Language Diffusion Models are Associative Memories Capable of Retrieving Unseen Data

Anonymous ACL submission

Abstract

Understanding when language diffusion models memorize training data versus generalize remains an open challenge. We address it by framing Uniform-based Discrete Diffusion Models (UDDMs) as Associative Memories (AMs) where both memorization and creativity coexist. Unlike Hopfield networks, which require an energy function to produce attractors, we show that attractors can be formed via conditional likelihood maximization. We observe that UDDMs treat both training and unseen test sequences as fixed points for denoising while retaining the capacity to synthesize novel texts. We demonstrate that, as training set size increases, attractor basins around training samples shrink while those around unseen test samples expand until they become indistinguishable. This transition can be detected via the conditional entropy of tokens, which vanishes during memorization. Our framework provides both a theoretical foundation and a practical metric to elucidate the generative regimes of UDDMs.

1 Introduction

Generative diffusion models (Sohl-Dickstein et al., 2015) have set new standards for image and video generation (Ho et al., 2020; Song and 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 generalization 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 and 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. For instance, 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, by producing distinct basins 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 having both factual recall and creative behaviors resembles the co-existence of memory attractors and novel attractors in AMs. For this reason, it is useful to describe the generalization phase of UDDMs from the perspective of AMs, by focusing on the

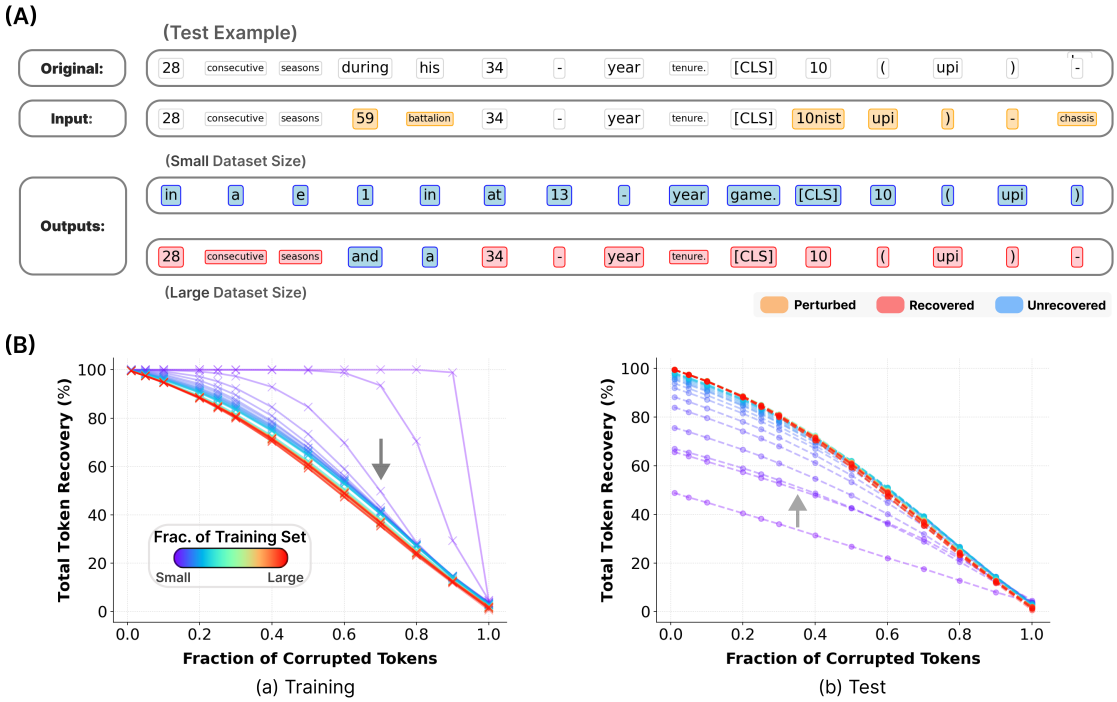
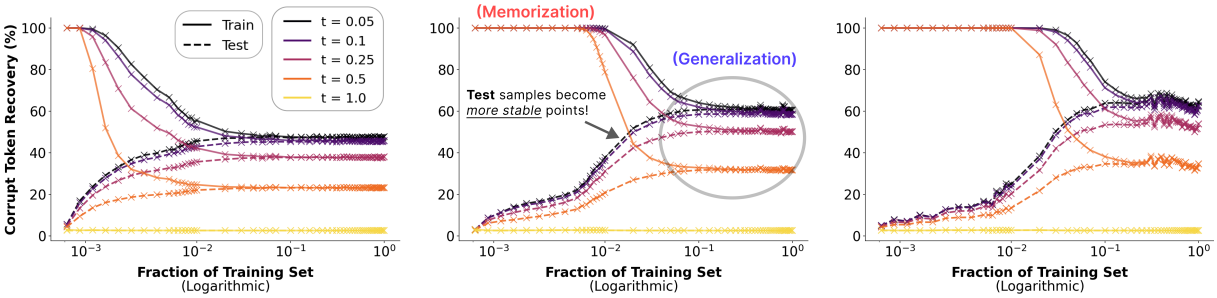


Figure 1: **Basins around training examples shrink and basins around test examples expand as the training dataset size increases**. **(A)** Textual examples showing two *Tiny* ($\sim$ 24**M**) UDDMs' token recovery at noise level $t = 0.2$, where each is trained on two different training dataset sizes. With a small training dataset, the model fails to recognize unseen test sequences and alters all tokens. With a larger training set, the tokens of unseen sequences however become stable and recoverable by 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 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 training set sizes (red curves) marks the transition from memorization to generalization. *Note: Deterministic (greedy) sampling was used across these experiments to isolate from stochastic noise.*

retrieval of seen and unseen data points. 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 function, as seen in Hopfield networks (Hopfield, 1982; Amari, 1972) and Dense Associative Memories (Krotov and Hopfield, 2016, 2021). However, this reliance on an energy function is not strictly necessary to guarantee attractor dynamics. As shown by (D'Amico et al., 2026), conditional likelihood maximization alone produces basins around the data points. Importantly, 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 the 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 attractor basins 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 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,



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

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 sequences (**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.

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 the 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 tokens of unseen test sequences 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 detect 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). Notably, conditional entropy has also been found to be crucial for scaling laws in autoregressive language models (Cagnetta et al., 2026), suggesting the versatility and usefulness of this metric in language modeling.

Finally, we validate our proposed metrics via the typical measurement of perplexity using an external critic model as commonly done in language modeling (Radford et al., 2019). We observe a strong alignment between these metrics, not just in the detection of memorization-to-generalization transition point, but also in the estimation of sample quality (see Fig. 5). The success of this validation indicates that out metrics are effective for assessing the generative capabilities of UDDMs, while also providing a principled geometric interpretation of such capabilities.

2 Uniform-State Discrete diffusion

Consider a clean token $\mathbf{x} \in \mathcal{V}$ drawn from the data distribution q_{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_{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; Gat et al., 2024; Sahoo et al., 2024, 2025):

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

where $\boldsymbol{\pi} \in \Delta$, $\text{Cat}(\cdot)$ denotes categorical distribution, and Δ 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

$$\mathbf{z}_s \sim q_{s|t}(\mathbf{z}_s|\mathbf{z}_t, \mathbf{x}) = \mathrm{Cat}\Big(\mathbf{z}_s; g(\mathbf{z}_t, \mathbf{x})\Big), \quad (2)$$

where $g(\mathbf{z}_t, \mathbf{x})$ parameterizes the categorical transition probabilities for the reverse step by combining $\mathbf{z}_t$, $\mathbf{x}$, and uniform noise according to the schedule coefficients. Here, 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), which we can formally define as the conditional probability distribution $p_\theta(\mathbf{x}|\mathbf{z}_t) = \mathbf{x}_\theta(\mathbf{z}_t, t)$. The network's parameters θ are optimized via the Negative Evidence Lower Bound (NELBO) objective (Austin et al., 2021; Sahoo et al., 2024):

$$\mathcal{L}_{\text{NELBO}} = \mathbb{E}_q\Big[\mathcal{L}_{\text{reconstruction}} + \mathcal{L}_{\text{diffusion}} + \mathcal{L}_{\text{prior}}\Big] \quad (3)$$

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. (C) for more details on this form of UDDMs, Eq. (2), and NELBO (3).

3 Associative Memories from Conditional Sampling

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* is that the denoising process of a sequence $\mathbf{z}^{1:L}$ of length L factorizes for each token $\mathbf{z}^\ell$ as

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

where $\psi^{\theta}_{s|t}$ denotes the conditional probability:

$$\psi^{\theta}_{s|t}(\mathbf{z}^{\ell}_s \mid \mathbf{z}^{1:L}_t) = \mathrm{Cat}\big(\mathbf{z}^{\ell}_s;\ f^{\ell}_{\theta}(\mathbf{z}^{1:L}_t)\big). \quad (5)$$

Here, $f^{\ell}_{\theta}(\mathbf{z}^{1:L}_t) = \mathrm{softmax}_K\big[\beta(t)\mathbf{u}^{\ell}_{\theta}(\mathbf{z}^{1:L}_t)\big]$, where $\mathbf{u}^{\ell}(\mathbf{z}^{1:L}_t)$ maps the sequence $\mathbf{z}^{1:L}_t$ to raw logits via a diffusion transformer (Peebles and Xie, 2023), and the softmax is applied over the K-categories per position ℓ. The inverse temperature $\beta(t)$ depends on the diffusion schedule $\alpha(t)$ and typically increases as $t \to 0$. These conditional probabilities enter the cross-entropy terms in NELBO (3). This reliance on cross-entropy establishes the connection to AMs, where minimizing this loss implicitly creates basins of attraction around data points during conditional sampling (D'Amico et al., 2026), and translates with minimal changes to UDDMs.

