GENERALIZATION IN CONTINUOUS FLOWS FOR CATEGORICAL DATA

Anonymous authors
Paper under double-blind review

ABSTRACT

Flow matching regresses a velocity field onto a finite training set, and the minimizer of that objective transports noise exactly onto the training points. Trained models nonetheless generalize, and the cause has been attributed in turn to the stochastic training target, to the sampler, and to the model. These explanations were developed for continuous data such as images, whose distributions concentrate near low-dimensional manifolds. Flow matching has recently been extended to categorical data such as text and protein sequences, whose support is a finite set of points with no such structure, and it is unclear whether and why it generalizes there. We study this question on discrete supports such as permutations and bracket languages. On these supports we derive the optimal denoiser in closed form and show that the flow map it induces is a lookup table from noise to training points. A one-step network regressed onto it is trained on an exact target and never integrates an ODE, yet it still produces valid samples absent from the training set. We measure memorization across training-set size, model depth, training steps and sampling steps, and find that at matched architecture and compute, flow matching generalizes at $10\times$ less data than masked diffusion and $100\times$ less than autoregression.

1 INTRODUCTION

Flow matching (Lipman et al., 2022; Liu et al., 2022; Albergo et al., 2023) has become the dominant paradigm for generative modeling in continuous domains, underpinning state-of-the-art systems for image and video synthesis (Rombach et al., 2022; Black Forest Labs, 2024; Polyak et al., 2024). Recently, the same construction has been carried over to categorical data such as language and protein sequences, either by placing interpolants on the probability simplex (Stark et al., 2024; Davis et al., 2024; Cheng et al., 2024; Williams et al., 2025) or directly in Euclidean space (Lee et al., 2026; Roos et al., 2026; Potaptchik et al., 2026; Davis et al., 2026).

Flow matching is trained by regressing a velocity field against a finite dataset, so its objective is minimized by the velocity field of the *empirical* distribution (Bertrand et al., 2026) rather than of the distribution the data was drawn from. Integrating that field transports noise exactly onto the training points, and a model that attained the optimum would therefore reproduce its training set (Gu et al., 2023). In practice, models nonetheless generalize (Kadkhodaie et al., 2024; Yoon et al., 2023; Zhang et al., 2023), producing samples absent from their training data.

Three explanations have been proposed for this gap. One attributes it to the training target: the single-sample estimate of the optimal velocity adds gradient noise that keeps optimization from the empirical optimum (Vastola, 2025). Another attributes it to the sampler: sampling trajectories never enter the regions in which the denoiser has memorized (Kaiser & Kollmann, 2026; Luo et al., 2026). The third places it in the model, whose capacity and inductive biases favor a smoother field than the empirical one (Kadkhodaie et al., 2024; Bertrand et al., 2026; Niedoba et al., 2024).

These explanations come from the image domain, where the data lie near a low-dimensional manifold (Pope et al., 2021; Loaiza-Ganem et al., 2024). Since neighboring pixels are correlated, the learned field is smooth (Kadkhodaie et al., 2024; Kamb & Ganguli, 2024; Niedoba et al., 2024), and the model generalizes once the data provide sufficient coverage (Yoon et al., 2023; Biroli et al., 2024; Kaiser & Kollmann, 2026). Categorical data has no such structure. Under a one-hot embedding the support is discrete, and every point between two data points is off the support. Whether flow matching generalizes on such data, and whether it does so for the same reasons, remains open.

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.

On images, a generated sample counts as a copy when it lies within a chosen distance of a training example under a pretrained embedding (Somepalli et al., 2023a;b; Zhang et al., 2018; Pizzi et al., 2022), and generalization is read off a distributional score such as FID (Heusel et al., 2017; Kynkäänniemi et al., 2019). On categorical data with a known support, such as permutations or well-nested bracket strings, neither surrogate is needed. A sample either appears in the training set or does not, and either satisfies the constraint or does not, so memorization and validity are counted exactly.

In this setting, we show that neither the stochastic target nor the sampler is necessary for generalization. Starting from the closed-form denoiser (Bertrand et al., 2026), we numerically integrate the probability flow ODE (Song et al., 2020) to obtain its flow map (Boffi et al., 2025; Geng et al., 2025; Song et al., 2023). For any input noise, this map returns one of the training points, so a model that fits it exactly will memorize. A one-step model regressed onto it generalizes nonetheless, producing valid samples absent from the training set in a *single step*. It is trained against an exact target and never integrates an ODE, so neither the target nor the sampler accounts for its generalization.

Moreover, the same supports are shared with autoregressive models (Brown et al., 2020; Radford et al., 2019) and discrete diffusion (Austin et al., 2021; Lou et al., 2023; Sahoo et al., 2024), the paradigms native to categorical data, so flow matching can be placed beside them at matched architecture, parameter count, training compute and training set. Under this comparison, flow matching generalizes at $10\times$ less data than masked diffusion and $100\times$ less than autoregression.

Our contributions are as follows:

1. **Memorization on categorical data.** We measure memorization and validity of flow matching across dataset size, model depth, training steps, sampling steps and different distributions.
2. **One-step generalization.** We show that a one-step model regressed onto the optimal flow map still generalizes, so neither the stochastic target nor the sampler is a necessary explanation.
3. **Cross-paradigm comparison.** At matched architecture, parameters and training budget, flow matching generalizes at $10\times$ less data than masked diffusion and $100\times$ less than autoregression.

2 Background

Flow Matching. Flow matching (Lipman et al., 2022; Liu et al., 2022; Albergo et al., 2023) learns a transport from a source distribution, here Gaussian noise, to the data distribution by regressing the velocity field of a probability flow ODE along a prescribed interpolant. We sample $x_1 \sim p_{\mathrm{data}}$ and $x_0 \sim \mathcal{N}(0, I)$ of the same shape and form the linear interpolant

$$I_t := (1-t)\, x_0 + t\, x_1, \qquad t \in [0,1]. \tag{1}$$

This interpolant admits a deterministic probability flow ODE

$$\dot{x}_t = b_t(x_t), \qquad b_t(x) = \mathbb{E}[\dot{I}_t \mid I_t = x] = \mathbb{E}[x_1 - x_0 \mid I_t = x], \tag{2}$$

whose solution transports samples from the source to the data distribution. The conditional expectation in Equation (2) is not available in closed form for p_{data}, so b_t is parameterized by a neural network and regressed onto the single-sample target $x_1 - x_0$. This loss differs from regression onto b_t by a constant and shares its minimizer (Lipman et al., 2022). One may instead regress the denoiser (Karras et al., 2022) $D_t(x) = \mathbb{E}[x_1 \mid I_t = x]$, from which the velocity is recovered by

$$b_t(x) = \frac{D_t(x) - x}{1 - t}. \tag{3}$$

We work with the denoiser throughout.

Flow Maps. Sampling from a trained velocity field requires integrating Equation (2) numerically, which costs many network evaluations and incurs discretization error. For a velocity field b_t, the flow map $X_{s,t}$ is the solution operator that transports the state at time s to the state at time t (Boffi et al., 2025),

$$X_{s,t}(x_s) = x_s + \int_s^t b_\tau(x_\tau)\, \mathrm{d}\tau = x_s + (t-s)\, v_{s,t}(x_s), \tag{4}$$

where $v_{s,t}$ is the average velocity over $[s,t]$ (Geng et al., 2025). A model that represents $X_{s,t}$ directly can jump between any two times in a single evaluation without integrating, and in particular $X_{0,1}$ carries a Gaussian sample to the endpoint of its trajectory in one step (Song et al., 2023; Frans et al., 2024; Lee et al., 2026).

Flow Matching on Categorical Data. To apply the construction above to sequences, we follow Lee et al. (2026) and embed them in Euclidean space. A sequence of length L over a vocabulary of size V is identified with its one-hot encoding

$$x \in \{0,1\}^{L\times V}, \tag{5}$$

whose ℓ-th row is the basis vector of the ℓ-th token. Noise is drawn as $x_0 \sim \mathcal{N}(0, I)$ of the same shape and interpolated with x_1 using Equation (1). Since D_t is a conditional expectation of one-hot sequences, its value at each position lies on the probability simplex. So we can parameterize the network using a softmax and train it via cross-entropy against x_1 (Lee et al., 2026). Generation integrates Equation (2) to $t = 1$ and decodes the result position-wise by the $\arg\max$ over the vocabulary (Lee et al., 2026; Roos et al., 2026; Potaptchik et al., 2026; Davis et al., 2026).

