Language models suffer from a curse of ambiguity

Nicolas Zucchet **Hyun Dong Lee** **Scott Linderman**
Stanford University
{nzucchet, hdlee, scott.linderman}@stanford.edu

Abstract

Large language models increasingly rely on sampling as a driver of their own improvement, making the fidelity of their learned distributions more critical than ever. Yet, not all distributions are equally easy to learn. In this work, we identify a curse of ambiguity: in large language models, and more broadly in all neural networks that produce discrete probability distributions, the more ambiguous a next-token distribution is, the harder it is to learn accurately. Through an extensive theoretical analysis, we trace this curse to architectural and learning roots. More ambiguous distributions require more capacity to be stored, larger embeddings to be represented, more steps to be fitted, and amplify token-sampling noise. We validate these findings on synthetic tasks with controlled ground truth and observe the same signatures in language models trained on real data. Our results provide a new perspective on the statistical capabilities of large language models and a practical framework for when to trust their output distribution.

1 Introduction

Autoregressive language models generate text one token at a time, sampling a conditional distribution given the tokens that came before [1–8]. We often use greedy decoding strategies to bias samples toward the most likely tokens [9–12], but such strategies have deleterious effects. Inference-time scaling methods [13–16] only succeed when the model spreads probability over diverse continuations, not just the top ones [17, 18]. Similarly, on-policy reinforcement learning post-training [19, 20] can only explore and reinforce trajectories the model actually samples [21]. As model-generated synthetic data [22, 23] and AI-generated content online [24, 25] become an increasingly large share of training corpora, a lack of diversity in the sampled distribution risks compounding across generations until the model collapses [26]. For all these reasons, the learned distribution needs to be well calibrated, not only capturing the most likely token but the full next-token distribution. This necessity is especially acute in ambiguous settings where several continuations are plausible.

In this work, we take a step toward understanding how well neural networks, and thus large language models, can model such uncertainty. We uncover a new phenomenon, the *curse of ambiguity*:

The more ambiguous the next-token distribution, the harder it is to learn.

The rest of the paper is dedicated to making this statement precise, to uncovering its roots, and to discussing its consequences for large language models. Our contributions are the following:

- We identify and formalize the curse of ambiguity in large language models.
- We trace the curse to architectural roots (Section 3)—more ambiguous distributions require more capacity to store and larger embeddings to represent—and to learning roots (Section 4)—they take more optimization steps to fit and amplify the noise from token sampling.
- We validate the curse empirically (Section 5), confirming our theoretical predictions in controlled synthetic settings, showing that the same signatures appear in language models trained on real data, and arguing that these results extend to larger models.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

2 Autoregressive language modeling as context classification

Most large language models are decoder-only Transformers [5–8] that learn to autoregressively predict the next token [1–4]. In our analysis, we view this process as a context-to-token classification problem, and we distinguish the many contexts that language models must classify by the ambiguity of the next token under the training distribution. This section introduces this formalism and motivates the simplified toy model we analyze theoretically in Sections 3 and 4.

From sequences to classification. An autoregressive language model factorizes the probability of a sequence $x_{1:t}$ as a product of conditional distributions, each predicting x_t given its preceding context $x_{1:t-1}$, that is

$$p_\theta(x_{1:t}) = \prod_{t'=1}^{t} p_\theta(x_{t'}|x_{1:t'-1}) \tag{1}$$

with θ the parameters of the model and p_θ the distribution it defines. Mechanistically, $p_\theta(x_t|x_{1:t-1})$ is computed by first letting a Transformer-based backbone map the context $x_{1:t-1}$ to an embedding $e_t = f_\theta(x_{1:t-1})$ of dimension h, and then applying a linear head W followed by a softmax to produce a distribution over the size-d vocabulary:

$$p_\theta(x_t = j \,|\, x_{1:t-1}) = \mathrm{softmax}(We_t)_j, \text{ where } e_t = f_\theta(x_{1:t-1}). \tag{2}$$

From this perspective, next token prediction is a classification problem: the backbone produces a representation of the context, and then the linear head predicts the next token with a multi-class logistic regression. Training jointly fits the backbone and the head by minimizing, on average over training contexts, the cross-entropy between the ground-truth and predicted distributions.