3.1 Simpler Setting: Binary Variables and Linear Logits

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:

$$\mathbf{s}^{\ell}_{\tau} = \mathrm{sgn}\Bigg(\sum_{m=1}^{L} \mathbf{W}^{\ell m}\, \mathbf{s}^{m}_{\tau+1}\Bigg), \quad (6)$$

where we adopted the notation of the discrete time τ 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 instead needed:

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

where the 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 and Derrida, 1988; Forrest, 1988; Benedetti et al., 2022). Larger κ 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),

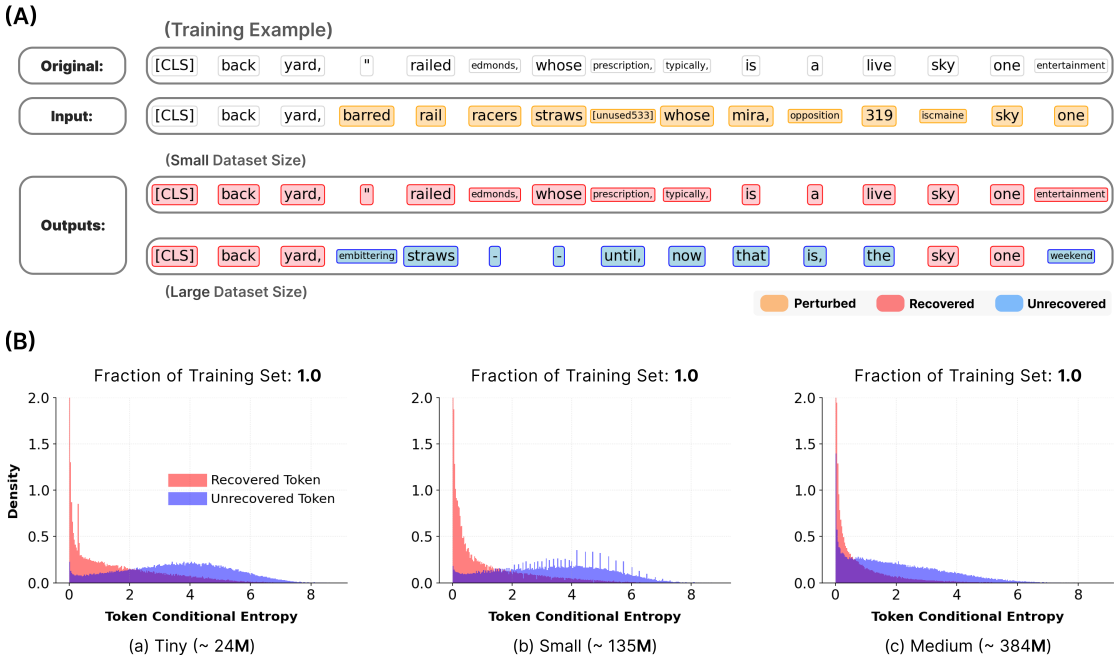


Figure 3: **Token-level conditional entropy highlights different token recovery behaviors**. **(A)** Textual examples showing how two *Medium* UDDMs, trained on two different training dataset sizes, recover corrupted training sequences at noise level $t = 0.5$. The model trained on a small training set 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 persist even during the generalization phase. *Note: The y-axis is clipped to better contrast these two token types.*

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:

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

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

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

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 a fixed inverse temperature β[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 that is also called pseudo-likelihood (Besag, 1974):

$$\begin{aligned}\mathcal{L}(W) &= -\frac{1}{P}\sum_{\mathbf{x}\in\Xi} \log \prod_{l=1}^{L} \psi(\mathbf{x}^l|\mathbf{x}^{1:L}; \mathbf{W}^l) \\ &= -\frac{1}{P}\sum_{\mathbf{x}\in\Xi}\sum_{l=1}^{L}\Big[\mathbf{x}^l f^l_{\mathbf{W}}(\mathbf{x}^{1:L}) \\ &\qquad - \log 2\cosh\left(f^l_{\mathbf{W}}(\mathbf{x}^{1:L})\right)\Big]\end{aligned} \tag{10}$$

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

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

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

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 $\mathbf{W}$:

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

Here, we denote the local classification margin as $M^\ell(\mathbf{x}^{1:L}) = \mathbf{x}^\ell f^\ell_{\mathbf{W}}(\mathbf{x}^{1:L})$. The Hebbian term is factored out via identity $\tanh(f^\ell_{\mathbf{W}}(\mathbf{x}^{1:L})) = \tanh(\mathbf{x}^\ell M^\ell(\mathbf{x}^{1:L})) = \mathbf{x}^\ell \tanh(M^\ell(\mathbf{x}^{1:L}))$, given $\mathbf{x}^\ell\mathbf{x}^\ell = 1$. This yields two distinct gradient terms: a classical Hebbian learning (Hebb, 1949) component for pattern storage and a margin-modifying component. Because 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, the objective concentrates learning on samples with the smallest margins. The loss is thus minimized when all margins are maximized, driving weight magnitudes to diverge in the separable regime (Soudry et al., 2018).

Main differences with UDDMs. The above 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 to the continuous setting (Ambrogioni, 2024; Pham et al., 2025). From this perspective, we suspect 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 Methods

As shown in D'Amico et al. (2026) and Kalaj et al. (2025), the AM produced by conditional-likelihood (10) is capable of generalization: when trained on a sufficiently large dataset, the system creates 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, found in AMs, can appear in UDDMs. Consequently, we design 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.

Experiment 1: Deterministic retrieval from reference examples. To emulate the zero-temperature retrieval dynamics of an AM without the confounding effects of stochastic generation noise, we establish a specialized baseline setting. As reported in Fig. (1), we initialize the reverse process from corrupted training and test examples, and replace the standard stochastic sampling with 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 always set to $t = 1$ while the fraction of corrupted input tokens is varied, allowing us to cleanly measure the token recovery rate as a function of the input noise level.

Experiment 2: Stochastic retrieval from reference examples. To understand how the standard diffusion dynamics alter the stability and structural boundaries of these basins under realistic operating conditions, we introduce stochasticity back into the UDDMs. 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 typical UDDM scheme, where each value of t determines both the fraction of corrupted tokens and the time step at which the reverse process is initialized, providing a dynamic look at token recovery across varying levels of input noise.

Experiment 3: Standard generative process. Finally, we run the full standard reverse dynamics starting from *noise*, independent of any reference sequences, e.g., training or test sequences, in Figs. (4) and (5). Because ground-truth samples are unavailable, token recovery rate is no longer a viable metric. Instead, we use conditional entropy (13) as our metric. In Fig. (5) we validate the performance of UDDM across the memorization-to-generalization transition by measuring the perplexity of the generated sequences with GPT-2 Large (Radford et al., 2019) used as the critic model.

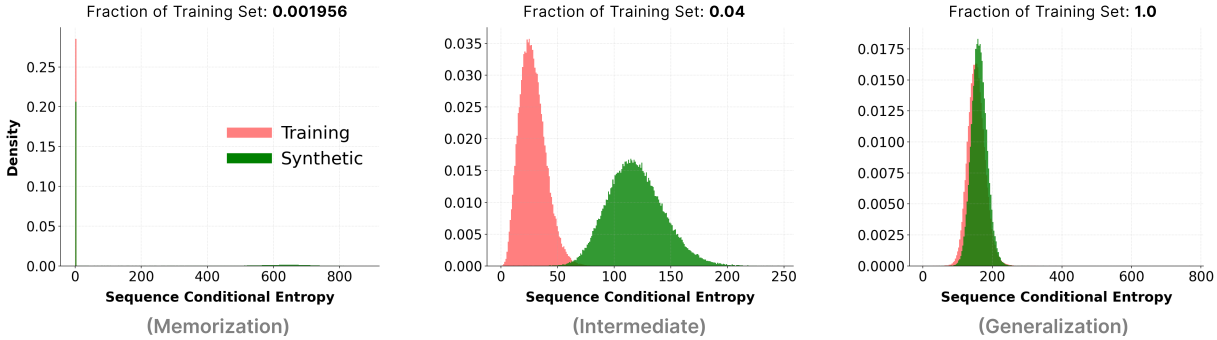


Figure 4: **Sequence conditional entropy highlights the memorization to generalization transition**. The histograms compare the sequence entropy distributions of training and synthetic (or generated) sequences respectively, for three selected fractions of the training set for the *Medium* UDDMs. At a 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. See Appx. (E) for the full range of histograms across various model sizes.

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) and Appx. (E), respectively.

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 $t \in (0, 1]$:

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

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), $\mathcal{M} = \{j : \delta(\mathbf{x}^j, \mathbf{z}_t^j) = 0\}$ is the set of indices in the input sequence $\mathbf{z}_t^{1:L}$ 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.

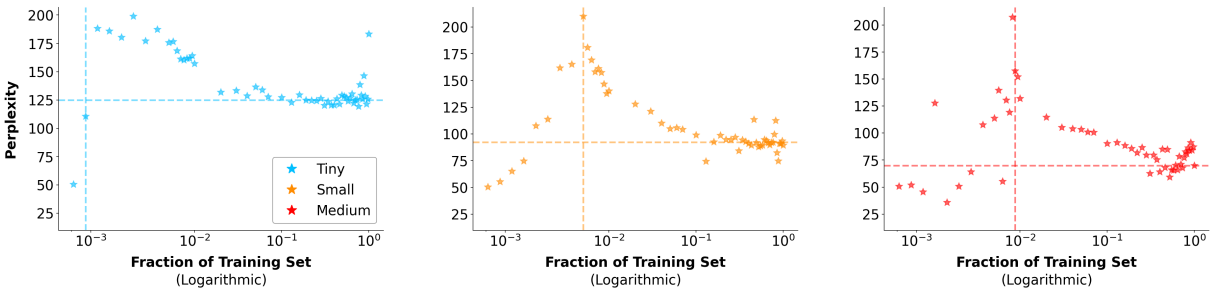
Conditional Entropy. While token recovery rates indicate whether the system successfully returns to an attractor, UDDMs also provide direct access to conditional likelihood, allowing us to probe the local geometry and sharpness of these basins. Here, the conditional entropy of an individual token $\mathbf{x}^\ell$ in a sequence, given its associated perturbed sequence $\mathbf{z}_t^{1:L}$ at some time t is defined as:

$$\mathcal{H}(\mathbf{x}^\ell | \mathbf{z}_t^{1:L}) = -\sum_{k=1}^{K} \left[\mathbf{x}_\theta^\ell(\mathbf{z}_t^{1:L}, t)\right]_k \log \left[\mathbf{x}_\theta^\ell(\mathbf{z}_t^{1:L}, t)\right]_k, \qquad (13)$$

where $\mathbf{x}_\theta^\ell(\mathbf{z}_t^{1:L}, t)$ is the output of the diffusion transformer for position ℓ 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., $\mathcal{H}(\mathbf{x}^{1:L} \mid \mathbf{z}_t^{1:L}) = \sum_{\ell=1}^{L} \mathcal{H}(\mathbf{x}^\ell \mid \mathbf{z}_t^{1:L})$. Please refer to Appx. (D) for our discussion on the relationship between conditional entropy and energy curvature (or sharpness).

