# Language Diffusion Models are Associative Memories Capable of Retrieving Unseen Data

**Bao Pham** [1] **Mohammed J. Zaki** [1] **Luca Ambrogioni** [2] **Dmitry Krotov** [3] **Matteo Negri** [4]

## Abstract

When do language diffusion models memorize their training data, and how to quantitatively assess their true generative regime? We address these questions by establishing that Uniform-based Discrete Diffusion Models (UDDMs) fundamentally behave as Associative Memories (AMs) *with emergent creative capabilities*. The core idea of an AM is to reliably recover stored data points as *memories* by establishing distinct basins of attraction around them. Historically, models like Hopfield networks use an explicit energy function to guarantee these stable attractors. We broaden this perspective by leveraging that energy is not strictly necessary, as basins of attraction can also be formed via conditional likelihood maximization. This usage of conditional dynamics enables a co-existence of factual recall, where the UDDM can recognize unseen test sequences as fixed points and recover their original tokens given their partially corrupted version, alongside the capability of synthesizing novel sentences. We show that, as the training dataset size increases, basins around training data points shrink while basins around unseen test data points expand, eventually becoming indistinguishable from one another. This memorization-to-generalization transition can be also detected also using the conditional entropy of predicted sequences, which vanish in the memorization regime.

## 1. Introduction

Generative diffusion models (Sohl-Dickstein et al., 2015) have set new standards for image and video generation (Ho et al., 2020; Song & Ermon, 2019; Song et al., 2021; Rombach et al., 2022). Yet, alongside their remarkable generative power, these models exhibit a well-documented tendency to reproduce their training data (Somepalli et al., 2023a;b; Carlini et al., 2023; Webster, 2023). Although recent studies have addressed the interplay between memorization and generation in diffusion models (DMs), they predominantly focus on the continuous image domain (Yoon et al., 2023; Kadkhodaie et al., 2023; Biroli et al., 2024; Kamb & Ganguli, 2024; Achilli et al., 2024; Wen et al., 2024; Jeon et al., 2024; Pham et al., 2025; Achilli et al., 2025). Their mechanics in the discrete domain, particularly for language modeling, remain poorly investigated.

A particularly compelling instance of these questions arises in the context of large language models. Works like (Brown et al., 2020) and (Kojima et al., 2022) have found that these models are capable of few-shot and even zero-shot capabilities, referring to the model's ability to perform novel tasks with minimal or even no task-specific examples, or alternatively the possibility of retrieving appropriate responses never encountered during training. The reasons behind the emergence of these capabilities remain unclear, and the push for models capable of both factual recall and creative behaviors makes it difficult to define and assess what *generalization* means in this regime.

Motivated by these phenomena, this work studies Uniform-based Discrete Diffusion Models (UDDMs) (Austin et al., 2021; Campbell et al., 2024; Gat et al., 2024; Sahoo et al., 2024) through the theoretical lens of Associative Memories (AMs) (Krotov et al., 2025). Fundamentally, a generative system functions as an AM by reliably retrieving stored data points, via producing distinct *basins of attraction* around them (Hopfield, 1982; Gardner, 1988). Historically, attempting to overload these systems with too many data points leads to a catastrophic *memory blackout*, where all meaningful attractors are destroyed (Amit et al., 1987). However, (Kalaj et al., 2025) has recently revealed a counterintuitive regime: overloading an AM can instead trigger a generalization phase where new attractors spontaneously form near unseen examples from the underlying data distribution, while the memorization of the training data points persists. The desired capability of language modeling of

[1]Department of Computer Science, RPI [2]Donders Institute for Brain, Cognition, and Behaviour, Radboud Universit [3]Dynamical Mind [4]Laboratoire de Physique Théorique et Modélisation, CY Cergy Paris Université. Correspondence to: Bao Pham <phamb1@rpi.edu>, Matteo Negri <matteo.negri1@cyu.fr>.

*Published as a paper at the 1st FoGen workshop, ICML 2026*, Seoul, South Korea, 2026. Copyright 2026 by the author(s).

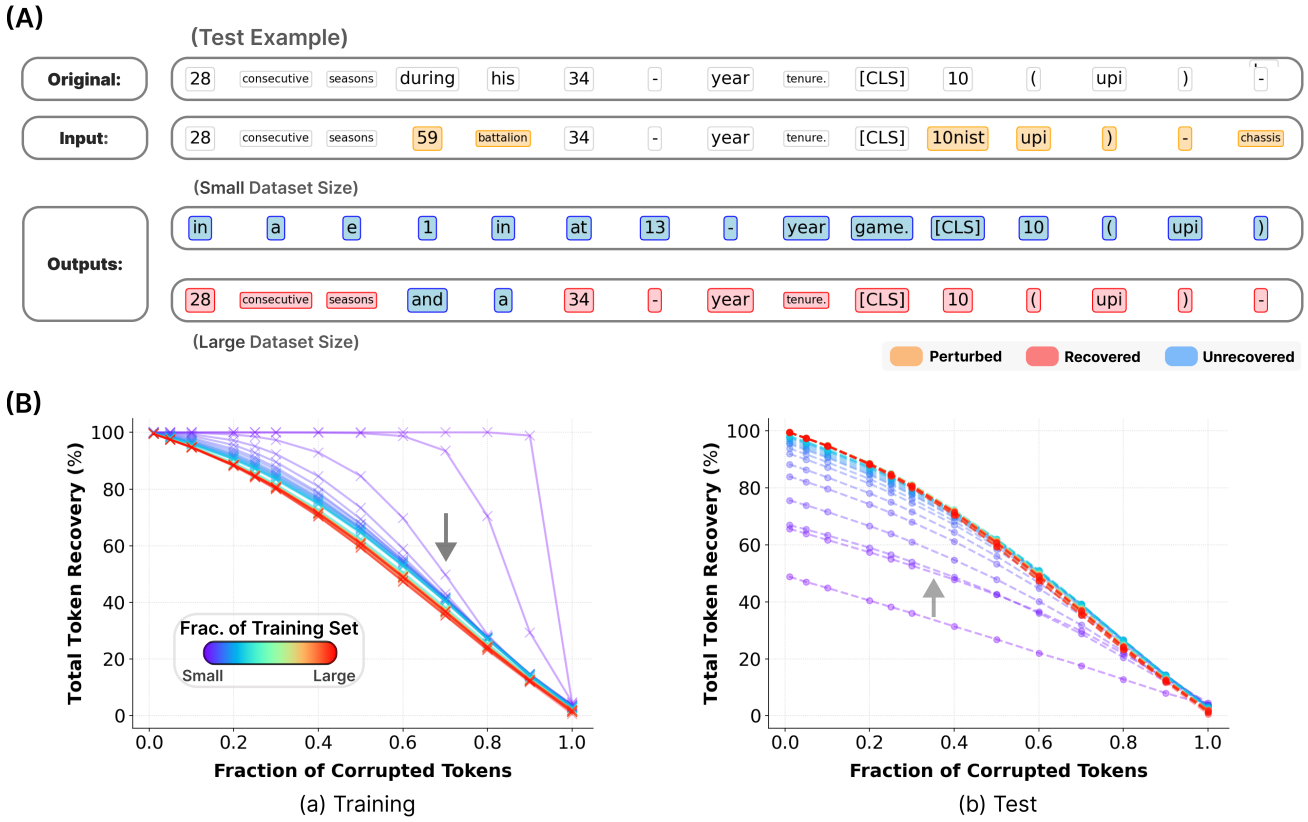



*Figure 1.* **Basins around training examples shrink and basins around test examples expand as the training dataset size increases**. **(A)** Textual examples showing two *Tiny* UDDMs' token recovery at noise level $t = 0.2$, where each is trained on two different training dataset sizes. With a small training dataset, the model fails to recognize unseen test tokens and alters them. With a larger training set, these unseen tokens however become stable and remain intact after the sampling process. **(B)** Average total token recovery rates (%), including both non-corrupt and corrupted tokens, for training and test sequences across varying corruption levels. Line colors indicate the fractions of the training dataset used (ranging from small to large). As data scales, the model's ability to flawlessly recover explicit training examples drops (indicating shrinking basins), while its recovery rate of unseen test examples improves (indicating expanding basins). The convergence of these rates at large dataset sizes (red curves) marks the sharp transition from memorization to generalization. *Note: Deterministic (greedy) sampling was used across these experiments to isolate from stochastic noise.*

having both factual recall and creative behaviors resembles the co-existence of memory attractors and novel attractors in AMs. For this reason, it seems useful to describe the generalization phase of UDDMs from the perspective of AMs, by focusing on the retrieval of seen and unseen test examples. Notably, the co-existence of these two seemingly contradicting capabilities has been studied in an analytically tractable teacher-student setting (Farné et al., 2026).

Typically, AMs are characterized with an explicit and well-defined energy functions, as seen in the Hopfield networks (Hopfield, 1982; Amari, 1972) and Dense Associative Memories (Krotov & Hopfield, 2016; Ramsauer et al., 2021; Krotov & Hopfield, 2021). However, this reliance on an explicit and well-defined energy function is not strictly necessary to guarantee attractor dynamics. As shown by (D'Amico et al., 2026), conditional likelihood maximization alone produces basins of attraction around the data points. Dropping the energy requirement is conceptually essential for extending the AM framework to deep feed-forward architectures: there is no reason to expect that a generic feed-forward network can be written as an energy-based model, since this interpretation restricts the class of admissible architectures due to its constraints on the symmetry of network's weights (Krotov, 2021; Hoover et al., 2023a; Kozachkov et al., 2025). On the contrary, the conditional likelihood structure already exists in many widely used architectures (such as the Transformer (Vaswani et al., 2017)), making it a natural connection with AMs.

**Contributions**. Leveraging conditional likelihood dynamics, our work establishes a connection between UDDMs and AMs by showing that basins of attraction can be formed through conditional likelihood maximization alone (via pseudo-likelihood (Besag, 1974)) and provide a benefit for AM. Specifically, in a simplified setting detailed in Sec. (3), we establish that maximizing conditional likelihood implicitly enforces Hebbian learning (Hebb, 1949) on data points

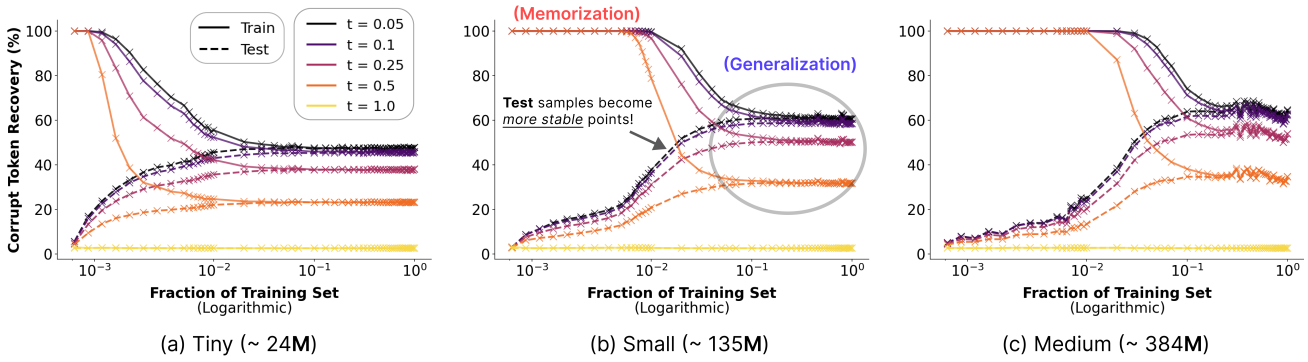



*Figure 2.* **The convergence of corrupted training and test token recovery rates marks the phase transition from memorization to generalization**. The plots display the average token recovery rate (%) as the fraction of the training dataset grows, comparing the model's ability to denoise perturbed tokens (using *stochastic sampling*) from the training set (**solid** lines) against unseen test samples (**dashed** lines) across various perturbation levels based on time $t$ on the LM1B dataset (Chelba et al., 2013). In the limited-data regime, the model exhibits memorization, perfectly recovering training samples while failing on test data. As the dataset expands, the recovery rates for training and test sequences converge to an identical recovery rate, demonstrating that unseen samples have become stable attractors. Notably, the model's size dictates the timing of this shift, where the *Medium* model requires a significantly larger fraction of data to trigger the transition, effectively prolonging the memorization phase.

while maximizing their classification margins, providing a formal bridge between AM and generative models (like UDDMs) which rely on conditional likelihood. Moreover, unlike previous works which connect continuous DMs and AMs (Hoover et al., 2023b; Ambrogioni, 2024; Pham et al., 2025), we extend their results to the discrete setting of language modeling.

This theoretical link of conditional likelihood maximization and AM allows us to interpret UDDMs as AM systems and reveals a sharp memorization-to-generalization transition governed by training dataset size. As the training set gets bigger, basins around training examples shrink while (partial) basins around unseen test examples expand, eventually converging to a regime where novel samples become stable attractors (see Figs. 1 and 2). During memorization, the UDDM fails to recognize unseen test tokens as stable points, frequently altering them during the reverse process. Once generalized, however, the model is likely to maintain these tokens, preserving them if unperturbed and successfully recovering them from partially corrupted sequences via the reverse process (see Figs. 1A and 3A for visual examples of test and training points' token recovery).

Empirically, we validate this transition using token recovery rates and conditional entropy as complementary probes for token stability and the memorization-to-generalization transition. Crucially, token-level conditional entropy distinguishes between different token recovery behaviors, with successfully recovered tokens exhibiting near-zero entropy (see Fig. 3). Meanwhile, sequence-level conditional entropy serves as a practical metric for detecting generalization, as the entropy distributions of training and synthetic sequences align perfectly once the transition is reached (see Fig. 4). Furthermore, we reveal that while scaling up the model's parameter count delays the onset of this phase transition, it ultimately narrows the average conditional "entropy gap" between training and synthetic (or generated) data, effectively increasing the model's confidence in its novel generations (see Fig. 5). Interestingly, conditional entropy has also been found to be crucial for scaling laws in autoregressive language models (Cagnetta et al., 2026), which suggests a wide range of usefulness of this metric in language modeling.