Focusing on the linear head for analytical tractability. Sections 3 and 4 study this last linear head in isolation, treating the context representations produced by the backbone as given. The linear head plays a central role in the network: it is the only component that maps embeddings to the output vocabulary, and it modulates all gradients flowing back into the backbone [27]. Studying it in isolation therefore probes the part of the model most directly responsible for fitting the next-token distribution. Representation learning is, in this respect, an important but secondary concern. Section 5 will then empirically show that similar results hold for the language model as a whole.

Since we view the last layer as a logistic regression, we can simplify our notation: the head sees n context embeddings e_i paired with target distributions p_i, and must learn to map one to the other. We study how it copes when these targets vary in ambiguity. By default, we treat the embeddings as sampled i.i.d. and uniformly from the unit sphere. In Section 3.2, we will instead ask what optimal embeddings would look like and what they can achieve at a fixed dimension h; in that regime, we jointly optimize the weights W and the embeddings e_i.

Modeling ambiguity through support size. In natural language, contexts often admit multiple plausible continuations: after "the cat sat on the", reasonable next tokens include "mat", "floor", "couch", etc. We model this ambiguity by assigning each context i a ground truth distribution p_i that is uniform on a random support S_i of size k. The probability of context i being followed by token j is

$$p_{i,j} = \begin{cases} \frac{1}{k} & \text{if } j \in S_i \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

The hyperparameter k controls the degree of ambiguity:[1] $k = 1$ corresponds to a context with a single possible continuation according to the data, while $k > 1$ means several tokens are equally likely. Although next-token distributions are obviously not uniform in practice, this model captures the key intuition: spreading probability mass over more tokens makes the prediction problem intrinsically harder. On the non-uniform distributions we encounter in our experiments in Section 5, we measure ambiguity through the effective support size defined as the exponential of the entropy of p_i. It is more robust than the raw support size in the sense that vanishing probabilities contribute vanishingly little, and it reduces to k on the uniform-on-support distributions we study in theory. The same quantity appears under different names in other fields, including perplexity in information theory [29] and multiplicity in statistical mechanics [30].

[1] We use "ambiguity" beyond its linguistic sense [28], as a term for distributions spreading mass over many tokens.

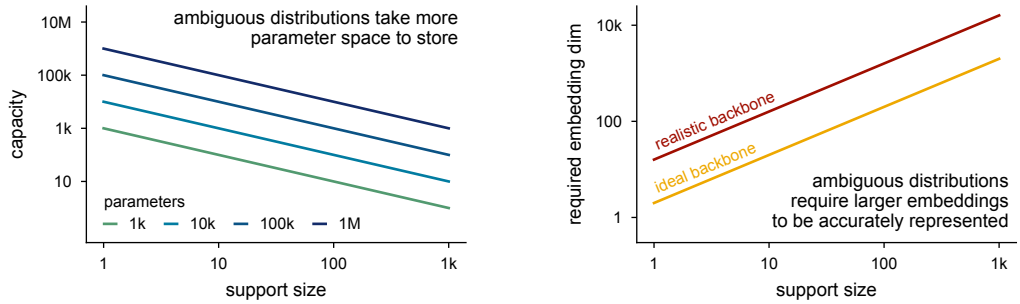

Figure 1: **Architectural roots of the curse of ambiguity: ambiguous distributions require more parameters to be stored and larger embeddings to be represented.** Schematic description of the theoretical analysis of Section 3, depicting how the network capacity (left) and the required embedding dimension (right) depend on our measure of ambiguity, which is the number of possible next tokens k (the support size) under the ground-truth distribution. The capacity is proportional to the number of parameters and inversely proportional to k, while the required embedding dimension grows proportionally with k, with a larger constant when the backbone cannot produce arbitrary embeddings (realistic vs. ideal backbone). Capacity predictions are empirically validated in Appendix A.6.

3 Architectural roots of the curse of ambiguity

The curse of ambiguity is partially rooted in the architectural properties of the model. More precisely, we show that, as they become more ambiguous, next token distributions require more capacity to be stored at a desired level of accuracy and larger embedding dimensions to be properly represented. We focus on gaining theoretical intuition for why this holds true using linear models.