5 Results

Our empirical analysis demonstrates that, as the training set size increases, basins around training examples shrink and basins around test examples expand (see Fig. 1). Changes in the structure of basins are reflected in token recovery rates: for larger datasets the token recovery rate for training examples drops (see Fig. 2, *solid* lines), while the recovery rate for unseen test samples increases (*dashed* lines in Fig. 2). For a certain size onward, the retrieval behaviors of both seen and unseen sequences converge to an identical recovery rate, mirroring the convergence of training and test basins and completing the transition from memorization to generalization. Scaling up the model size delays this transition, consistently with continuous DMs (Yoon et al., 2023; Pham et al., 2025). Moreover, as shown in Fig. (3), successful token recovery is char-



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

Figure 5: **Perplexity trends during the memorization-to-generalization transition**. Evaluation of UDDMs across different model sizes as the fraction of the training set increases. The *vertical* dashed lines roughly illustrate the transition point where the model switches from memorization to generalization: perplexity starts at a very low value during the memorization regime, increases as the training dataset size grows, and it then decreases to a lower value upon reaching generalization. The *horizontal* dashed lines highlight the converging perplexity value, showing that scaling up the model leads to improved perplexity at the end of this transition. The precise position of each vertical line is the last point where the recovery rate is 100% for training tokens in Fig. (2). *Note: Perplexity is computed using GPT-2 Large (Radford et al., 2019) as the critic model.*

acterized by near-zero conditional entropy, whereas unrecovered tokens exhibit higher and non-zero conditional entropy value. Interestingly, a distinct fraction of highly stable and low-entropy tokens persists even at the later generalization phase, suggesting that there are tokens which are very stable and less likely to change through the generative process. Using the average *sequence-level* conditional entropy of training and generated sequences, we observe a conditional *entropy gap* at the intermediate regime of the transition (see Fig. 4): at very low fractions of the training set, the conditional entropy distributions of training and generated sequences coincide and peak near zero. In the intermediate regime, these distributions however separate from each other until the training set size is large enough, and once again overlap each other (but at *non-zero* values of conditional entropy).

Perplexity benchmarks using GPT-2 Large validate these phase transitions (see Fig. 5): we observe that the region where perplexity increases (to the *left* of the transition) corresponds to the memorization phase, which suggest that the perplexity is initially low due to the model's duplication of training sequences. Conversely, the region where perplexity decreases (to the *right* of the transition) corresponds to the generalization phase where basins around test sequences start forming, suggesting that these novel basins are related to the generative capabilities of the model. The correspondence between token recovery rate in Fig. (2) and perplexity in Fig. (5) also holds for the rightmost region where the perplexity plateaus: the length of the plateaus is similar in both figure. The plateaus of lower perplexity correspond to the converging token recovery rates, see Fig. (16) in Appx. (E) for a combined version of Figs. (2) and (5). This alignment suggests that token recovery rate captures the same underlying phenomenon: it identifies the generative regime without an external model and offers a geometrical interpretation, where generalization is defined by the evolution of basins around training and test sequences.

6 Conclusion

By framing UDDMs as AMs, we show that an explicit energy function is not required to guarantee attractor dynamics. Instead, basins can form through conditional likelihood maximization alone, which reveals a sharp memorization-to-generalization transition governed by the training set size. As training dataset scales up, basins around training examples shrink while basins around unseen test sequences expand until they become indistinguishable. This evolution allows the model to recover both seen and unseen sequences from perturbations, showing a form of co-existence of memory and creativity. Furthermore, token recovery rate and conditional entropy serve as complementary geometric probes of this transition: conditional entropy vanishes during memorization but remains finite during generalization, providing a measurable entropy gap between training and synthetic sequences that narrows with model scale. Larger models delay this transition, requiring more data to generalize before both metrics stabilize past the phase boundary.

Limitations

This work opens some directions for future investigation. First, how good the new proposed metrics are in practically relevant cases remains an open question: while we showed a promising consistent picture with the perplexity measurement, an extensive analysis of how token recovery rate and conditional entropy correlate with standard evaluation metrics would be needed to consolidate 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 conceptual and practical challenges beyond the scope of this work.

References

Beatrice Achilli, Luca Ambrogioni, Carlo Lucibello, Marc Mézard, and Enrico Ventura. 2025. Memorization and generalization in generative diffusion under the manifold hypothesis. *arXiv preprint arXiv:2502.09578*.

Beatrice Achilli, Enrico Ventura, Gianluigi Silvestri, Bao Pham, Gabriel Raya, Dmitry Krotov, Carlo Lucibello, and Luca Ambrogioni. 2024. Losing dimensions: Geometric memorization in generative diffusion. *arXiv preprint arXiv:2410.08727*.

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

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

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

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

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

Julian Besag. 1974. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, 36(2):192–236.

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

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Francesco Cagnetta, Allan Raventós, Surya Ganguli, and Matthieu Wyart. 2026. Deriving neural scaling laws from the statistics of natural language. *arXiv preprint arXiv:2602.07488*.

Andrew Campbell, Jason Yim, Regina Barzilay, Tom Rainforth, and Tommi Jaakkola. 2024. Generative flows on discrete state-spaces: Enabling multimodal flows with applications to protein co-design. *arXiv preprint arXiv:2402.04997*.

Nicholas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramèr, Borja Balle, Daphne Ippolito, and Eric Wallace. 2023. Extracting training data from diffusion models. In *Proceedings of the 32nd USENIX Conference on Security Symposium*, SEC '23, USA. USENIX Association.

Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. 2013. One billion word benchmark for measuring progress in statistical language modeling. *arXiv preprint arXiv:1312.3005*.

Francesco D'Amico, Dario Bocchi, Luca Maria Del Bono, Saverio Rossi, and Matteo Negri. 2026. Pseudo-likelihood produces associative memories able to generalize, even for asymmetric couplings. *Physica A: Statistical Mechanics and its Applications*, 692:131497.

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

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

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

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

Itai Gat, Tal Remez, Neta Shaul, Felix Kreuk, Ricky TQ Chen, Gabriel Synnaeve, Yossi Adi, and Yaron Lipman. 2024. Discrete flow matching. *Advances in Neural Information Processing Systems*, 37:133345–133385.

Donald Olding Hebb. 1949. The organization of behavior: A neuropsychological theory.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*.

Benjamin Hoover, Yuchen Liang, Bao Pham, Rameswar Panda, Hendrik Strobelt, Duen Horng Chau, Mohammed Zaki, and Dmitry Krotov. 2023a. Energy transformer. In *Advances in Neural Information Processing Systems*, volume 36, pages 27532–27559. Curran Associates, Inc.

Benjamin Hoover, Hendrik Strobelt, Dmitry Krotov, Judy Hoffman, Zsolt Kira, and Duen Horng Chau. 2023b. Memory in plain sight: A survey of the uncanny resemblances between diffusion models and associative memories. *arXiv preprint arXiv:2309.16750*.

John J. Hopfield. 1982. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8):2554–2558.

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

Zahra Kadkhodaie, Florentin Guth, Eero P Simoncelli, and Stéphane Mallat. 2023. Generalization in diffusion models arises from geometry-adaptive harmonic representation. *arXiv preprint arXiv:2310.02557*.

Silvio Kalaj, Clarissa Lauditi, Gabriele Perugini, Carlo Lucibello, Enrico M. Malatesta, and Matteo Negri. 2025. Random features hopfield networks generalize retrieval to previously unseen examples. *Physica A: Statistical Mechanics and its Applications*, 678:130946.

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

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Leo Kozachkov, Jean-Jacques Slotine, and Dmitry Krotov. 2025. Neuron–astrocyte associative memory. *Proceedings of the National Academy of Sciences*, 122(21):e2417788122.

Dmitry Krotov. 2021. Hierarchical associative memory. *arXiv preprint 2107.06446*.

Dmitry Krotov. 2023. A new frontier for Hopfield networks. *Nature Reviews Physics*, 5(7):366–367.

Dmitry Krotov, Benjamin Hoover, Parikshit Ram, and Bao Pham. 2025. Modern methods in associative memory. *arXiv preprint arXiv:2507.06211*.

Dmitry Krotov and John J. Hopfield. 2016. Dense associative memory for pattern recognition. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

Dmitry Krotov and John J Hopfield. 2021. Large associative memory problem in neurobiology and machine learning. In *International Conference on Learning Representations*.

Andrea Montanari, Yiqiao Zhong, and Kangjie Zhou. 2024. Tractability from overparametrization: The example of the negative perceptron. *Probability Theory and Related Fields*, 188(3–4):805–910. ArXiv:2110.15824.

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

Bao Pham, Gabriel Raya, Matteo Negri, Mohammed J Zaki, Luca Ambrogioni, and Dmitry Krotov. 2025. Memorization to generalization: Emergence of diffusion models from associative memory. *arXiv preprint arXiv:2505.21777*.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

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

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

Subham Sekhar Sahoo, Justin Deschenaux, Aaron Gokaslan, Guanghan Wang, Justin T Chiu, and Volodymyr Kuleshov. 2025. The diffusion duality. *International Conference on Machine Learning*, 42.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*.

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

Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. 2023b. Understanding and mitigating copying in diffusion models.

Advances in Neural Information Processing Systems, 36:47783–47803.

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

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. 2021. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*.

Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. 2018. The implicit bias of gradient descent on separable data. *Journal of Machine Learning Research*, 19(70):1–57.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

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

Yuxin Wen, Yuchen Liu, Chen Chen, and Lingjuan Lyu. 2024. Detecting, explaining, and mitigating memorization in diffusion models. In *The Twelfth International Conference on Learning Representations*.

TaeHo Yoon, Joo Young Choi, Sehyun Kwon, and Ernest K Ryu. 2023. Diffusion probabilistic models generalize when they fail to memorize. In *ICML 2023 Workshop on Structured Probabilistic Inference Generative Modeling*.

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)[3]. For more details on this variant of UDDMs, please refer to the discussion in Appx. (C) below. Meanwhile, the backbone of our trained UDDMs is the diffusion transformer architecture from (Peebles and Xie, 2023). The configurations of our three variations of UDDMs are described in Tab. (1) [4].

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 linear spacing (of 17 points, inclusively) starting at 10^{-4} to 10^{-2}, and another set of points using linear 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 5×10^3 samples belonging to their respective training set (in accordance with their fraction of training dataset size) and also 5×10^3 sequences from the test set. We performed the perturbation of these sequences using the forward process (20) 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 (20), 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 5×10^3 samples (per training and per test sequences), respectively. Meanwhile, for the results in Fig. (4) and Fig. (6), 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. (6) 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. (13)-(15) for the full histograms of conditional entropy on training versus generated samples, and Figs. (10)-(12) for more conditional entropy histograms of the unrecovered versus recovered tokens shown in Fig. (2).