Optimal Velocity. Once p_{data} is replaced by the empirical distribution $\hat{p}_1$ on a finite training set $\mathcal{D} = \{x^{(1)}, \dots, x^{(|\mathcal{D}|)}\}$, the minimizer of the flow matching objective, which we call the *optimal velocity*, is available in closed form (Karras et al., 2022; Scarvelis et al., 2023; Bertrand et al., 2026),

$$b_t^\star(x) = \sum_{i=1}^{|\mathcal{D}|} \lambda_i(x,t)\,\frac{x^{(i)} - x}{1-t}, \qquad \lambda_i(x,t) = \operatorname{softmax}_i\!\left(-\frac{\|x - t\,x^{(i)}\|^2}{2(1-t)^2}\right). \tag{6}$$

The weights λ_i form a posterior over which training point produced x and sharpen as $t \to 1$, so the field points at a single example. Integrating it to $t = 1$ returns a training point, and a model that attained this optimum would memorize exactly. Bertrand et al. (2026) use this to test whether generalization arises from the variance of the single-sample regression target (Vastola, 2025), and find that a model regressed directly onto $b_t^\star$ still generalizes.

Memorization in Diffusion Models. Diffusion and flow models reproduce training images when the training set is small or contains duplicates (Somepalli et al., 2023a; Carlini et al., 2023). The transition to memorization has been tied to training-set size, model capacity and training time (Gu et al., 2023; Yoon et al., 2023; Kadkhodaie et al., 2024; Biroli et al., 2024), with models generalizing first and memorizing later in training (Yoon et al., 2023). On categorical data, Luo et al. (2026) study masked diffusion on language and find that finer sampling recovers more of the training set.

Measuring Memorization. On natural data, memorization and generalization are not directly observable, so both must be estimated. A generated sample x is declared a copy when its distance to the training set,

$$\operatorname{dist}(x, \mathcal{D}) = \min_{x^{(i)}\in\mathcal{D}} d\big(\phi(x), \phi(x^{(i)})\big), \tag{7}$$

falls below a threshold, where ϕ is a pretrained embedding and d is a distance (Carlini et al., 2023). Bertrand et al. (2026) take ϕ from LPIPS (Zhang et al., 2018), whereas the copy-detection literature uses self-supervised descriptors (Pizzi et al., 2022; Somepalli et al., 2023a). Generalization is assessed through a distributional surrogate such as FID (Heusel et al., 2017; Kynkäänniemi et al., 2019).

3 Method

We study two objects defined for the empirical distribution of a finite training set. The optimal denoiser $D_t^\star$ is the minimizer of the flow matching objective and has a closed form. The optimal flow map $X_{0,1}^\star$ is the one-step map the denoiser induces, and we show that it returns a training point wherever it is defined. We call the models regressed onto these targets FLM-D* and FLM-X*, and evaluate them against FLM on a family of distributions on which training-set membership and validity are decidable. Sec. 3.1 introduces these distributions, and Sec. 3.2 and 3.3 derive the two targets and give their training and inference recipes.

3.1 Distributions and Metrics

To study memorization, we define each distribution by a set $\mathcal{S} \subseteq \{1, \dots, V\}^L$ of valid sequences, and select those where validity can be checked easily and $|\mathcal{S}|$ is known (Hewitt et al., 2020; Allen-Zhu & Li, 2023; Delétang et al., 2022). We draw the training set $\mathcal{D} = \{x^{(1)}, \dots, x^{(|\mathcal{D}|)}\}$ uniformly from $\mathcal{S}$ without replacement, so $\hat{p}_1$ is uniform on $\mathcal{D}$ and p_{data} is uniform on $\mathcal{S}$.

Every generated sample x then falls into exactly one of three classes:

$$x \in \mathcal{D} \text{ (memorized)}, \qquad x \in \mathcal{S} \setminus \mathcal{D} \text{ (novel)}, \qquad x \notin \mathcal{S} \text{ (invalid)}. \tag{8}$$

For a set $\mathcal{G}$ of generated samples we report validity and memorization as:

$$\text{val}(\mathcal{G}) = \frac{|\mathcal{G} \cap \mathcal{S}|}{|\mathcal{G}|}, \qquad \text{mem}(\mathcal{G}) = \frac{|\mathcal{G} \cap \mathcal{D}|}{|\mathcal{G} \cap \mathcal{S}|}. \tag{9}$$

We normalize memorization as $(\text{mem} - |\mathcal{D}|/|\mathcal{S}|)/(1 - |\mathcal{D}|/|\mathcal{S}|)$, so that uniform sampling over $\mathcal{S}$, which lands in $\mathcal{D}$ with probability $|\mathcal{D}|/|\mathcal{S}|$, scores 0 and reproducing the training set scores 1. A model that generalizes should have high validity and low memorization. We use three distributions: permutations of v symbols, Dyck words with k bracket types (Hewitt et al., 2020), and strings of the random hierarchy model (RHM) (Cagnetta et al., 2024; Sclocchi et al., 2025). We provide further details in Appendix B.

3.2 Optimal Denoiser

Regressing onto the single sample x_1 has the same minimizer as regressing onto $D_t(x) = \mathbb{E}[x_1 \mid I_t = x]$ but adds gradient noise, which Vastola (2025) proposes as the reason models do not reach the memorizing optimum. To study its impact, we remove the noise by instead regressing onto the empirical optimum. We define the *optimal denoiser* $D_t^\star$ as the denoiser of the optimal velocity (Equation (6)) under Equation (3), which is a convex combination of training points for every t.

On one-hot data the optimal denoiser simplifies to a softmax over inner products with the training set.

Proposition 3.1. *Let $\mathcal{D}$ consist of one-hot sequences and $\hat{p}_1$ be uniform on $\mathcal{D}$. Then the optimal denoiser admits the closed form*

$$D_t^\star(x) = \sum_{i=1}^{|\mathcal{D}|} \lambda_i(x,t)\, x^{(i)}, \qquad \lambda_i(x,t) = \text{softmax}_i\Big(\tfrac{t}{(1-t)^2}\langle x, x^{(i)}\rangle\Big). \tag{10}$$

The proof is in App. A. The weights λ_i are a posterior over which training point generated x, sharpened by the factor $t/(1-t)^2$ as $t \to 1$. At small t the target averages the training set and as $t \to 1$ it points at a single training point, so a model that fits it exactly memorizes.

Training. The optimal-denoiser model FLM-D* is FLM with one substitution (Alg. 1). At each step we draw t uniformly, form the interpolant $x_t = (1-t)x_0 + tx_1$, and regress the network's per-position distribution onto $D_t^\star(x_t)$ by cross-entropy, instead of the one-hot x_1. The interpolant, backbone, loss and Euler sampler are otherwise identical, so the two models differ only in whether the target carries sampling noise. Evaluating $D_t^\star$ costs one inner product per training point and is done without gradient, so the substitution adds compute linear in $|\mathcal{D}|$ but no trainable parameters.

Inference. FLM-D* is sampled exactly as FLM: starting from $x_0 \sim \mathcal{N}(0, I)$, the velocity is integrated with S Euler steps to $t = 1$ and the result is decoded by the position-wise $\arg\max$.

Algorithm 1 FLM-D* training and sampling.

```
# D_star: optimal denoiser, Eq. (10)
# f: denoiser network
t = rand()
x_1 = sample(pool)
x_t = (1- t) * randn(L, V) + t * x_1
# exact target replaces x_1
target = D_star(x_t, t, pool)
loss = CE(f(x_t, t), target)

# inference: S Euler steps
x = randn(L, V)
for t in linspace(0, 1, S+ 1)[:- 1]:
   v = (f(x, t) - x) / (1- t)
   x = x + v / S
return argmax(x, dim=- 1)
```

3.3 Optimal Flow Map

A model fitted to Equation (10) is still sampled by integrating the probability flow ODE in finitely many steps, so its trajectories need not land on training points, and the sampler remains a possible explanation for any generalization we observe. The flow map of Equation (4) lets us remove the sampling trajectory, since it carries a noise sample to the endpoint of its trajectory in one evaluation.

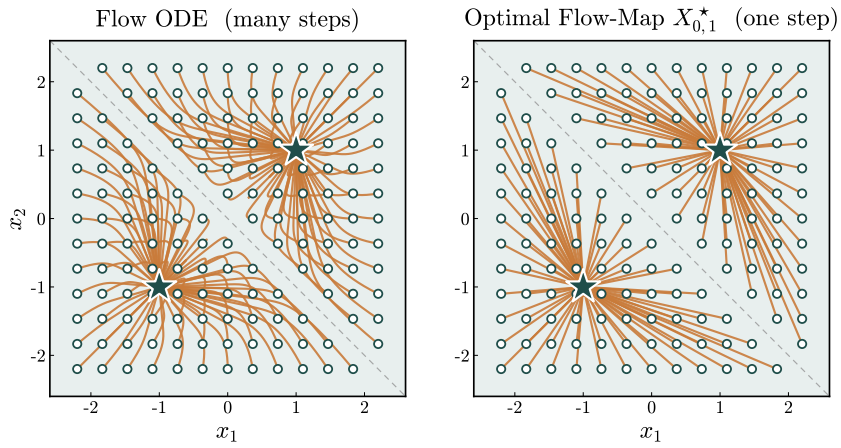


Figure 1: **Optimal Flow Map.** Both panels start from the same grid of noise samples on a two-point training set (stars). *Left*, the probability flow ODE bends each sample onto a training point over many evaluations. *Right*: $X^\star_{0,1}$ sends each sample to the same point in a single evaluation.