3.1 Model capacity: ambiguous distributions require more parameters to be stored

Intuitively, modeling k possible continuations is like solving k different context-to-token associations at once, which should take k times more capacity. We now show that this is indeed the case. We consider the linear head model of Equation 2 with random embeddings e_i and ask how the number of context-dependent distributions p_i (as defined in Equation 3) it can store depends on k. By "store" we informally mean the maximum number of contexts that the model can accurately recall; Appendix A.1 gives a precise definition. This can be viewed as an associative memory problem, connecting to a long line of work on neural associative memories, historically on Hopfield networks [31–33] and more recently on language models [34–40].

A Hebbian model. Following previous work [41, 42], we study a Hebbian-like construction where the learned weight matrix is approximated by,

$$W = \beta \sum_{i=1}^{n} z_i \, e_i^\top \quad \text{with } z_{i,j} = \begin{cases} 1 - k/d & \text{if } j \in S_i \\ -k/d & \text{otherwise} \end{cases}. \tag{4}$$

Here, z_i are the target logits, and they are designed such that the difference between on- and off-support tokens is one and their corresponding probability distributions are close to p_i. We can think of W as approximating the result of online gradient descent to minimize $\|z_i - We_i\|^2$ with learning rate β, assuming the embeddings are approximately orthogonal. Minimizing the actual cross-entropy loss by gradient descent would yield different weights with greater capacity, so this analysis only provides a lower bound on capacity. However, the bound is close to tight: the Hebbian model captures the same scaling with k as a linear head trained with Adam on random embeddings (Appendix A.6).

Ambiguity requires more capacity. To understand how well the Hebbian model predicts p_i given embedding e_i of size h, we decompose its output logits into a signal and a noise term:

$$\ell_i := We_i = \beta\big(\underbrace{z_i e_i^\top e_i}_{\text{signal}} + \underbrace{\textstyle\sum_{j \neq i} z_j \, (e_j^\top e_i)}_{\text{noise}}\big). \tag{5}$$

Since the embeddings are unit vectors, the signal term has fixed magnitude and recovers z_i exactly. The noise term comes from interference between embeddings. It has zero mean and per-component variance $\sigma^2 \approx \frac{nk}{hd}$ in the regime $k \ll d$ we focus on. The variance grows with the number of patterns n and the support size k: Each target z_i affects k output coordinates, so any given prediction

direction is interfered with by more patterns as ambiguity increases. The Hebbian model should be able to store all n patterns as long as this noise stays smaller than the signal. Solving for n yields the capacity scaling limit, our first root cause of the curse of ambiguity:

Root 1 $$\text{capacity} \propto \frac{hd}{k}. \tag{6}$$

The left panel of Figure 1 illustrates this scaling, and Appendix A formally proves it up to logarithmic factors under two different notions of capacity. A size-k distribution requires k times more capacity than a deterministic one, so more ambiguity directly requires more network capacity. Appendix A.6 verifies the hd and $1/k$ scalings empirically. The k scaling is tight up to logarithmic factors: Appendix A.5 establishes a matching upper bound on capacity that applies to any linear-softmax model, via a sign-pattern counting argument on the model's degrees of freedom. The linear model also recovers the proportionality to the total number of parameters hd, a now well-established universal measure of language model capacity, both empirically [38, 40] and theoretically across different architectures [41, 42]. The fact that it recovers this universal scaling suggests that its dependence on ambiguity extends to more general models as well.

3.2 Embedding dimension: ambiguous distributions require larger embeddings

We now move to another property of the architecture that matters just as much: the embedding dimension h. We show that, to represent all possible target distributions, h must grow at least linearly with the ambiguity level k. The intuition is that an h-dimensional embedding can only point in a few vocabulary directions at once, so once h falls below the ambiguity level there is no room left to single out the top-k tokens of the target distribution.

