CLOSING THE HALLUCINATION WINDOW IN DIFFUSION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion models learn structure from noise, yet not always all of it, and this leads to hallucination. While the memorization and generalization regimes of their training dynamics have been well studied, we study how hallucination fits into these dynamics. We believe that by better understanding how hallucination occurs from an optimization perspective, we can better design regularization methods and, by extension, architectures for diffusion models that could offset hallucination during training. We find that denoising loss has an elbow, i.e., it plateaus significantly before the model actually generalizes $T_{elb} << T_{gen}$. Consequently, hallucination can be distinguished from generalization only as an asynchrony in the optimization dynamics when the denoising loss is hierarchically resolved and rescaled. This gap, which we call the hierarchical lag, is the training dynamics signature of hallucination. We further show for a diffusion transformer trained on a random hierarchy model how a simple regularization to decrease this unintended "lag" can accelerate generalization and shrink the hallucination window in training time by an order of a magnitude.

1 INTRODUCTION

Diffusion models(Ho et al., 2020; Sohl-Dickstein et al., 2015) are a leading paradigm in generative modeling, finding their use in different use cases from typical modalities like images(Dhariwal & Nichol, 2021), videos(Ho et al., 2022), and texts (Austin et al., 2021) to more scientific ones in drug discovery (Watson et al., 2023) and climate modeling(Watt & Mansfield, 2024). Despite their realistic outputs, these models are known to produce samples that violate constraints present in the underlying distribution structurally(Kim et al., 2024), logically(Han et al., 2025), or syntactically(Lu et al., 2025); we refer to all such violations as hallucinations.

Similar to prior works (Lu et al., 2025; Favero et al., 2025), we define a hallucinated sample as a generated sample that is "locally valid but globally incoherent" or, more formally, a sample that is valid under all low-level (local) structural constraints of the underlying distribution but violates at least one high-level (global) constraint. It is hence different from a) a generalized sample, a novel sample that satisfies all the constraints of the distribution (Wang et al., 2026); b) a badly generated sample, a sample that violates even low-level constraints; and c) a memorized sample, a sample that closely matches a training instance(Carlini et al., 2023). Together these four states form a four-way partition of the generated instances in any generative task. While training a diffusion model (denoiser), the composition of generated samples, i.e., the fraction of samples falling into each class, shifts, and this can be used to demarcate regimes of diffusion models. For example, the onset of the memorization regime is marked by the fraction of memorized samples crossing a predefined threshold (Bonnaire et al., 2025). However, this compositional view is secondary to, and should agree with, the primary and more classical picture of these regimes given by the training and test loss curves (George et al., 2025; Favero et al., 2026).

As seen in Fig. 1 a), memorization regime occurs when a denoiser overfits to its training data at high effective capacity (gradient steps). This regime is characterized by a low training loss (empirical risk) and a high test loss, i.e., in classical learning theory terms, a large generalization gap between empirical and population risk, signaling high variance. Similarly, we know that there is a counterpart to memorization at the start of training where a denoiser underfits the training data, a regime characterized by both a high training loss and test loss. Both generalization and hallucination lie

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

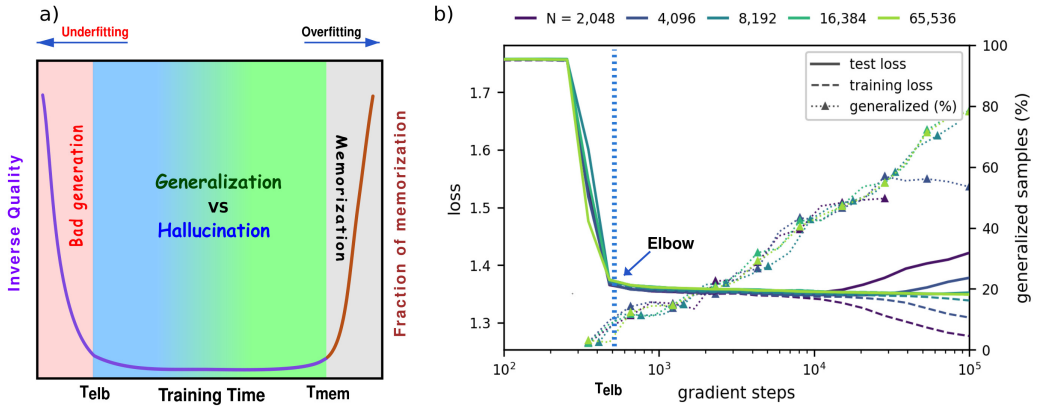


Figure 1: a) Illustration of inverse quality + fraction of memorization vs. training time. The right extreme corresponds to memorization regime; the left extreme corresponds to bad generation dominated regime. Both generalization and/or hallucination seem to exist in the plateau between them. b) Evolution of test denoising loss, training denoising loss (averaged over diffusion time), and percentage of valid samples during training of a diffusion transformer on RHM setting. Both test loss and training loss across different training sizes plateau at around 5×10^2 gradient steps, characterized by an elbow in both the loss curves. The percentage of valid samples generated by the model, however, is low at the elbow and considerably increases with more training steps, even though both losses only change marginally.

in the plateau between these two extremes. However, we cannot distinguish between them solely based on the training loss or test loss, as they can partition the plot into at most three regions from a bias-variance perspective, while there are four states to separate. On a synthetic task where the four states can be decided exactly, as seen in Fig. 1 b) both loss curves flatten at a shared elbow T_{elb} after around 10^2 gradient steps, while the fraction of valid samples is still near zero at that point and keeps rising for another two orders of magnitude in training time. Hence, whatever separates a hallucinating model from a generalizing one happens after the loss has stopped visibly changing. This is our motivation for attempting to "vectorize" the loss to obtain a training-dynamics signature that separates it and leads to the following two contributions:

1. **A training-dynamics signature of hallucination**. We show that by using belief-propagation oracles for a random hierarchy model, we can decompose the denoising loss into comparable n_{global} and n_{local} risk components corresponding to global and local correlations. This reveals the existence of a hierarchical lag, a gap by which the global component trails the local one. We show that the interval over which the lag persists, between the loss elbow T_{elb} and the onset of global minimization T_{global}, defines the hallucination window in which hallucinated samples peak.
2. **A regularizer that closes the hallucination window.** We show how a simple regularization on the hierarchical lag could shrink the hallucination window, showing that it reduces the fraction of hallucinated samples, and accelerating generalization.

2 Related Works

Generalization and memorization in diffusion models. Kadkhodaie et al. (2024) showed that diffusion models trained on enough data generalize because of their inherent bias to geometry-adaptive harmonic representations. Bonnaire et al. (2025) formalized the notion of memorization and generalization regimes in diffusion, showing through random features network that the training time at which memorization T_{mem} sets in grows with dataset size, while the generalization time T_{gen} is independent of the training size. Our work refines the picture from the side of generalization: within the window where the model is not yet memorizing, we attempt to characterize the part where the model hallucinates.