We define the *optimal flow map* $X^\star_{0,1}$ as the one-step flow map of the optimal denoiser, and for it that endpoint is a training point. Fig. 1 shows this on a two-point training set. The next proposition shows that this holds for any training set of distinct one-hot sequences.

Proposition 3.2. *Let $\mathcal{D}$ consist of distinct one-hot sequences, $\hat{p}_1$ be uniform on $\mathcal{D}$, and let x_t solve the probability flow ODE induced by Prop. 3.1,*

$$\dot{x}_t = \frac{D^\star_t(x_t) - x_t}{1-t}, \qquad X^\star_{0,1}(x_0) := \lim_{t\to 1} x_t. \tag{11}$$

If the limit exists and $i^\star(x_0) = \arg\max_i \langle X^\star_{0,1}(x_0), x^{(i)} \rangle$ is unique, then

$$X^\star_{0,1}(x_0) = x^{(i^\star(x_0))}. \tag{12}$$

The proof is in App. A. Equation (12) says that the optimal flow map is a simple lookup: it partitions the noise space into $|\mathcal{D}|$ cells and returns one training point per cell.

Training. The optimal-flow-map model FLM-X* regresses a one-step network onto $X^\star_{0,1}$ (Alg. 2). At each step we draw fresh noise x_0, integrate Equation (11) numerically to find the training point it flows to (Sec. 4.2 chooses the step count), and train the network to output that point from x_0 by cross-entropy. The network is not conditioned on time.

Inference. FLM-X* is sampled in one evaluation: $x_0 \sim \mathcal{N}(0, I)$ is passed through f and the output is decoded by the position-wise arg max.

Algorithm 2 FLM-X* training and sampling.

```
# X_star: optimal flow map, Eq. (12)
# f: one- step network
x_0 = randn(L, V)
# the training point x_0 flows to
target = X_star(x_0, pool)
loss = CE(f(x_0), target)

# inference: one forward pass
x_0 = randn(L, V)
return argmax(f(x_0), dim=- 1)
```

4 EXPERIMENTS

Our experiments ask whether flow matching generalizes on categorical data, which parts of training are necessary for it, and how it compares with discrete paradigms. Sec. 4.1 measures memorization in flow matching across training-set size, training steps, sampling steps and depth, and finds a sharp threshold in training-set size. Sec. 4.2 trains FLM-D* and FLM-X* against the exact targets of Sec. 3 and finds that both generalize. Sec. 4.3 places flow matching beside autoregression and masked diffusion at matched architecture, parameters and training steps.

Setup. Unless stated otherwise, every model is a diffusion transformer (Peebles & Xie, 2023) of 8 blocks and width 512, trained with batch size 256 for 200k steps, with hyperparameters following Agarwal et al. (2026). Full hyperparameters are in App. C.

4.1 MEMORIZATION IN FLOW MATCHING

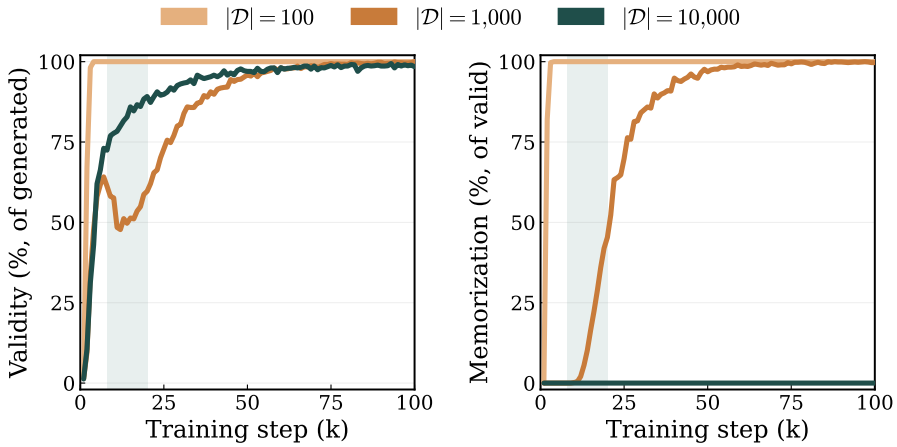


Figure 2: **Delayed memorization at $|\mathcal{D}| = 10^3$.** Validity and memorization over the first 100k training steps, for FLM on Dyck at three training-set sizes $|\mathcal{D}|$. $|\mathcal{D}| = 10^2$ memorizes within the first few thousand steps and $|\mathcal{D}| = 10^4$ stays at zero memorization throughout. $|\mathcal{D}| = 10^3$ generalizes first and memorizes later, and its validity dips during the transition.

Memorization threshold in $|\mathcal{D}|$. Sweeping the training-set size $|\mathcal{D}|$ from 10^2 to 10^6 on each distribution, every valid sample lies in $\mathcal{D}$ at $|\mathcal{D}| \leq 10^3$ and none does at $|\mathcal{D}| \geq 10^4$, so the transition from memorization to generalization happens inside a narrow band of $|\mathcal{D}|$ (Fig. 3), similar to what is seen on images (Yoon et al., 2023; Kadkhodaie et al., 2024; Biroli et al., 2024). The three distributions cross inside the same band despite their different constraints.

Delayed memorization. Resolving the same runs over training separates the training-set sizes into three regimes (Fig. 2, Dyck). At $|\mathcal{D}| = 10^2$ the model memorizes within a few thousand steps, and at $|\mathcal{D}| = 10^4$ memorization stays at zero for the entire run. At $|\mathcal{D}| = 10^3$ the early samples are valid and novel, and as training proceeds the model starts memorizing (Varma et al., 2023), with validity dipping during the transition.

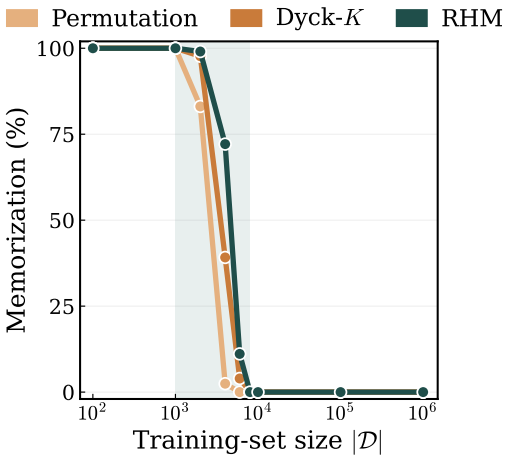


Figure 3: **A sharp threshold in $|\mathcal{D}|$.** Final memorization against training-set size $|\mathcal{D}|$, for FLM on all three distributions.

When does the sampler matter? Varying the number of Euler steps S from 8 to 512 at four training-set sizes spanning the regimes (Tab. 1), memorization is complete at $|\mathcal{D}| = 1000$ and zero at 6000 for every S, and inside the transition it increases with S, a similar dependence on sampling resolution that Luo et al. (2026) prove for masked diffusion. Validity also increases with S, as expected, but the models inside the transition are by far the most sensitive to it: at $S = 8$ they produce valid samples less than half the time, while the models on either side stay near or above 75%. The sampler therefore decides how much is memorized only inside the transition.

	Sampling steps S						
$\lvert\mathcal{D}\rvert$	8	16	32	64	128	256	512
Memorization							
1000	100.0	100.0	100.0	100.0	100.0	100.0	100.0
2000	68.8	78.7	80.6	80.2	80.5	81.2	81.9
4000	1.9	1.5	2.8	3.1	3.6	3.8	3.5
6000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Validity							
1000	96.4	99.8	100.0	100.0	100.0	100.0	100.0
2000	32.5	65.1	86.2	93.8	97.1	98.0	98.8
4000	40.4	84.7	97.2	99.8	99.8	100.0	99.9
6000	74.9	99.4	100.0	100.0	100.0	100.0	100.0

Table 1: **Sampling steps matter only inside the transition.** Final memorization and validity (%) against sampling steps S, for FLM on permutation ($v = 16$).