## 2. Uniform-State Discrete diffusion

Consider a clean token $\mathbf{x} \in \mathcal{V}$ drawn from the data distribution $q_{\text{data}}$ with the vocabulary $\mathcal{V} = \{\mathbf{x} \in \{0,1\}^K : \sum_{i=1}^K \mathbf{x}_i = 1\}$. In the DDM framework, $q_{\text{data}}$ is mapped into a simple distribution through a sequence of Markov states via a forward process that is somewhat akin to the continuous diffusion framework (Austin et al., 2021; Sahoo et al., 2024; 2025):

$$\mathbf{z}_t \sim q_t(\mathbf{z}_t|\mathbf{x};\alpha_t) = \text{Cat}(\mathbf{z}_t;\alpha_t\mathbf{x} + (1-\alpha_t)\boldsymbol{\pi}), \quad (1)$$

where $\boldsymbol{\pi} \in \Delta$, $\text{Cat}(\cdot)$ denotes categorical distribution, and $\Delta$ denotes $K$-simplex. Here, $\mathbf{z}_t$ denotes the perturbed token at a time $t \in (0,1]$, where $\mathbf{z}_0 = \mathbf{x}$. The diffusion parameter $\alpha_t \in [0,1]$ is a strictly decreasing $t$-dependent function with the boundary conditions: $\alpha_{t=0} \approx 1$ and $\alpha_{t=1} \approx 0$.

In UDDM, as shown by (Austin et al., 2021) and (Campbell et al., 2024), the true reverse posterior of a previous timestep $s < t$ corresponding to the forward process (1) is

$$\mathbf{z}_s \sim q_{s|t}(\mathbf{z}_s|\mathbf{z}_t,\mathbf{x}) = \text{Cat}\left(\mathbf{z}_s; \frac{K\alpha_t\mathbf{z}_t \odot \mathbf{x} + (\alpha_{t|s}-\alpha_t)\mathbf{z}_t}{K\alpha_t\langle\mathbf{z}_t,\mathbf{x}\rangle + (1-\alpha_t)} + \frac{(\alpha_s-\alpha_t)\mathbf{x} + (1-\alpha_{t|s})\frac{\mathbf{1}}{K}}{K\alpha_t\langle\mathbf{z}_t,\mathbf{x}\rangle + (1-\alpha_t)}\right), \quad (2)$$

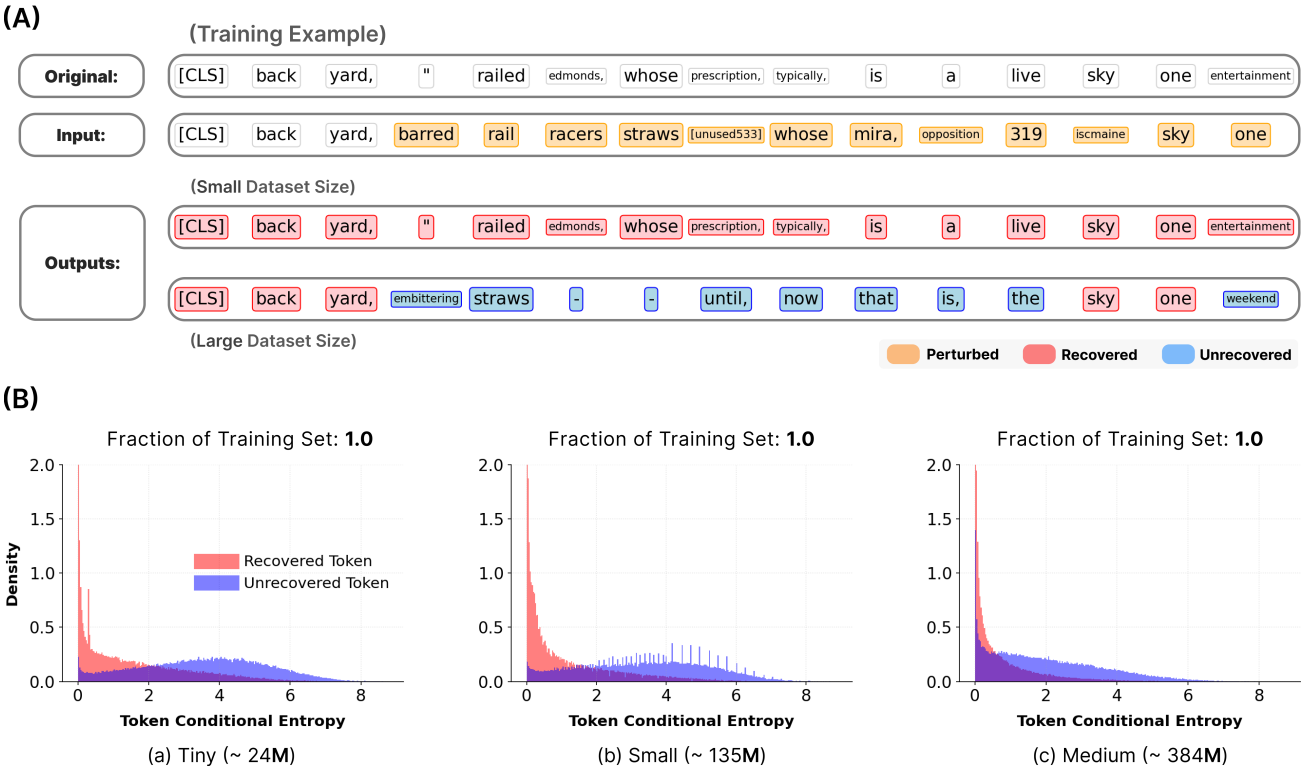



*Figure 3.* **Token-level conditional entropy highlights different token recovery behaviors**. **(A)** Textual examples showing how two *Medium* UDDMs, trained on two different training dataset sizes, recover corrupted training sequences at noise level $t = 0.5$. The model trained on a small training set perfectly memorizes and restores the original text. In contrast, the model trained on a large training set recovers some original words but actively alters others to synthesize a novel sentence. **(B)** Density histograms of conditional entropy for individual tokens from Fig. (2). Successfully recovered tokens consistently demonstrate much lower entropy, indicating higher model confidence. While this gap between recovered and unrecovered tokens narrows in larger models, a surprising number of highly stable, low-entropy tokens persists even during the generalization phase. *Note: The $y$-axis is clipped to better contrast these two token types.*

where $\langle \cdot, \cdot \rangle$ denotes the dot product, $\odot$ denotes Hadamard product, the relative diffusion parameter is $\alpha_{t|s} = \frac{\alpha_t}{\alpha_s}$, and we have a uniform prior over $\mathcal{V}(\boldsymbol{\pi} = \mathbf{1}/K)$ (Sahoo et al., 2024; 2025).

Since the true clean token $\mathbf{x}$ is unknown during the generative process (2), the approximate reverse posterior is defined as $p^{\theta}_{s|t}(\mathbf{z}_s|\mathbf{z}_t) = q_{s|t}(\mathbf{z}_s|\mathbf{z}_t, \mathbf{x} = \mathbf{x}_\theta(\mathbf{z}_t, t))$. The neural network $\mathbf{x}_\theta(\mathbf{z}_t, t) \approx \mathbf{x}$ is trained to predict the clean token $\mathbf{x}$ at any time $t$ according to Eq. (2), where we can formally define this output as the conditional probability distribution $p_\theta(\mathbf{x}|\mathbf{z}_t) = \mathbf{x}_\theta(\mathbf{z}_t, t)$. The network's parameters $\theta$ are optimized via the Negative Evidence Lower Bound (NELBO) objective (Austin et al., 2021; Sahoo et al., 2024):

$$\mathcal{L}_{\text{NELBO}} = \mathbb{E}_q \Big[ \underbrace{-\log p_\theta(\mathbf{x}|\mathbf{z}_0)}_{\mathcal{L}_{\text{reconstruction}}} + \underbrace{\sum_{s<t} D_{\text{KL}}\big[q(\mathbf{z}_s|\mathbf{z}_t, \mathbf{x}) \,||\, p_\theta(\mathbf{z}_s|\mathbf{z}_t)\big]}_{\mathcal{L}_{\text{diffusion}}} \Big] + \underbrace{D_{\text{KL}}\big[q(\mathbf{z}_1|\mathbf{x}) \,||\, p_\theta(\mathbf{z}_1)\big]}_{\mathcal{L}_{\text{prior}}}. \tag{3}$$

By framing the network's prediction as a categorical probability distribution, the cross-entropy terms within $\mathcal{L}_{\text{NELBO}}$ naturally enforce a finite classification margin and therefore store training data points. As we will explore next, this reliance on cross-entropy provides a mathematical link between discrete diffusion dynamics and AM systems. Lastly, the form of UDDMs we used is based on (Sahoo et al., 2025). Please see Appx. (B) for more details on this form of UDDMs.

## 3. Associative Memories from Conditional Sampling

**Conditional sampling in UDDMs**. To establish a connection between AMs and UDDMs, we rely on a specific aspect that emerges from modeling sequences of discrete variables in DMs: the *core assumption* (Austin et al., 2021; Hoogeboom et al., 2021; Lou et al., 2024; Sahoo et al., 2024) is that the denoising process of a sequence $\mathbf{z}^{1:L}$ of length $L$

factorizes for each token $\mathbf{z}^\ell$ as

$$p^\theta_{s|t}(\mathbf{z}^{1:L}_s \mid \mathbf{z}^{1:L}_t) = \prod_{\ell=1}^{L} \psi^\theta_{s|t}(\mathbf{z}^\ell_s \mid \mathbf{z}^{1:L}_t), \tag{4}$$

where $\psi^\theta_{s|t}$ denotes the conditional probability

$$\psi^\theta_{s|t}(\mathbf{z}^\ell_s \mid \mathbf{z}^{1:L}_t) = \mathrm{Cat}\big(\mathbf{z}^\ell_s;\, \mathrm{softmax}_K\big[\beta(t)\, f^\ell_\theta(\mathbf{z}^{1:L}_t)\big]\big), \tag{5}$$

with $f^\ell_\theta(\cdot)$ being the logits produced from a diffusion transformer (Vaswani et al., 2017; Peebles & Xie, 2023) and the softmax is applied over the $K$-categories, which produces a probability distribution per position $\ell$. Here, $\beta(t)$ is a time-dependent inverse temperature, dependent on the diffusion variable $\alpha(t)$, typically increasing as $t \to 0$. These conditional probabilities enter the cross-entropy terms of NELBO (3) and are also used in practice during the denoising process. This reliance on cross-entropy is what provides the connection to AMs: as shown below and by (D'Amico et al., 2026), the cross-entropy loss produces *basins of attraction around the training data points* in the dynamics of conditional sampling for a basic AM, and translates with minimal changes to UDDMs.

### 3.1. Simpler Setting: Binary Variables and Linear Logits

**Associative Memory using classification margins**. Consider an AM of $L$ binary neurons $\mathbf{s}^\ell \in \{\pm 1\}$ with a non-symmetric coupling matrix $\mathbf{W} \in \mathbb{R}^{L\times L}$, where its *diagonal entries are zero*, we have the following deterministic update rule:

$$\mathbf{s}^\ell_\tau = \mathrm{sgn}\Big(\sum_{m=1}^{L} \mathbf{W}^{\ell m}\, \mathbf{s}^m_{\tau+1}\Big), \tag{6}$$

where we adopted the notation of the time $\tau$ running backwards to highlight the connection with the backward dynamics of DMs. Consider also a set of $P$ examples $\boldsymbol{\Xi} \in \{\pm 1\}^{P\times L}$, where $\mathbf{x}^{1:L} \in \boldsymbol{\Xi}$ is a binary vector of length $L$. To build an AM, it is not sufficient to find an optimal coupling matrix $\mathbf{W}^*$ such that $\forall \mathbf{x}^{1:L} \in \boldsymbol{\Xi}$ is a fixed point of Eq. (6), because it would not guarantee finite basins of attraction around the examples. To create such basins, a stronger condition is needed:

$$\mathbf{x}^\ell = \mathrm{sgn}\Big(\sum_{m=1}^{L} \mathbf{W}^{\ell m}\mathbf{x}^m + \kappa\Big), \qquad \forall \ell = 1,\dots,L \tag{7}$$

so that a classification margin $\kappa \in \mathbb{R}^+$ ensures that each fixed point is robust to a finite amount of variable flips from the deterministic update rule (Gardner, 1988; Gardner & Derrida, 1988; Forrest, 1988; Benedetti et al., 2022). Larger $\kappa$ implies larger basins. Given the load $\gamma = P/L$, there exists a maximum margin $\kappa_{\max}(\gamma)$. As shown by (Soudry et al., 2018) and (Montanari et al., 2024), training a Perceptron with the cross-entropy loss in the separable regime implicitly solves Eq. (7) with $\kappa = \kappa_{\max}(\gamma)$, suggesting how to produce large basins of attraction around the training examples.

**Conditional sampling**. To connect the cross-entropy loss with conditional sampling, it is useful to interpret this deterministic update (6) as:

$$\mathbf{s}^\ell_\tau = \arg\max_{\mathbf{s}^\ell}\ \psi_{\tau|\tau+1}\big(\mathbf{s}^\ell_\tau \mid \mathbf{s}^{1:L}_{\tau+1}; \mathbf{W}^\ell\big) \tag{8}$$

where we select the most probable state under the conditional distribution (between position $\ell$ and its *neighborhood* or $1\dots L$ positions in the spin vector excluding $\ell$[1]):

$$\psi_{\tau|\tau+1}(\mathbf{s}^\ell_\tau \mid \mathbf{s}^{1:L}_{\tau+1}; \mathbf{W}^\ell) = \frac{\exp\Big(\mathbf{s}^\ell_\tau f^\ell_{\mathbf{W}}(\mathbf{s}^{1:L}_{\tau+1})\Big)}{2\cosh\Big(f^\ell_{\mathbf{W}}(\mathbf{s}^{1:L}_{\tau+1})\Big)}, \tag{9}$$

where $f^\ell_{\mathbf{W}}(\mathbf{s}^{1:L}_{\tau+1}) = \beta \sum_{m=1}^{L} \mathbf{W}^{\ell m}\, \mathbf{s}^m_{\tau+1}$ with the inverse temperature $\beta$[2]. For binary variables, Eq. (9) yields a logistic form. However, for generic categorical variables, $f_\ell(\mathbf{s}^{1:L})$ are logits inside $\mathrm{softmax}(\cdot)$ like that of Eq. (5).

