) PBMC68k

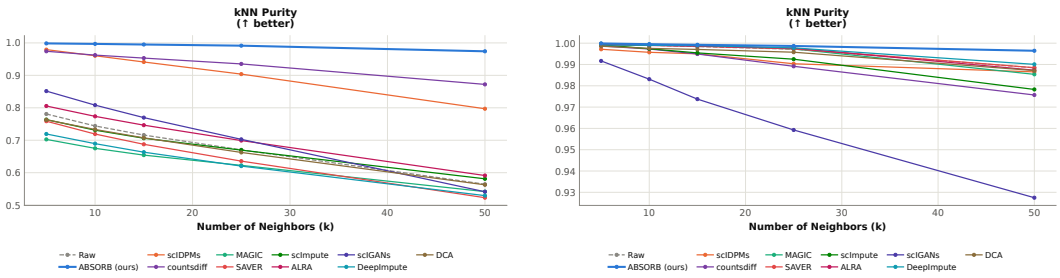


(d) Human Brain



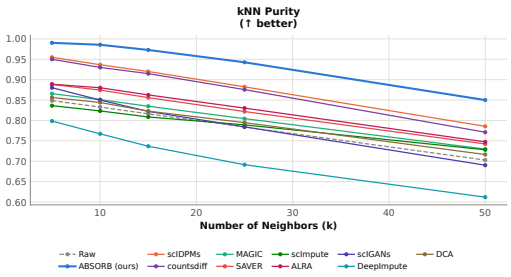
(e) Dentate Gyrus

Figure 12: **Leiden resolution sweep, all five datasets.** Leiden community detection is run on the shared embedding at resolutions $\{0.1, 0.2, 0.3, 0.5, 0.7, 1.0, 1.5, 2.0\}$ and scored by ARI against the reference labels. Unlike k-means this fixes no cluster count, so the curve tests whether a method encodes the right structure at *some* scale rather than only at the scale it was handed. Peak ARI and the resolution at which it occurs: **Bone Marrow**, ABSORB 0.378 at $\rho = 2.0$ (25 communities against 19 reference clusters), then countsdiff 0.262, scImpute 0.181, ALRA 0.176, and Raw 0.152, so ABSORB is $2.5\times$ Raw and $1.4\times$ the best baseline. **PBMC68k** (Louvain reference), ABSORB 0.759 at $\rho = 1.5$, recovering exactly 11 communities, narrowly ahead of scIDPMs 0.757 and countsdiff 0.728, with Raw at 0.578; against the bulk-label reference the ordering reverses and Raw peaks highest at 0.430 against 0.344 for ABSORB. **Human Brain**, scIDPMs 0.849 at $\rho = 0.7$ (recovering exactly 8 communities), then ABSORB 0.793 at $\rho = 0.5$, countsdiff 0.777, and Raw 0.708. **Dentate Gyrus**, countsdiff 0.599 at $\rho = 2.0$, then ABSORB 0.562, scImpute 0.395, and Raw 0.338, so the two count-native diffusion models are clearly separated from the rest of the panel. **VISp** reproduces the granularity inversion of Figure 11: at the class level ABSORB peaks highest at 0.735 ($\rho = 0.3$, 7 communities) against 0.620 for Raw, but at the subclass level it peaks at only 0.299 against 0.477 for DeepImpute and 0.472 for scImpute. Across datasets the peak consistently falls where the recovered community count is closest to the reference count, which is why the methods that over-merge peak early and low.