Belief propagation in Diffusion. The random hierarchy model (RHM) was introduced by Petrini et al. (2023) to study how deep networks learn compositional data, and they establish the sample complexity of learning each level of the hierarchy. Sclocchi et al. (2024) show that for diffusion on the RHM, the probability of reconstructing the root undergoes a sharp phase transition at a critical noise level. Mei (2024) shows that U-Nets can efficiently approximate belief propagation (Mézard & Montanari, 2009) in such generative hierarchical models. Similarly, Garnier-Brun et al. (2025) show that transformers approximate BP while learning structured data. Favero et al. (2025) find that diffusion models learn the hierarchy level by level during training, partially generalizing and producing locally consistent but globally incoherent samples at intermediate training times, the phenomenon we call as hallucination. Garnier-Brun et al. (2025) use the same BP substrate to track the divergence between the model and filtered BP oracles to show that models have a bias towards training data even before they actually memorize. We build on this filtered-oracle construction to levelwise resolve the scalar losses.

Hallucination and rule learning. Lu et al. (2025); Han et al. (2025) report a strong local inductive bias as the cause of hallucination. Supporting this Kamb & Ganguli (2025) indicates that even an analytical denoiser with local inductive bias imposed on it can show structural hallucinations (Kim et al., 2024). Wang et al. (2026) identify two separable clocks in generative models T_{rule} and T_{mem} whose gap defines an innovation window in which rule-consistent novel samples are produced. They also show the existence of a training window where the model generates samples that lie on the hypercube but don't follow the rule present in the training samples. While they study singular rules on a hypercube, we study a hierarchy of compositional rules on sequences.

3 Random hierarchy model

An RHM is explicitly designed to replicate the hierarchical structure that can be seen in real-world data. It defines a probabilistic context-free grammar (PCFG) in terms of a set of hierarchically ordered production rules, which can be used to draw a sample satisfying the PCFG. We use the original RHM from Carlini et al. (2023) with uniform production probabilities, except that a single rule set is shared by all levels, so global and local rules differ only in the scale at which they act. Knowing the underlying data-generating process lets us use the tree as a training-free Bayes-optimal denoiser (in essence an oracle) for a noised sample through belief propagation. The oracle gives us two things. First, it makes it possible to find the most local level at which a production rule is broken, i.e., the violation depth for any given arbitrary sequence. This helps us in classifying the novel samples into one of the three states; Bad generation: local rule violation, hallucinated sample: global rule violation, and generalized sample: no rule violation. Meanwhile, memorized (non-novel) samples can be identified by a simple look up in the training set. Second, truncating the tree at depth $k \neq 0$ flattens the k levels nearest the root ($\ell < k$) making the nodes at that level conditionally independent given the root. This leads to a family of filtered oracles BP_k that resolve correlation only up to a given scale.Throughout the rest of this paper, we would be using an RHM with the specifications v=6, m=4, s=2, and L=4 (as detailed in App.A). As shown in 2, for the purpose of state classification, we consider ℓ=1,2 as levels corresponding to global rules and ℓ=3,4 as levels corresponding to local rules.

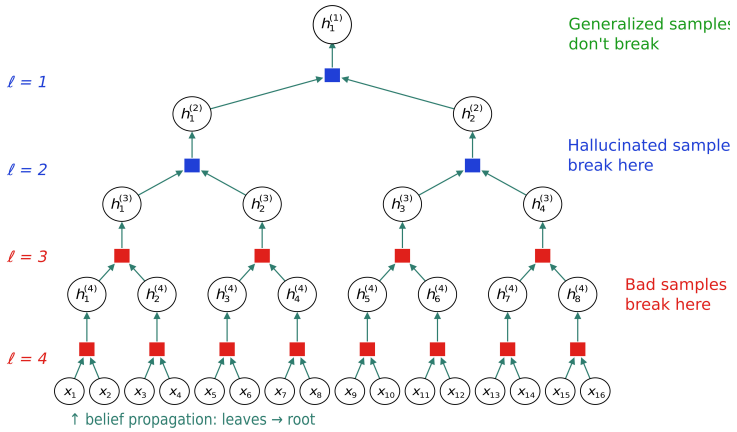
4 Level Resolving the scalar risk

In the RHM setting, as shown in App.B the population objective $J(\theta)$ decomposes into

$$J(\theta) \;=\; \mathrm{bp}_0(\theta) \;+\; H(\mathrm{BP}_0)$$

where BP_0 is the exact oracle posterior, evaluated under the full marginal distribution over the diffusion time t used in training, and $H(\mathrm{BP}_0)$ is its entropy and independent of θ. So the only term in $J(\theta)$ that training can actually decrease is $\mathrm{bp}_0(\theta)$ the forward KL divergence from the full tree oracle to the model's predicted posterior averaged over diffusion time and data.

More generally, for any truncation depth $k \in \{0, \ldots, L\}$, BP_k denotes the corresponding filtered oracle posterior. The denoising risk for a given level k is the forward KL divergence from the filtered



H

Figure 2: A random hierarchy model with branching factor s=2 and depth L=4 and how belief propagation can be used to convert it into a violation depth estimator and, hence, a state classifier.The leaf nodes x_i together denote the sequence, whereas $h_i^{(j)}$ denote the hidden nodes at level j.

oracle to the model's predicted posterior, averaged over the diffusion time and data.

$$\mathrm{bp}_k(\theta) \;:=\; \mathbb{E}_{t,\,x_0,\,x_t|x_0,t}\Big[\mathrm{D_{KL}}\big(\mathrm{BP}_k(\cdot \mid x_t,t)\,\big\|\,p_\theta(\cdot \mid x_t,t)\big)\Big], \tag{1}$$

$\mathrm{bp}_k(\theta)$ for $k \in \{0,\ldots,L\}$ can be thought of as a hierarchical ladder of denoising risks of the model at different resolutions of the true posterior. To see how this ladder looks like for the oracle itself, it is possible to define the forward KL divergence between the exact and the filtered oracle at a given level k, averaged over the diffusion time and data.

$$\mathrm{bp}_k^{\mathrm{orc}} \;:=\; \mathbb{E}_{t,\,x_0,\,x_t|x_0,t}\Big[\mathrm{D_{KL}}\big(\mathrm{BP}_k(\cdot \mid x_t,t)\,\big\|\,\mathrm{BP}_0(\cdot \mid x_t,t)\big)\Big]. \tag{2}$$

This ladder formed by the oracle is a model-agnostic property of the grammar alone and $\mathrm{bp}_0^{\mathrm{orc}} = 0$ by construction. For the filtered oracles ($k \neq 0$) , the magnitude of their denoising risk $\mathrm{bp}_k(\theta)$ on its own is misleading, as even a model that has exactly recovered the true posterior, $p_\theta = \mathrm{BP}_0$, still has $\mathrm{bp}_k(\theta) = \mathrm{bp}_k^{\mathrm{orc}} > 0$. Since BP_k is not the true posterior, the hierarchical filtering does introduce an error reflecting the amount of information being lost. We therefore get the excess denoising risk at truncation depth k by subtracting Eq.2 from Eq.1

$$g_k(\theta) \;=\; \mathrm{bp}_k(\theta) - \mathrm{bp}_k^{\mathrm{orc}}, \quad \text{with} \quad g_0(\theta) = \mathrm{bp}_0(\theta).$$

which vanishes identically for a perfect model at every k. Since g_k is a cumulative term for all levels under k i.e., all levels from k to L contribute to g_k, we telescope g_k to isolate the individual excess risk for each level:

$$R_L = g_L(\theta), \qquad R_\ell = g_\ell(\theta) - g_{\ell+1}(\theta) \;\; (\ell = 0,\ldots,L-1)$$

so that $\sum_{\ell=0}^{L} R_\ell \;=\; g_0(\theta) \;=\; \mathrm{bp}_0(\theta)$. R_L is the model's excess risk relative to the naive-Bayes oracle BP_L, which retains none of the rules. Each R_ℓ for $\ell < L$ isolates the excess risk contributed specifically by resolving correlations at level ℓ. $(R_0,\ldots,R_L)$ is, hence, the level resolved excess risk vector, and it sums exactly to the reducible part of the population risk.