Deeper models memorize more. Holding the width and the training steps fixed and varying only the number of transformer blocks shifts the threshold (Tab. 2). At $|\mathcal{D}| = 6000$, memorization is 0%, 49%, 81% and 99% for 4, 8, 16 and 32 blocks, and the 32-block model still returns training points for 84% of valid samples at 10^4. Validity stays above 89% in every cell (Tab. 7), so the deeper model memorizes more without leaving the support, as seen in images (Gu et al., 2023; Yoon et al., 2023) and in language (Tirumala et al., 2022; Pham et al., 2026). The trend is monotone at the depths we test, and a non-monotone regime may appear at larger scale, as in double descent (Belkin et al., 2019; Nakkiran et al., 2021).

	Training-set size $\lvert\mathcal{D}\rvert$					
Blocks	1000	2000	4000	6000	8000	10000
4	100.0	96.6	14.3	0.0	0.0	0.0
8	100.0	100.0	95.1	49.2	6.8	0.2
16	100.0	100.0	98.3	80.6	51.0	19.9
32	100.0	100.0	100.0	99.4	93.2	84.1

Table 2: **Depth moves the threshold.** Final memorization (%) against training-set size $|\mathcal{D}|$, for FLM on permutation ($v = 16$) at four depths and fixed width 512. Models are trained with batch size 1024.

4.2 Closed-Form Targets

Do exact targets memorize? If the stochastic target or the sampler were what produces generalization, training against the exact targets of Propositions 3.1 and 3.2 should remove it. We repeat the sweep on permutation ($v = 10$) and on Dyck-K ($L = 10$), for FLM, FLM-D* and FLM-X* (Tab. 3).

FLM-D* tracks FLM on both distributions, memorizing at $|\mathcal{D}| \leq 10^3$ and leaving the training set by 10^4 on permutation and 10^5 on Dyck, so removing the variance of the target does not move the threshold. FLM-X* has neither a stochastic target nor any sampler, yet it leaves the training set in the same band of $|\mathcal{D}|$ as FLM.

	Validity				Memorization			
$\lvert\mathcal{D}\rvert$	10^2	10^3	10^4	10^5	10^2	10^3	10^4	10^5
Permutation ($v = 10$)								
FLM	100.0	100.0	100.0	100.0	100.0	100.0	0.2	0.2
FLM-D*	100.0	100.0	100.0	100.0	100.0	99.8	1.0	0.2
FLM-X*	96.1	74.0	13.6	10.3	100.0	98.7	12.0	0.0
Dyck-K ($L = 10$, $K = 8$)								
FLM	100.0	100.0	99.8	100.0	100.0	99.9	48.9	0.9
FLM-D*	100.0	100.0	99.7	100.0	100.0	100.0	72.9	0.0
FLM-X*	97.1	83.1	34.0	39.3	100.0	98.7	54.3	0.0

Table 3: **Exact targets still generalize.** Final validity and memorization (%) against training-set size $|\mathcal{D}|$, for FLM and the two exact-target models, on permutation ($v = 10$) and Dyck-K ($L = 10$, $K = 8$). Memorization is normalized as in Equation (9). Highlighted cells are the $|\mathcal{D}|$ at which FLM has left the training set.

One-step generation is difficult. While the validity of FLM-X* falls with $|\mathcal{D}|$, we note that it stays far above chance. At $|\mathcal{D}| = 10^5$ it produces valid and novel samples 10.3% of the time on permutation and 39.3% of the time on Dyck, where a uniformly random string is valid with probability 0.04% and 10^{-4}% respectively. The one-step model therefore generalizes, though with lower validity than the multi-step model. Fig. 7 shows qualitative samples on both distributions.

Are samples close to the training set? We measure the Hamming distance from each valid sample to its nearest training point on Dyck at $|\mathcal{D}| = 10^5$, and compare it with the same distance for words drawn uniformly from $\mathcal{S}$ (Fig. 4). FLM, FLM-D* and FLM-X* all match the uniform reference: about 7% at distance 0, which is $|\mathcal{D}|/|\mathcal{S}|$, none at 1, and the rest at 2 and 3. The valid samples of all three models therefore relate to the training set exactly as a uniform sample from the support does.

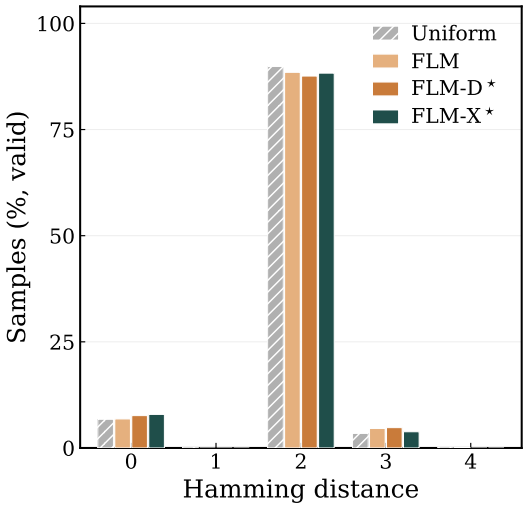


Figure 4: **Distance to the training set.** Hamming distance from each valid sample to its nearest training point, for the three models on Dyck-K ($L = 10$) at $|\mathcal{D}| = 10^5$.

Number of integration steps. FLM-X* needs $X^\star_{0,1}$ for fresh noise at every training step, and each evaluation integrates Equation (11) against the full training set. Therefore, the cost is linear in $|\mathcal{D}|$ and in the number of steps. To pick the step count, we compute the target for $1{,}024$ fixed noise samples with different steps and record how often the training point matches the $1{,}024$-step reference (Fig. 5).

One step matches on under 5% of samples, two steps on most, and eight on 99% at $|\mathcal{D}| \leq 10^4$ and 97% at $|\mathcal{D}| = 10^5$. Agreement at a fixed step count decreases slowly with $|\mathcal{D}|$. We use eight steps throughout, and do not run FLM-X* at $|\mathcal{D}| = 10^6$, where the target computation becomes prohibitive. Full numbers, including Dyck, are in Tab. 8.

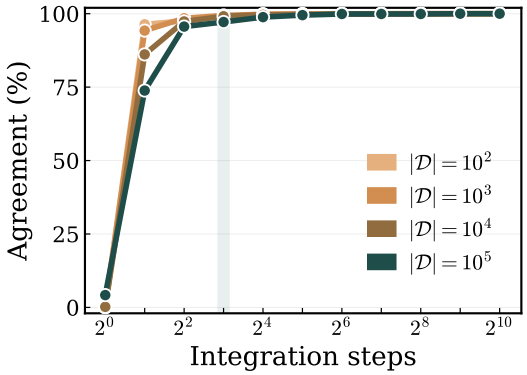


Figure 5: **Eight steps suffice.** Fraction of noise samples whose target matches a $1{,}024$-step reference for permutation ($v = 10$).

Random pairings do not generalize. We train the same one-step network on a target that assigns each noise sample a training point by hashing its sign pattern (App. C), a lookup with no relation between a noise sample and the point it maps to. At every $|\mathcal{D}|$ the network collapses to the marginal token frequencies of the training set and produces no valid samples. Therefore, the pairing induced by the flow is what makes $X^\star_{0,1}$ learnable.

4.3 Comparison Across Paradigms

Which paradigm generalizes first? We train autoregressive (AR; Radford et al., 2019), masked diffusion (MDLM; Sahoo et al., 2024) and flow matching (FLM; Lee et al., 2026) on each distribution with the same backbone, parameter count, and training steps, varying only $|\mathcal{D}|$ (Fig. 6). All three memorize at $|\mathcal{D}| \leq 10^3$ and generalize once $|\mathcal{D}|$ is large enough, but at different sizes: FLM by 10^4 on every distribution, MDLM by 10^5, still returning training points at 10^4 on Dyck and RHM, and AR only by 10^6. Flow matching therefore generalizes at $10\times$ less data than MDLM on Dyck and RHM, and at $100\times$ less than AR on all three distributions.

Corruption as data augmentation. Prabhudesai et al. (2026); Ni et al. (2025); Gao et al. (2025) attribute the data efficiency of masked diffusion over autoregression to implicit augmentation: random masking exposes the model to 2^L corrupted views of each training sequence. Flow matching corrupts each position into a point of $\mathbb{R}^V$, so each sequence is seen under infinitely many corrupted views. We hypothesize that this is a stronger form of the same augmentation and accounts for the ordering of thresholds. We leave its quantification to future work.