Textual Examples. In Figs. (1) and (7), 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 utilize 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

[3]The code-base used in our experiments is publicly available at https://github.com/s-sahoo/duo under the Apache 2.0 license, and we use it solely for research purposes consistent with its intended use.

[4]The LM1B dataset (Chelba et al., 2013) is publicly available at https://www.statmt.org/lm-benchmark/ and is intended for research purposes in statistical language modeling, which is consistent with our use.

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.

recovery experiments. Please see Figs. (3A), (8), and (9) for the textual examples that collected using the typical stochastic process.

Perplexity. Similarly to our results on conditional entropy, we utilized the same sets of 10^5 generated sequences belonging to each trained UDDM, respectively. The code to compute perplexity is also from Sahoo et al. (2025). Keep in mind, the reported conditional entropy results are computed from our UDDMs, unlike perplexity, which requires the reference or critic model of the pre-trained GPT2-Large (Radford et al., 2019) as mentioned previously.

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 Usage Of LLM Disclaimer

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

C Uniform-state Discrete Diffusion and Duality with Gaussian

Further Details on UDDMs. As mentioned in Sec. (2), the true reverse posterior of a previous timestep $s < t$ corresponding to the forward process (1) is

$$\begin{aligned}\mathbf{z}_s \sim q_{s|t}(\mathbf{z}_s|\mathbf{z}_t, \mathbf{x}) &= \text{Cat}\Big(\mathbf{z}_s; g(\mathbf{z}_t, \mathbf{x})\Big) \\ &= \text{Cat}\left(\mathbf{z}_s; \frac{K\alpha_t \mathbf{z}_t \odot \mathbf{x} + (\alpha_{t|s} - \alpha_t)\mathbf{z}_t}{K\alpha_t\langle \mathbf{z}_t, \mathbf{x}\rangle + (1-\alpha_t)} + \frac{(\alpha_s - \alpha_t)\mathbf{x} + (1-\alpha_{t|s})\frac{\mathbf{1}}{K}}{K\alpha_t\langle \mathbf{z}_t, \mathbf{x}\rangle + (1-\alpha_t)}\right),\end{aligned} \tag{14}$$

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 (14), 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 $p_\theta(\mathbf{x}|\mathbf{z}_t) = \mathbf{x}_\theta(\mathbf{z}_t, t)$. The network's parameters θ are optimized via the Negative Evidence Lower Bound (NELBO) objective (Austin et al., 2021; Sahoo et al., 2024):

$$\mathcal{L}_{\text{NELBO}} = \mathbb{E}_q\Big[\underbrace{-\log p_\theta(\mathbf{x}|\mathbf{z}_0)}_{\mathcal{L}_{\text{reconstruction}}} + \underbrace{\sum_{s<t} D_{\text{KL}}\big[q(\mathbf{z}_s|\mathbf{z}_t, \mathbf{x}) \,||\, p_\theta(\mathbf{z}_s|\mathbf{z}_t)\big]}_{\mathcal{L}_{\text{diffusion}}}\Big] + \underbrace{D_{\text{KL}}\big[q(\mathbf{z}_1|\mathbf{x}) \,||\, p_\theta(\mathbf{z}_1)\big]}_{\mathcal{L}_{\text{prior}}}. \tag{15}$$

As discussed in Sec. (3), the term $\mathcal{L}_{\text{reconstruction}}$ is more related to our simple setting in that section due to the fact that it only deals with a single finite temperature value (or it does not rely on temperature annealing). However, we suspect that the other terms in NELBO (15) further help the maximization of classification margin (or basin) of the data points.

Duality of Uniform and Gaussian. In continuous diffusion modeling, we typically rely on the diffusion mapping of a data distribution q_{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:

$$\mathbf{w}_t \sim \tilde{q}_t(\mathbf{w}_t|\mathbf{x}; \tilde{\alpha}_t) = \mathcal{N}(\mathbf{w}_t; \tilde{\alpha}_t\mathbf{x}, (1-\tilde{\alpha}_t^2)\mathbf{I}_K), \tag{16}$$

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 $\arg\max(\mathbf{w}) = \arg\max_{\mathbf{z}\in\mathcal{V}} \mathbf{z}^\top \mathbf{w}$. Then, we can define the discrete marginals to be

$$\mathbf{z}_t = \arg\max(\mathbf{w}_t), \tag{17}$$

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

$$\mathbf{z}_t \sim P_t\big(\mathbf{z}_t|\mathbf{x}; \mathcal{T}(\tilde{\alpha}_t)\big) = \text{Cat}\left(\mathbf{z}_t; \mathcal{T}(\tilde{\alpha}_t)\mathbf{x} + (1-\mathcal{T}(\tilde{\alpha}_t))\frac{\mathbf{1}}{K}\right), \tag{18}$$

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

$$\alpha_t = \mathcal{T}(\tilde{\alpha}_t) = \frac{K}{K-1}\left[\int_{-\infty}^{\infty} \phi\left(z - \frac{\tilde{\alpha}_t}{\sqrt{1-\tilde{\alpha}_t^2}}\right)\Phi^{K-1}(z)\mathrm{d}z - \frac{1}{K}\right] \tag{19}$$

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

$$\mathbf{z}_t \sim q_t(\mathbf{z}_t|\mathbf{x}; \mathcal{T}(\tilde{\alpha}_t)) = [\arg\max]_* \, \tilde{q}_t(\mathbf{w}_t|\mathbf{x}; \tilde{\alpha}_t) \tag{20}$$

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. For full details, please refer to (Sahoo et al., 2025).

D Conditional Entropy and Curvature

In the more common setting, where AM have an energy landscape, the sharpness of the basin dictates retrieval dynamics (Krotov and 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).

In this section, inspired by Biroli et al. (2024) and D'Amico et al. (2026), we relate entropy and the curvature (or sharpness) of the energy in the continuous setting, using local approximation, to show the 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 $E(\mathbf{x}; \mathbf{z}_t)$:

$$p(\mathbf{x}|\mathbf{z}_t) = \frac{1}{Z(\mathbf{z}_t)} e^{-E(\mathbf{x};\, \mathbf{z}_t)}, \tag{21}$$

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 $\mathbf{x}^*(\mathbf{z}_t)$:

$$E(\mathbf{x}; \mathbf{z}_t) \approx E(\mathbf{x}^*; \mathbf{z}_t) + (\mathbf{x} - \mathbf{x}^*)^\top \nabla_{\mathbf{x}} E(\mathbf{x}^*; \mathbf{z}_t) + \frac{1}{2}(\mathbf{x} - \mathbf{x}^*)^\top \mathbf{H}(\mathbf{z}_t)(\mathbf{x} - \mathbf{x}^*). \tag{22}$$

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. (22) 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 $Z(\mathbf{z}_t)$:

$$\begin{aligned} Z(\mathbf{z}_t) &\approx \int e^{-\left(E(\mathbf{x}^*; \mathbf{z}_t) + \frac{1}{2}(\mathbf{x}-\mathbf{x}^*)^\top \mathbf{H}(\mathbf{z}_t)(\mathbf{x}-\mathbf{x}^*)\right)} \mathrm{d}\mathbf{x} \\ &= e^{-E(\mathbf{x}^*; \mathbf{z}_t)} \int e^{-\frac{1}{2}(\mathbf{x}-\mathbf{x}^*)^\top \mathbf{H}(\mathbf{z}_t)(\mathbf{x}-\mathbf{x}^*)} \mathrm{d}\mathbf{x} \\ &= (2\pi)^{\frac{d}{2}} \, e^{-E(\mathbf{x}^*; \mathbf{z}_t)} \det\big(\mathbf{H}(\mathbf{z}_t)\big)^{-\frac{1}{2}} \end{aligned} \tag{23}$$

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

$$\begin{aligned} \mathcal{H}(\mathbf{x}|\mathbf{z}_t) &= \mathop{\mathbb{E}}_{\mathbf{x} \sim p(\mathbf{x}|\mathbf{z}_t)} \Big[E(\mathbf{x}; \mathbf{z}_t) + \log Z(\mathbf{z}_t) \Big] \\ &= \mathop{\mathbb{E}}_{\mathbf{x} \sim p(\mathbf{x}|\mathbf{z}_t)} \Big[E(\mathbf{x}; \mathbf{z}_t) + \frac{d}{2}\log(2\pi) - E(\mathbf{x}^*; \mathbf{z}_t) - \frac{1}{2}\log\big[\det\big(\mathbf{H}(\mathbf{z}_t)\big)\big] \Big] \end{aligned} \tag{24}$$

If we substitute Eq. (22) into the term $E(\mathbf{x}; \mathbf{z}_t)$ in Eq. (24), we then have

$$\begin{aligned} \mathcal{H}(\mathbf{x}|\mathbf{z}_t) &\approx \mathop{\mathbb{E}}_{\mathbf{x} \sim p(\mathbf{x}|\mathbf{z}_t)} \Big[\cancel{E(\mathbf{x}^*; \mathbf{z}_t)} + \frac{d}{2}\log(2\pi) - \cancel{E(\mathbf{x}^*; \mathbf{z}_t)} - \frac{1}{2}\log\big[\det\big(\mathbf{H}(\mathbf{z}_t)\big)\big] + C \Big] \\ &\approx \mathop{\mathbb{E}}_{\mathbf{x} \sim p(\mathbf{x}|\mathbf{z}_t)} \Big[-\frac{1}{2}\log\big[\det\big(\mathbf{H}(\mathbf{z}_t)\big)\big] + C \Big] \end{aligned} \tag{25}$$

where C is a constant involving the omitted terms from our substitution of Eq. (22).

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.