Compared to the total parameter count, the role of the embedding dimension in language model performance is less understood [43–46]. Yet, we are not the first to point out the softmax bottleneck [47, 48, 27]: the embedding dimension constrains the family of distributions a language model can represent and learn. Chang and McCallum [49] observe that ambiguous distributions are a particularly hard case. We sharpen these observations into a precise linear scaling with the ambiguity level. We build on a large literature on such questions [50–54], discussed in Appendix B.1.

The ideal backbone assumption. As highlighted in Section 2, a large language model can be split into a backbone that produces context-dependent embeddings and a linear head that must reconstruct the output distribution from these embeddings. For the embedding dimension analysis, we assume the backbone can produce close to optimal embeddings for the task. We ask how large the embedding dimension h needs to be for the linear head to exactly recover the target distribution. This idealization says nothing about how the backbone achieves such embeddings; it isolates a purely representational constraint that no learning can circumvent, and therefore provides a lower bound on the embedding dimension that a successful model must exceed.

A compressed sensing analysis. This representation question falls under the umbrella of compressed sensing, which broadly studies when sparse signals in high-dimensional spaces can be exactly or approximately recovered from few linear measurements [55–57]. The main takeaway of our analysis is that the embedding dimension is a representational root of the curse of ambiguity as ambiguous distributions require larger embeddings to be properly represented. We prove this through two theorems, formally stated and proven in Appendix B. The first shows that, in the idealized setup above, the linear-softmax model can exactly represent every family of size-k distributions as long as the embedding dimension satisfies

Root 2 $$h \geq 2k. \tag{7}$$

Its proof relates the representation problem to a classical question on convex polytopes, namely k-neighborliness, and invokes classical results from that literature (namely the Radon-type upper bound on neighborliness [58] and the cyclic-polytope construction [59]) to precisely quantify the threshold. The second theorem relaxes this idealization by allowing context-dependent perturbations of the embedding, reflecting the fact that real backbones can only produce embeddings that are close to, rather than exactly, optimal. In this robust regime, random-weight constructions from the compressed-sensing literature still yield an accurate representation up to a small increase of the KL-divergence, at the cost of an extra $\log(d/k)$ factor on h. The embedding dimension thus remains proportional to k, only with a larger overall constant, as illustrated in the right panel of Figure 1.

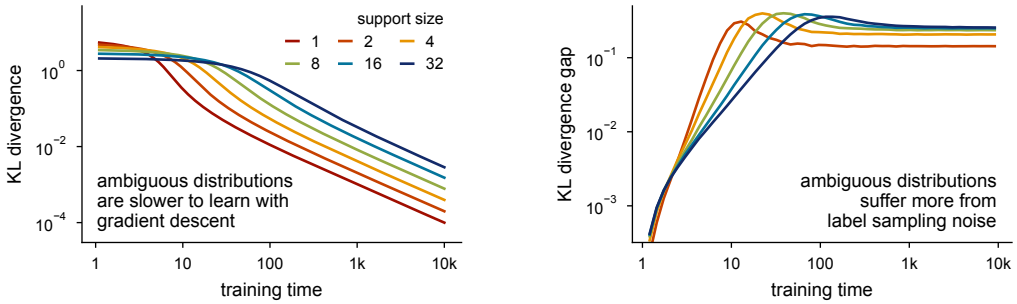

Figure 2: **Learning roots of the curse of ambiguity: ambiguous distributions require more optimization steps to be learned and amplify the noise coming from next-token sampling.** Simulation of the dynamics theoretically analyzed in Section 4. Increasing the support size k slows down initial gradient descent on the expected loss and yields larger losses in the long run (left), and widens the gap between sampled-batch trajectories and the expected-loss dynamics (right). The insights from this simplified one-context toy model extend to a linear head trained on many contexts of mixed ambiguity, as shown in Appendices C.2 and D.6.

4 Learning-based roots of the curse of ambiguity

Section 3 characterized which distributions models efficiently represent and how many they can store, but said nothing about how fast they learn them. We show that there is an additional learning-based curse of ambiguity: high ambiguity distributions take longer to optimize and are inherently noisier. Appendix E complements our learning-centric perspective with classical sampling results.