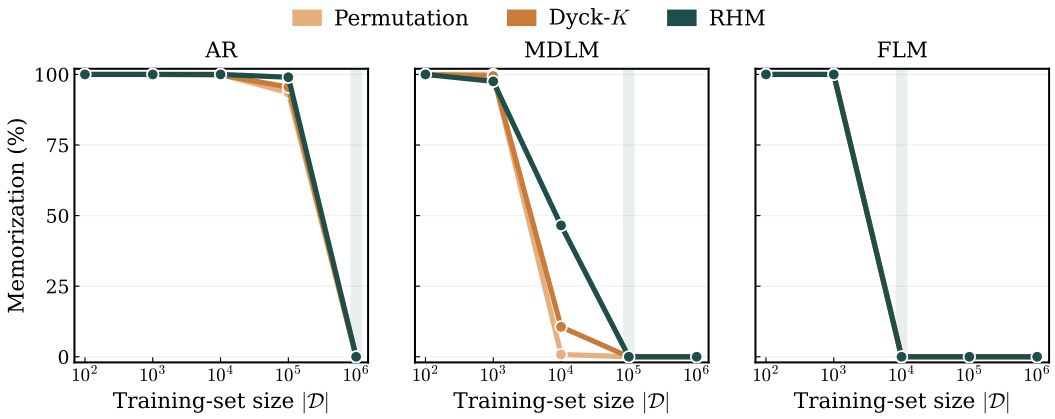


Figure 6: **Paradigms differ in their threshold.** Final memorization against training-set size $|\mathcal{D}|$ for autoregressive (AR), masked-diffusion (MDLM) and flow-matching (FLM) models on all three distributions, at fixed architecture, parameter count, and training steps. FLM generalizes at $10\times$ less data than MDLM and $100\times$ less than AR. Full numbers, including validity, are in Tab. 6.

Token dropout. We apply data augmentation to AR directly, masking a fixed fraction of input tokens during training as in Gao et al. (2025) (Tab. 4). Memorization at $|\mathcal{D}| = 10^5$ drops from 100% to 69% at ratio 0.5, but at 10^4 every valid sample is still memorized, so masking alone does not close the gap to MDLM or FLM. Further tuning of AR, including the learning rate, is shown in App. E.

	Validity		Memorization	
$\lvert\mathcal{D}\rvert$	10^4	10^5	10^4	10^5
none	100.0	99.6	100.0	99.9
token dropout 0.3	99.9	95.3	100.0	94.6
token dropout 0.5	99.2	85.6	100.0	68.8

Table 4: **Token dropout.** Final validity and memorization (%) for AR on permutation with a fraction of input tokens masked in training.

5 Conclusion

We asked whether flow matching generalizes on categorical data, where the support is discrete and the explanations developed for images need not apply, and whether it does so for the same reasons. On distributions where we can study memorization and validity exactly, we derived the optimal denoiser in closed form and the optimal flow map it induces, and used them to remove the stochastic training target and the sampler in turn. A one-step model regressed onto the flow map, a target that is a retrieval of training points, still generalizes, so neither is necessary. Across training-set size, training steps, sampling steps and depth, flow matching switches from memorization to generalization at a sharp threshold, and at matched architecture, parameters and training steps it generalizes at 10× less data than masked diffusion and 100× less than autoregression.

Future work. Regressing onto $D_t^\star$ removes the variance of the single-sample target. Xu et al. (2023) and De Bortoli et al. (2024) exploit this effect with lower-variance targets to improve score matching, and a similar object could be used for discrete domains. Separately, understanding what produces the generalization gap between paradigms, and whether it carries over at scale, could make flow matching the paradigm of choice in data-scarce regimes. Finally, whether memorized sequences can be extracted from flow models on categorical data, as they can from autoregressive and diffusion models (Carlini et al., 2021; 2023), is open.

Limitations. Our data distributions are synthetic and our experiments are small in scale. We show that a one-step model regressed onto the optimal flow map generalizes, but not which properties of the target allow it to, and a closer study of this would be valuable. Its validity also falls with training-set size, and ablations that raise it would allow a stronger claim.

AI USE STATEMENT

Among the tasks with required disclosure, we used generative AI tools to check the correctness of our two proofs and for coding assistance with the data-generation and plotting scripts. We did not use generative AI tools to design experiments, develop the theoretical framework, or interpret results. Other required categories, such as translation and qualitative data analysis, do not apply to this work. Among the tasks with recommended disclosure, we used generative AI tools to polish prose and identify relevant literature. We did not use them for other recommended categories.

We have reviewed all AI-assisted work: generated code was tested, and suggested edits were incorporated only after we checked them. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies generative models on synthetic distributions. It involves no human subjects, personal data, or external datasets. Memorization of training data is a privacy concern for models trained on real data (Carlini et al., 2021; 2023), and understanding when it occurs is a step toward preventing it.

REPRODUCIBILITY STATEMENT

Propositions 3.1 and 3.2 are proved in App. A. App. B defines each distribution and its support size, and App. C gives the backbone, optimizer, schedule, sampling procedure and evaluation protocol for every experiment, including the hashed-lookup control. We will release the code and the generated training sets on publication.

REFERENCES

Manan Agarwal, Sheel Shah, Chanhyuk Lee, Jaehoon Yoo, Jerry Huang, Seunghoon Hong, Aditi Raghunathan, Jinwoo Kim, and Nicholas M Boffi. Posterior refinement: Fast language generation via any-order flow maps. *arXiv preprint arXiv:2606.24773*, 2026.

Michael S Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv preprint arXiv:2303.08797*, 2023.

Zeyuan Allen-Zhu and Yuanzhi Li. Physics of language models: Part 1, learning hierarchical language structures. *arXiv preprint arXiv:2305.13673*, 2023.

Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in neural information processing systems*, 34:17981–17993, 2021.

Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019.

Quentin Bertrand, Anne Gagneux, Mathurin Massias, and Rémi Emonet. On the closed-form of flow matching: Generalization does not arise from target stochasticity. *Advances in neural information processing systems*, 38:8522–8549, 2026.

Giulio Biroli, Tony Bonnaire, Valentin De Bortoli, and Marc Mézard. Dynamical regimes of diffusion models. *Nature Communications*, 15(1):9957, 2024.

Black Forest Labs. FLUX.1 [dev]: A 12 billion parameter rectified flow transformer, 2024. Model available on Hugging Face.

Nicholas M. Boffi, Michael S. Albergo, and Eric Vanden-Eijnden. Flow map matching with stochastic interpolants: A mathematical framework for consistency models. *arXiv:2406.07507*, 2025.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020. URL https://arxiv.org/abs/2005.14165.

Francesco Cagnetta, Leonardo Petrini, Umberto M Tomasini, Alessandro Favero, and Matthieu Wyart. How deep neural networks learn compositional data: The random hierarchy model. *Physical Review X*, 14(3):031001, 2024.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. In *30th USENIX security symposium (USENIX Security 21)*, pp. 2633–2650, 2021.

Nicolas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramer, Borja Balle, Daphne Ippolito, and Eric Wallace. Extracting training data from diffusion models. In *32nd USENIX security symposium (USENIX Security 23)*, pp. 5253–5270, 2023.

Chaoran Cheng, Jiahan Li, Jian Peng, and Ge Liu. Categorical flow matching on statistical manifolds. *Advances in Neural Information Processing Systems*, 37:54787–54819, 2024.

Oscar Davis, Samuel Kessler, Mircea Petrache, İsmail İ Ceylan, Michael Bronstein, and Avishek J Bose. Fisher flow matching for generative modeling over discrete data. *Advances in Neural Information Processing Systems*, 37:139054–139084, 2024.

Oscar Davis, Anastasiia Filippova, Pierre Ablin, Victor Turrisi, Amitis Shidani, Marco Cuturi, and Louis Béthune. Scaling categorical flow maps. *arXiv preprint arXiv:2605.07820*, 2026.

Valentin De Bortoli, Michael Hutchinson, Peter Wirnsberger, and Arnaud Doucet. Target score matching. *arXiv preprint arXiv:2402.08667*, 2024.

Grégoire Delétang, Anian Ruoss, Jordi Grau-Moya, Tim Genewein, Li Kevin Wenliang, Elliot Catt, Chris Cundy, Marcus Hutter, Shane Legg, Joel Veness, et al. Neural networks and the chomsky hierarchy. *arXiv preprint arXiv:2207.02098*, 2022.

Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One step diffusion via shortcut models. *arXiv preprint arXiv:2410.12557*, 2024.

Zitian Gao, Haoming Luo, Lynx Chen, Jason Klein Liu, Ran Tao, Joey Zhou, and Bryan Dai. What makes diffusion language models super data learners? *arXiv preprint arXiv:2510.04071*, 2025.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, J Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. *arXiv preprint arXiv:2505.13447*, 2025.

Xiangming Gu, Chao Du, Tianyu Pang, Chongxuan Li, Min Lin, and Ye Wang. On memorization in diffusion models. *arXiv preprint arXiv:2310.02664*, 2023.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

John Hewitt, Michael Hahn, Surya Ganguli, Percy Liang, and Christopher D Manning. Rnns can generate bounded hierarchical languages with optimal memory. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1978–2010, 2020.