E Additional Results

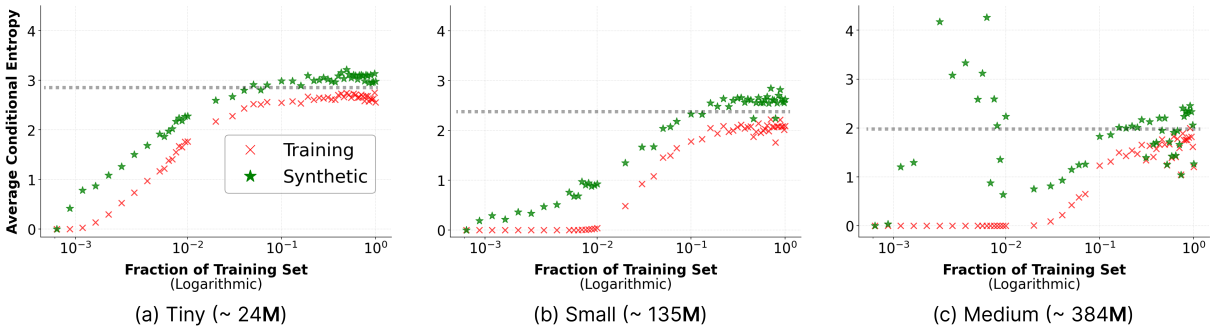
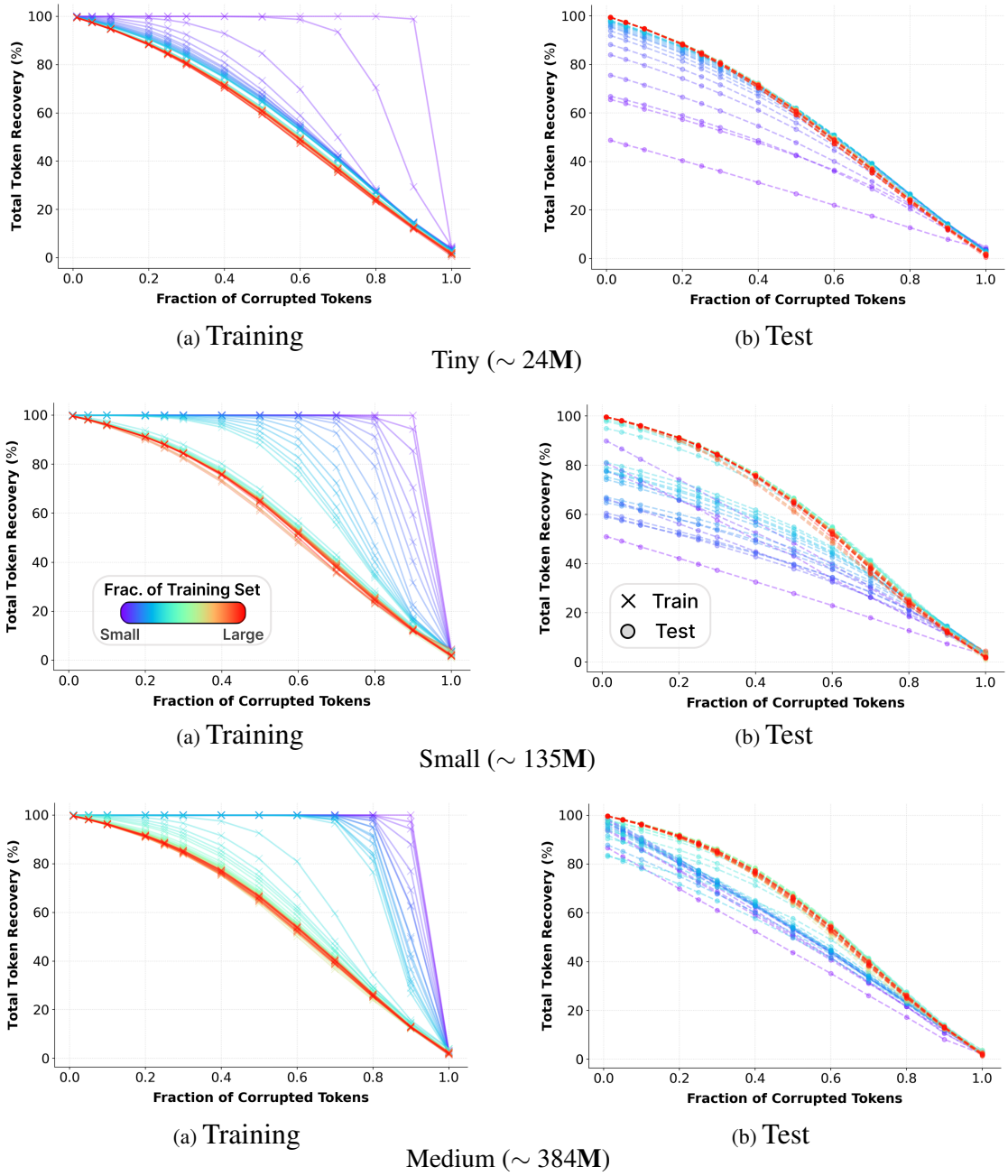


Figure 6: **Average conditional entropy for training versus synthetic sequences**. As the training dataset size increases, the average (over sequences and tokens) conditional entropy for both training and synthetic (or generated) sequences naturally rises. Here, we plot the dashed horizontal line in 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.

E.1 Shrinkage and Expansion of Basins



(a) Training (b) Test

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

(a) Training (b) Test

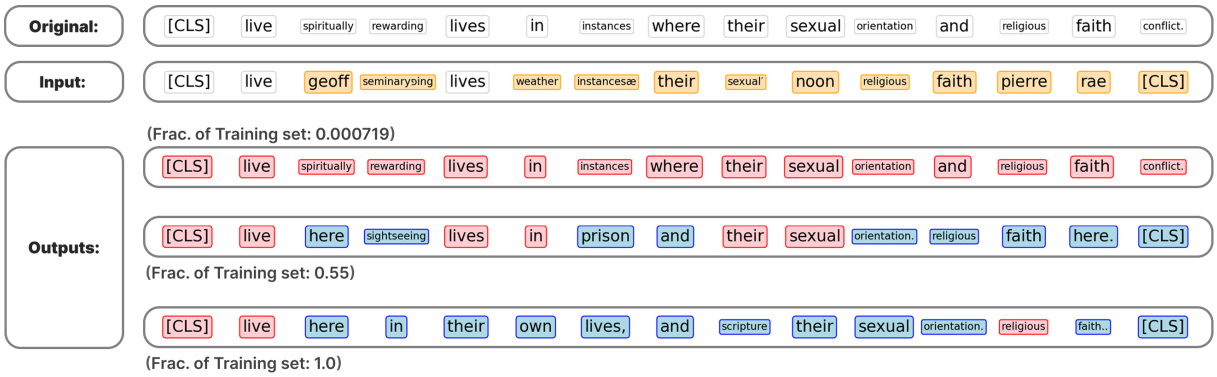
Small ($\sim$ 135**M**)

(a) Training (b) Test

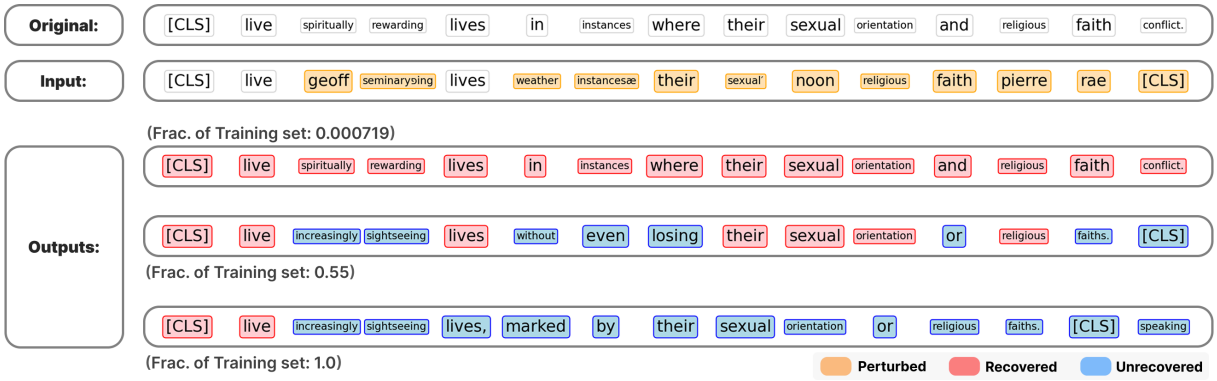
Medium ($\sim$ 384**M**)

Figure 7: **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.

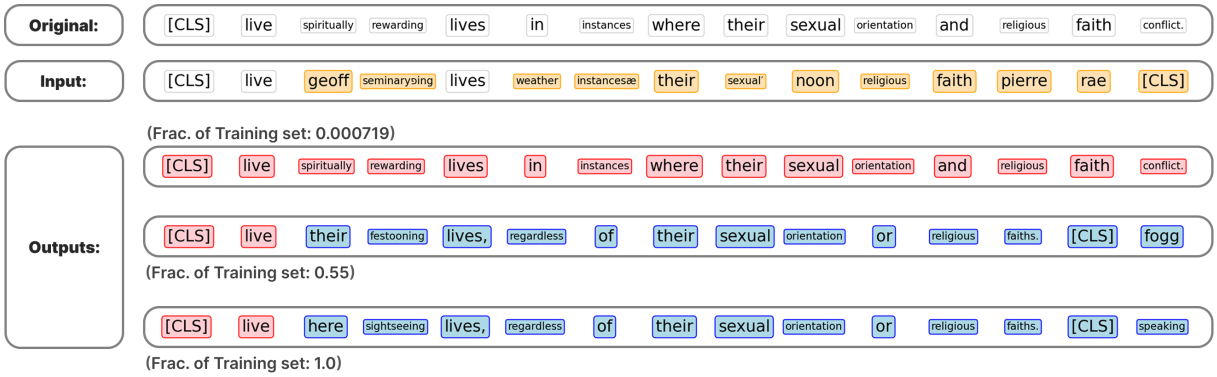
E.2 Visualizations of Text Recovery Examples



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

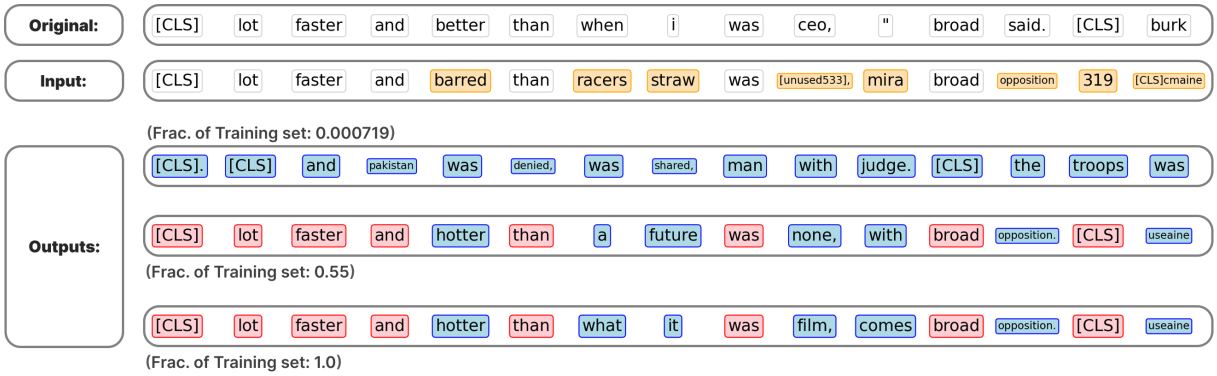


(b) Small ($\sim$ 135**M**)