4.1 Optimization: ambiguous outputs take longer to learn

We study gradient flow on the expected cross-entropy loss of the linear model of Equation 2, that is, on the average KL divergence between the ground-truth conditional distributions and the model's outputs, up to a constant entropy term. Gradient flow has a long tradition of being used as a proxy for actual training dynamics [60, 61], outside of a few documented edge cases [62, 63]. Our analysis shows that ambiguity slows down initial learning, since each output token is seen less frequently and thus provides a weaker signal, and yields higher losses late in training.

Reduction to a scalar dynamics. The linear head must solve a logistic regression problem to map contexts to corresponding output probabilities. Far below the capacity threshold of Section 3.1, these contexts decouple and we can study each in isolation as gradient flow on its logits. For a single context with target distribution p uniform on a support of size k, all on-support logits remain equal at all times (and likewise for off-support logits), so the dynamics collapse onto a single scalar, $\Delta\ell$, the gap between any on-support and any off-support logit. As shown in Appendix C, $\Delta\ell$ obeys the implicit equation

$$\frac{k^2}{d}\left(\exp(\Delta\ell) - 1\right) + \frac{k(d-k)}{d}\Delta\ell = t, \tag{8}$$

with initial condition $\Delta\ell(0) = 0$, which relates the logit gap, training time t and ambiguity k.

Two phases of learning. Equation 8 reveals two distinct phases of learning. Early on, $\Delta\ell$ is close to 0 and grows linearly at rate $1/k$, so the KL divergence decreases as

Root 3.a
$$D_{\mathrm{KL}}(t) \approx \log\left(\frac{d}{k}\right) - \frac{t}{k} \quad \text{for small } t, \tag{9}$$

that is, ambiguity slows down initial learning. Late in training, the exponential term in Equation 8 dominates as $\Delta\ell$ grows to infinity and

Root 3.b
$$D_{\mathrm{KL}}(t) \approx \frac{k(d-k)}{td} \quad \text{for large } t. \tag{10}$$

The convergence rate $1/t$ no longer depends on k, but the prefactor scales (approximately linearly in k for $k \ll d$) with the support size: at any fixed training time, more ambiguous distributions sit at a higher loss. Figure 2 confirms that these two phases together provide a comprehensive description

of the true gradient flow trajectory. Despite its simplicity, this single-context toy model accurately captures the learning dynamics of a linear head trained on many contexts of mixed ambiguity below the capacity threshold, with only a slow-down from cross-context competition as the number of contexts grows. Above the threshold, the capacity perspective becomes the relevant one: contexts that require less capacity are better modeled (Appendix C.2).

4.2 Stochastic optimization: ambiguous outputs have noisier gradients

Our analysis so far has assumed access to the ground-truth conditional distributions p_i. In practice, large language models only observe a single next-token sample $y_i \sim p_i$ for each context i, and the cross-entropy loss is computed against this one-hot target rather than against p_i itself. We focus on this output sampling noise, conditional on the context. This noise incurs an additional curse of ambiguity: a deterministic target produces no noise at all, whereas increasing the support size k increases output sampling noise and thus impairs learning. The analysis below quantifies this effect.

Integrating output sampling noise in the gradient flow dynamics. In our analysis in Section 4.1, we optimized the expected cross-entropy loss, averaged over all possible outputs from the ground-truth distributions, which is unobserved in practice. In this section, we lift this limitation by introducing label sampling noise into the dynamics. To this end, we leverage a well-established line of work [64, 65] which shows that, for a small enough learning rate, the stochastic gradient descent iterates are closely tracked by a stochastic differential equation that characterizes how the trajectory deviates from gradient flow. This framework lets us cleanly isolate how label sampling noise perturbs the deterministic dynamics of Section 4.1. Writing the iterates as $\ell_t = \ell_t^\star + \delta_t$ where $\ell_t^\star$ is the gradient-flow trajectory, the deviation δ_t obeys an Ornstein–Uhlenbeck process driven by the per-step gradient noise covariance and damped by the local Hessian of the loss.