Zahra Kadkhodaie, Florentin Guth, Eero Simoncelli, and Stéphane Mallat. Generalization in diffusion models arises from geometry-adaptive harmonic representations. In *International Conference on Learning Representations*, volume 2024, pp. 46543–46567, 2024.

Tim Kaiser and Markus Kollmann. Diffusion models memorize in training–and generalize in inference. *arXiv preprint arXiv:2603.13419*, 2026.

Mason Kamb and Surya Ganguli. An analytic theory of creativity in convolutional diffusion models. *arXiv preprint arXiv:2412.20292*, 2024.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577, 2022.

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. *Advances in neural information processing systems*, 32, 2019.

Chanhyuk Lee, Jaehoon Yoo, Manan Agarwal, Sheel Shah, Jerry Huang, Aditi Raghunathan, Seunghoon Hong, Nicholas M. Boffi, and Jinwoo Kim. Flow map language models: One-step language modeling via continuous denoising, 2026. preprint.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

Gabriel Loaiza-Ganem, Brendan Leigh Ross, Rasa Hosseinzadeh, Anthony L Caterini, and Jesse C Cresswell. Deep generative models through the lens of the manifold hypothesis: A survey and new connections. *arXiv preprint arXiv:2404.02954*, 2024.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. *arXiv preprint arXiv:2310.16834*, 2023.

Xiaoyu Luo, Wenrui Yu, Qiongxiu Li, and Johannes Bjerva. Characterizing memorization in diffusion language models: Generalized extraction and sampling effects. *arXiv preprint arXiv:2603.02333*, 2026.

Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt. *Journal of Statistical Mechanics: Theory and Experiment*, 2021(12):124003, 2021.

Jinjie Ni, Qian Liu, Longxu Dou, Chao Du, Zili Wang, Hang Yan, Tianyu Pang, and Michael Qizhe Shieh. Diffusion language models are super data learners. *arXiv preprint arXiv:2511.03276*, 2025.

Matthew Niedoba, Berend Zwartsenberg, Kevin Murphy, and Frank Wood. Towards a mechanistic explanation of diffusion model generalization. *arXiv preprint arXiv:2411.19339*, 2024.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4195–4205, 2023.

Bao Pham, Mohammed J Zaki, Luca Ambrogioni, Dmitry Krotov, and Matteo Negri. Language diffusion models are associative memories capable of retrieving unseen data. *arXiv preprint arXiv:2604.26841*, 2026.

Ed Pizzi, Sreya Dutta Roy, Sugosh Nagavara Ravindra, Priya Goyal, and Matthijs Douze. A self-supervised descriptor for image copy detection. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14512–14522. IEEE, 2022.

Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, et al. Movie gen: A cast of media foundation models. *arXiv preprint arXiv:2410.13720*, 2024.

Phillip Pope, Chen Zhu, Ahmed Abdelkader, Micah Goldblum, and Tom Goldstein. The intrinsic dimension of images and its impact on learning. *arXiv preprint arXiv:2104.08894*, 2021.

Peter Potaptchik, Jason Yim, Adhi Saravanan, Peter Holderrieth, Eric Vanden-Eijnden, and Michael S. Albergo. Discrete flow maps. 2026.

Mihir Prabhudesai, Mengning Wu, Amir Zadeh, Katerina Fragkiadaki, and Deepak Pathak. Diffusion beats autoregressive in data-constrained settings. *Advances in Neural Information Processing Systems*, 38:10581–10606, 2026.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Daan Roos, Oscar Davis, Floor Eijkelboom, Michael Bronstein, Max Welling, İsmail İlkan Ceylan, Luca Ambrogioni, and Jan-Willem van de Meent. Categorical flow maps. *arXiv preprint arXiv:2602.12233*, 2026.

Danielle Rothermel, Margaret Li, Tim Rocktäschel, and Jakob Foerster. Don't sweep your learning rate under the rug: A closer look at cross-modal transfer of pretrained transformers. *arXiv preprint arXiv:2107.12460*, 2021.

Subham Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. *Advances in Neural Information Processing Systems*, 37:130136–130184, 2024.

Christopher Scarvelis, Haitz Sáez de Ocáriz Borde, and Justin Solomon. Closed-form diffusion models. *arXiv preprint arXiv:2310.12395*, 2023.

Antonio Sclocchi, Alessandro Favero, and Matthieu Wyart. A phase transition in diffusion models reveals the hierarchical nature of data. *Proceedings of the National Academy of Sciences*, 122(1): e2408799121, 2025.

Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Diffusion art or digital forgery? investigating data replication in diffusion models. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6048–6058. IEEE, 2023a.

Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Understanding and mitigating copying in diffusion models. *Advances in Neural Information Processing Systems*, 36:47783–47803, 2023b.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *International Conference on Machine Learning*, pp. 32211–32252. PMLR, 2023.

Hannes Stark, Bowen Jing, Chenyu Wang, Gabriele Corso, Bonnie Berger, Regina Barzilay, and Tommi Jaakkola. Dirichlet flow matching with applications to dna sequence design. *arXiv preprint arXiv:2402.05841*, 2024.

Kushal Tirumala, Aram Markosyan, Luke Zettlemoyer, and Armen Aghajanyan. Memorization without overfitting: Analyzing the training dynamics of large language models. *Advances in Neural Information Processing Systems*, 35:38274–38290, 2022.

Vikrant Varma, Rohin Shah, Zachary Kenton, János Kramár, and Ramana Kumar. Explaining grokking through circuit efficiency. *arXiv preprint arXiv:2309.02390*, 2023.

John Vastola. Generalization through variance: how noise shapes inductive biases in diffusion models. In *International Conference on Learning Representations*, volume 2025, pp. 34922–34960, 2025.

Bernardo Williams, Victor M Yeom-Song, Marcelo Hartmann, and Arto Klami. Simplex-to-euclidean bijections for categorical flow matching. *arXiv preprint arXiv:2510.27480*, 2025.

Yilun Xu, Shangyuan Tong, and Tommi Jaakkola. Stable target field for reduced variance score estimation in diffusion models. *arXiv preprint arXiv:2302.00670*, 2023.

TaeHo Yoon, Joo Young Choi, Sehyun Kwon, and Ernest K Ryu. Diffusion probabilistic models generalize when they fail to memorize. In *ICML 2023 workshop on structured probabilistic inference & generative modeling*, 2023.

Huijie Zhang, Jinfan Zhou, Yifu Lu, Minzhe Guo, Peng Wang, Liyue Shen, and Qing Qu. The emergence of reproducibility and generalizability in diffusion models. *arXiv preprint arXiv:2310.05264*, 2023.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE/CVF conference on computer vision and pattern recognition*, pp. 586–595. IEEE, 2018.

A THEORETICAL DETAILS

Proposition 3.1. *Let $\mathcal{D}$ consist of one-hot sequences and $\hat{p}_1$ be uniform on $\mathcal{D}$. Then the optimal denoiser admits the closed form*

$$D_t^\star(x) = \sum_{i=1}^{|\mathcal{D}|} \lambda_i(x,t)\, x^{(i)}, \qquad \lambda_i(x,t) = \mathrm{softmax}_i\Big(\tfrac{t}{(1-t)^2}\langle x, x^{(i)}\rangle\Big). \tag{10}$$

Proof. We take as given the closed-form optimal velocity for the empirical distribution $\hat{p}_1$, Equation (6),

$$b_t^\star(x) = \sum_{i=1}^{|\mathcal{D}|} \lambda_i(x,t)\, \frac{x^{(i)} - x}{1-t}, \qquad \lambda_i(x,t) = \mathrm{softmax}_i\left(-\frac{\|x - t x^{(i)}\|^2}{2(1-t)^2}\right).$$

The velocity and denoiser parameterizations are related by $b_t^\star(x) = \big(D_t^\star(x) - x\big)/(1-t)$, so rearranging and substituting,

$$D_t^\star(x) = x + (1-t)\, b_t^\star(x) = x + \sum_{i=1}^{|\mathcal{D}|} \lambda_i(x,t)\big(x^{(i)} - x\big) = \sum_{i=1}^{|\mathcal{D}|} \lambda_i(x,t)\, x^{(i)},$$

where the last equality uses $\sum_i \lambda_i(x,t) = 1$, the λ_i being softmax weights.