**Conditional-likelihood maximizes classification margins**. Eq. (9) suggests a way to train the model: by minimizing the following loss function (the negative logarithm of the conditional-likelihood, also called pseudo-likelihood (Besag, 1974)):

$$\begin{aligned}\mathcal{L}(\mathbf{W}) &= -\frac{1}{P}\sum_{\mathbf{x}\,\in\,\boldsymbol{\Xi}} \log \prod_{\ell=1}^{L} \psi(\mathbf{x}^\ell|\mathbf{x}^{1:L}; \mathbf{W}^\ell) \\ &= -\frac{1}{P}\sum_{\mathbf{x}\,\in\,\boldsymbol{\Xi}}\sum_{\ell=1}^{L}\Big[\mathbf{x}^\ell f^\ell_{\mathbf{W}}(\mathbf{x}^{1:L}) - \log 2\cosh(f^\ell_{\mathbf{W}}(\mathbf{x}^{1:L}))\Big].\end{aligned} \tag{10}$$

It has been shown by (D'Amico et al., 2026) that the above objective (10) produces *basins of attraction around the training data points* in the dynamics of the conditional sampling. Unlike classical AM, the couplings $\mathbf{W}$ induced by conditional likelihood do not need to be symmetric, and therefore *no explicit global energy function is required*. The existence of attractor-like behavior follows directly from the structure of the conditional probabilities. To have an intuition on why the loss (10) promotes classification margins, we can derive it with respect to the coupling matrix $\mathbf{W}$:

$$\frac{\mathrm{d}\,\mathcal{L}(\mathbf{W})}{\mathrm{d}\,\mathbf{W}^{\ell m}} \propto -\frac{1}{P}\sum_{\mathbf{x}\,\in\,\boldsymbol{\Xi}} \underbrace{\mathbf{x}^\ell\mathbf{x}^m}_{\text{Hebbian}}\Big[\underbrace{1-\tanh(M^\ell(\mathbf{x}^{1:L}))}_{\text{Penalty}}\Big], \tag{11}$$

[1]Since the diagonal entries of our coupling matrix $\mathbf{W}$ are zeroed, position $\ell$ *does not attend to itself*.

[2]Eq. (9) is obtained using $\psi(\mathbf{s}^\ell|\mathbf{s}^{1:L}) \propto \exp(\beta \mathbf{s}^\ell \sum_m \mathbf{W}^{\ell m}\mathbf{s}^m)$.

where we highlighted the local classification margin $M^\ell(\mathbf{x}^{1:L}) = \mathbf{x}^\ell f^\ell_{\mathbf{W}}(\mathbf{x}^{1:L})$ and factored out the Hebbian term using $\tanh(f^\ell_{\mathbf{W}}(\mathbf{x}^{1:L})) = \tanh(\mathbf{x}^\ell M^\ell(\mathbf{x}^{1:L})) = \mathbf{x}^\ell \tanh(M^\ell(\mathbf{x}^{1:L}))$ and $\mathbf{x}^\ell\mathbf{x}^\ell = 1$. We have two gradient terms, one that involves the typical Hebbian learning (Hebb, 1949) used for storing training data points in the Hopfield network, while the other involves modifying the classification margin around those points. The gradient penalty $1 - \tanh(M^\ell(\mathbf{x}^{1:L})) \approx 2e^{-2M^\ell(\mathbf{x}^{1:L})}$ decays exponentially for correctly classified patterns with wide margins. Consequently, objective (10) concentrates learning on patterns with the smallest margins, and it is minimized when all margins are as large as possible (the weight magnitudes diverge in the separable regime (Soudry et al., 2018)).

**Main differences with UDDMs**. The conditional-likelihood objective (10) is directly related only to the term $\mathcal{L}_{\text{reconstruction}}$ in Eq. (3), since AMs traditionally rely on a fixed temperature rather than annealing the temperature during its dynamics. By following this analogy, the reverse diffusion process can be interpreted as a stochastic AM retrieval dynamics for categorical variables where we also anneal the temperature, similarly in the continuous setting (Ambrogioni, 2024; Pham et al., 2025). From this perspective, we conjecture that the additional terms in Eq. (3) are useful to enlarge basins of attraction when conditional probabilities are parametrized with deep architectures (like a transformer), but we leave this study for future work.

## 4. Memorization to Generalization

From the previous section, the AM produced by conditional-likelihood (10) is capable of generalization (and memorization) (D'Amico et al., 2026; Kalaj et al., 2025). When trained on a sufficiently large dataset, the system is able to create new attractors that are strongly correlated with both training and test examples. In this section, taking inspirations from these results, we can ask whether similar regimes of memorization and generalization can appear in UDDMs.

To understand how deep the AM analogy is, we designed three experiments of increasing realism from the UDDM perspective, progressively bridging the gap between the AM retrieval setting and the standard generative process of a UDDM. Across these experiments, we explore the memorization-to-generalization transition as a function of the training dataset size and model scale. We analyze these experiments on the LM1B dataset (Chelba et al., 2013) via two metrics, as a function of the training dataset size: *token recovery rate* and *conditional entropy*, which serve as proxies for the stability and geometry of the attractors, respectively. For more details and results, please refer to Appx. (A).

**Corrupt Token Recovery**. We define the corrupt token recovery rate as the accuracy with which the model recovers a target or original sequence of length $L$, either a training or a test example, via applying the reverse process on a noisy sequence defined at time $t \in (0, 1]$:

$$R(\mathbf{x}^{1:L}, \hat{\mathbf{x}}^{1:L}) = \frac{1}{|\mathcal{M}|} \sum_{j \in \mathcal{M}} \delta(\mathbf{x}^j, \hat{\mathbf{x}}^j), \tag{12}$$

where $\mathbf{x}^{1:L}$ is the original sequence, $\hat{\mathbf{x}}^{1:L}$ is the recovered sequence after running a denoising process (either greedy or stochastic), $\mathcal{M} = \{j : \delta(\mathbf{x}^j, \mathbf{z}^j_t) = 0\}$ is the set of indices in the input sequence $\mathbf{z}^{1:L}_t$ denoting the positions where the tokens of $\mathbf{x}^{1:L}$ have been changed after applying the forward process (1), and $\delta(\cdot, \cdot)$ denotes the Kronecker delta function.

**Conditional Entropy**. While token recovery rates indicate whether the system successfully returns to an attractor, UDDMs uniquely provide direct access to conditional likelihood, allowing us to probe some information about the local geometry and sharpness of these basins. In the more common setting where AM have an energy landscape, the sharpness of the basin dictates retrieval dynamics (Krotov & Hopfield, 2016; Krotov, 2023; Krotov et al., 2025). Low entropy implies more deterministic attractors (*memorization*), while high entropy signals a flatter landscape with distributed probability mass (Biroli et al., 2024). This flatness facilitates *generalization*, where basins widen and merge to capture the underlying data manifold, enabling the synthesis of novel patterns (Pham et al., 2025). See Appx. (C) for further discussion on the relationship between conditional entropy and energy curvature. The conditional entropy of an individual token $\mathbf{x}^\ell$ in a sequence, given its associated perturbed sequence $\mathbf{z}^{1:L}_t$ at some time $t$, is defined as:

$$\begin{aligned}\mathcal{H}(\mathbf{x}^\ell|\mathbf{z}^{1:L}_t) &= -\sum_{k=1}^{K} \left[p_\theta(\mathbf{x}^\ell \mid \mathbf{z}^{1:L}_t)\right]_k \log \left[p_\theta(\mathbf{x}^\ell \mid \mathbf{z}^{1:L}_t)\right]_k \\ &= -\sum_{k=1}^{K} \left[\mathbf{x}^\ell_\theta(\mathbf{z}^{1:L}_t, t)\right]_k \log \left[\mathbf{x}^\ell_\theta(\mathbf{z}^{1:L}_t, t)\right]_k,\end{aligned} \tag{13}$$

where $\mathbf{x}^\ell_\theta(\mathbf{z}^{1:L}_t, t)$ is the output of the diffusion transformer for position $\ell$ of the sequence over a vocabulary of size $K$. Meanwhile, the conditional entropy of a sequence is the sum of each token's conditional entropy for all positions from 1 to $L$, i.e., $\mathcal{H}(\mathbf{x}^{1:L} \mid \mathbf{z}^{1:L}_t) = \sum_{\ell=1}^{L} \mathcal{H}(\mathbf{x}^\ell \mid \mathbf{z}^{1:L}_t)$.

**Experiment 1: Deterministic retrieval from reference examples shown in Fig. (1)**. In the setting closest to a classical AM, we initialize the reverse process from corrupted training and test examples, and replace the standard stochastic sampling with a deterministic greedy dynamics: at each step, we take the $\arg\max$ of the conditional probabilities rather than sampling from them. In this setting, the perturbation level is decoupled from the diffusion time, which is

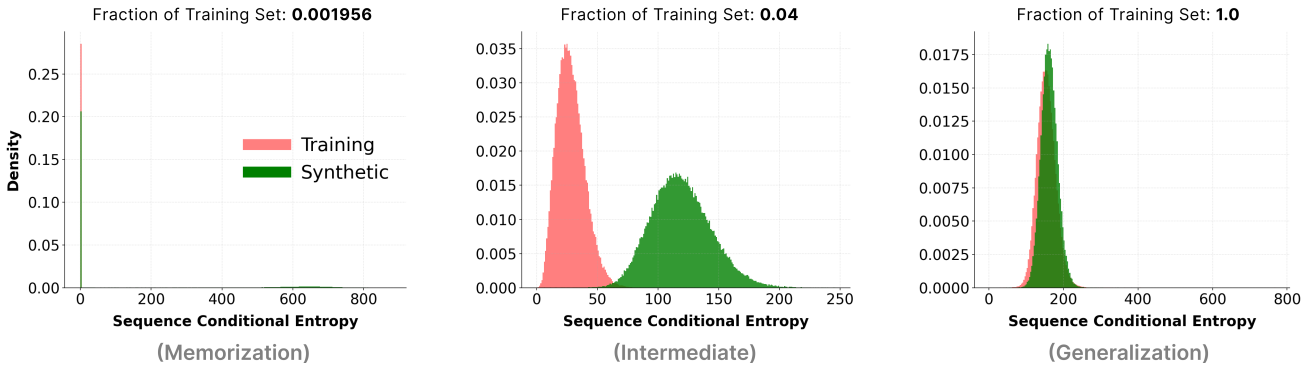



*Figure 4.* **Sequence conditional entropy highlights the memorization to generalization transition**. The histograms compare the sequence entropy distributions of training and synthetic (or generated) sequences respectively, for three hand-selected fractions of the training set for the *Medium* UDDMs. At a very low fraction of the training set, the conditional entropy of training sequences is near zero, aligning very well alongside the conditional entropy of many generated sequences, indicating the regime of memorization. However, as the fraction of the training set reaches the total, the synthetic and training distributions of the sequence conditional entropy converge, signifying generalization.

always set to $t = 1$ while the fraction of corrupted input tokens is varied. Specifically, we measure the token recovery rate as a function of the input noise level, while emulating the zero-temperature retrieval dynamics of an AM.

**Experiment 2: Stochastic retrieval from reference examples shown in Figs. (2) and (3).** We retain the initialization from training and test examples, but restore the standard stochastic reverse dynamics. The perturbation level is now tied to the diffusion time $t$ as in the UDDM scheme: each value of $t$ determines both the fraction of corrupted tokens and the time step at which the reverse process is initialized. We again measure the token recovery rate across varying levels of input noise.

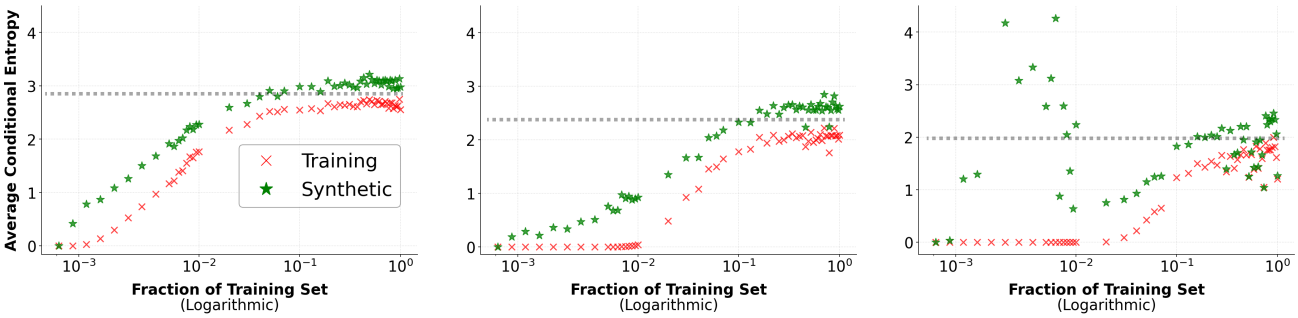
**Experiment 3: Standard generative process shown in Figs. (4) and (5).** In this final setting, we run the full standard reverse dynamics starting from a random initial condition, independent of any reference sequences (e.g., training or test examples). Because ground-truth configurations are unavailable, token recovery rate is no longer a defined metric. Instead, we analyze the conditional entropy of both training and generated sequences. This experiment evaluates the UDDM's standard generative behavior, demonstrating that conditional entropy serves as a practical probe for the memorization-to-generalization transition, eliminating the need to explicitly verify whether generated samples are duplicates of the training dataset as its size increases.

## 5. Results

Based on our results, we observe that, as the training set size grows, the recovery rate for training examples drops (Fig. 2, *solid* lines), reflecting a shrinkage of the basins of attraction around such points shown in Fig. (1A). Simultaneously, the recovery rate for unseen test samples improves (as seen in Fig. 2, *dashed* lines), demonstrating an expansion of basins around test examples examples shown in Fig. (1B). Crucially, the recovery rate of test and training examples, as well as the basins, converge for large datasets. For example, in Fig. (2), the recovery rate approaches zero at $t = 1$, whereas at $t = 0.25$, the model can recover roughly 50% of the corrupted (training and unseen test) tokens for the *Tiny* UDDM. Scaling up the model's parameter size delays this transition: larger models require a significantly greater fraction of the training set to trigger the transition, prolonging the memorization phase. This aspect is also similarly observed in (Yoon et al., 2023; Pham et al., 2025) for the continuous DM setting. For textual examples of token recovery, please refer to Fig. (3A). Also, see Appx. (E) for more textual examples of token recovery alongside additional figures illustrating the shrinkage of the basins across various UDDMs' sizes.