Noise introduces within-support variance. Applied to the dynamics of Section 4.1, the stochastic differential equation collapses onto a single scalar degree of freedom: the within-on-support variance of the logits,

$$V := \frac{1}{k}\sum_{j\in S}(\ell_j - \bar{\ell}_S)^2, \text{ with } \bar{\ell}_S := \frac{1}{k}\sum_{j\in S}\ell_j. \tag{11}$$

This variance is zero under gradient flow and is the new degree of freedom unlocked by sampling. At each step, only one on-support coordinate is incremented, scattering the on-support logits around their mean. As derived in Appendix D, the expected variance obeys the differential equation

$$\frac{\mathrm{d}\,\mathbb{E}V}{\mathrm{d}t} = -\frac{2\,\hat{p}_S^\star(t)}{k}\,\mathbb{E}V + \frac{k-1}{k^2}, \tag{12}$$

where $\hat{p}_S^\star(t)$ is the on-support mass along the gradient-flow trajectory. Noise is injected at rate $(k-1)/k^2$ and damped at rate $2\hat{p}_S^\star(t)/k$ by the curvature of the loss.

Convergence rate vs. noise floor. Equation 12 cleanly separates the two effects of sampling. The mean trajectory of the logits is untouched at leading order in the learning rate: label sampling noise does not slow down convergence, it only scatters the logits around the gradient-flow trajectory.[2] However, sampling changes the asymptotic value of the loss. A second-order KL expansion around the deterministic trajectory, evaluated at the stationary value of Equation 12, yields the noise floor

Root 4 $$\lim_{t\to\infty}\left(\mathbb{E}[D_{\mathrm{KL}}^{\mathrm{SGD}}(t)] - D_{\mathrm{KL}}^{\mathrm{GF}}(t)\right) = \frac{k-1}{4k} \tag{13}$$

which is zero for deterministic targets and asymptotes to $1/4$ as k grows to infinity. This is another manifestation of the curse of ambiguity, compounding with the slowdown of Section 4.1: early in training the $1/k$ rate dominates; as training progresses, the deterministic loss decays as $k(d-k)/(td)$ until it falls below the noise floor, after which stochastic gradient descent plateaus while gradient flow keeps decreasing. Figure 2 confirms this behavior in simulation. Both the onset of the plateau and its value worsen with k. Appendix D.6 confirms that this k-dependent gap also appears when training the linear head with sampled labels on mixed-ambiguity data.

[2] This is a statement of the leading-order stochastic differential equation limit. Discrete stochastic gradient descent at finite step size does exhibit rate slowdowns, a standard stochastic optimization result that we review in Appendix D.

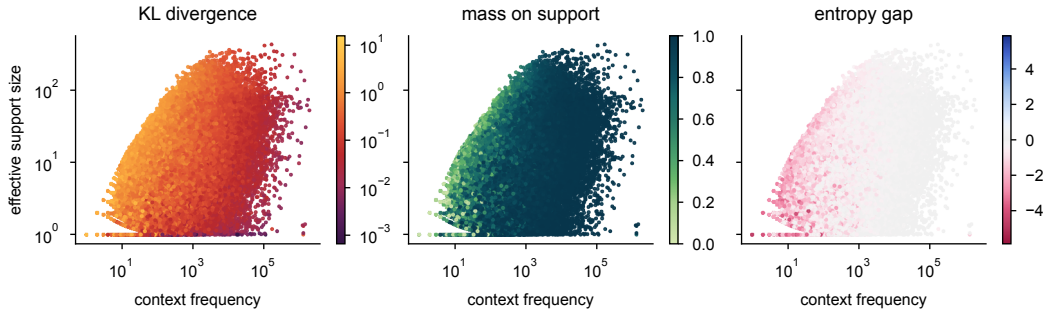

Figure 3: **Transformers trained autoregressively on a 3-gram task suffer from the curse of ambiguity.** The data follows the empirical 3-gram distributions from the TinyStories dataset, at the word level. Each dot is a different context (a pair of words), characterized by how many times it appears in the dataset (context frequency) and the effective support size of the true next-token distribution. For each context, we report the average KL divergence between the model's prediction and the true distribution (left), the average probability mass assigned to the support (middle), and the average entropy gap between the true and the predicted distribution (right). The first two panels confirm that the model suffers from the curse of ambiguity: performance degrades as effective support size increases at fixed context frequency. The model's failure mode systematically consists of leaking probability mass outside the support.