To match with the classification based on violation depth that we defined in Sec.3, we collapse the $L+1$ entry ladder into three groups— global correlations, local correlations, and base. By grouping consecutively R_ℓ at a single split point $\ell^\star \in \{1,\ldots,L-1\}$:

$$D_{\mathrm{global}} = g_0(\theta) - g_{\ell^\star}(\theta), \qquad D_{\mathrm{local}} = g_{\ell^\star}(\theta) - g_L(\theta), \qquad D_{\mathrm{base}} = g_L(\theta),$$

an exact repartition of the same telescoped sum, $D_{\mathrm{global}} + D_{\mathrm{local}} + D_{\mathrm{base}} \;=\; \mathrm{bp}_0(\theta)$. $\ell^\star = 2$ in our setting. While g_k has been corrected for asymmetric floor, it still has not been corrected for asymmetric scale. The scale of D_{global} and D_{local} is again a property of the grammar, as the number

of production rules at a given level ℓ is proportional to s^{l-1}, so the local levels contain far more rule instances than the global ones and can carry more risk for that reason alone. We therefore rescale them by the corresponding gap in the oracle case, resulting in

$$n_{global} = \frac{D_{\text{global}}}{\left|\text{bp}_{\ell^\star}^{\text{orc}} - \text{bp}_0^{\text{orc}}\right|}, \qquad n_{local} = \frac{D_{\text{local}}}{\left|\text{bp}_L^{\text{orc}} - \text{bp}_{\ell^\star}^{\text{orc}}\right|}.$$

n_{global} and n_{local} are the global and local components of the model's excess risk, measured in units of the oracle's own gap for that group of levels. Unlike D_{global} and D_{local} they are comparable and can be used to differentiate how much global versus local correlation remains unlearned. By extension, there also exists a n_{base} which corresponds to the base component of risk that remains once all rules are removed. However, we don't study it further; as we primarily concern ourselves with hallucination and hence with distinguishing global versus local correlations. We evaluate n_{global} and n_{local} on heldout test samples analogous to the test loss, $\hat{n}_{global}$ and $\hat{n}_{local}$ are their training counterparts evaluated on training samples that are later used for regularization in Sec.4.2

4.1 The Hallucination Window and Heirarchical Lag

After levelwise resolving and scaling the test loss, on comparing the rescaled global risk component n_{global} and the rescaled local risk component n_{local} in Fig.3, we see a stark difference between the optimization of the global correlations and that of local correlations. This appears either as an absence of global risk minimization in low-data settings or as a delay in the onset of it as denoted by T_{global} in high-data settings.

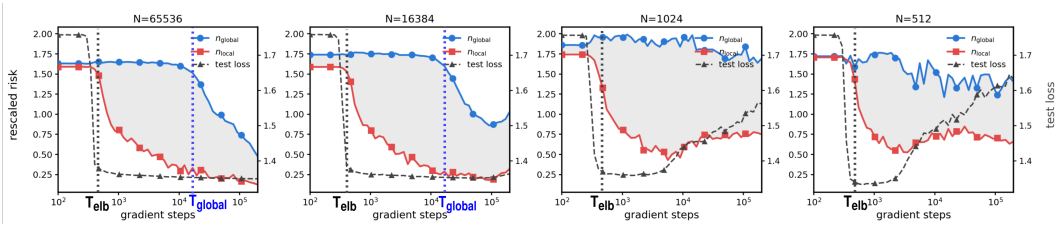


Figure 3: Comparison of the evolution of the rescaled global risk component n_{global} and the rescaled local risk component n_{local} overlaid with test loss in the RHM setting. For all training sizes N, the elbow of the test loss T_{elb} coincides with the onset of minimization of the rescaled local risk. Left: In a high data setting for training sizes N=65535 and N=16384, we can observe the delay in the onset of minimization of rescaled global risk as denoted by T_{global} as compared to the onset of minimization of rescaled local risk as denoted by T_{elb}. Right: In a low data setting for training sizes N=1024 and N=512, there is an absence of any coherent minimization of the global risk component.

This imbalance in the optimization is not evident from the scalar empirical risk (training loss), and hence any typical empirical risk minimization-based optimization like SGD (Robbins & Monro, 1951) or Adam (Kingma & Ba, 2015) does not account for it, which leads to hallucination. Further, the elbow of the test denoising loss T_{elb} coincides with the onset of minimization of the local-risk component. Hence, the window of training steps

$$W = [T_{elb}\,, T_{global}]$$

is an optimizable gap that delays generalization, and it is what we define as the hallucination window, whereas the difference between n_{global} and n_{local} at any given gradient step is defined as the hierarchical lag. Within W, the model has started resolving, or has resolved, the local structure while the global structure remains unresolved. As shown in the state composition plot (Fig.5) this window corresponds to the steps where the hallucinated samples peak. For $N = 65536$, the fraction of hallucinated samples rises from 24% at T_{elb} to a maximum of 46% when the lag is largest, just before T_{global}, and declines thereafter. At low N, the window never closes, and the training passes directly from hallucination to memorization.

Since generalization cannot be achieved before global correlations start getting resolved, generalization time T_{gen} must satisfy $T_{\text{gen}} \geq \tau_{\text{global}} = T_{\text{elb}} + |W|$, which means $T_{\text{elb}} + |W|$ is a floor on the generalization time, and shrinking the hallucination window $|W|$ means reducing T_{gen} itself as long as $T_{\text{gen}} - \tau_{\text{global}}$ does not grow to compensate. Hence, shrinking the window amounts to accelerating generalization.