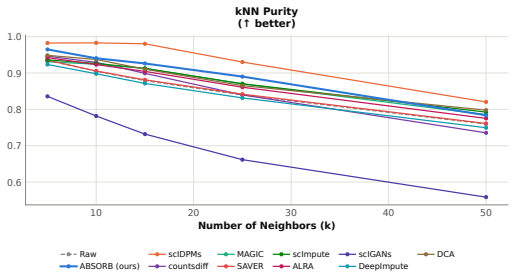


(a) Bone Marrow

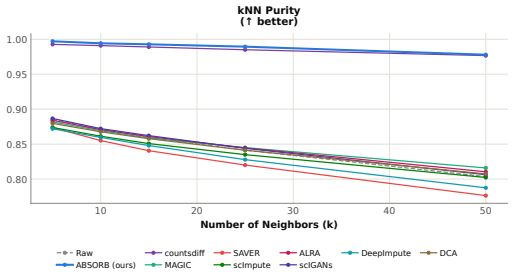
(b) VISp



(c) PBMC68k

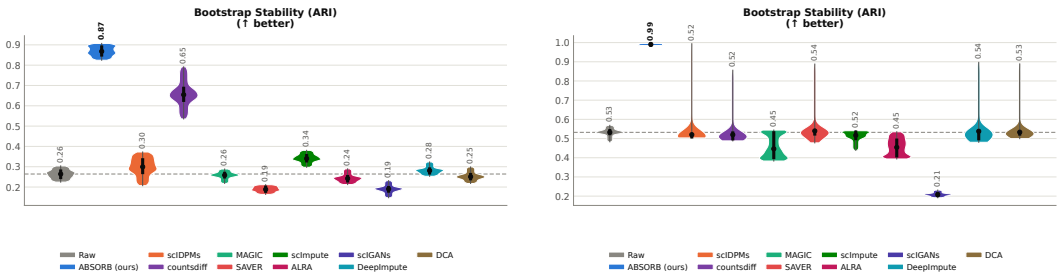


(d) Human Brain



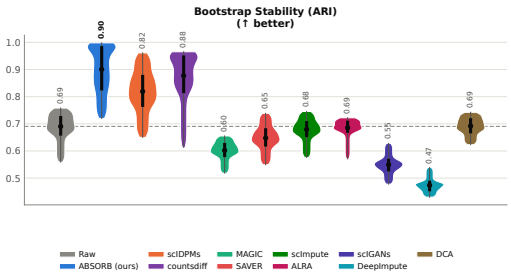
(e) Dentate Gyrus

Figure 13: **kNN purity sweep, all five datasets.** For each cell we compute the fraction of its k nearest neighbours in the embedding that carry the same reference label, for $k \in \{5, 10, 15, 25, 50\}$. The value at $k = 5$ measures whether the immediate neighbourhood is clean; the decay out to $k = 50$ measures how far the correct neighbourhood extends, and is the more discriminative statistic because a locally denoised but globally scrambled embedding decays fast. ABSORB is highest at every k on three datasets and decays the most slowly. **Bone Marrow**: $0.999 \to 0.974$ from $k = 5$ to $k = 50$, a loss of only 2.4 points, against countsdiff $0.974 \to 0.872$ (10 points) and Raw $0.781 \to 0.565$ (22 points). **Dentate Gyrus**: $0.997 \to 0.978$ (1.9 points) against countsdiff $0.993 \to 0.977$ and Raw $0.886 \to 0.804$. **PBMC68k** (Louvain): $0.990 \to 0.850$ against scIDPMs $0.955 \to 0.786$, countsdiff $0.950 \to 0.771$, and Raw $0.849 \to 0.703$; scored instead against the 10 bulk labels every method lands within 0.577–0.660 at $k = 50$ and the spread is not meaningful. **Human Brain**: scIDPMs is highest throughout ($0.982 \to 0.820$), followed by DCA ($0.949 \to 0.798$), scImpute, and MAGIC; ABSORB has the second best $k = 5$ value at 0.965 but decays to 0.784, placing it fifth at $k = 50$, consistent with its mid-pack Silhouette on this dataset. **VISp** again depends on granularity: against the 4 major classes ABSORB is highest at every k ($0.9999 \to 0.9964$), though nearly all methods exceed 0.97 there and the margin is small; against the 23 subclasses ABSORB falls to $0.869 \to 0.613$, below Raw at $0.984 \to 0.902$, which is the neighbourhood-level signature of the same subclass merging seen in Figures 11 and 12.