5 The curse of ambiguity in large language models

Having dissected the roots of the curse of ambiguity in toy settings, we now establish that large language models suffer from it. We first study a synthetic n-gram task in which each component of our theory can be ablated in isolation, then turn to pretrained transformers on natural data.

5.1 Synthetic n-gram data

We start with a synthetic language modeling task whose ground-truth conditional distributions are known exactly, letting us measure precisely how predictions deviate from them. To that end, we construct a word-level 3-gram dataset derived from the TinyStories corpus [66]. Training sequences are sampled autoregressively from the empirical 3-gram distribution, and by default, we train a 4-layer transformer with 4M parameters on 100M tokens; see Appendix F for details.

Small Transformer models suffer from the curse of ambiguity. In this synthetic setting, we know exactly which part of the input is relevant for next-token prediction, namely the previous two tokens, and can therefore characterize the complexity of each context precisely. Ignoring correlations between contexts in the n-gram table, we summarize this complexity by the number of times the context appears in the training data, which we refer to as its context frequency. We then analyze how well the model predicts the ground-truth distribution as contexts become more ambiguous, that is, as their effective support size (exponential of the entropy, cf. Section 2) grows, at fixed context frequency. Figure 3 summarizes this per-context evaluation. These results confirm that transformer models suffer from the curse of ambiguity, as more ambiguous contexts are harder to predict (left panel). The middle and right panels further show that the model fails by systematically leaking probability mass outside the support, leading the sequence-level distribution to have more entropy than the ground truth.

Ablations qualitatively match the toy-model predictions. To probe whether the predictions of our toy-model theory hold for actual transformers, we ablate the key components identified in Sections 3 and 4: model capacity, embedding dimension, training duration, and whether we minimize the sampled cross-entropy or the full KL divergence to the true conditional distribution. To make this comparison tractable, we summarize each configuration by a single iso-loss contour in the effective support size, context frequency plane; see Appendix F for how this contour is computed. Each intervention then either shifts the contour, indicating a uniform improvement across all effective support sizes, or rotates it, indicating a different effect on small versus large effective support sizes.

Figure 4 shows the result. Increasing the number of training steps produces a uniform shift, consistent with the late-phase regime of Section 4.1 where the convergence rate $1/t$ no longer depends on k.

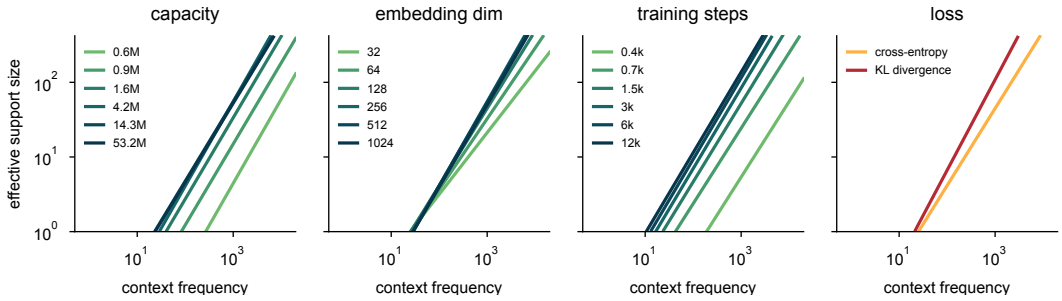

Figure 4: **Transformers suffer from the curse of ambiguity in the same way as linear models.** Each line is an iso-loss contour at $D_{\mathrm{KL}} = 0.5$, obtained from a log-log linear fit of the per-context loss of Figure 3 (left) in effective support size and context frequency, which describes the measurements well (Appendix F). Increasing capacity or the number of training steps improves performance uniformly across all effective support sizes. Increasing the embedding dimension or removing sampling noise by directly optimizing the expected loss over outputs (KL divergence) significantly improves modeling abilities at large effective support sizes, but barely affects deterministic next-token distributions, consistent with the theory developed for the linear case. Setup is the same as in Figure 3; see Appendix F for experimental details.