4.2 The Hierarchical Lag Regularizer

The ideal regularizer would be one that could shift the minimization trajectory n_{global} to the left without affecting that of n_{local}. This ideal is a statement about entire trajectories and cannot be imposed at a single training step, so we propose a simple regularizer to shrink the hallucination window by reducing the hierarchical lag between the global and local risk minimizations. We express it as a constrained problem that at every step, the global risk component should not be less than the local risk component.

$$\min_{\theta} \; J_S(\theta) \quad \text{s.t.} \quad \text{lag}(\theta) := \hat{n}_{global}(\theta) - \hat{n}_{local}(\theta) \; \leq \; 0$$

where $J_S(\theta)$ is the training objective over a training set S (as defined in App.B) and $\text{lag}(\theta)$ is the hierarchical lag defined over the training samples. This constrained problem can be converted into an unconstrained one via a quadratic penalty, giving the regularized loss.

$$\min_{\theta} \; J_S(\theta) + \lambda \left[\max(0, \text{lag}(\theta)) \right]^2$$

The $\max(0, \text{lag}(\theta))$ term in the penalty term can be implemented via a ReLU function, which makes the penalty exactly zero whenever the constraint holds, so it only acts inside the hallucination window. and its gradient is given by

$$\nabla_\theta \Big[\lambda \left(\text{ReLU}(\text{lag}) \right)^2 \Big] = 2\lambda \max(0, \text{lag}) \cdot \nabla_\theta \text{lag} = 2\lambda \max(0, \text{lag}) \Big(\nabla_\theta \hat{n}_{global} - \nabla_\theta \hat{n}_{local} \Big)$$

When the constraint is violated, the optimizer can reduce the penalty either by decreasing $\hat{n}_{\text{global}}$ or by increasing $\hat{n}_{\text{local}}$. To "freeze" $\hat{n}_{\text{local}}$ for a given gradient step, we use a stop-gradient over $\hat{n}_{\text{local}}$. The regularizer thus enforces only a one-sided push of $\hat{n}_{\text{global}}$ toward a fixed $\hat{n}_{\text{local}}$, rather than a symmetric two-way pull. The regularized loss thus becomes

$$\mathcal{L}_{\mathcal{S}}(\theta) \; = \; J_S(\theta) \; + \; \lambda \Big[\text{ReLU}\big(\hat{n}_{global}(\theta) - sg\big[\hat{n}_{local}(\theta) \big] \big) \Big]^2 \tag{3}$$

The penalty being L2 ensures that the magnitude of the correction on n_{global} scales linearly with the amount of violation, ensuring the correction decays when close to n_{local}. An L1 penalty would instead apply a correction of constant magnitude and could overshoot the correction.

5 Closing the Window for the RHM

We train a diffusion transformer on the RHM data for different training sizes N 512,1024,2048,4096,8192,16384,65536 with the regularized loss (3) using an Adam optimizer with a initial learning rate of 3×10^{-4} for over 2×10^5 gradient steps with a batch size of 512. All reported values are averaged over 2 seeds.

Shrinkage of Hierarchical Lag. As a direct consequence of the hierarchical lag regularization, we observe in Fig.4 that the hallucination window, i.e., the gap, between T_{global} and T_{elb} shrinks. Across all values of N, T_{global} moves from around 10^4 gradient steps to around 10^3 gradient steps , so the hallucination window shrinks by an order of magnitude. In the smaller training sets N=4096,8192 , the gap reopens late in training steps. This is consistent with the fact that in the memorization/overfitting regime present at the late training steps in low-data setting, the regularization we apply on $\hat{n}_{global}$ doesn't generalize to n_{global}.

Impact on State Composition. A nontrivial consequence of the regularization can be seen in the state composition before (Fig.5) and after regularization (Fig.6). The regularizer only acts on the rescaled risk components; it never sees the sample itself. If the composition changes, it is evidence that the hierarchical lag is causally involved in hallucination. Quantitatively at all values of N, we see a consistent drop in percentage of hallucination at all gradient steps. For example, for N=65536,

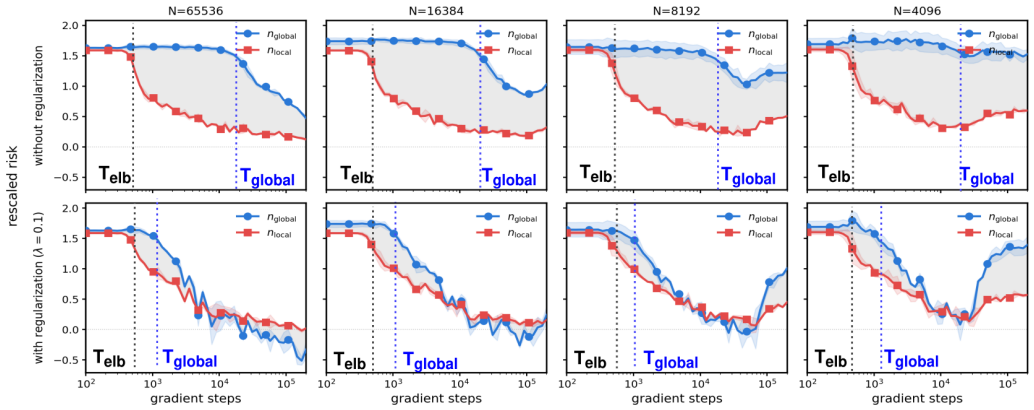


Figure 4: Comparison of the evolution of the rescaled global risk component n_{global} and the rescaled local risk component n_{local} in RHM before (top) and after (bottom) regularization for training sizes N= 65536,16384,8192,4096. We can see that after regularization ($\lambda = 0.1$) the width of the hallucination window $|W|$, i.e, the gap between T_{elb} and T_{global} reduces.

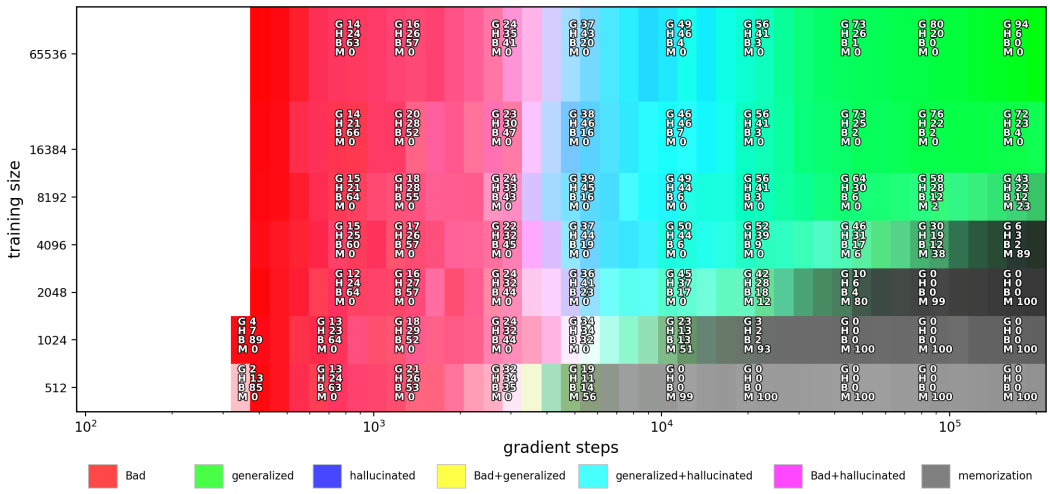


Figure 5: State composition of the generated sequences at different training regimes (number of training samples x gradient steps). The text in each cell top-to-bottom reports the percentage of generalized (G) hallucinated (H) and bad-generation (B) and (M) memorized samples among 1000 generated sequences. At early (around 10^3) gradient steps bad generation dominates; at intermediate (around 10^4) gradient steps hallucinated samples dominates before settling into mostly generalized samples at late (around 10^5) timesteps in the high data setting or collapsing into memorization in the low data regime. Cells around 10^2 steps are faint because the model produces few distinct sequences there (opacity encodes percentage of distinct samples), so their percentages are not informative.