In Fig. (3B), we observe that *successful token recovery is characterized by near-zero conditional entropy*, whereas much of the unsuccessfully recovered tokens are characterized by non-zero conditional entropy. We note that, in the generalization phase, there is a surprising fraction of low-entropy tokens, suggesting that there are tokens which are very stable and less likely to change through the generative process.

Meanwhile, we compare the conditional entropy of training examples and generated samples in Fig. (4). We see that the sequence conditional entropy detects the memorization-to-generalization transition for UDDMs: when the fraction of the training set is sufficiently small for memorization, conditional entropy of both training and generated samples is distributed near zero. As the fraction of the training set increases, the two distributions differentiate from each other and shift to positive values. When the training dataset size




(a) Tiny (~ 24M) (b) Small (~ 135M) (c) Medium (~ 384M)

*Figure 5.* **Average conditional entropy for training versus synthetic sequences**. As the training dataset size increases, the average conditional entropy for both training and synthetic (or generated) sequences naturally rises. Here, we plot the dashed horizontal line in each plot to show the location of the separation between the average conditional entropies of training and synthetic sequences. Initially, there exists an "entropy gap" which separates these two sample types given small fractions of the training set, depicting the UDDM's uncertainty during generation. However, scaling up the model's size narrows this entropy gap, demonstrating that higher parameter counts increase the model's confidence in its own generated text during the generalization regime.

becomes sufficiently large enough for generalization, these two distributions overlap again, now centering on large positive values.

Finally, we compare the *average* conditional entropy of training and synthetic sequences for the whole range of training dataset sizes in Fig. (5). Here, we averaged conditional entropy values across all tokens and sequences. We observe an "entropy gap" as the dataset size increases, implying that the model's uncertainty of generated samples is higher than that of the training examples. As we scale up the model's size, this entropy gap is reduced and the average conditional entropy at the full training dataset size is lower. Please refer to Appx. (E) for similar histograms alongside those which display the conditional entropy of the unrecoverable and recoverable tokens.

## 6. Conclusion

By interpreting UDDMs through the lens of AMs, we show that an explicit energy function is not necessary to guarantee attractor dynamics: basins of attraction can form through conditional likelihood maximization alone, which naturally extends the AM framework to architectures like the diffusion transformer. This perspective reveals a memorization-to-generalization transition governed by training dataset size: as the dataset grows, basins around training examples shrink while basins around unseen test examples expand, until both converge.

Token recovery rate and conditional entropy serve as complementary probes of this transition: the former tracks basin stability directly, the latter provides access to basin geometry. In the memorization regime, conditional entropy vanishes. But, in the generalization regime, it remains finite, producing a measurable entropy gap between training and synthetic sequences that narrows with model scale. Larger models delay the transition, requiring more data before generalization emerges, and both metrics stabilize past that point, possibly suggesting diminishing returns from further training. Since conditional entropy is rather efficient to compute, it offers a practical diagnostic for deployed models.

This work opens several directions for future investigation. First, we proposed a notion of generalization (and an associated metric) that is task-independent and applicable to individual samples. How good these are in practical cases remains an open question: while we showed some promising results, an extensive analysis of how token recovery rate and conditional entropy correlate with standard evaluation metrics would be needed to establish their validity as proxies for generalization. Second, UDDMs provide a convenient and tractable setting for this study, but extending these ideas to regimes where factual recall appears in practice would require significantly larger models, up to and including large language models, which would introduce both conceptual and practical challenges beyond the scope of this work.

## Acknowledgements

Matteo Negri thanks Lenka Zdeborová for insightful discussions on the role of factual recall in generative models. Bao Pham and Mohammed Zaki acknowledge the funding provided by the RPI-IBM Future of Computing Research Collaboration. Part of the work of Dmitry Krotov was performed during his employment at IBM.

## References

Achilli, B., Ventura, E., Silvestri, G., Pham, B., Raya, G., Krotov, D., Lucibello, C., and Ambrogioni, L. Losing dimensions: Geometric memorization in generative diffusion. *arXiv preprint arXiv:2410.08727*, 2024.

Achilli, B., Ambrogioni, L., Lucibello, C., Mézard, M., and Ventura, E. Memorization and generalization in generative diffusion under the manifold hypothesis. *arXiv preprint arXiv:2502.09578*, 2025.

Amari, S.-I. Learning patterns and pattern sequences by self-organizing nets of threshold elements. *IEEE Transactions on computers*, 100(11):1197–1206, 1972.

Ambrogioni, L. In search of dispersed memories: Generative diffusion models are associative memory networks. *Entropy*, 26(5):381, 2024.

Amit, D. J., Gutfreund, H., and Sompolinsky, H. Statistical mechanics of neural networks near saturation. *Annals of physics*, 173(1):30–67, 1987.

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

Benedetti, M., Ventura, E., Marinari, E., Ruocco, G., and Zamponi, F. Supervised perceptron learning vs unsupervised hebbian unlearning: Approaching optimal memory retrieval in hopfield-like networks. *The Journal of Chemical Physics*, 156(10), 2022.

Besag, J. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, 36(2):192–236, 1974. ISSN 00359246. URL http://www.jstor.org/stable/2984812.

Biroli, G., Bonnaire, T., de Bortoli, V., and Mézard, M. Dynamical regimes of diffusion models. *Nature Communications*, 15(1), November 2024. ISSN 2041-1723. doi: 10.1038/s41467-024-54281-3.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.

Cagnetta, F., Raventós, A., Ganguli, S., and Wyart, M. Deriving neural scaling laws from the statistics of natural language. *arXiv preprint arXiv:2602.07488*, 2026.

Campbell, A., Yim, J., Barzilay, R., Rainforth, T., and Jaakkola, T. Generative flows on discrete state-spaces: Enabling multimodal flows with applications to protein co-design. *arXiv preprint arXiv:2402.04997*, 2024.

Carlini, N., Hayes, J., Nasr, M., Jagielski, M., Sehwag, V., Tramèr, F., Balle, B., Ippolito, D., and Wallace, E. Extracting training data from diffusion models. In *Proceedings of the 32nd USENIX Conference on Security Symposium*, SEC '23, USA, 2023. USENIX Association. ISBN 978-1-939133-37-3.

Chelba, C., Mikolov, T., Schuster, M., Ge, Q., Brants, T., Koehn, P., and Robinson, T. One billion word benchmark for measuring progress in statistical language modeling. *arXiv preprint arXiv:1312.3005*, 2013.

D'Amico, F., Bocchi, D., Bono, L. M. D., Rossi, S., and Negri, M. Pseudo-likelihood produces associative memories able to generalize, even for asymmetric couplings. *Physica A: Statistical Mechanics and its Applications*, 692:131497, 2026. ISSN 0378-4371. doi: https://doi.org/10.1016/j.physa.2026.131497. URL https://www.sciencedirect.com/science/article/pii/S0378437126002335.

Farné, G., Boncoraglio, F., and Zdeborová, L. The rules-and-facts model for simultaneous generalization and memorization in neural networks. *arXiv preprint arXiv:2603.25579*, 2026.

Forrest, B. Content-addressability and learning in neural networks. *Journal of Physics A: Mathematical and General*, 21(1):245, 1988.

Gardner, E. The space of interactions in neural network models. *Journal of physics A: Mathematical and general*, 21(1):257, 1988.

Gardner, E. and Derrida, B. Optimal storage properties of neural network models. *Journal of Physics A*, 21:271–284, 1988.

Gat, I., Remez, T., Shaul, N., Kreuk, F., Chen, R. T., Synnaeve, G., Adi, Y., and Lipman, Y. Discrete flow matching. *Advances in Neural Information Processing Systems*, 37:133345–133385, 2024.

Hebb, D. O. The organization of behavior: A neuropsychological theory. 1949. URL https://api.semanticscholar.org/CorpusID:144400005.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 2020.

Hoogeboom, E., Nielsen, D., Abdolshahi, A., and Vahdat, A. Argmax flows and multinomial diffusion: Learning categorical distributions. *Advances in Neural Information Processing Systems*, 34, 2021. URL https://arxiv.org/abs/2102.05379.

Hoover, B., Liang, Y., Pham, B., Panda, R., Strobelt, H., Chau, D. H., Zaki, M., and Krotov, D. Energy transformer. In Oh, A., Neumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural*

*Information Processing Systems*, volume 36, pp. 27532–27559. Curran Associates, Inc., 2023a.

Hoover, B., Strobelt, H., Krotov, D., Hoffman, J., Kira, Z., and Chau, D. H. Memory in plain sight: A survey of the uncanny resemblances between diffusion models and associative memories. *arXiv preprint arXiv:2309.16750*, 2023b.

Hopfield, J. J. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8):2554–2558, 1982. doi: 10.1073/pnas.79.8.2554. URL https://www.pnas.org/doi/abs/10.1073/pnas.79.8.2554.

Jeon, D., Kim, D., and No, A. Understanding memorization in generative models via sharpness in probability landscapes. *arXiv preprint arXiv:2412.04140*, 2024.

Kadkhodaie, Z., Guth, F., Simoncelli, E. P., and Mallat, S. Generalization in diffusion models arises from geometry-adaptive harmonic representation. *arXiv preprint arXiv:2310.02557*, 2023.

Kalaj, S., Lauditi, C., Perugini, G., Lucibello, C., Malatesta, E. M., and Negri, M. Random features hopfield networks generalize retrieval to previously unseen examples. *Physica A: Statistical Mechanics and its Applications*, 678:130946, 2025. ISSN 0378-4371. doi: https://doi.org/10.1016/j.physa.2025.130946. URL https://www.sciencedirect.com/science/article/pii/S0378437125005989.

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

Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., and Iwasawa, Y. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35: 22199–22213, 2022.

Kozachkov, L., Slotine, J.-J., and Krotov, D. Neuron–astrocyte associative memory. *Proceedings of the National Academy of Sciences*, 122(21):e2417788122, 2025.

Krotov, D. Hierarchical associative memory. *arXiv preprint 2107.06446*, 2021. URL https://arxiv.org/abs/2107.06446.

Krotov, D. A new frontier for Hopfield networks. *Nature Reviews Physics*, 5(7):366–367, July 2023. doi: 10.1038/s42254-023-00595-y.

Krotov, D. and Hopfield, J. J. Dense associative memory for pattern recognition. In Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper_files/paper/2016/file/eaae339c4d89fc102edd9dbdb6a28915-Paper.pdf.

Krotov, D. and Hopfield, J. J. Large associative memory problem in neurobiology and machine learning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=X4y_10OX-hX.

Krotov, D., Hoover, B., Ram, P., and Pham, B. Modern methods in associative memory. *arXiv preprint arXiv:2507.06211*, 2025.

Lou, A., Meng, C., and Ermon, S. Masked diffusion models are masked language models. *arXiv preprint arXiv:2406.07524*, 2024. URL https://arxiv.org/abs/2406.07524.

Montanari, A., Zhong, Y., and Zhou, K. Tractability from overparametrization: The example of the negative perceptron. *Probability Theory and Related Fields*, 188(3–4):805–910, 2024. doi: 10.1007/s00440-023-01248-y. URL https://doi.org/10.1007/s00440-023-01248-y. arXiv:2110.15824.

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

Pham, B., Raya, G., Negri, M., Zaki, M. J., Ambrogioni, L., and Krotov, D. Memorization to generalization: Emergence of diffusion models from associative memory. *arXiv preprint arXiv:2505.21777*, 2025.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language models are unsupervised multitask learners. 2019.

Ramsauer, H., Schäfl, B., Lehner, J., Seidl, P., Widrich, M., Gruber, L., Holzleitner, M., Adler, T., Kreil, D., Kopp, M. K., Klambauer, G., Brandstetter, J., and Hochreiter, S. Hopfield networks is all you need. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=tL89RnzIiCd.

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

Sahoo, S., Arriola, M., Schiff, Y., Gokaslan, A., Marroquin, E., Chiu, J., Rush, A., and Kuleshov, V. Simple and effective masked diffusion language models. *Advances in Neural Information Processing Systems*, 37:130136–130184, 2024.

Sahoo, S. S., Deschenaux, J., Gokaslan, A., Wang, G., Chiu, J. T., and Kuleshov, V. The diffusion duality. *International Conference on Machine Learning*, 42, 2025. URL https://openreview.net/forum?id=9P9Y8FOSOk.

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, 2015.

Somepalli, G., Singla, V., Goldblum, M., Geiping, J., and Goldstein, T. Diffusion art or digital forgery? investigating data replication in diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6048–6058, 2023a.

Somepalli, G., Singla, V., Goldblum, M., Geiping, J., and Goldstein, T. Understanding and mitigating copying in diffusion models. *Advances in Neural Information Processing Systems*, 36:47783–47803, 2023b.

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

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

Soudry, D., Hoffer, E., Nacson, M. S., Gunasekar, S., and Srebro, N. The implicit bias of gradient descent on separable data. *Journal of Machine Learning Research*, 19 (70):1–57, 2018.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Webster, R. A reproducible extraction of training images from diffusion models. *arXiv preprint arXiv:2305.08694*, 2023.

Wen, Y., Liu, Y., Chen, C., and Lyu, L. Detecting, explaining, and mitigating memorization in diffusion models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=84n3UwkH7b.

Yoon, T., Choi, J. Y., Kwon, S., and Ryu, E. K. Diffusion probabilistic models generalize when they fail to memorize. In *ICML 2023 Workshop on Structured Probabilistic Inference Generative Modeling*, 2023.

*Table 1.* A table showing hyperparameters of the UDDMs for the *Tiny*, *Small*, and *Medium* sets. These variables are obtained from (Sahoo et al., 2025).

| | Model Type | | |
|---|---|---|---|
| **Hyperparameters** | Tiny | Small | Medium |
| Hidden Size | 256 | 768 | 1024 |
| Conditioning Size | 128 | 128 | 128 |
| Length | 128 | 128 | 128 |
| Num. of Blocks | 8 | 12 | 24 |
| Num. of Heads | 8 | 12 | 16 |
| Scale by Sigma[1] | True | True | True |
| Dropout | 0.1 | 0.1 | 0.1 |

[1] Note: scale by sigma indicates that the model takes the inverse temperature or the appropriate diffusion scheduling parameter at time $t$ instead of the typical approach of conditioning on $t$.