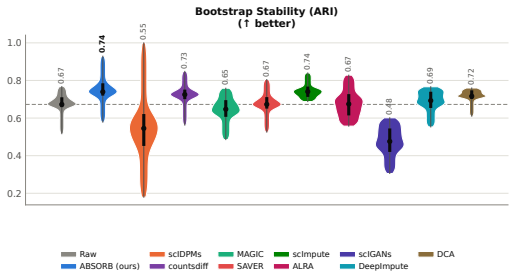


(a) Bone Marrow

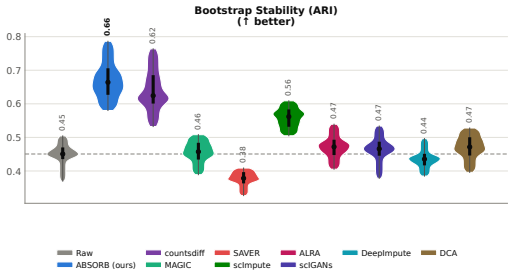
(b) VISp



(c) PBMC68k

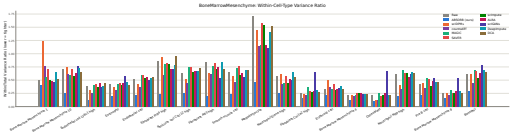


(d) Human Brain

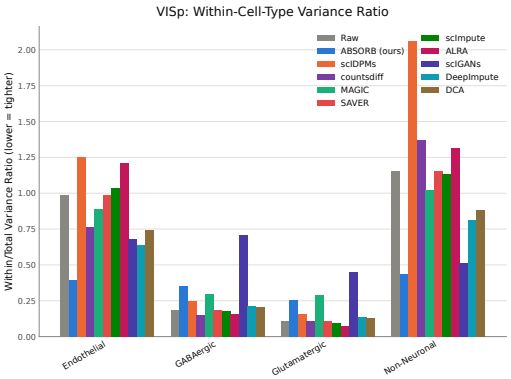


(e) Dentate Gyrus

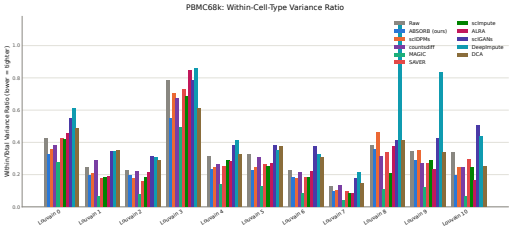
Figure 14: **Bootstrap stability of the recovered partition, all five datasets.** k-means is refitted on 50 independent draws of 80% of the cells and each partition is scored by ARI against the reference labels; the distribution answers whether the structure a method reports is a property of the biology or of the particular cells that happened to be sampled. **VISp** (4 classes): ABSORB 0.990 ± 0.003, effectively deterministic under resampling, against Raw 0.532, DeepImpute 0.538, DCA 0.533, and scIGANs 0.208. **Bone Marrow**: ABSORB 0.868 ± 0.023 with interquartile range 0.850–0.890, against countsdiff 0.654 ± 0.052, scImpute 0.340, and Raw 0.264. **PBMC68k**: ABSORB 0.900 ± 0.077, against countsdiff 0.877, scIDPMs 0.819, and Raw 0.690. **Dentate Gyrus**: ABSORB 0.664 ± 0.044, against countsdiff 0.624, scImpute 0.562, and Raw 0.451. **Human Brain** is the one dataset where ABSORB does not lead outright: scImpute reaches 0.741 ± 0.029 and ABSORB 0.739 ± 0.068, a difference well inside the spread of either, ahead of countsdiff 0.725, DCA 0.716, and Raw 0.673. The spread is as informative as the mean: scIDPMs on Human Brain averages 0.545 with standard deviation 0.150 over a range spanning 0.179 to 1.000, so its partition is essentially a function of which cells were drawn, and its favourable Leiden peak in Figure 12 should be read with that in mind. High stability alone is not evidence of correctness, since a method that collapsed all cells into a few blobs would also be stable; the claim here rests on ABSORB combining top stability with the highest kNN purity on the same datasets (Figure 13). As elsewhere, the VISp figure is at the 4-class granularity.