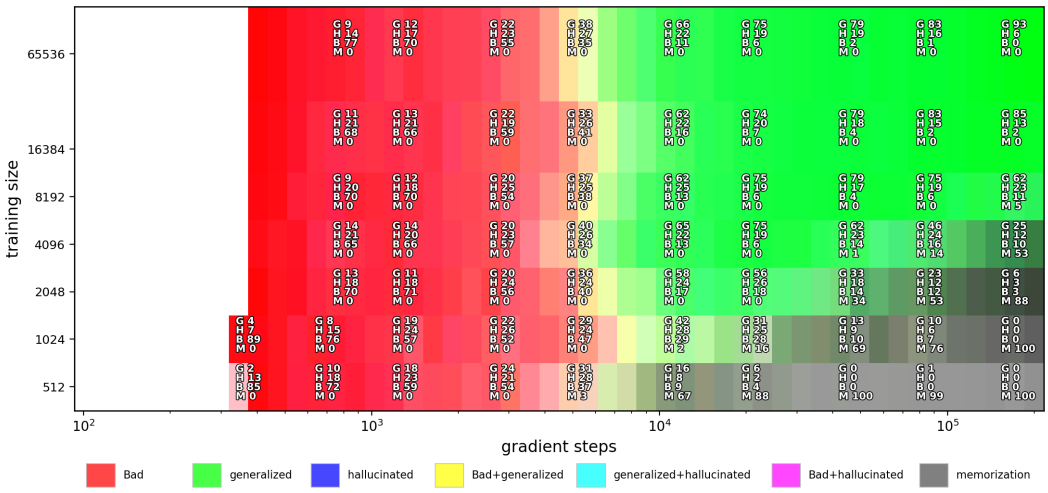


Figure 6: State composition plot, with the lag regularizer ($\lambda = 0.1$). The hallucination-dominant (blue) band visible in 5 disappears at every N: the composition transitions from a bad generation-dominant band to a generalization-dominant band denoted by the yellow cells separating them.

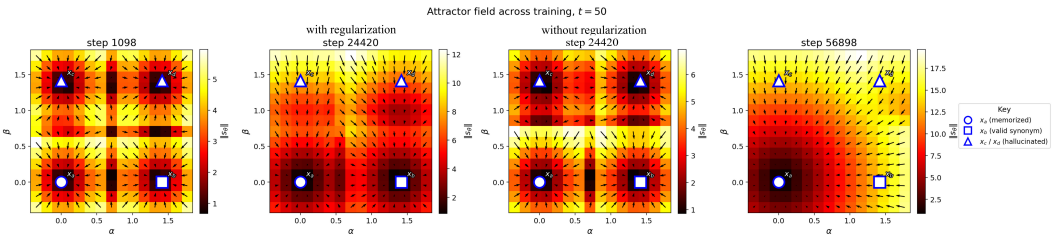


Figure 7: Score field $\|s_\theta\|$ (heatmap) and direction (arrows) at diffusion time $t = 50$, on the 2D plane through a training sequence x_a, a generalized synonym x_b, and two hallucinated sequences x_c, x_d, before and after hierarchical lag regularization ($\lambda = 0.1$), for $N = 4096$. At intermediate gradient steps without regularization, the hallucinated sequences x_c and x_d behaved like attractor basins in the score field as strong as the valid sequences x_a, x_b; after regularization x_c and x_d no longer behaved like attractor basins in the same range ($2 \times 10^4 - 5 \times 10^4$).

the peak percentage of hallucination percentage drops from 46% to 27%, and the generalization percentage around 10^4 steps rises from 49% to 66%. More holistically, the band of gradient steps where hallucinated samples are dominant no longer exists. The training dynamics shift directly from a bad dominant band to a generalized dominant band. This effect is not free at early steps; the training dynamics of the regularized loss spend longer in bad generation as the extra push on the global correlations does compete with the optimization of the local correlations.

Impact on Score field. In 7 we study the more mechanistic impact of the regularizer in the denoiser's score field, extending the vector field visualization from Wang et al. (2026). During sampling, the reverse process follows the score, so points where its magnitude vanishes act as attractor basins. We take a training sequence x_a, a generalized synonym x_b, and two hallucinated sequences x_c, x_d, and evaluate the score on the 2D plane through them. Without regularization, the hallucinated sequences are basins as deep as the valid ones well into the window, so the sampler has no reason to prefer valid sequences over hallucinated ones nearby. With regularization, the field around x_c and x_d no longer converges on them in the same range of training. At later steps only the training sequence x_a remains a sink, a sign of memorization consistent with the late reopening of the gap at N=4096 in Fig.3.

6 LIMITATIONS

Dependence on the oracle. The results are on a random hierarchy model which makes the four-way partition exactly decidable and supplies an exact Bayes-optimal oracle; the same property is what limits the scope of the method. Both hierarchical lag as a diagnostic and the regularizer we impose on it require the filtered oracle's BP_k. For real data, no exact oracle is available, so transferring to real-world data like images, proteins, or language would require approximate, learned substitutes for BP_k, and whether the hierarchical lag and the regularizer could survive that approximation is untested.

Specificity of regularizer. The regularizer is designed to only target the hallucination window and nothing else. It does not prevent memorization: at the smaller dataset sizes, the gap between n_{global} and n_{local} reopens late in training as memorization sets in. It reduces the hallucinated fraction but does not eliminate it, since the constraint $n_{global} \leq n_{local}$ only asks global structure to keep pace with local structure, not to be fully learned.

7 CONCLUSION

In this work, we show that the training loss (or test loss) of a diffusion model cannot differentiate between hallucination and generalization; both live on the same plateau, and the loss flattens, i.e., has an elbow considerably before the model actually generalizes. We then show that resolving the reducible part of the loss level by level against filtered belief-propagation oracles, grouping them, and rescaling each group by the oracle's own gap, makes the difference visible. The component corresponding to local correlations starts getting minimized at T_{elb}, while the global component is minimized one to two orders of magnitude later at T_{global}, or not at all when data is scarce. We find that between T_{elb} and T_{global} lies the hallucination window $W = [T_{elb}, T_{global}]$, in which the model has learned local correlations but not started learning global correlations and in which the hallucinated fraction of generated samples peaks. We define the difference between the rescaled global risk component and the local risk component in this window as the hierarchical lag.