# Appendix

## A. Additional Details on Memorization to Generalization

**Setup**. For our experiments, showcased in Sec. (4), we trained three sets of UDDMs, labeled as *Tiny*, *Small*, and *Medium*, utilizing the code base and approach of (Sahoo et al., 2025). For more details on this variant of UDDMs, please refer to the discussion in Appx. (B) below. Meanwhile, the backbone of our trained UDDMs is the diffusion transformer architecture from (Peebles & Xie, 2023). The configurations of our three variations of UDDMs are described in Tab. (1).

Meanwhile, there are a total of $162$ models, or $54$ models for each of the three UDDM sets we have trained. All models are trained up to 1 million training iterations following the procedures detailed in (Sahoo et al., 2025). For the selection of the fraction of the training dataset sizes, we initially start with the fraction $n = 10^{-2}$ and increment it by $\Delta n = 0.03$ all the way to the full dataset. However, to further magnify the memorization phase, we train more points using linearly spacing (of 17 points, inclusively) starting at $10^{-4}$ to $10^{-2}$, and another set of points using linearly spacing (of 7 points, inclusively) starting from $10^{-2}$ to $0.07$. Lastly, our models are trained on the LM1B dataset (Chelba et al., 2013), using `GPT-2` tokenizer (Radford et al., 2019), where our model handles the block size (or sequence length) of $128$, and all of them are initialized from the same random seed.

**Token Recovery Rate**. To obtain the results in Fig. (2), we utilized our trained models, from the three sets corresponding to *Tiny*, *Small*, and *Medium* UDDMs. Here, we computed the analysis of randomly chosen $5 \times 10^3$ samples belonging to their respective training set (in accordance with their fraction of training dataset size) and also $5 \times 10^3$ sequences from the test set. We performed the perturbation of these sequences using the forward process (18) at time $t$, and run the reverse process (2) starting at or near the same time $t$ back to a small terminal time $\epsilon = 10^{-5}$. Finally, to measure our recovery rate, we applied Eq. (12) from the main text, which measures the rate of perturbed tokens being recovered.

Meanwhile, for Fig. (1), we are interested in the shrinkage of the basins of our training and unseen test sequences. To obtain a more comprehensive view of this shrinkage without worrying about the stochastic noise added during each reverse process (2) step as $t \to 0$, we utilized the greedy or deterministic sampling dynamics instead. Specifically, in Eq. (2), to obtain $\mathbf{z}_s$ as we traverse back in time, we are effectively sampling from a categorical distribution $\text{Cat}(\cdot)$. To convert this stochastic process to a deterministic one, we simply replaced $\text{Cat}(\cdot)$ with the $\arg\max(\cdot)$ operation. Similarly to the previous experiment, we applied perturbation using the forward process (18), and utilized the greedy process to denoise our sequences. The timesteps we used are $\{0.1, 0.2, 0.3, \ldots, 1.0\}$ and the same values of $t$ we used in Fig. (2).

**Conditional Entropy**. In Fig. (3), we computed the token conditional entropy following Eq. (13) for unrecovered and recovered tokens at $t = 10^{-5}$ identified in Fig. (2) using $5 \times 10^3$ samples (per training and per test sequences), respectively. Meanwhile, for the results in Fig. (4) and Fig. (5), we generated a synthetic set of $10^5$ samples for each of the trained models and performed our analyses – alongside using at most $10^5$ sequences belonging to the training set that each model was trained with. It is important to note that the results in Fig. (5) are the average conditional entropy computed across sequences and their respective tokens, while Fig. (4) showcases the sequence conditional entropy – which is the sum of all of the tokens' conditional entropy in a given sequence. Please see Figs. (15)-(17) for the full histograms of conditional entropy on

training versus generated samples, and Figs. (12)-(14) for more conditional entropy histograms of the unrecovered versus recovered tokens shown in Fig. (2).

**Textual Examples**. In Figs. (1) and (9), we are interested in studying the stability of unseen test tokens across different perturbation levels (although we are much more interested in the case where there is little noise for these experiments). For the textual examples illustrated in Figs. (1A) and (3A), we used the models that have been trained with the fractions of the training dataset: $0.000719$ and $1.0$. Although these two figures are utilizing different sampling dynamics (i.e., greedy versues stochastic), in general we do not spot much differences between the greedy and stochastic processes for our token recovery experiments. Please see Figs. (3A), (10), and (11) for the textual examples that collected using the typical stochastic process.

**Hardware**. The training of the UDDMs are done using NVIDIA Tesla V100 GPUs. Each GPU has 32GB of memory and is linked with Power9 processors, clocking at 3.15 GHz maximum. For each model, we used 4 GPUs and an effective total (or global) batch size of $512$ samples. For each GPU, the local batch size is set as $64$, requiring $2$ gradient accumulation steps.

## B. Uniform-state Discrete Diffusion and Duality with Gaussian

**Duality of Uniform and Gaussian**. In continuous diffusion modeling, we typically rely on the diffusion mapping of a data distribution $q_{\text{data}}$ to a simple prior distribution that is often the standard Gaussian distribution $\mathcal{N}(0, \mathbf{I}_K)$. The marginal distribution of the noisy latent variable $\mathbf{w}_t \sim \tilde{q}_t(\cdot|\mathbf{x})$ at time $t$ is defined as:

$$\mathbf{w}_t \sim \tilde{q}_t(\mathbf{w}_t|\mathbf{x}; \tilde{\alpha}_t) = \mathcal{N}(\mathbf{w}_t; \tilde{\alpha}_t \mathbf{x}, (1 - \tilde{\alpha}_t^2)\mathbf{I}_K), \tag{14}$$

where $\tilde{\alpha}_t \in [0, 1]$ is the diffusion parameter that is a monotonically decreasing function in $t$. The boundary conditions are $\tilde{q}_{t=0} \approx q_{\text{data}}$ and $\tilde{q}_{t=1} = \mathcal{N}(0, \mathbf{I}_K)$.

However, as shown in (Sahoo et al., 2025), there exists a connection between the Gaussian and Uniform diffusion processes for the discrete setting. Specifically, we can utilize the operator, $\arg\max : \mathbb{R}^K \to \mathcal{V}$, to map a continuous vector $\mathbf{w} \in \mathbb{R}^K$ to the one-hot vector corresponding to $\arg\max(\mathbf{w}) = \underset{\mathbf{z} \in \mathcal{V}}{\arg\max}\ \mathbf{z}^\top \mathbf{w}$. Then, we can define the discrete marginals to be

$$\mathbf{z}_t = \arg\max(\mathbf{w}_t), \tag{15}$$

and the conditional probability mass function $p_t(\mathbf{z}_t|\mathbf{x})$ marginalized over $\mathbf{w}_t \sim \tilde{q}_t(\mathbf{w}_t|\mathbf{x}; \tilde{\alpha}_t)$ such that

$$\mathbf{z}_t \sim P_t\big(\mathbf{z}_t|\mathbf{x}; \mathcal{T}(\tilde{\alpha}_t)\big) = \text{Cat}\left(\mathbf{z}_t; \mathcal{T}(\tilde{\alpha}_t)\mathbf{x} + (1 - \mathcal{T}(\tilde{\alpha}_t))\frac{\mathbf{1}}{K}\right), \tag{16}$$

where $\mathcal{T} : [0, 1] \to [0, 1]$ is the Gaussian Diffusion Transformation operator. This operator is defined as the following Gaussian integral:

$$\alpha_t = \mathcal{T}(\tilde{\alpha}_t) = \frac{K}{K-1}\left[\int_{-\infty}^{\infty} \phi\left(z - \frac{\tilde{\alpha}_t}{\sqrt{1 - \tilde{\alpha}_t^2}}\right)\Phi^{K-1}(z)\mathrm{d}z - \frac{1}{K}\right] \tag{17}$$

where $\phi(z) = \frac{\exp(-z^2)}{\sqrt{2\pi}}$ is the standard Normal distribution and $\Phi(z) = \int_{-\infty}^{z} \phi(t)\mathrm{d}t$ is the respective cumulative distribution.

Overall, there exists a fundamental connection between Uniform-state discrete and Gaussian diffusion processes, shown in (Sahoo et al., 2025). Specifically, this formal connection is expressed as

$$\mathbf{z}_t \sim q_t(\mathbf{z}_t|\mathbf{x}; \mathcal{T}(\tilde{\alpha}_t)) = [\arg\max]_* \, \tilde{q}_t(\mathbf{w}_t|\mathbf{x}; \tilde{\alpha}_t) \tag{18}$$

where $*$ denotes the push-forward of the $K$-dimensional Gaussian density $\tilde{q}_t$ under $\arg\max$ which yields a categorical distribution of $K$ categories.

## C. Conditional Entropy and Curvature

In this section, inspired by (Biroli et al., 2024) and (D'Amico et al., 2026), we relate entropy and the curvature of the energy in the continuous setting, using local approximation, to show there exists a connection between these two ideas. Here, assume that the clean data $\mathbf{x} \in \mathbb{R}^d$ and its perturbed version $\mathbf{z}_t \in \mathbb{R}^d$ at time $t$.

**Proof Sketch**. Consider the conditional distribution $p(\mathbf{x}|\mathbf{z}_t)$ defined by an energy function $E(\mathbf{x};\mathbf{z}_t)$:

$$p(\mathbf{x}|\mathbf{z}_t) = \frac{1}{Z(\mathbf{z}_t)} e^{-E(\mathbf{x};\,\mathbf{z}_t)}, \tag{19}$$

where $Z(\mathbf{z}_t) = \int e^{-E(\mathbf{y},\mathbf{z}_t)} \mathrm{d}\mathbf{y}$ is the partition function. We assume the distribution is peaked around a mode $\mathbf{x}^*$, representing the most likely clean data point given the noisy observation $\mathbf{z}_t$.

To analyze the local geometry, we perform a second-order Taylor expansion of the energy $E(\mathbf{x};\mathbf{z}_t)$ with respect to $\mathbf{x}$, centered around the mode $\mathbf{x}^*(\mathbf{z}_t)$:

$$E(\mathbf{x};\mathbf{z}_t) \approx E(\mathbf{x}^*;\mathbf{z}_t) + (\mathbf{x}-\mathbf{x}^*)^\top \nabla_\mathbf{x} E(\mathbf{x}^*;\mathbf{z}_t) + \frac{1}{2}(\mathbf{x}-\mathbf{x}^*)^\top \mathbf{H}(\mathbf{z}_t)(\mathbf{x}-\mathbf{x}^*). \tag{20}$$

Since $\mathbf{x}^*$ is a local minimum of the energy surface defined by $\mathbf{z}_t$, the gradient $\nabla_\mathbf{x} E(\mathbf{x}^*;\mathbf{z}_t)$ vanishes. The matrix $\mathbf{H}(\mathbf{z}_t) = \nabla^2_\mathbf{x} E(\mathbf{x}^*;\mathbf{z}_t)$ is the Hessian of the energy, representing the local curvature or *sharpness* of the energy basin conditioned on $\mathbf{z}_t$.

Using Eq. (20) and assuming that we are at the minimum where $\nabla_\mathbf{x} E(\mathbf{x}^*;\mathbf{z}) = \mathbf{0}$, we can perform Laplace approximation for the partition function $Z(\mathbf{z}_t)$:

$$\begin{aligned} Z(\mathbf{z}_t) &\approx \int e^{-\left(E(\mathbf{x}^*;\mathbf{z}_t) + \frac{1}{2}(\mathbf{x}-\mathbf{x}^*)^\top \mathbf{H}(\mathbf{z}_t)(\mathbf{x}-\mathbf{x}^*)\right)} \mathrm{d}\mathbf{x} \\ &= e^{-E(\mathbf{x}^*;\mathbf{z}_t)} \int e^{-\frac{1}{2}(\mathbf{x}-\mathbf{x}^*)^\top \mathbf{H}(\mathbf{z}_t)(\mathbf{x}-\mathbf{x}^*)} \mathrm{d}\mathbf{x} \\ &= (2\pi)^{\frac{d}{2}}\, e^{-E(\mathbf{x}^*;\mathbf{z}_t)} \det\big(\mathbf{H}(\mathbf{z}_t)\big)^{-\frac{1}{2}} \end{aligned} \tag{21}$$

The conditional entropy $\mathcal{H}(\mathbf{x}|\mathbf{z}_t)$ can now be defined. Using the relationship $\log p = -E - \log Z$ and substitute it into $\mathcal{H}(\mathbf{x}|\mathbf{z}_t)$, we have

$$\begin{aligned} \mathcal{H}(\mathbf{x}|\mathbf{z}_t) &= \mathop{\mathbb{E}}_{\mathbf{x}\sim p(\mathbf{x}|\mathbf{z}_t)} \Big[ E(\mathbf{x};\mathbf{z}_t) + \log Z(\mathbf{z}_t) \Big] \\ &= \mathop{\mathbb{E}}_{\mathbf{x}\sim p(\mathbf{x}|\mathbf{z}_t)} \Big[ E(\mathbf{x};\mathbf{z}_t) + \frac{d}{2}\log(2\pi) - E(\mathbf{x}^*;\mathbf{z}_t) - \frac{1}{2}\log\big[\det\big(\mathbf{H}(\mathbf{z}_t)\big)\big] \Big] \end{aligned} \tag{22}$$

If we substitute Eq. (20) into the term $E(\mathbf{x};\mathbf{z}_t)$ in Eq. (22), we then have

$$\begin{aligned} \mathcal{H}(\mathbf{x}|\mathbf{z}_t) &\approx \mathop{\mathbb{E}}_{\mathbf{x}\sim p(\mathbf{x}|\mathbf{z}_t)} \Big[ \cancel{E(\mathbf{x}^*;\mathbf{z}_t)} + \frac{d}{2}\log(2\pi) \cancel{- E(\mathbf{x}^*;\mathbf{z}_t)} - \frac{1}{2}\log\big[\det\big(\mathbf{H}(\mathbf{z}_t)\big)\big] + C \Big] \\ &\approx \mathop{\mathbb{E}}_{\mathbf{x}\sim p(\mathbf{x}|\mathbf{z}_t)} \Big[ -\frac{1}{2}\log\big[\det\big(\mathbf{H}(\mathbf{z}_t)\big)\big] + C \Big] \end{aligned} \tag{23}$$

where $C$ is a constant involving the omitted terms from our substitution of Eq. (20).