It remains to bring the weights into inner-product form. Expanding the square,

$$-\frac{\|x - t x^{(i)}\|^2}{2(1-t)^2} = -\frac{\|x\|^2}{2(1-t)^2} + \frac{t}{(1-t)^2}\langle x, x^{(i)}\rangle - \frac{t^2\|x^{(i)}\|^2}{2(1-t)^2}.$$

A softmax is unchanged by any shift of its arguments that is constant in the index i. The first term is such a shift, and since every one-hot sequence satisfies $\|x^{(i)}\|^2 = L$, so is the third. Discarding both leaves

$$\lambda_i(x,t) = \mathrm{softmax}_i\left(\frac{t}{(1-t)^2}\langle x, x^{(i)}\rangle\right),$$

which is Equation (10). □

Proposition 3.2. *Let $\mathcal{D}$ consist of distinct one-hot sequences, $\hat{p}_1$ be uniform on $\mathcal{D}$, and let x_t solve the probability flow ODE induced by Prop. 3.1,*

$$\dot{x}_t = \frac{D_t^\star(x_t) - x_t}{1-t}, \qquad X_{0,1}^\star(x_0) := \lim_{t\to 1} x_t. \tag{11}$$

If the limit exists and $i^\star(x_0) = \arg\max_i \langle X_{0,1}^\star(x_0), x^{(i)}\rangle$ is unique, then

$$X_{0,1}^\star(x_0) = x^{(i^\star(x_0))}. \tag{12}$$

Proof. Throughout, $\|x^{(i)}\|^2 = L$ for every i, and for $i \neq k$ the inner product $\langle x^{(i)}, x^{(k)}\rangle$ counts the positions on which the two sequences agree, so it is an integer at most $L-1$.

Let $s = \dfrac{t}{1-t}$, a bijection from $[0,1)$ to $[0,\infty)$ with $\dfrac{ds}{dt} = \dfrac{1}{(1-t)^2}$, and set

$$w(s) = \frac{x_t}{1-t} = (1+s)\, x_t.$$

Then

$$\frac{dw}{ds} = (1-t)^2 \frac{d}{dt}\left[\frac{x_t}{1-t}\right] = (1-t)^2\left[\frac{\dot{x}_t}{1-t} + \frac{x_t}{(1-t)^2}\right] = D_t^\star(x_t),$$

since the two x_t terms cancel. Moreover $\dfrac{t}{(1-t)^2} = s(1+s)$ and $x_t = \dfrac{w}{1+s}$, so the exponent in Equation (10) becomes

$$\frac{t}{(1-t)^2}\langle x_t, x^{(i)}\rangle = s\langle w(s), x^{(i)}\rangle.$$

The dynamics are therefore autonomous in w up to the explicit factor s,

$$\frac{dw}{ds} = \sum_{i=1}^{|\mathcal{D}|} \lambda_i(s)\, x^{(i)}, \qquad \lambda_i(s) = \mathrm{softmax}_i\Big(s\langle w(s), x^{(i)}\rangle\Big), \qquad w(0) = x_0, \tag{13}$$

and, since $x_t = w(s)/(1+s)$,

$$X^\star_{0,1}(x_0) = \lim_{s\to\infty} \frac{w(s)}{s} = \lim_{s\to\infty} \frac{1}{s}\int_0^s \frac{dw}{d\sigma}\, d\sigma, \tag{14}$$

whenever either limit exists. The right-hand side is a Cesàro average of points of $\mathrm{conv}\{x^{(i)}\}$, so any limit lies in that hull. Write $u(s) = w(s)/s$ and $a_i(s) = \langle u(s), x^{(i)}\rangle$, so that $\lambda_i(s) = \mathrm{softmax}_i\big(s^2 a_i(s)\big)$ and, from Equation (13),

$$\frac{du}{ds} = \frac{1}{s}\Big(\sum_{i=1}^{|\mathcal{D}|} \lambda_i(s)\, x^{(i)} - u(s)\Big). \tag{15}$$

Suppose $u(s) \to z$. For $i \neq j$,

$$s\langle w(s), x^{(i)} - x^{(j)}\rangle = s^2\big(\langle z, x^{(i)} - x^{(j)}\rangle + o(1)\big),$$

so if the maximizer $i^\star = \mathrm{argmax}_i \langle z, x^{(i)}\rangle$ is unique, every competing logit gap diverges to $+\infty$ and $\lambda_{i^\star}(s) \to 1$. Then $dw/ds \to x^{(i^\star)}$, and by Equation (14), $z = x^{(i^\star)}$. The limit is thus a fixed point of the self-consistency condition

$$z = x^{(i^\star)}, \qquad i^\star = \operatorname*{argmax}_i \langle z, x^{(i)}\rangle.$$

Take $z = x^{(k)}$ for some k. By Cauchy–Schwarz and the equal-norm property, $\langle x^{(k)}, x^{(i)}\rangle \leq \|x^{(k)}\|\|x^{(i)}\| = L = \langle x^{(k)}, x^{(k)}\rangle$, with equality only if $x^{(i)} = c x^{(k)}$ for some $c > 0$, which together with $\|x^{(i)}\| = \|x^{(k)}\|$ forces $x^{(i)} = x^{(k)}$ and hence $i = k$ by distinctness. So the maximizer at $z = x^{(k)}$ is unique, and every training point is a fixed point. Since a fixed point equals $x^{(i^\star)}$ by the condition itself, there are no others. Hence the limit, whenever it exists and its maximizer is unique, is one of the $x^{(i)}$. □

B Distributions

Permutation. $\mathcal{S}$ is the set of orderings of v symbols, one token per symbol, so $L = V = v$ and $|\mathcal{S}| = v!$. A sequence is valid when every symbol appears exactly once, a global constraint that no window of positions can check on its own. We use $v = 16$ throughout and $v = 10$ in Tab. 3.

Dyck. $\mathcal{S}$ is the set of well-nested bracket words built from p pairs over k bracket types. We have $L = 2p$, $V = 2k$ and $|\mathcal{S}| = \mathrm{Cat}(p)\, k^p$. A sequence is valid when every close matches the most recent open of its type. We use $p = 8$ throughout and $p = 5$ in Tab. 3, with $k = 8$.

Random hierarchy model. $\mathcal{S}$ is the language of a random fixed-shape grammar (Cagnetta et al., 2024) with branching factor s, depth d, v symbols per level and m productions per symbol, so $L = s^d$, $V = v$ and $|\mathcal{S}| = v\, m^{(s^d-1)/(s-1)}$. The productions are drawn once and frozen, and a sequence is valid when it can be derived from the root. We use $s = 2$, $d = 4$, $v = 16$, $m = 4$ throughout.

C Hyperparameters

Backbone. Every model is a diffusion transformer (Peebles & Xie, 2023) with 8 blocks, width 512, 8 heads, learned positional embeddings and dropout 0.1, following the implementation of Sahoo et al. (2024). The depth sweep of Tab. 2 varies the number of blocks over $\{4, 8, 16, 32\}$ at fixed width. FLM-X* and MDLM use the backbone without time conditioning, and AR uses it with causal attention.

Training. All models are trained with AdamW at learning rate 3×10^{-4}, $\beta = (0.9, 0.999)$, no weight decay, a constant schedule after $2{,}500$ warmup steps, gradient clipping at 1.0, global batch size 256 and bf16 precision, for 200k steps. Time is drawn uniformly with antithetic sampling for FLM, FLM-D* and MDLM. Evaluation uses an exponential moving average of the weights with decay 0.9999. The AR, MDLM and FLM models of Fig. 6 share every setting above and differ only in the training objective.

Evaluation. FLM and FLM-D* are sampled with 128 Euler steps unless stated otherwise, and FLM-X* with a single forward pass. MDLM is sampled with L steps, unmasking one uniformly chosen position per step, and AR by ancestral decoding at temperature 1. All decoding is position-wise $\arg\max$ for the flow models and categorical sampling for MDLM and AR. Validity and memorization are computed on $1{,}024$ generated samples per checkpoint.

Hashed lookup. The random-lookup baseline of Sec. 4.2 assigns each noise sample $x_0 \in \mathbb{R}^{L\times V}$ a training index by hashing its sign pattern. We draw a fixed integer key $k \in \{0, \dots, 2^{31}-1\}^{LV}$ once, compute $h(x_0) = \sum_j k_j \mathbb{1}[x_{0,j} > 0]$, and set the target to $x^{(h(x_0) \bmod |\mathcal{D}|)}$. The map is deterministic and constant on each orthant of $\mathbb{R}^{L\times V}$, every orthant has the same Gaussian mass, and the induced index distribution is uniform over $\mathcal{D}$. The network, optimizer and training steps are those of FLM-X*.

Tab. 5 lists the architecture, optimization, and inference hyperparameters used for each distribution. Unless stated otherwise, all reported FLM results use these settings.