We then show how a simple regularizer derived from hierarchical lag shrinks the window by about an order of magnitude in training time. Acting only on the rescaled risk curves, it consistently lowers the hallucinated fraction across all training sizes and gradient steps, eliminating the hallucination-dominant band in the training dynamics. More broadly, these results suggest that hallucination in diffusion models is a problem of timing rather than of capacity: the model can learn global long-range correlations, but ordinary training gives it little reason to do so at the same pace as local correlations. Objectives and architectures that account for when each level of structure is learned and not only for the total loss are a natural route to diffusion models that hallucinate less. Since models failing to follow global constraints are seen across different generative settings from video generation (Kang et al., 2025) to drug discovery (Christopher et al., 2025), overcoming the dependence on the exact oracle Sec.6 and extending the hierarchical lag diagnostic and regularizer to real-world data is the next step. Training choices guided by this diagnostic could fundamentally improve generation across these settings, from reducing visual artifacts in videos to improving the physical realizability of designed proteins.

AI USE STATEMENT

In this work, we used generative AI tools for assisting with the implementation of the regularization and the different visualization plots used in the paper. We have not used generative AI tools for helping develop theoretical models or conceptual frameworks, designing or providing feedback on research methodology or experiments, interpreting results. Generating synthetic data sets, formulating mathematical claims, assisting in the writing of proofs, proposing or refining hypotheses, providing critical ingredients for proving mathematical claims, assisting with translation, cleaning and reformatting datasets, and supporting qualitative and thematic data analysis are not applicable to this work. Additionally, we used generative AI tools for creating the illustrations used on the paper and for aiding with polishing the paper. We have reviewed all AI-assisted work. LLM-generated code was verified and tested for correctness by the first author, and the training pipeline derived from Garnier-Brun et al. (2026) was used without modification. All illustrations have been checked by

the first author for factual errors. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

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

Tony Bonnaire, Raphaël Urfin, Giulio Biroli, and Marc Mézard. Why diffusion models don't memorize: The role of implicit dynamical regularization in training. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/ceb7f3cc876a6dcb15130a645b5a4507-Abstract-Conference.html.

Nicholas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramèr, Borja Balle, Daphne Ippolito, and Eric Wallace. Extracting training data from diffusion models. In Joseph A. Calandrino and Carmela Troncoso (eds.), *32nd USENIX Security Symposium, USENIX Security 2023, Anaheim, CA, USA, August 9-11, 2023*, pp. 5253–5270. USENIX Association, 2023. URL https://www.usenix.org/conference/usenixsecurity23/presentation/carlini.

Jacob K. Christopher, Austin Seamann, Jingyi Cui, Sagar Khare, and Ferdinando Fioretto. Constrained diffusion for protein design with hard structural constraints. *CoRR*, abs/2510.14989, 2025. doi: 10.48550/ARXIV.2510.14989. URL https://doi.org/10.48550/arXiv.2510.14989.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems*, volume 34, pp. 8780–8794, 2021.

Alessandro Favero, Antonio Sclocchi, Francesco Cagnetta, Pascal Frossard, and Matthieu Wyart. How compositional generalization and creativity improve as diffusion models are trained. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025. URL https://proceedings.mlr.press/v267/favero25a.html.

Alessandro Favero, Antonio Sclocchi, and Matthieu Wyart. Bigger isn't always memorizing: Early stopping overparameterized diffusion models. *Trans. Mach. Learn. Res.*, 2026, 2026. URL https://openreview.net/forum?id=P6r9OxZ1vm.

Jerome Garnier-Brun, Marc Mézard, Emanuele Moscato, and Luca Saglietti. How transformers learn structured data: Insights from hierarchical filtering. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025. URL https://proceedings.mlr.press/v267/garnier-brun25a.html.

Jerome Garnier-Brun, Luca Biggio, Davide Beltrame, Marc Mézard, and Luca Saglietti. Biased generalization in diffusion models. *CoRR*, abs/2603.03469, 2026. doi: 10.48550/ARXIV.2603.03469. URL https://doi.org/10.48550/arXiv.2603.03469.

Anand Jerry George, Rodrigo Veiga, and Nicolas Macris. Denoising score matching with random features: Insights on diffusion models from precise learning curves. *CoRR*, abs/2502.00336, 2025. doi: 10.48550/ARXIV.2502.00336. URL https://doi.org/10.48550/arXiv.2502.00336.

Yujin Han, Andi Han, Wei Huang, Chaochao Lu, and Difan Zou. Can diffusion models learn hidden inter-feature rules behind images? In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025. URL https://proceedings.mlr.press/v267/han25b.html.

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

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 8633–8646, 2022.

Zahra Kadkhodaie, Florentin Guth, Eero P. Simoncelli, and Stéphane Mallat. Generalization in diffusion models arises from geometry-adaptive harmonic representations. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=ANvmVS2Yr0.

Mason Kamb and Surya Ganguli. An analytic theory of creativity in convolutional diffusion models. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025. URL https://proceedings.mlr.press/v267/kamb25a.html.

Bingyi Kang, Yang Yue, Rui Lu, Zhijie Lin, Yang Zhao, Kaixin Wang, Gao Huang, and Jiashi Feng. How far is video generation from world model: A physical law perspective. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025. URL https://proceedings.mlr.press/v267/kang25g.html.

Seunghoi Kim, Chen Jin, Tom Diethe, Matteo Figini, Henry F. J. Tregidgo, Asher Mullokandov, Philip Teare, and Daniel C. Alexander. Tackling structural hallucination in image translation with local diffusion. In *European Conference on Computer Vision (ECCV)*. Springer, 2024.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org.

Rui Lu, Runzhe Wang, Kaifeng Lyu, Xitai Jiang, Gao Huang, and Mengdi Wang. Towards understanding text hallucination of diffusion models via local generation bias. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=SKW10XJlAI.

Song Mei. U-nets as belief propagation: Efficient classification, denoising, and diffusion in generative hierarchical models. *CoRR*, abs/2404.18444, 2024. doi: 10.48550/ARXIV.2404.18444. URL https://doi.org/10.48550/arXiv.2404.18444.

Marc Mézard and Andrea Montanari. *Information, Physics, and Computation*. Oxford University Press, 2009.

Leonardo Petrini, Francesco Cagnetta, Umberto M. Tomasini, Alessandro Favero, and Matthieu Wyart. How deep neural networks learn compositional data: The random hierarchy model. *CoRR*, abs/2307.02129, 2023. doi: 10.48550/ARXIV.2307.02129. URL https://doi.org/10.48550/arXiv.2307.02129.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407, 1951. doi: 10.1214/aoms/1177729586.

Antonio Sclocchi, Alessandro Favero, and Matthieu Wyart. A phase transition in diffusion models reveals the hierarchical nature of data. *CoRR*, abs/2402.16991, 2024. doi: 10.48550/ARXIV.2402.16991. URL https://doi.org/10.48550/arXiv.2402.16991.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 2256–2265. PMLR, 2015.

Binxu Wang, Emma Lucia Byrnes Finn, and Bingbin Liu. The two clocks and the innovation window: When and how generative models learn rules. *CoRR*, abs/2605.10019, 2026. doi: 10.48550/ARXIV.2605.10019. URL https://doi.org/10.48550/arXiv.2605.10019.