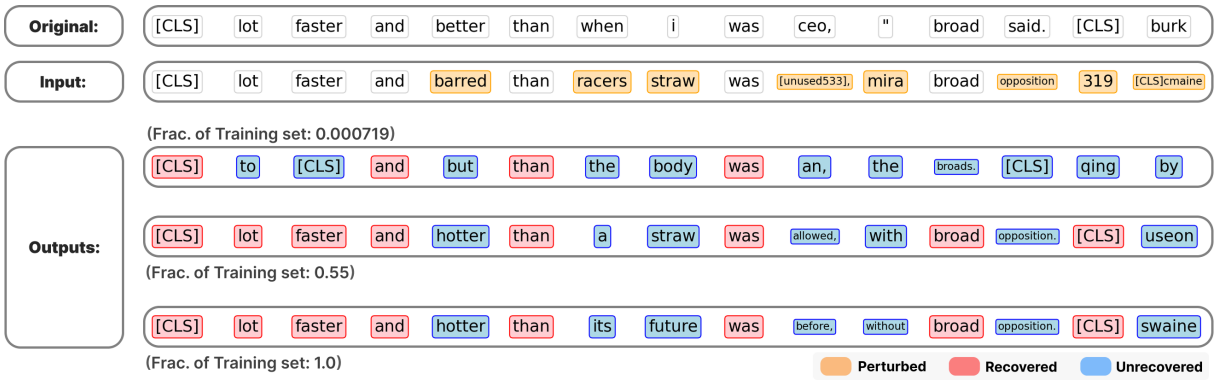


(c) Medium ($\sim$ 384**M**)

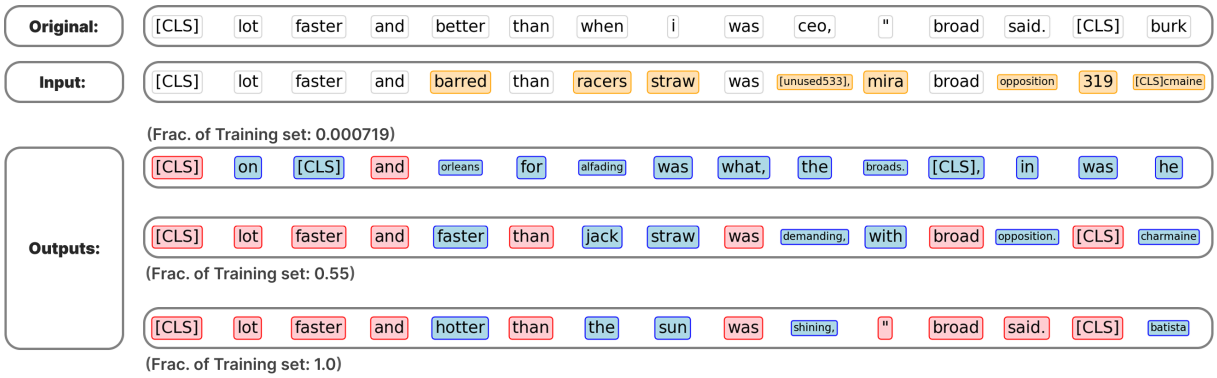
Figure 8: An illustration of the model's ability to recover tokens from perturbed sequences on **training examples** at three different fractions of the training dataset and sizes of the UDDMs. Perturbation is computed at $t = 0.5$ and the typical stochastic reverse process is performed afterwards. As 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 9: 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

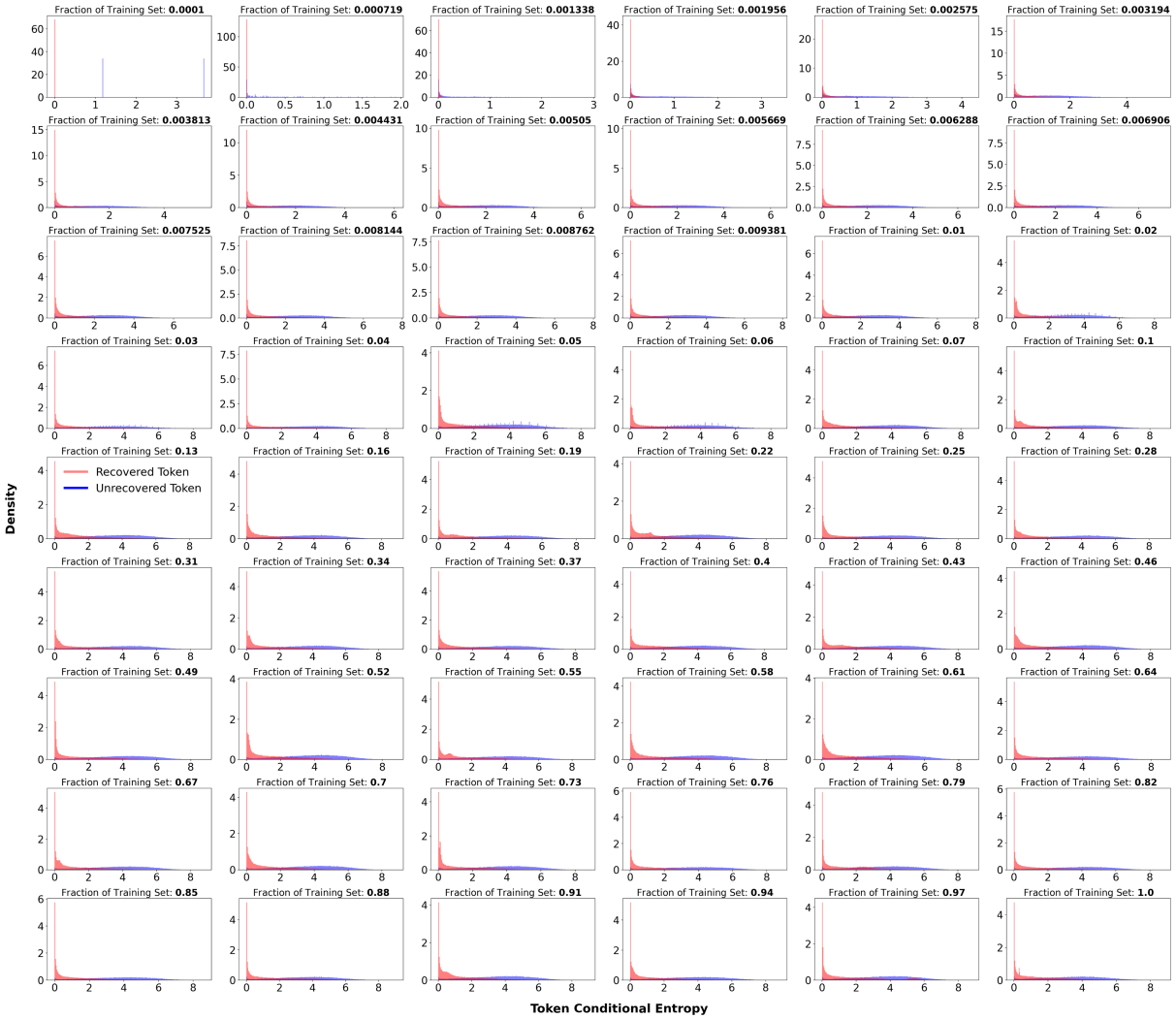


Figure 10: An illustration of the density of conditional entropy for two categories of tokens, *recovered* and *unrecovered*, computed at $t = 0.25$ for the *Tiny* model. The subplots are ordered by the fraction of training dataset, ranging from 10^{-4} (top-left) to 1.0 (bottom-right). As the fraction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

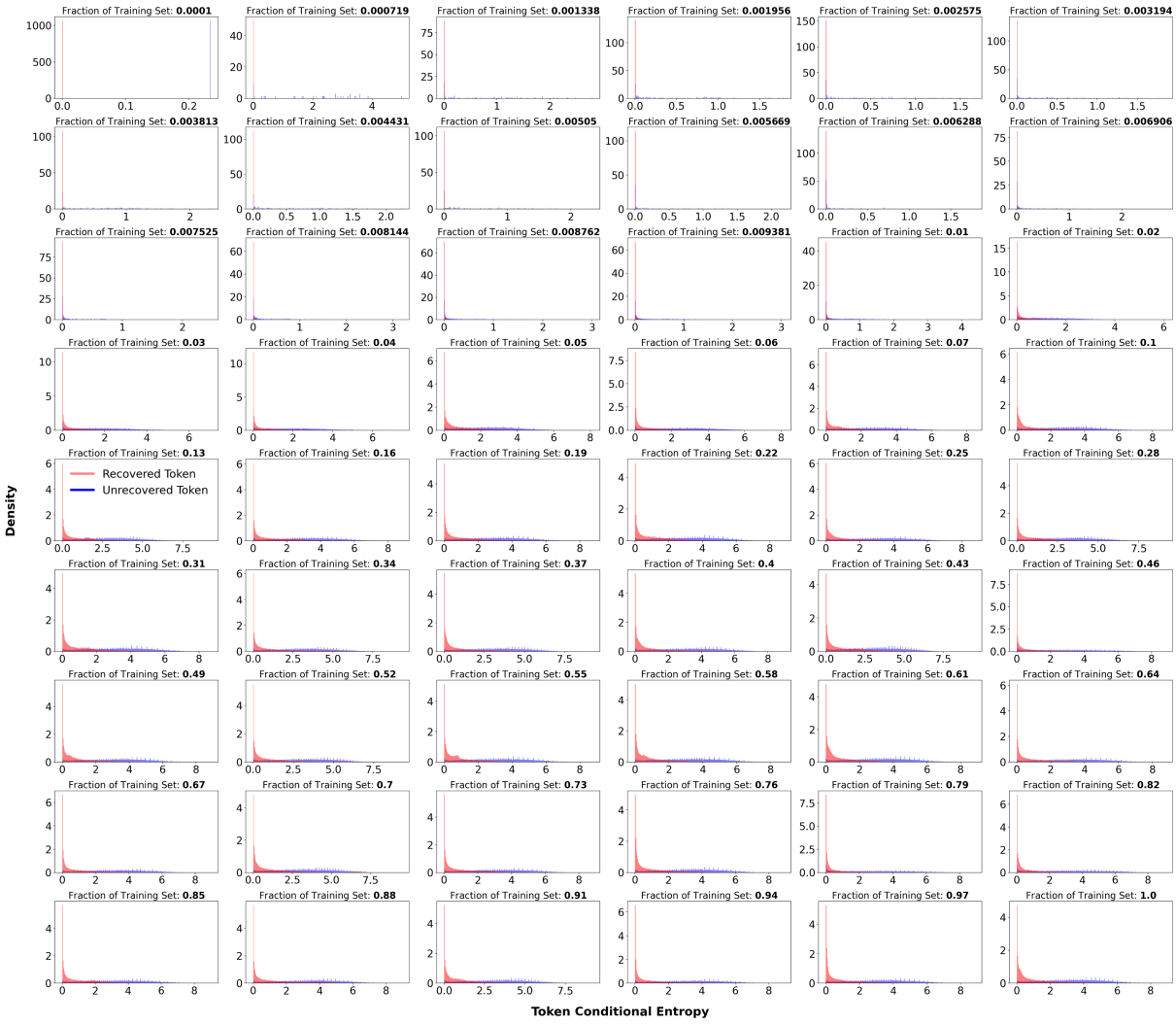


Figure 11: An illustration of the density of conditional entropy for two categories of tokens, *recovered* and *unrecovered*, computed at $t = 0.25$ for the *Small* model. The subplots are ordered by the fraction of training dataset, ranging from 10^{-4} (top-left) to 1.0 (bottom-right). As the fraction of training data increases, recovered tokens concentrate near zero entropy (high confidence), while unrecovered tokens exhibit a broad distribution at higher entropy.