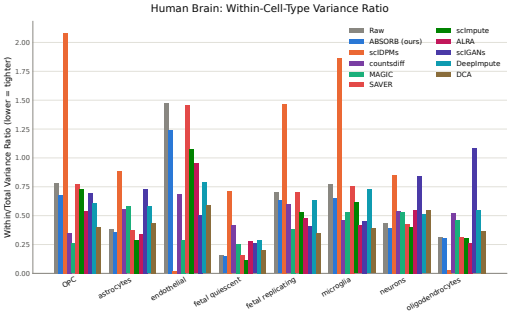
(a) Bone Marrow



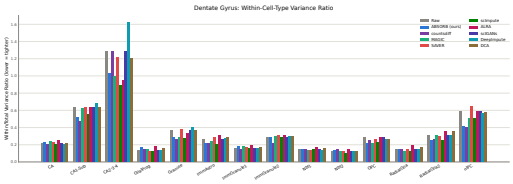
(b) VISp



(c) PBMC68k



(d) Human Brain



(e) Dentate Gyrus

Figure 15: **Within-cluster variance ratio, all five datasets.** For each reference cell type we compute the mean within-cluster variance in the shared embedding and divide by the global variance of the same matrix; the bars show this ratio averaged over cell types. A ratio near 1 means a cell type is no more homogeneous than the dataset as a whole, and lower values indicate noise removed from within populations. ABSORB gives the tightest populations on **Bone Marrow** (0.226 against countsdiff 0.434, scImpute 0.527, Raw 0.568, SAVER 0.603) and on **VISp** (0.358 against DeepImpute 0.450, DCA 0.490, Raw 0.611). On **Dentate Gyrus** the top three are within 0.02 of each other, scImpute 0.289, ABSORB 0.300, and countsdiff 0.306, all below Raw 0.353. On **PBMC68k** MAGIC is lowest at 0.145 with ABSORB second at 0.259 and Raw at 0.341, and on **Human Brain** DCA (0.410) and MAGIC (0.412) lead with ABSORB at 0.551, still below Raw at 0.628. This metric should not be read as monotonically better-is-lower. MAGIC and DCA reach their low ratios by diffusing expression across neighbouring cells, which is the same mechanism behind their leading Silhouette on exactly these two datasets in Figure 9, and it costs them local neighbourhood fidelity on Bone Marrow and Dentate Gyrus, where MAGIC's kNN purity at $k = 50$ is 0.542 and 0.816 against 0.974 and 0.978 for ABSORB. The failure mode in the other direction is also visible: scIDPMs is worst on both VISp (0.930) and Human Brain (0.990), with per-type ratios above 2 on the VISp Non-Neuronal class, meaning it injects more variance into a cell type than the dataset carries overall.

4000-gene datasets, where gene-axis attention is $\mathcal{O}(G^2)$; see §1.4), with 500 warmup steps followed by cosine decay, gradient clipping at 1.0, mixed precision, and an exponential moving average of the weights (decay 0.999) used for inference. Decoding uses $S = 24$ passes at temperature 1. The linear head is zero-initialised. An optional cell-type embedding can be added for classifier-free guidance but is disabled by default.

3.2 Compute Resources

All experiments were run on a single NVIDIA A100 (40 GB) GPU. Training one model takes 15–20 minutes on the 1000-gene simulated datasets (batch size 64) and 25–35 minutes on the 4000-gene real datasets (batch size 32, where gene-axis attention is $\mathcal{O}(G^2)$). Imputing the full matrix ($S = 24$ passes, $n_{\text{samples}} = 5$) takes 1–2 minutes for a 500×1000 matrix and 3–5 minutes at 4000 genes, both wall-clock on the same GPU. Peak GPU memory is approximately 4 GB at $G = 1000$ and 18 GB at $G = 4000$, both in mixed precision.

Software. PyTorch 2.2.1, Python 3.10.13, CUDA 12.1, NumPy 1.23.5, pandas 1.5.3, scipy 1.12.0, scikit-learn 1.4.1. Scanpy 1.9.8 is needed only when cell-type labels are not supplied and Leiden clustering is computed instead. Mixed precision (`torch.cuda.amp`) is enabled by default; all reported numbers use it. Pinned versions are listed in `requirements.txt` and `environment.yml` in the code archive.