Joseph L Watson, David Juergens, Nathaniel R Bennett, Brian L Trippe, Jason Yim, Helen E Eisenach, Woody Ahern, Andrew J Borst, Robert J Ragotte, Lukas F Milles, et al. De novo design of protein structure and function with RFdiffusion. *Nature*, 620(7976):1089–1100, 2023.

Robbie A. Watt and Laura A. Mansfield. Generative diffusion-based downscaling for climate. *arXiv preprint arXiv:2404.17752*, 2024.

A Data model and oracle Construction

We use a random hierarchy model (Petrini et al., 2023) in which one rule set is shared by all levels, along with filtered belief-propagation oracles construction from (Garnier-Brun et al., 2025). The specifications of the RHM we use are

- $v = 6$ (vocabulary size): every symbol, at every depth, takes one of v values.
- $s = 2$ (branching factor): each production has s children.
- $m = 4$ (productions per symbol): each symbol can be rewritten into exactly m distinct s-tuples.
- $L = 4$ (depth): the number of rule levels between the root and the leaves, giving sequences of length $s^L = 16$.

Production rules. The rules are encoded in a tensor $M \in \mathbb{R}_+^{v\times v\times v}$, where M_{abc} is the probability that symbol a is rewritten as the pair (b, c). The $v^2 = 36$ possible pairs are shuffled and split into v disjoint blocks of v pairs, one block per parent; each parent uses m pairs of its block, each with uniform probability $1/m$. Blocks are disjoint, so $M_{abc}M_{a'bc} = 0$ for $a \neq a'$: every admissible pair has a unique parent and parsing is deterministic. The same M is applied at every level. The standard RHM instead draws an independent rule set per level. With a shared M, global and local levels apply the same productions and differ only in the span over which they are composed.

Size of the grammar. The grammar admits $v\, m^{(s^L-1)/(s-1)} = 6\cdot 4^{15} \approx 6.4\times 10^9$ valid sequences, about 0.2% of the $v^{s^L} = 6^{16}$ possible ones. Even the largest training set ($N = 65536$) covers a vanishing fraction of it, so a novel valid sample reflects generalization rather than recall.

A.1 Sampling data and Noising

A sample is generated top-down: the root $h_1^{(1)}$ is drawn uniformly from $[v]$, and M is applied at every node down to the 16 leaves. With uniform productions and a uniform root, all valid sequences are equally likely. Training sets are sampled without duplicates, and held-out test samples are drawn from the same grammar. Each sequence is one-hot encoded, $x_0 = \mathrm{onehot}_v(x) \in \{0, 1\}^d$ with $d = s^L v = 96$, so that position i is a block $x_{0,i} \in \{e_1, \dots, e_v\}$. Gaussian noise is added coordinate-wise,

$$x_t = \sqrt{\alpha_t}\, x_0 + \sqrt{1-\alpha_t}\, \xi, \qquad \xi \sim \mathcal{N}(0, I_d), \qquad \alpha_t = \prod_{i=1}^{t}(1-\beta_i).$$

The denoiser outputs, for each position, a softmax over the v symbols, interpreted as the posterior marginal $p_{\theta,i}(\cdot \mid x_t, t)$, and is trained with cross-entropy against the clean one-hot sequence.

A.2 Construction of Oracles

Because the noise is independent across positions, each noisy block enters belief propagation as a field on its leaf. Since $\|e_a\|^2 = 1$ for every symbol,

$$p(x_{t,i} \mid x_{0,i} = e_a) \;\propto\; \exp\Big(-\tfrac{\|x_{t,i}-\sqrt{\alpha_t}\,e_a\|^2}{2(1-\alpha_t)}\Big) \;\propto\; \exp\Big(\tfrac{\sqrt{\alpha_t}}{1-\alpha_t}\,x_{t,i}[a]\Big),$$

so the leaf field is $\phi_i = \mathrm{softmax}\big(\frac{\sqrt{\alpha_t}}{1-\alpha_t}\,x_{t,i}\big)$. An upward pass combines the fields of sibling nodes through the rule tensor,

$$\hat{\nu}_{\to p}(a) \;\propto\; \sum_{b,c} M_{abc}\,\nu_{\mathrm{left}}(b)\,\nu_{\mathrm{right}}(c),$$

up to the root, where it meets the uniform root prior; a downward pass returns the messages to the leaves. The product of incoming messages at each leaf is the exact posterior marginal $\mathrm{BP}_{0,i}(\cdot \mid x_t, t) = p(x_{0,i} = \cdot \mid x_t, t)$, the Bayes-optimal target for the cross-entropy objective.

The filtered model at level k removes the hierarchy between the root and depth k: the s^k nodes at that depth are drawn independently given the root, each through the composition of the left- and right-child marginals of M, $M^{\mathrm{L}}_{ab} = \sum_c M_{abc}$ and $M^{\mathrm{R}}_{ac} = \sum_b M_{abc}$, along its path from the root, while the usual branching process applies below. BP_k is exact BP on this filtered tree with the same leaf fields ϕ_j. It keeps all correlations within blocks of s^{L-k} leaves and removes those at larger scales, apart from what is mediated by the root; $k = 0$ is the full model and $k = L$ is naive Bayes (leaves independent given the root).

B Population and training risks

Population risk. We know for each position i, the model outputs a distribution $p_{\theta,i}(\cdot \mid x_t, t)$ over the v symbols and is trained with the cross-entropy summed over positions. The population risk is

$$J(\theta) = \mathbb{E}_{t,\,x_0,\,x_t|x_0,t}\Big[-\sum_{i=1}^{s^L} \log p_{\theta,i}(x_{0,i} \mid x_t, t)\Big].$$

To decompose it, we sample (t, x_t) first and then draw x_0 from its posterior $p(x_0 \mid x_t, t)$, whose i-th marginal is the exact oracle $\mathrm{BP}_{0,i}(\cdot \mid x_t, t)$:

$$\begin{aligned}
J(\theta) &= \mathbb{E}_{t,\,x_t}\,\mathbb{E}_{x_0|x_t,t}\Big[-\sum_i \log p_{\theta,i}(x_{0,i} \mid x_t, t)\Big] \\
&= \mathbb{E}_{t,\,x_t}\Big[\sum_i\; \mathbb{E}_{x_{0,i}\sim \mathrm{BP}_{0,i}}\big[-\log p_{\theta,i}(x_{0,i} \mid x_t, t)\big]\Big] \\
&= \mathbb{E}_{t,\,x_t}\Big[\sum_i \mathrm{CE}\big(\mathrm{BP}_{0,i},\, p_{\theta,i}\big)\Big] \\
&= \mathbb{E}_{t,\,x_t}\Big[\sum_i H\big(\mathrm{BP}_{0,i}\big) + D_{\mathrm{KL}}\big(\mathrm{BP}_{0,i} \,\|\, p_{\theta,i}\big)\Big] \\
&= \underbrace{\mathbb{E}_{t,\,x_t}\Big[\sum_i H\big(\mathrm{BP}_{0,i}\big)\Big]}_{H(\mathrm{BP}_0)} \;+\; \underbrace{\mathbb{E}_{t,\,x_t}\Big[\sum_i D_{\mathrm{KL}}\big(\mathrm{BP}_{0,i} \,\|\, p_{\theta,i}\big)\Big]}_{\mathrm{bp}_0(\theta)}.
\end{aligned}$$