	Permutation	Dyck	RHM
Distribution			
Parameters	$v = 16$	$p = 8, k = 8$	$s = 2, d = 4, v = 16, m = 4$
Sequence length L	16	16	16
Vocabulary V	16	16	16
Support $\lvert\mathcal{S}\rvert$	$16! \approx 2.1\times10^{13}$	$\mathrm{Cat}(8)\,8^8 \approx 2.4\times10^{10}$	$16 \cdot 4^{15} \approx 1.7\times10^{10}$
Training-set size $\lvert\mathcal{D}\rvert$	$\{10^2, 10^3, 10^4, 10^5, 10^6\}$, with $\{2, 4, 6, 8\}\times10^3$ inside the transition		
Architecture (DiT)			
Hidden size	512		
adaLN cond. dim	128		
Blocks	8		
Attention heads	8		
FFN multiplier	4		
Dropout	0.1		
Positional embedding	Learned		
Time conditioning	Yes		
Parameters (approx.)	29M		
Optimization			
Optimizer	AdamW (β_1=0.9, β_2=0.999, ϵ=10^{-8})		
Gradient clipping	1.0		
Schedule	Constant w/ linear warmup		
Warmup steps	2 500		
Peak LR	3×10^{-4}		
Total steps	200 000		
Global batch size	256		
Time sampling	Uniform, antithetic		
Loss	Cross-entropy on x_1		
Precision	bfloat16 mixed		
EMA decay	0.9999		
Inference			
Solver	Euler		
Sampling steps S	128		
Samples per checkpoint	1 024		

Table 5: **FLM hyperparameters.** Architecture, optimization and inference settings for the FLM runs at sequence length 16 on the three distributions. The depth sweep of Tab. 2 varies only the number of blocks, and the sampling-steps sweep of Tab. 1 varies only S.

D EXPERIMENT DETAILS

Paradigm comparison. Tab. 6 lists the validity and memorization behind Fig. 6.

		Validity			Memorization		
	$\lvert\mathcal{D}\rvert$	AR	MDLM	FLM	AR	MDLM	FLM
Permutation	10^2	99.9	99.9	100.0	100.0	100.0	100.0
	10^3	100.0	81.8	100.0	100.0	100.0	100.0
	10^4	100.0	96.4	100.0	100.0	0.8	0.0
	10^5	99.6	100.0	100.0	93.5	0.0	0.0
	10^6	99.4	100.0	100.0	0.0	0.0	0.0
Dyck-K	10^2	99.9	99.7	100.0	100.0	99.9	100.0
	10^3	100.0	82.5	99.9	100.0	99.4	100.0
	10^4	100.0	71.0	97.3	99.8	10.6	0.0
	10^5	100.0	99.9	99.8	95.5	0.0	0.0
	10^6	99.9	99.9	100.0	0.0	0.0	0.0
RHM	10^2	99.8	99.9	100.0	100.0	100.0	100.0
	10^3	99.7	83.4	99.7	100.0	97.5	100.0
	10^4	99.6	22.5	23.0	100.0	46.5	0.0
	10^5	98.4	96.8	38.4	99.0	0.0	0.0
	10^6	99.3	93.3	36.7	0.0	0.0	0.0

Table 6: **Paradigm comparison, full numbers.** Final validity and memorization (%) against training-set size $\lvert\mathcal{D}\rvert$ for AR, MDLM and FLM on the three distributions, at the settings of App. C.

Depth sweep. Tab. 7 gives the validity for the runs of Tab. 2. It stays above 89% in every cell. Validity *decreases* with depth at large $\lvert\mathcal{D}\rvert$, where the deeper models are still inside their transition.

	Training-set size $\lvert\mathcal{D}\rvert$					
Blocks	1000	2000	4000	6000	8000	10000
4	100.0	94.5	89.4	98.6	98.6	99.3
8	100.0	100.0	95.6	91.7	98.0	99.8
16	100.0	100.0	98.6	93.8	92.0	94.4
32	100.0	100.0	100.0	99.2	94.0	89.1

Table 7: **Depth sweep, validity.** Final validity (%) for the runs of Tab. 2, FLM on permutation ($v = 16$) at four depths and fixed width 512, batch size 1024.

Integration steps. Tab. 8 gives the agreement behind Fig. 5, and the same sweep on Dyck.

		Integration steps									
	$\lvert\mathcal{D}\rvert$	1	2	4	8	16	32	64	128	256	512
Permutation	10^2	0.4	96.4	97.6	99.1	99.8	99.9	100.0	100.0	100.0	100.0
	10^3	0.0	94.2	98.4	99.4	99.8	99.8	99.8	100.0	100.0	100.0
	10^4	0.2	86.1	97.4	99.1	99.6	99.8	99.8	99.9	100.0	100.0
	10^5	4.2	73.8	95.6	97.2	98.8	99.5	99.9	99.9	99.9	100.0
Dyck-K	10^2	1.2	96.1	97.9	99.1	99.4	99.6	99.8	99.8	100.0	100.0
	10^3	0.0	90.3	97.0	98.1	98.9	99.4	99.6	99.9	100.0	100.0
	10^4	0.2	78.5	95.1	97.6	98.7	99.3	99.9	99.9	100.0	100.0
	10^5	0.1	61.0	91.7	95.8	97.5	98.5	99.3	99.6	99.9	100.0

Table 8: **Integration steps.** Fraction (%) of $1{,}024$ fixed noise samples whose target $X^{\star}_{0,1}$ under a given number of Euler steps matches the reference, on permutation ($v = 10$) and Dyck-K ($L = 10$).

E Additional Results

The threshold of AR in Fig. 6 is measured at the learning rate shared by all paradigms. The gaps measured at a shared learning rate can be artifacts of that choice (Rothermel et al., 2021). We therefore sweep the learning rate over $\{10^{-4}, 3\times10^{-4}, 10^{-3}, 3\times10^{-3}\}$ for AR on permutation ($v = 16$) at $|\mathcal{D}| = 10^5$ (Tab. 9).

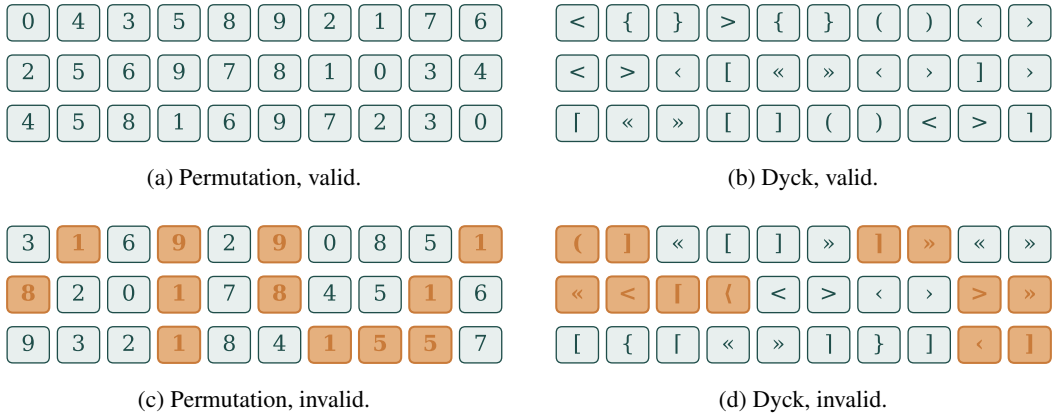
Every run that converges memorizes at least 98% of its valid samples, and the largest learning rate diverges, so no learning rate moves AR into the band where MDLM and FLM have generalized.

Learning rate	Validity	Memorization
10^{-4}	99.7	98.1
3×10^{-4}	99.9	99.9
10^{-3}	99.8	99.9
3×10^{-3}	0.0	–

Table 9: **AR learning-rate sweep.** Final validity and memorization (%) for AR on permutation ($v = 16$) at $|\mathcal{D}| = 10^5$, on 4,096 samples. The largest learning rate diverges.

F Qualitative samples

Fig. 7 shows samples from FLM-X* on permutation and Dyck. All valid samples shown are novel, and invalid ones fail on one or a few positions.



(a) Permutation, valid.

(b) Dyck, valid.

(c) Permutation, invalid.

(d) Dyck, invalid.

Figure 7: **One-step samples from FLM-X*.** Samples generated in a single evaluation by FLM-X* at $|\mathcal{D}| = 10^5$ on permutation ($v = 10$) and Dyck-K ($L = 10$, $K = 8$). Valid samples are the first three outside the training set, and invalid samples the first three that violate the constraint. Tokens that break the constraint are highlighted: repeated symbols on permutation, and mismatched or unmatched brackets on Dyck. Each bracket type is drawn with its own glyph pair.

Tab. 10 counts the violating tokens in each invalid sample of FLM-X* at $|\mathcal{D}| = 10^5$. Two violating tokens is the minimum under either constraint, and 70% of invalid permutations and 60% of invalid Dyck words sit at that minimum, only one edit away from valid sequences.

		Violating tokens (% of invalid samples)				
	Invalid	2	3	4	6	≥ 8
Permutation	887	69.6	0.6	28.1	1.8	0.0
Dyck-K	606	59.9	0.0	28.5	9.1	2.5

Table 10: **Violating tokens.** Distribution of violating tokens per invalid sample at $|\mathcal{D}| = 10^5$.