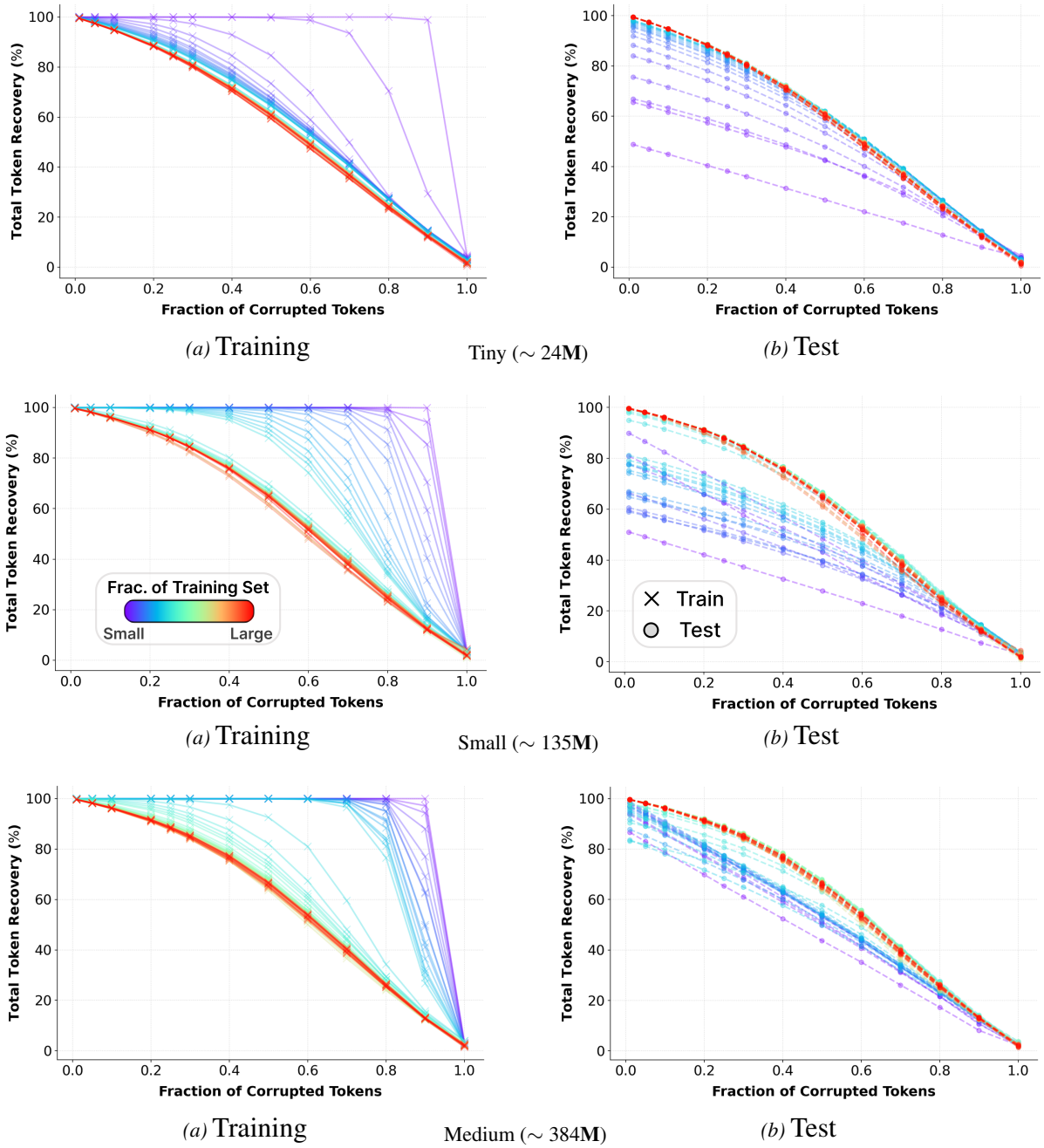
**Discussion**. Overall, this derivation highlights that the conditional entropy is inversely proportional to the log-determinant of the Hessian at the mode, and aligns well to the findings of (Biroli et al., 2024) where a collapse in entropy corresponds to the system getting trapped in small-disjoint regions of the configuration space. However, in this work, we are exploring UDDMs, which are not continuous DMs. Thus, we lack the formulations that attempt to link up their conditional entropy with the sharpness in the discrete setting of language or text modeling. But we suspect that the connection between Uniform and Gaussian distributions in the discrete setting, laid out by (Sahoo et al., 2025), provides some clues to further this link between conditional entropy and energy in a future work.

## D. Usage Of LLM Disclaimer

LLMs were utilized in part of the writing of the manuscript, specifically for fine-tuning sentences and syntax correction.

## E. Additional Results

### E.1. Shrinkage and Expansion of Basins



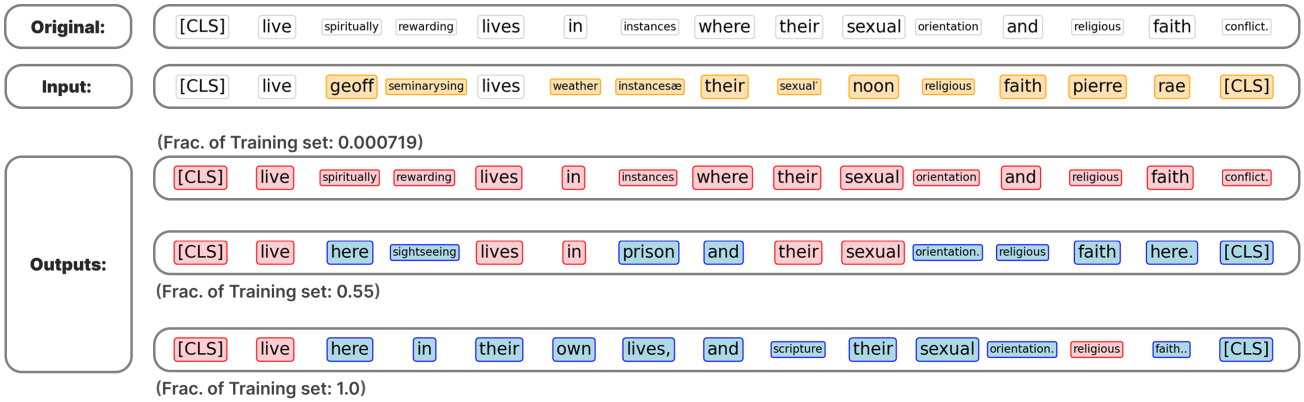

*(a)* Training Tiny (∼ 24**M**) *(b)* Test

*(a)* Training Small (∼ 135**M**) *(b)* Test

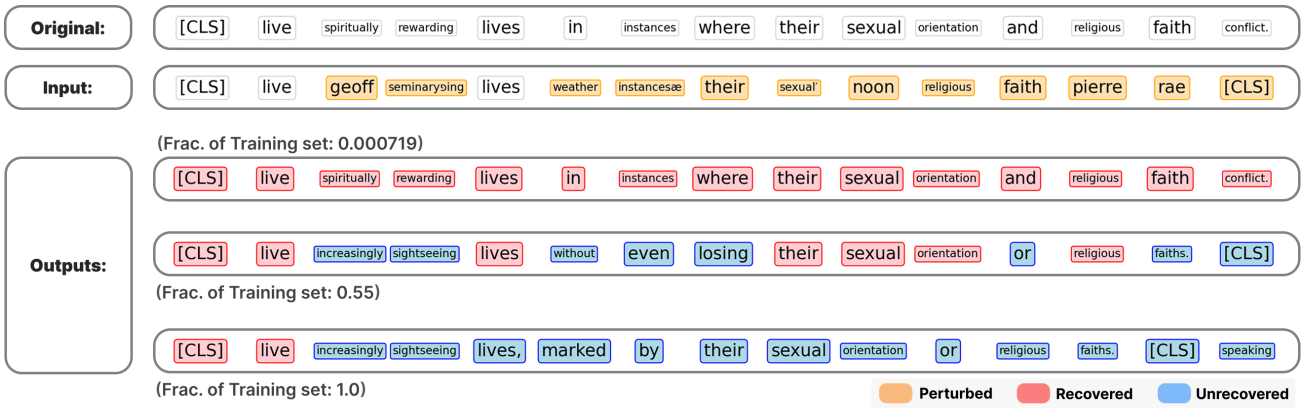
*(a)* Training Medium (∼ 384**M**) *(b)* Test

*Figure 9.* **Shrinkage and expansion of training and test samples' basins of attraction during the memorization-to-generalization transition for various UDDM sizes**. Total token recovery rates (%), including non-corrupt and corrupted, are shown across varying levels of corruption for (**a**) training sequences and (**b**) unseen test samples. The color gradient represents the fraction of the training dataset used, ranging from small to large. As the training dataset grows, the recovery rate for training examples diminishes, reflecting a shrinkage in the basins of attraction around explicitly memorized points. Simultaneously, the recovery rate for unseen test samples improves, demonstrating an expansion of the basins of attraction around novel examples within the broader data distribution. The convergence of these recovery rates at large dataset sizes (red curves) signifies that unseen test samples have effectively become stable attractors, marking the shift from pure memorization to generalization.

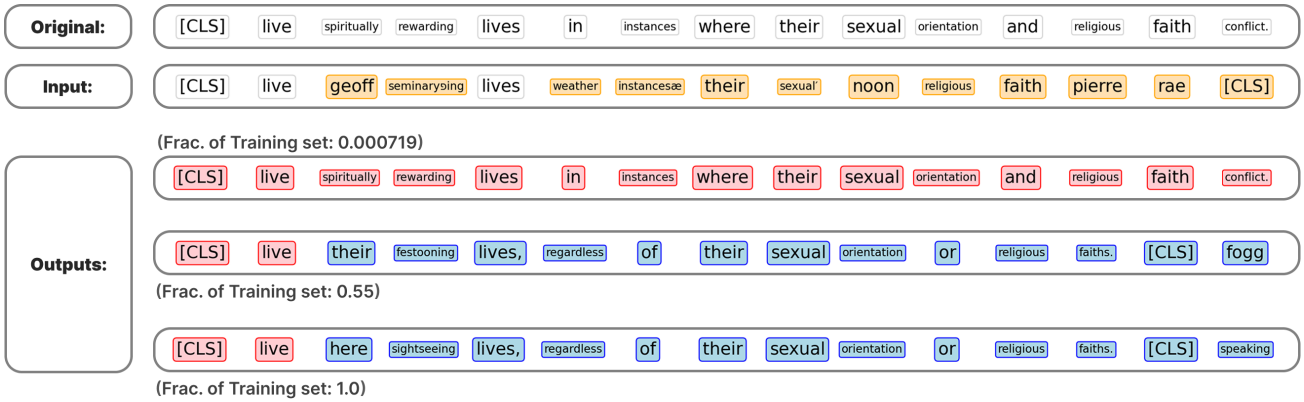
## E.2. Visualizations of Text Recovery Examples




*(a)* Tiny ($\sim$ 24**M**)

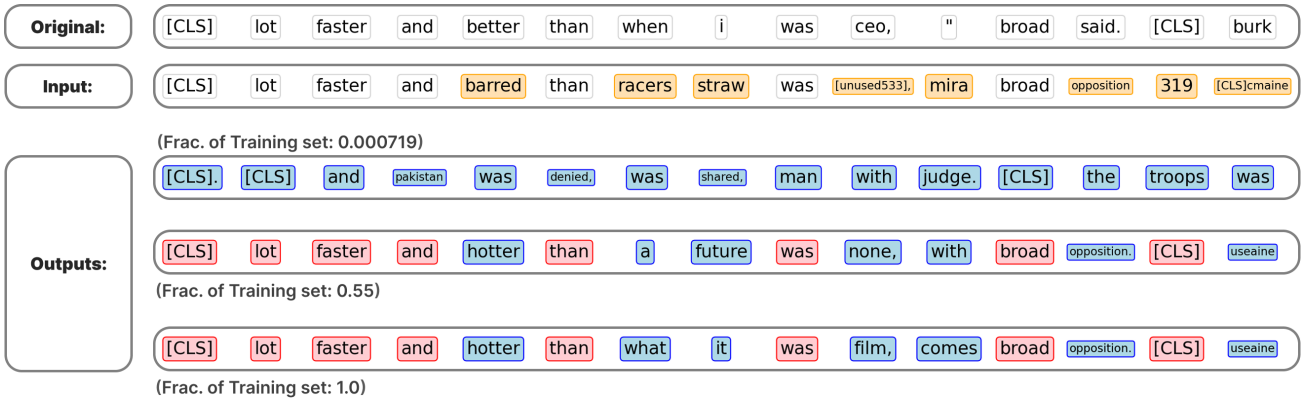



*(b)* Small ($\sim$ 135**M**)

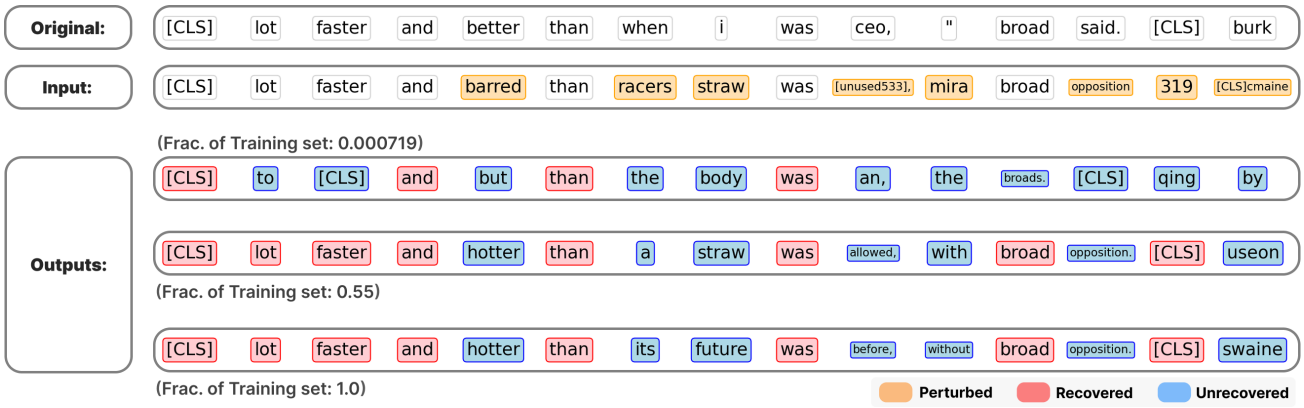



*(c)* Medium ($\sim$ 384**M**)