Training risk. Training minimizes the empirical counterpart over the training set S,

$$J_S(\theta) = \frac{1}{N}\sum_{x_0\in S} \mathbb{E}_{t,\,x_t|x_0,t}\Big[-\sum_i \log p_{\theta,i}(x_{0,i} \mid x_t, t)\Big]. \tag{4}$$

Similar to the population risk but with the empirical distribution in place of the grammar; the clean sequence given (t, x_t) is now distributed as the empirical posterior $\hat{p}_S(x_0 \mid x_t, t) \propto q(x_t \mid x_0, t)\,\mathbf{1}[x_0 \in S]$, so

$$J_S(\theta) = H(\hat{p}_S) + \mathbb{E}_{t,\,x_t\sim\hat{q}_S}\Big[\sum_j D_{\mathrm{KL}}\big(\hat{p}_{S,i} \,\|\, p_{\theta,i}\big)\Big],$$

where $\hat{q}_S$ is the distribution of noisy training samples. The minimizer of J_S is the empirical posterior, which reproduces the training set. Because $\hat{p}_S$ is supported on only N sequences, it is typically much more concentrated than BP_0, so $H(\hat{p}_S)$ lies below $H(\mathrm{BP}_0)$ and the training loss can fall below the population floor. The generalization gap $\varepsilon_S(\theta) = J_S(\theta) - J(\theta)$ is small while the model tracks BP_0 and grows once it starts fitting $\hat{p}_S$, which is the onset of memorization.

C SAMPLE COMPOSITION ALONG LOSS CURVES

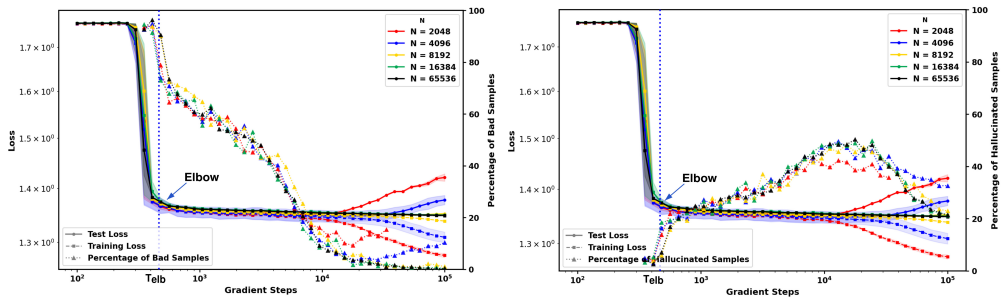


Figure 8: Percentage composition of bad (left) and hallucinated (right) across 2×10^5 gradient steps overlaid with training and test loss curves

Fig. 8 overlays the test and training loss with the fraction of bad and hallucinated samples, for all training-set sizes. It complements the elbow analysis of Fig. 1 b): the loss flattens at T_{elb} steps for every N, but generation keeps changing well after that. Badly generated samples keep decreasing, while hallucinated samples, rise after the elbow and peak just before T_{global} (of high data setting). Both changes happen while the loss is essentially constant, so the scalar loss cannot tell a model that is still hallucinating from one that has generalized.

D COMPARISON PLOTS FOR $\hat{n}_{global}$ AND $\hat{n}_{local}$

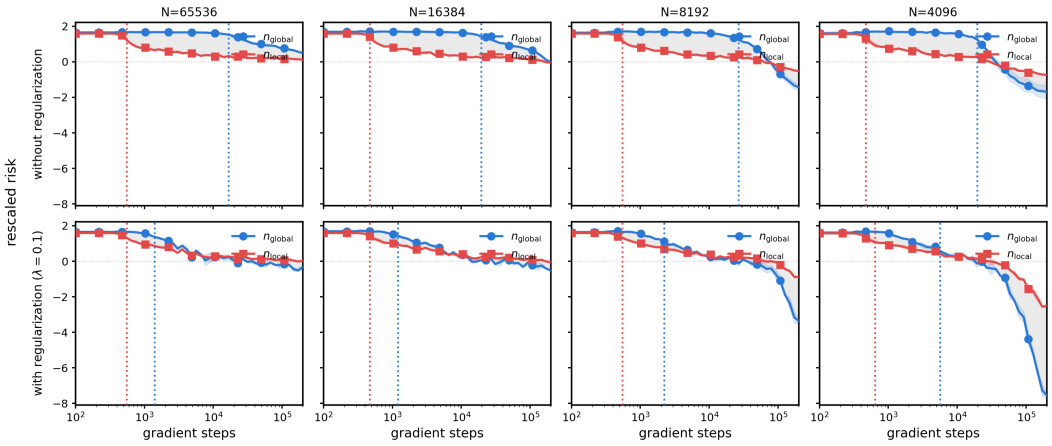


Figure 9: The empirical counterparts of the before and after regularization rescaled risk curves in Fig.4. The dashed lines represent empirical counterparts of T_{elb} (red) and T_{global} (blue).

Fig. 9 shows the empirical counterparts $\hat{n}_{\mathrm{global}}$ and $\hat{n}_{\mathrm{local}}$, computed on training batches. After regularization, the lag between $\hat{n}_{\mathrm{global}}$ and $\hat{n}_{\mathrm{local}}$ closes shortly after the elbow, and $\hat{n}_{\mathrm{global}}$ later

falls below $\hat{n}_{\text{local}}$. Late in training, both curves become negative, with the flip being earlier and much larger for smaller N. The large negative value of hierarchical lag means that, on training samples, the model is further from the coarser oracle than the exact posterior is: it has become sharper than BP_0 by fitting the empirical posterior $\hat{p}_S$ (App. B).

E TRAINING SETUP.

We train the denoiser with the pipeline of Garnier-Brun et al. (2026), without modification. The denoiser is a transformer with 8 layers, hidden dimension 512 and 4 attention heads, using standard positional embeddings together with sinusoidal embeddings of the diffusion time. It takes the noisy one-hot sequence x_t and outputs, for each position, a softmax over the v symbols, trained with cross-entropy against the clean sequence. The forward process uses $T = 500$ diffusion steps with a linear schedule from $\beta_1 = 2 \times 10^{-4}$ to $\beta_T = 4 \times 10^{-2}$, and each training example is paired with a time $t \sim \mathcal{U}\{1, \dots, T\}$. We optimize with Adam at a learning rate of 3×10^{-4} and a batch size of 512, for training-set sizes $N \in \{512, \dots, 65536\}$, up to 2×10^5 gradient steps. For the regularized runs we add the penalty of Eq. 3 with $\lambda = 0.1$ for all N, leaving everything else unchanged. Each run is checkpointed at 45 points on a logarithmic grid of gradient steps, from step 100 to 2×10^5, and results are averaged over 2 seeds, with shaded regions showing the range across seeds.