Reproducibility budget. The four simulated datasets each require one training run (4×20 min) plus scoring (< 1 min each), totalling ≈ 1.5 GPU-hours. Each real dataset requires two runs (one per masking regime) plus clustering, totalling ≈ 1.5 GPU-hours per dataset. The full result set reported in the paper; four-level simulated data plus three real datasets under two protocols each plus clustering, is reproducible in under 8 GPU-hours on one A100.

4. Datasets

4.1 Simulated Dataset Construction

Four datasets are generated with Splatter Zappia et al. (2017) by sweeping the dropout midpoint while holding everything else fixed. The simulation reproduces the scIMC benchmark configuration Dai et al. (2022) used by scIDPMs; the scIMC server does not distribute the matrices, so they are regenerated from the documented recipe. The full parameter set is:

Parameter	Value
`nGenes`	1000
`batchCells`	500
`group.prob`	$(0.25, 0.25, 0.25, 0.25)$
`mean.shape`	0.3
`mean.rate`	0.02
`de.prob`	0.5
`de.facLoc`	0.5
`dropout.shape`	0.5
`dropout.type`	`"experiment"`
`seed`	42

Table 3: Splatter parameters shared across the severity sweep. Only `dropout.mid` varies.

The severity sweep varies `dropout.mid` $\in \{2, 3, 4, 5\}$, producing four matrices at realised zero rates 71.9%, 64.4%, 56.9%, and 50.0% respectively; these are presented in the paper as D1 (hardest) through D4 (easiest). The naming is offset by one from Splatter's internal index: the paper's D1 corresponds to `dropout.mid=2`, and so on. `dropout.mid=1` and 6 (zero rates 78.8% and 44.0%) are generated by the script as well but are not used in the paper.

Each generated directory contains three files: `counts.csv` (the observed, dropout-corrupted matrix that every method trains on), `truth.csv` (the simulator's pre-dropout matrix, used *only* for

scoring and never revealed to any method), and `label.csv` (the simulator's group assignments). The scoring set is $\mathcal{H} = \{(i, g) : C_{ig}^{\text{obs}} = 0 \wedge C_{ig}^{\text{true}} > 0\}$, exactly the entries the simulator zeroed out. Because the reference is the true value and never the observed input, a method that predicts zero everywhere scores no credit.

The R script `data/generate_simulated.R` in the code archive reproduces these matrices in one command; `summary.csv` records the realised zero rate for verification.

4.2 Real Dataset Preprocessing

Five real datasets are used, each as a cells×genes matrix with accompanying cell-type labels. Table 4 reports both the dimensions of the public matrix as distributed and the dimensions of the filtered matrix that every method actually receives.

Table 4: Real datasets used for held-out recovery and clustering evaluation. "As distributed" is the public matrix before any of our filtering; "after preprocessing" is the matrix handed to every method. "Labels" is the number of reference groups; the same partition is used to build the observation mask M, as the clustering ground truth, and to set k for k-means: annotated cell types for Bone Marrow, Human Brain and Dentate Gyrus, the 4 major classes for VISp (23 transcriptomic subclasses are also reported), and Louvain clusters for PBMC68k (10 coarser bulk labels are also reported). All matrices are raw integer counts except PBMC68k, which is distributed only in normalised, gene-scaled form.

		As distributed		After preprocessing		
Dataset	Tissue	Cells	Genes	Cells	Genes	Labels
VISp	Mouse visual cortex	15,413	45,768	1,681	4,000	4
PBMC68k	Human blood	68,579	32,738	700	765	11
Bone Marrow	Mouse marrow stroma	7,390	20,190	1,698	1,000	19
Human Brain	Human cortex	420	22,085	420	1,000	8
Dentate Gyrus	Mouse hippocampus	18,213	27,998	2,996	4,000	14

Data sources. VISp is the Allen Brain Atlas mouse primary visual cortex dataset (Tasic et al., 2018); PBMC68k is the 10x Genomics peripheral blood dataset (Zheng et al., 2017), used via the `pbmc68k_reduced` release in Scanpy; Bone Marrow is the Mouse Cell Atlas bone-marrow stroma sample GSM2906404 (Han et al., 2018); Human Brain is GEO series GSE67835 (Darmanis et al., 2015); and Dentate Gyrus is from Hochgerner et al. (2018).