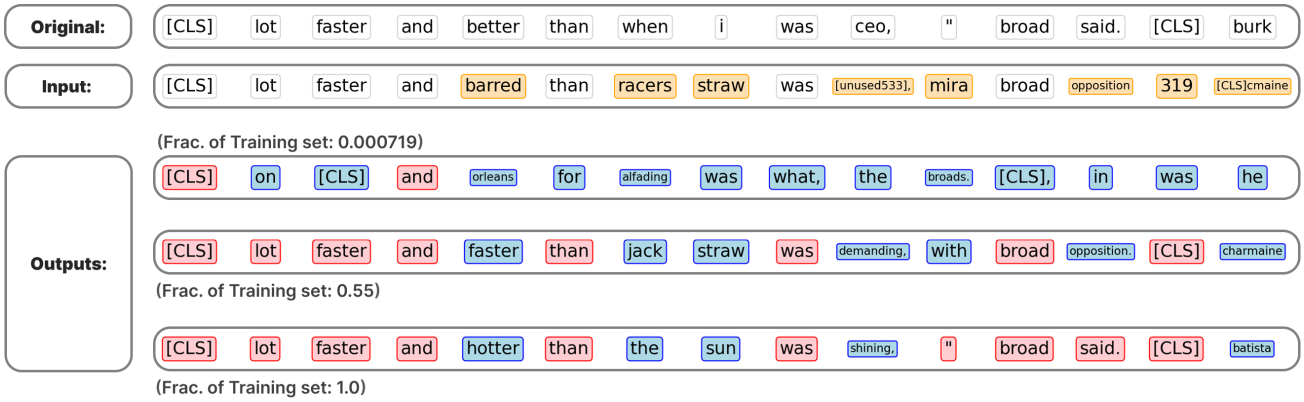
*Figure 10.* An illustration of the model's ability to recover tokens from perturbed sequences on **training examples** at three different fractions of the training dataset and sizes of the UDDMs. Perturbation is computed at $t = 0.5$ and the typical stochastic reverse process is performed afterwards. As the training dataset size increases, the model's ability to recover perturbed tokens becomes worse and in contrast, its generative ability improves. Meanwhile, there is no distinction among the various UDDMs' sizes, where the overall trend of memorization and generalization in relation to the training dataset size persists.




*(a)* Tiny ($\sim$ 24**M**)




*(b)* Small ($\sim$ 135**M**)




*(c)* Medium ($\sim$ 384**M**)

*Figure 11.* An illustration of the model's ability to recover tokens from perturbed sequences on **test examples** at different fractions of the training dataset and sizes of UDDMs. Perturbation is computed at $t = 0.5$ and the stochastic reverse process is performed afterwards. In the beginning, the model is unable to recognize unperturbed unseen test tokens, where it often change them to another tokens. However, as the training dataset size increases, the UDDM is more likely to maintain the unperturbed test tokens rather than 'flipping' them.

## E.3. Conditional Entropy Histograms

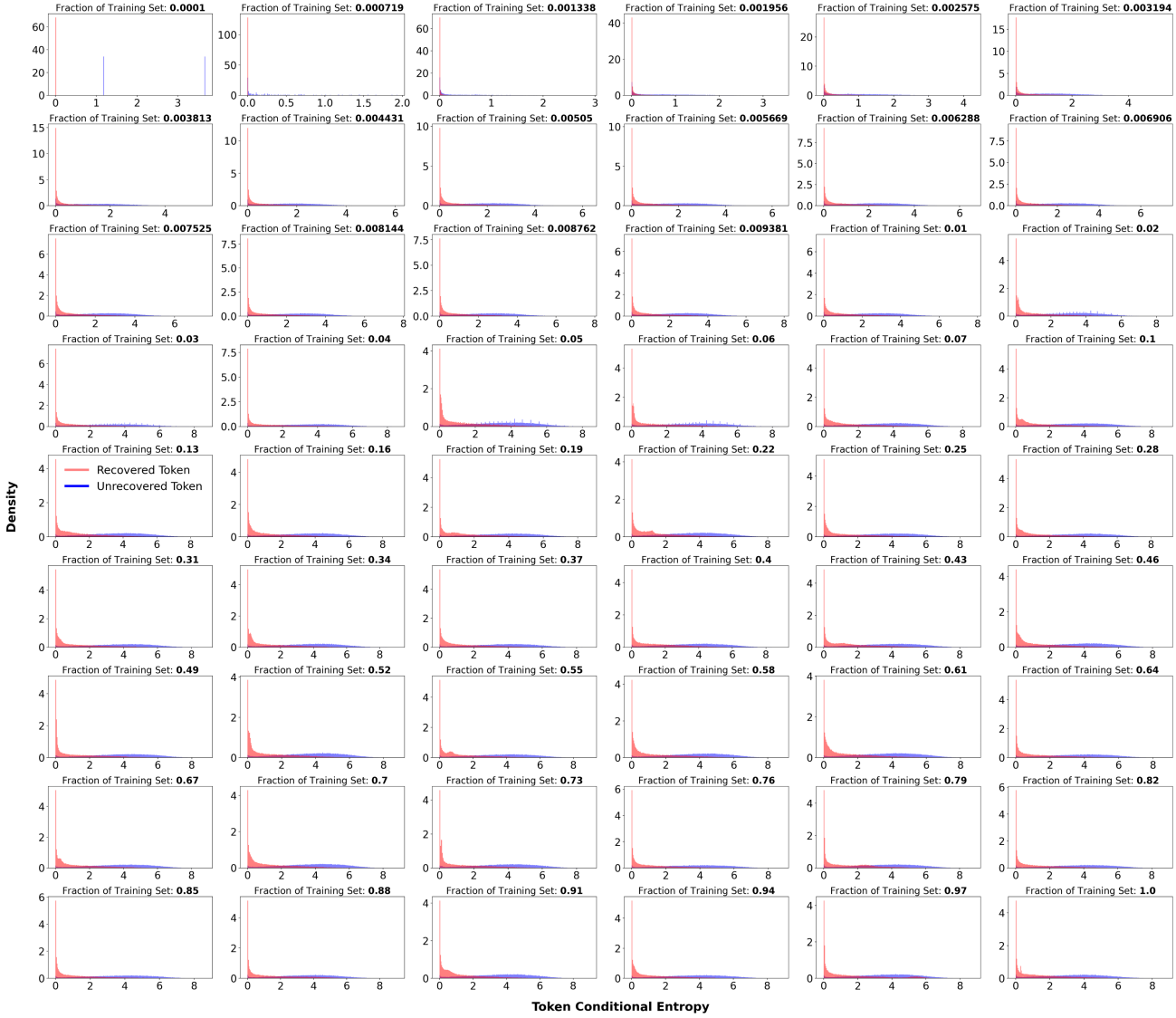



*Figure 12.* An illustration of the density of conditional entropy for two categories of tokens, *recovered* and *unrecovered*, computed at $t = 0.25$ for the *Tiny* model. The subplots are ordered by the fraction of training dataset, ranging from $10^{-4}$ (top-left) to 1.0 (bottom-right). As the fraction of training data increases, recovered tokens concentrate near zero entropy (high confidence), while unrecovered tokens exhibit a broad distribution at higher entropy.

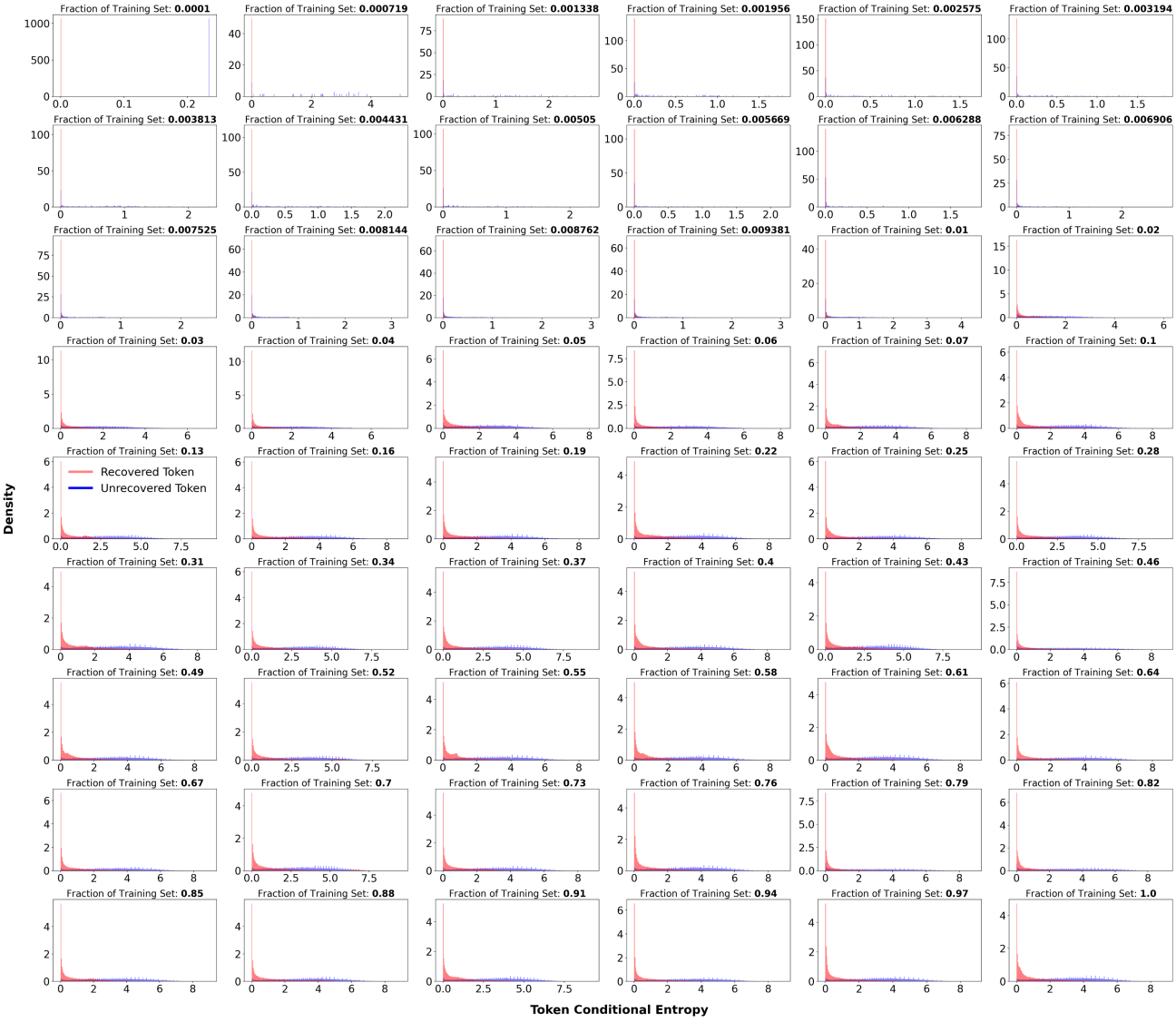



*Figure 13.* An illustration of the density of conditional entropy for two categories of tokens, *recovered* and *unrecovered*, computed at $t = 0.25$ for the *Small* model. The subplots are ordered by the fraction of training dataset, ranging from $10^{-4}$ (top-left) to 1.0 (bottom-right). As the fraction of training data increases, recovered tokens concentrate near zero entropy (high confidence), while unrecovered tokens exhibit a broad distribution at higher entropy.

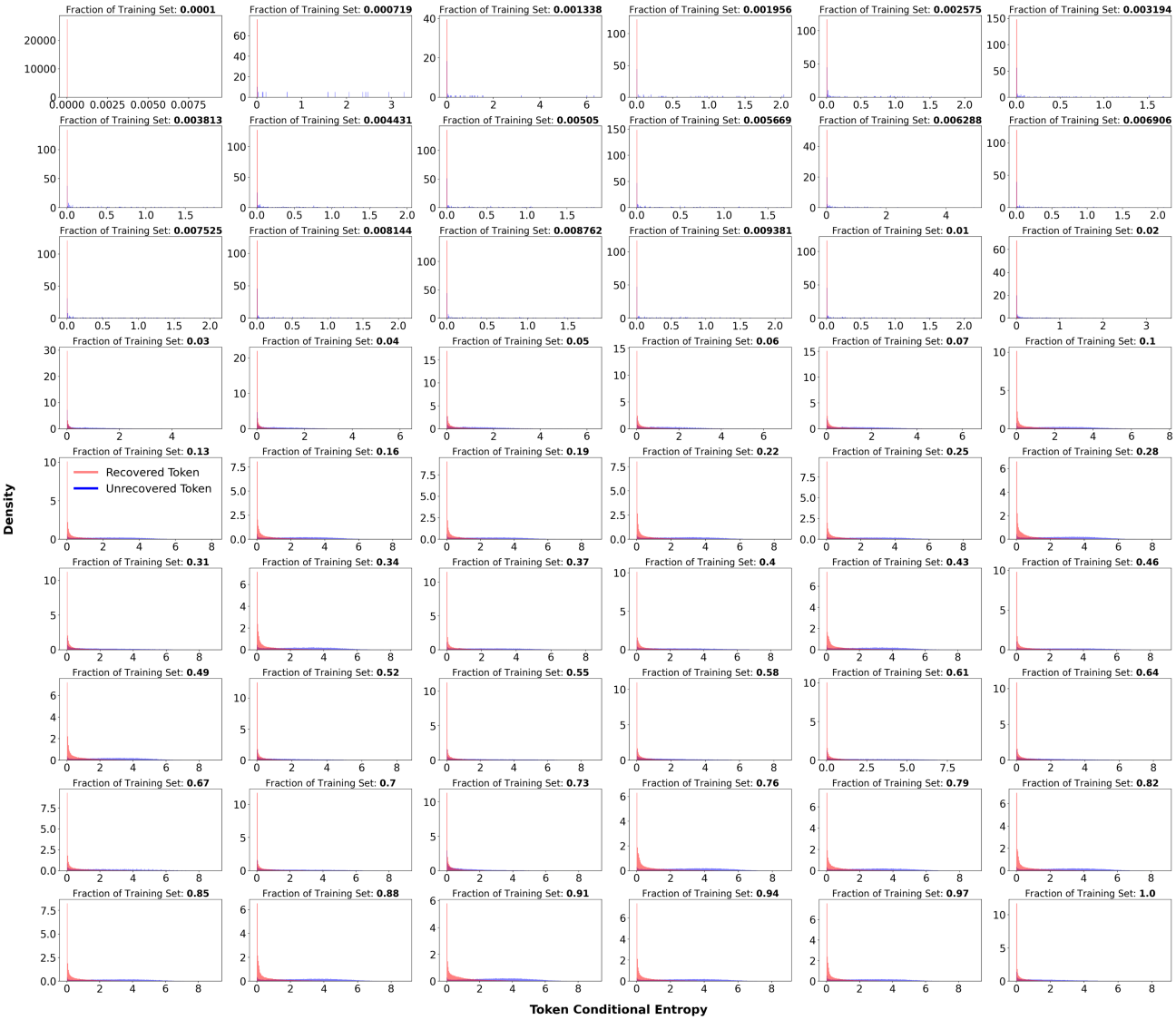



*Figure 14.* An illustration of the density of conditional entropy for two categories of tokens, *recovered* and *unrecovered*, computed at $t = 0.25$ for the *Medium* model. The subplots are ordered by the fraction of training dataset, ranging from $10^{-4}$ (top-left) to 1.0 (bottom-right). As the fraction of training data increases, recovered tokens concentrate near zero entropy (high confidence), while unrecovered tokens exhibit a broad distribution at higher entropy.