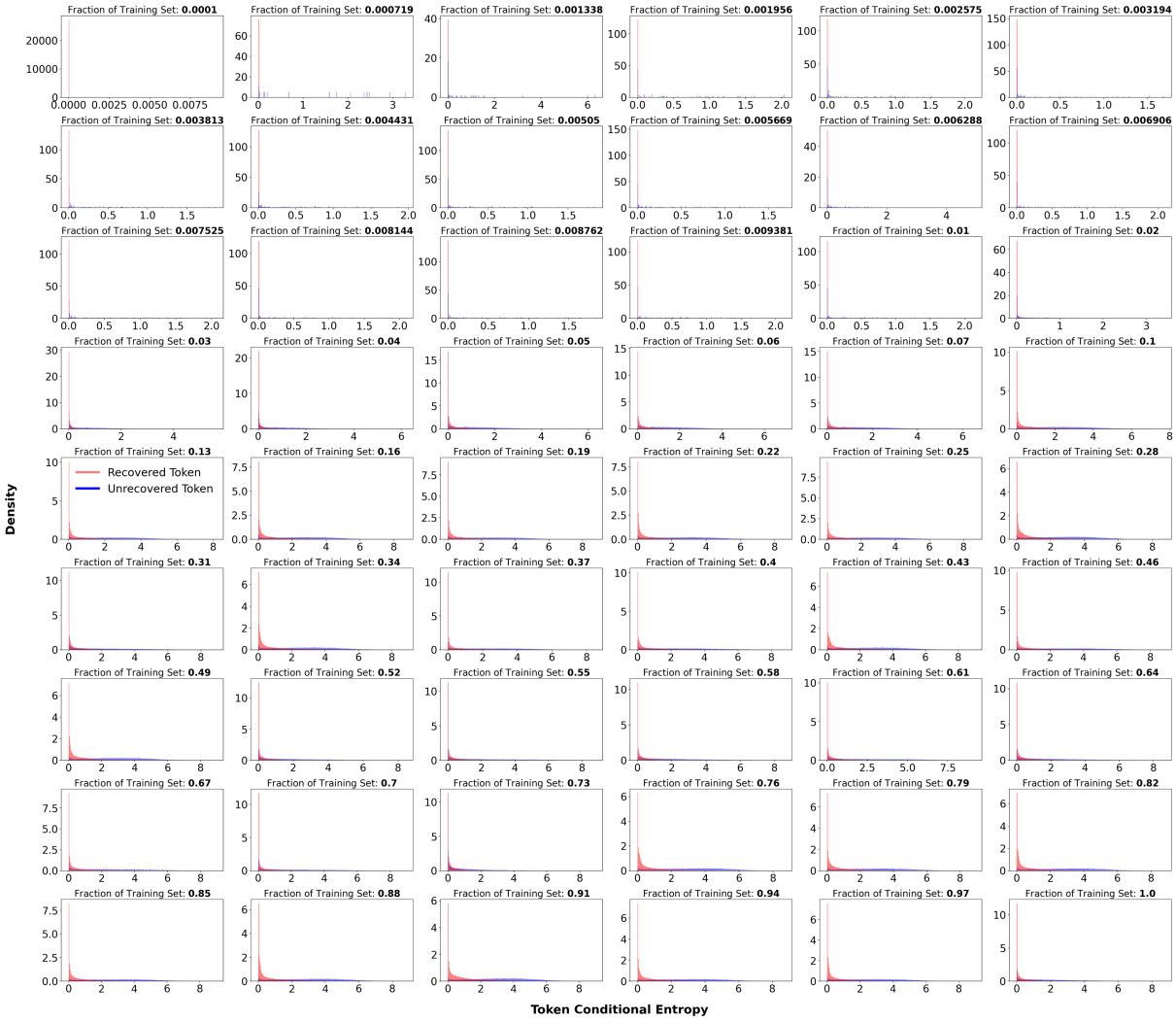


Figure 12: An illustration of the density of conditional entropy for two categories of tokens, *recovered* and *unrecovered*, computed at $t = 0.25$ for the *Medium* model. The subplots are ordered by the fraction of training dataset, ranging from 10^{-4} (top-left) to 1.0 (bottom-right). As the fraction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

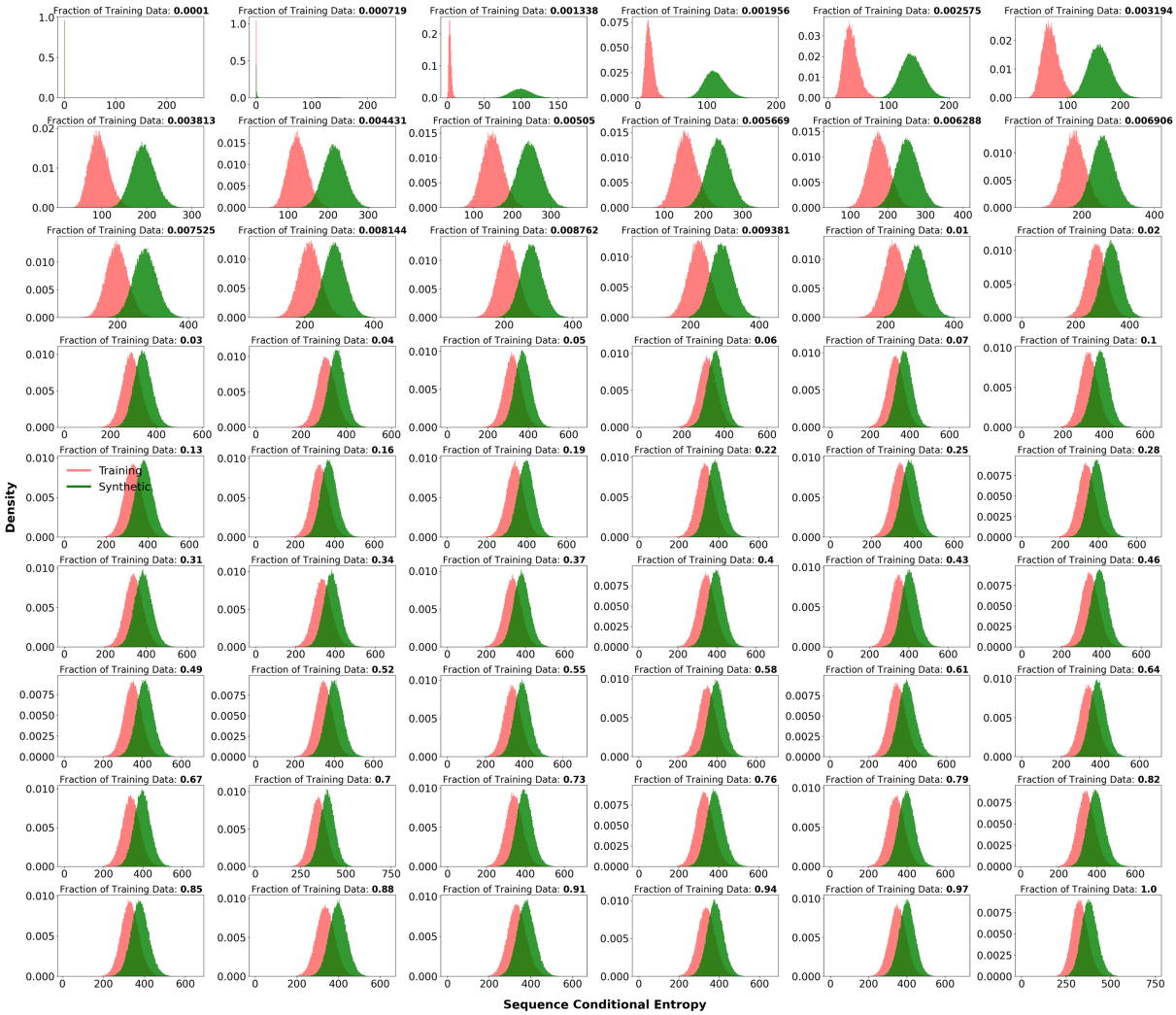


Figure 13: An illustration of the evolution of the density of the average conditional entropy for the probabilities of training and synthetic sequences respectively, computed at $t = 10^{-5}$ using the *Tiny* models, as the training dataset size grows. When the fraction of training set is small, there 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.