Gene filtering. Genes expressed in fewer than 10 cells are discarded first (this removes $45{,}768 \to 34{,}041$ genes on VISp, $20{,}190 \to 16{,}322$ on Bone Marrow, and $22{,}085 \to 15{,}652$ on Human Brain). The survivors are then ranked by raw-count dispersion $\text{disp}_g = \text{var}_g/(\text{mean}_g + \varepsilon)$ with $\varepsilon = 10^{-9}$, and the top G highly variable genes are retained: $G = 4000$ for VISp and Dentate Gyrus, and $G = 1000$ for Bone Marrow and Human Brain. PBMC68k is the one exception: we consume the `pbmc68k_reduced` release shipped with Scanpy, which is already reduced to its 765 most informative genes, so no further gene selection is applied. All subsequent steps, namely mask estimation, tokenizer fitting, training, imputation, and scoring, operate on this filtered matrix.

Cell subsampling. Where the source matrix is larger than needed for the sweep, cells are subsampled without replacement, balanced across reference labels, with a fixed seed. VISp is capped at 500 cells per major class (4 classes, of which Endothelial supplies only 181, giving 1,681); Bone Marrow is capped at a 2,000-cell budget spread over its 19 clusters (giving 1,698); and Dentate Gyrus at a 3,000-cell budget over its 14 cell types (giving 2,996). Human Brain and PBMC68k are used in full at 420 and 700 cells respectively.

Value scale. Four of the five matrices are raw integer UMI or read counts, which is the regime our tokenizer is designed for. PBMC68k is distributed only after library-size normalisation and per-gene scaling, so its entries are continuous and can be negative; following the released preprocessing, each gene is shifted by its column minimum to restore non-negativity before any method sees it. Its

held-out recovery numbers are therefore measured on normalised expression, not on counts, and are not directly comparable in absolute magnitude to the other four datasets.

Observation mask. The observation mask M is estimated per dataset by Algorithm 1: cells are clustered using the provided cell-type labels, and within each cluster a zero entry at a gene that is non-zero in at least one other cell of the cluster is flagged as a candidate dropout ($M_{ig} = 0$). The mask is computed once and shared across all methods.

Held-out masking protocols. Two independent masking protocols are applied per dataset:

- **MCAR50.** 50% of non-zero entries are hidden uniformly at random (missing completely at random), providing a large, unbiased scoring set.
- **MNAR25.** 25% of non-zero entries are hidden with probability biased toward low counts, via $w_i \propto \exp(-\beta\, C_{ig} / \max_j C_{jg})$ with $\beta = 4$. This is closer to how real dropout behaves (low-abundance transcripts are more likely to be missed) and is the harder test.

In both cases, only non-zero entries are eligible (`observed="nonzero"`), and the same random seed (`seed` $= 0$) is used for all methods. The masked matrix is written once as `counts.csv` and handed identically to every method, so all methods see the same evidence and are scored at the same held-out coordinates against the true values stored in `mask.npz`.

Clustering evaluation pipeline. All methods, including the unimputed "Raw" baseline, are embedded identically before clustering: clip negative values at 0, apply $\log(1 + \cdot)$, z-score per gene, then PCA to 50 components. Three clustering algorithms are applied (k-means with k equal to the number of reference cell types, Ward agglomerative, spectral with 15-neighbour affinity), and three label-agreement scores are computed on the k-means partition (AMI, FMI, V-measure). Bootstrap stability is measured as the mean k-means ARI over 50 draws of 80% of cells. Ward and spectral clustering use the first 20 PCs; all other views use all 50. k-means ARI is averaged over 50 random initialisations.