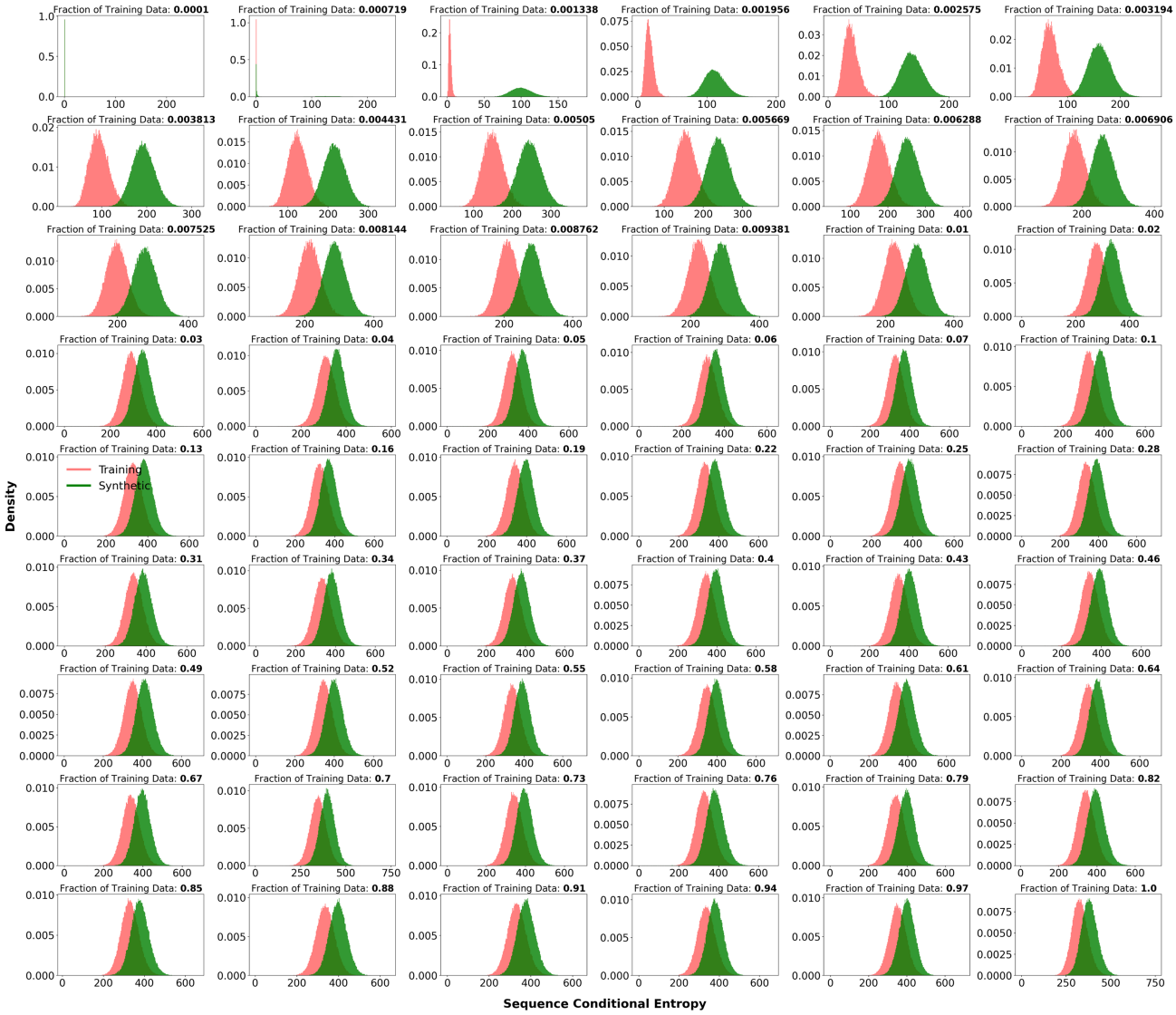



*Figure 15.* An illustration of the evolution of the density of the average conditional entropy for the probabilities of training and synthetic sequences respectively, computed at $t = 10^{-5}$ using the *Tiny* models, as the training dataset size grows. When the fraction of training set is small, there exists a separation in the average conditional entropies of training and synthetic samples. However, as the training dataset size grows, this separation is reduced and the conditional entropy of synthetic samples becomes similar to that of the training samples.

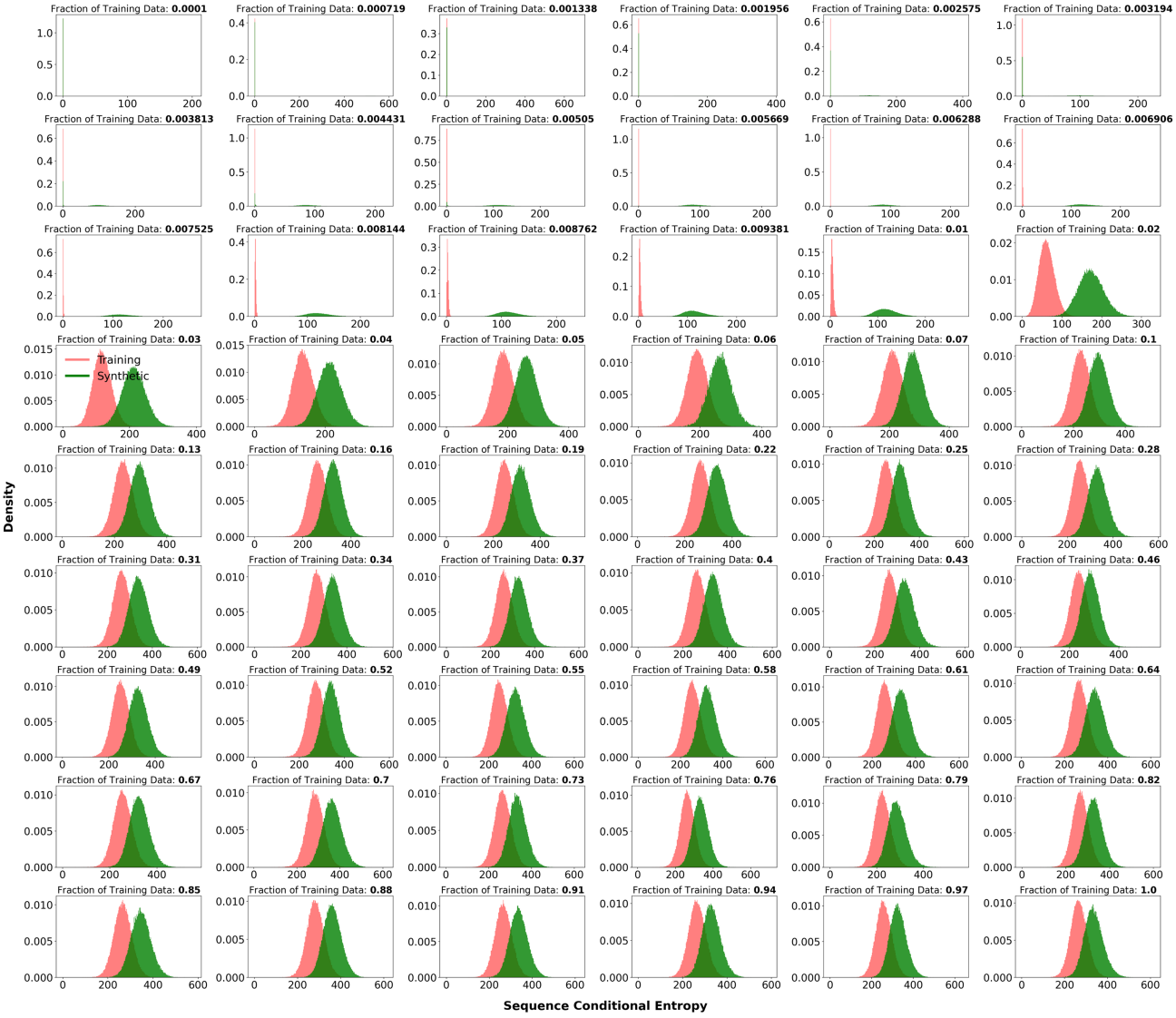



*Figure 16.* An illustration of the evolution of the density of the average conditional entropy for the probabilities of training and synthetic sequences respectively, computed at $t = 10^{-5}$ using the *Small* models, as the training dataset size grows. When the fraction of training set is small, there exists a separation in the average conditional entropies of training and synthetic samples. However, as the training dataset size grows, this separation is reduced and the conditional entropy of synthetic samples becomes similar to that of the training samples.

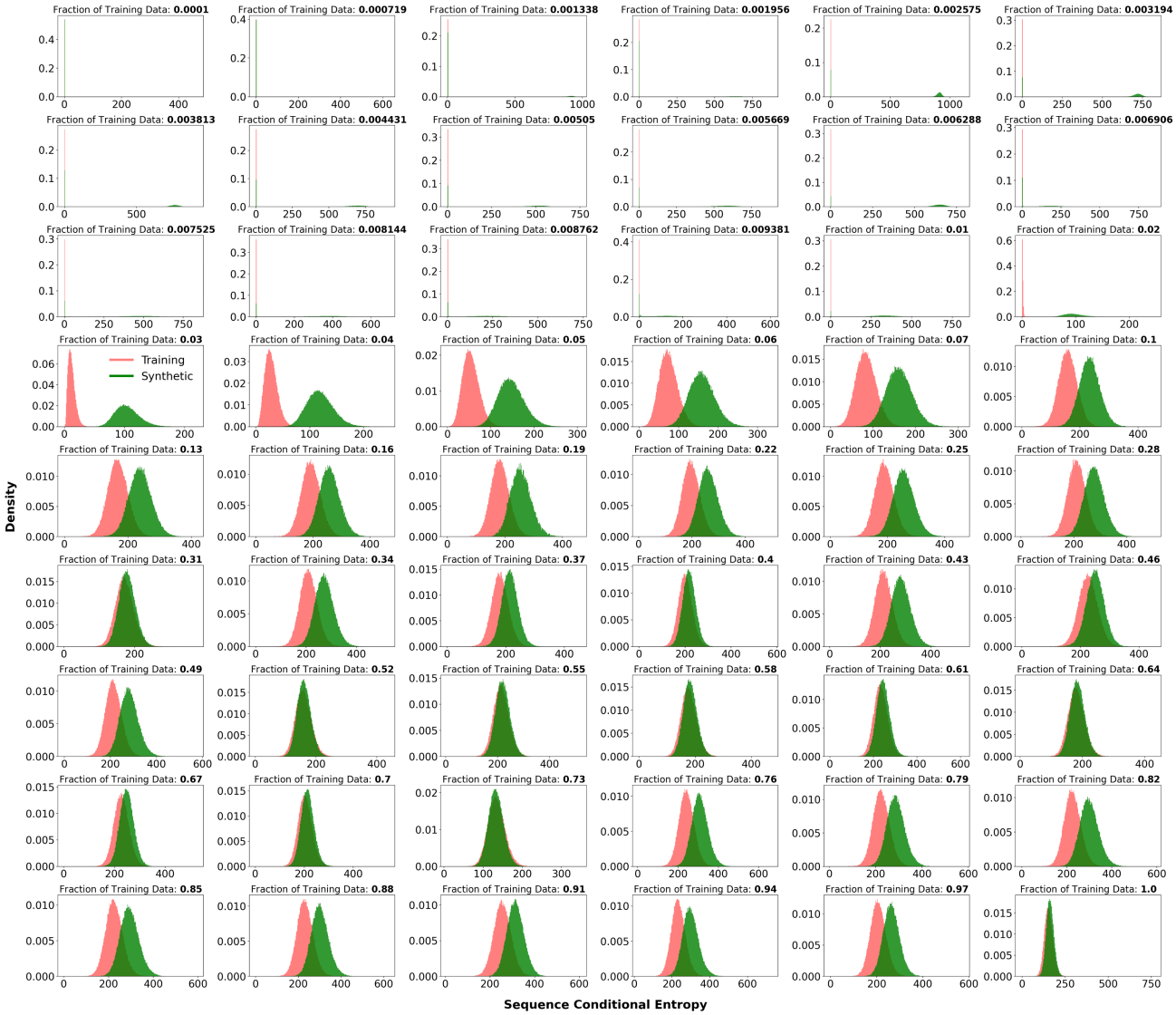



*Figure 17.* An illustration of the evolution of the density of the average conditional entropy for the probabilities of training and synthetic sequences respectively, computed at $t = 10^{-5}$ using the *Medium* models, as the training dataset size grows. When the fraction of training set is small, there exists a separation in the average conditional entropies of training and synthetic samples. However, as the training dataset size grows, this separation is reduced and the conditional entropy of synthetic samples becomes similar to that of the training samples. Due to larger model size, the UDDM exhibits stronger memorization (and chaotic) behaviors, where the two distributions initially merge at certain fractions of the training dataset and later diverge. Nonetheless, with the full training set, both distributions overlap each other at the end.