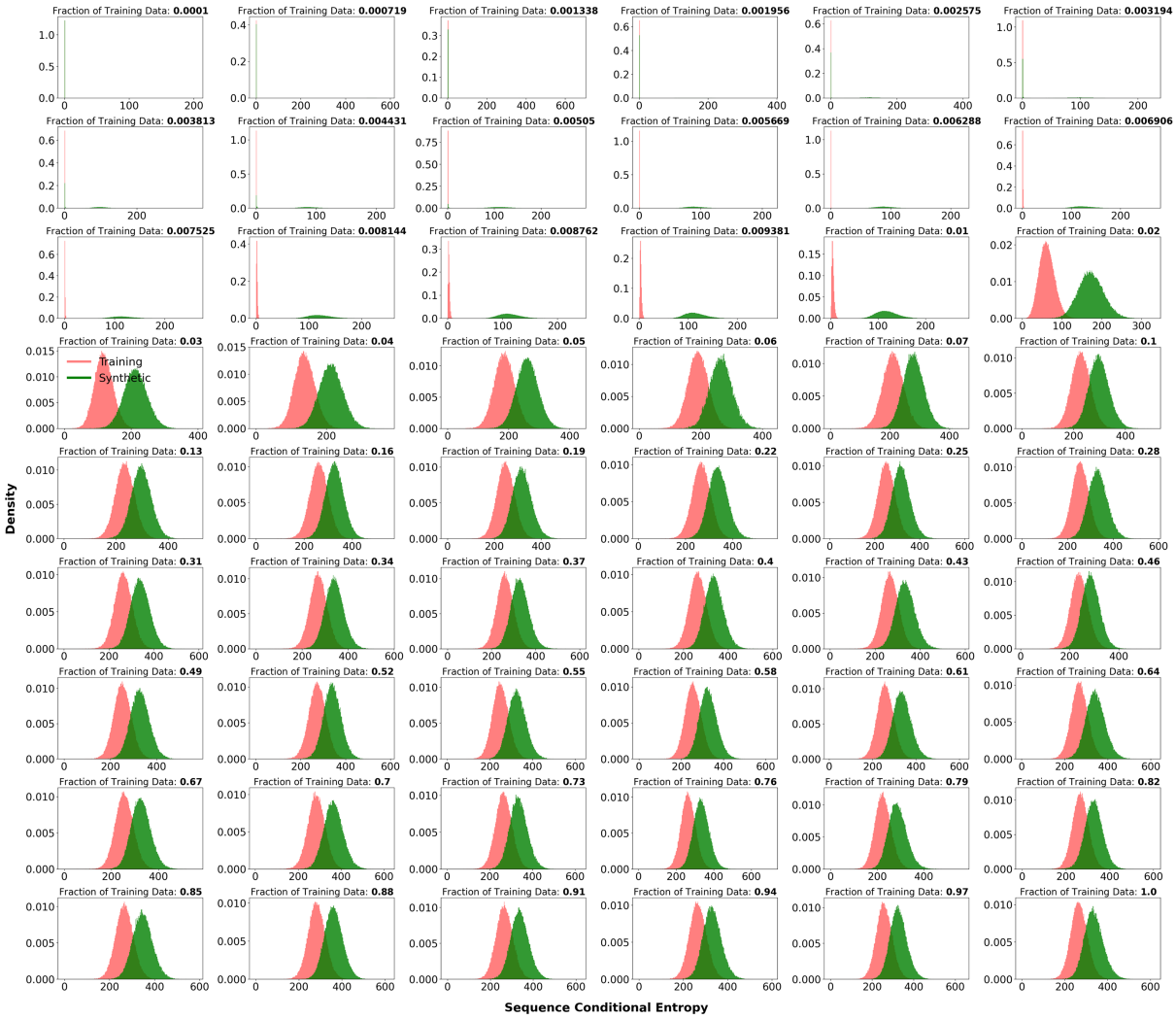


Figure 14: An illustration of the evolution of the density of the average conditional entropy for the probabilities of training and synthetic sequences respectively, computed at $t = 10^{-5}$ using the *Small* models, as the training dataset size grows. When the fraction of training set is small, there 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.

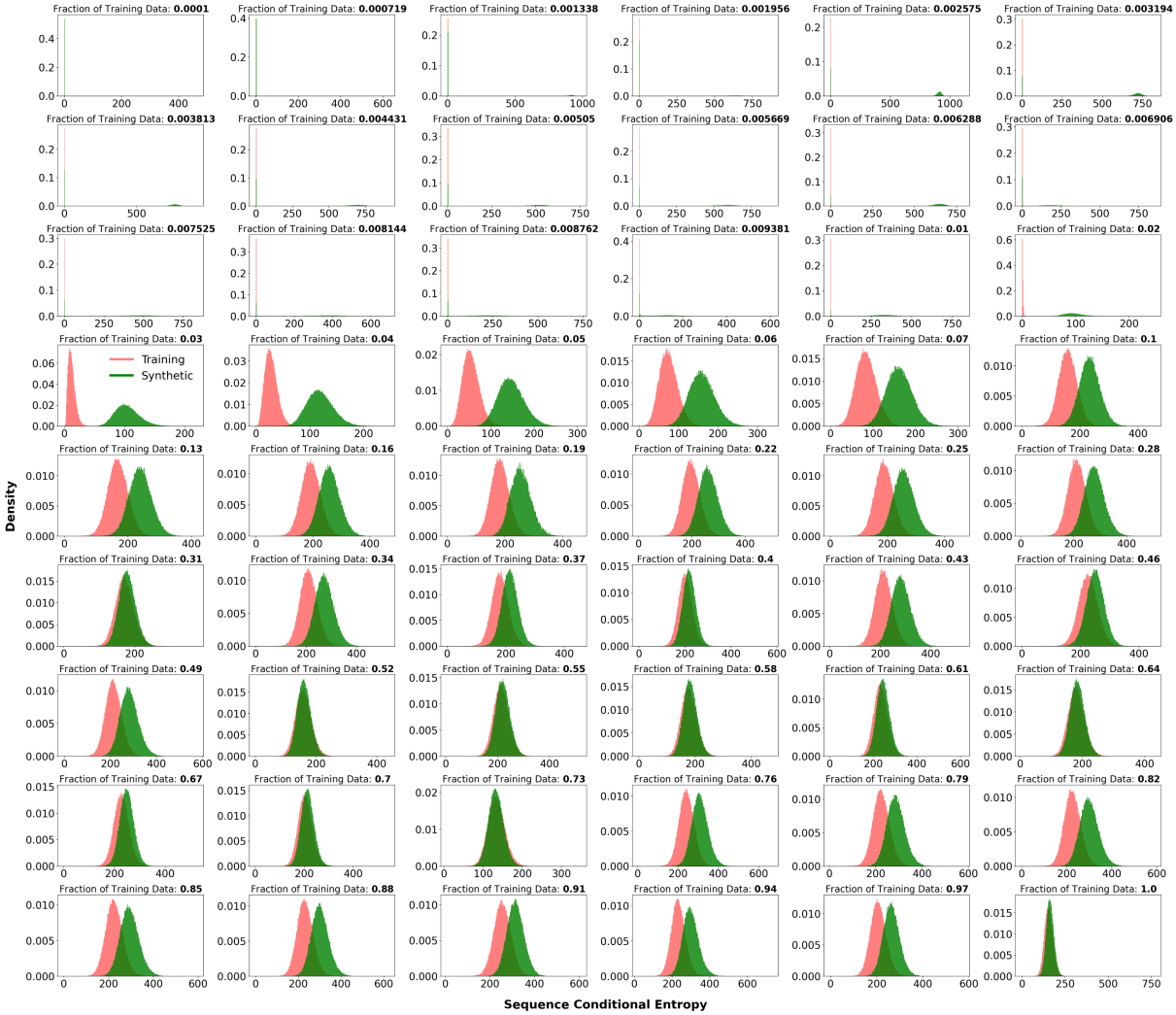
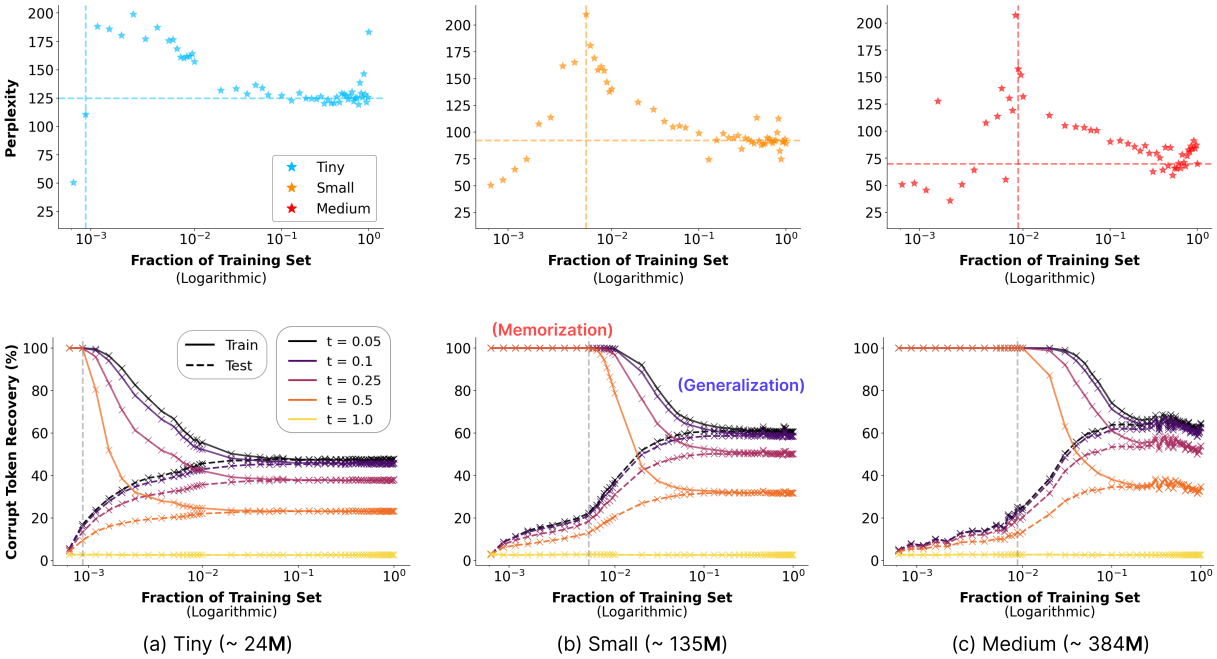


Figure 15: An illustration of the evolution of the density of the average conditional entropy for the probabilities of training and synthetic sequences respectively, computed at $t = 10^{-5}$ using the *Medium* models, as the training dataset size grows. When the fraction of training set is small, there 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.



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

Figure 16: **Perplexity trends during the memorization-to-generalization transition together with corrupted token recovery**. This figure demonstrates the alignment of phase transitions across perplexity (top) and the average corrupt token recovery rate (bottom) for three model sizes. The *vertical* dashed lines roughly represent the critical transitional points where the perplexity (calculated using GPT-2 Large (Radford et al., 2019)) aligns with the initial decrease of corrupt token recovery rate for training and the increase of test tokens' recovery rate. This alignment of the two metrics showcases the usefulness of the token recovery rate in determining whether generalization has been reached in UDDMs.