We see no trace of the early-dynamics regime in which large k should be learned much more slowly, presumably because even our shortest run already lies past that early phase. Capacity also shifts the line uniformly, with saturation at the largest models since training duration is held fixed. This is consistent with theory, as more parameters change the total capacity but not its optimal allocation. In contrast, increasing the embedding dimension and removing sampling noise by directly optimizing the KL divergence rather than a sampled cross-entropy both rotate this line, improving large effective support sizes while leaving deterministic contexts essentially unchanged. This is consistent with the embedding dimension analysis of Section 3.2 and the sampling analysis of Section 4.2.

5.2 Real-world models

We now show that large language models also suffer from the curse of ambiguity.

Proxies for the ground truth and the context frequency. The main obstacle to this analysis is that, on natural data, neither the ground-truth next-token distribution nor the frequency of each training context is observable: past a few tokens, every context appears at most a handful of times. We therefore replace each by a proxy. We use a large language model, OLMo-2 13B [67], as a stand-in for the ground-truth next-token distribution and evaluate the smaller OLMo-2 1B against it; both checkpoints are taken before any post-training. From the predictions of the larger model we read off the effective support size of each context by taking the exponential of the entropy of the predicted probability distribution. To estimate context frequency, we exploit the fact that similar contexts produce similar embeddings. We extract the last-layer hidden state of OLMo-2 13B at every position of $100{,}000$ sequences of length 512 sampled from the Dolma v1.7 corpus [68], and count how many other embeddings are close by; see Appendix G.1 for details. This yields an estimate of the number of semantically similar contexts seen during training, which we call the *semantic context frequency*. Both proxies are biased, since OLMo-2 13B is itself far from the true distribution of natural text and embedding distance is a coarse measure of context similarity. We nevertheless show in Appendix G.1 that, on the 3-gram task of the previous section where the ground truth is known, the semantic context frequency correlates with the true context frequency.

Large language models suffer from the curse of ambiguity. Figure 5 reports the per-context KL divergence between OLMo-2 13B and OLMo-2 1B in the estimated effective support size, semantic context frequency plane. At fixed semantic context frequency, more ambiguous contexts are harder to model, mirroring Figure 3; Figure 23 in the Appendix quantifies this trend. One qualitative difference with the synthetic setup is worth flagging: the low-frequency region contains a long tail of outliers, contexts seen rarely yet on which the two models agree closely. Inspecting them shows that they rely on in-context copying mechanisms that generalize to unseen sequences; see Appendix G.2.

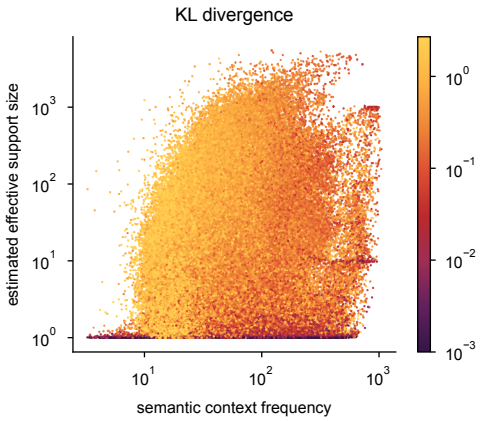

Figure 5: **Large language models suffer from the curse of ambiguity.** Same per-context evaluation as Figure 3 (left), except that we now evaluate OLMo-2 1B [67] on natural data sampled from Dolma v1.7 [68]. The effective support size and semantic context frequency are estimated from the larger OLMo-2 13B model, a proxy for the ground-truth distribution. Overall, we recover the same behavior as in Figure 3: at fixed frequency, more ambiguous contexts are harder to predict. Appendix G.3 quantitatively shows that the log KL divergence and the log effective support size at fixed semantic context frequency are correlated. See Section 5.2 and Appendix G.2 for a discussion of what the outliers in the bottom left of the plot correspond to, and Appendix G.1 for experimental details.

6 Discussion

The power of toy models. Our paper illustrates how studying simplified models and data can uncover phenomena that are hard to isolate at scale, by exposing controlled axes of complexity that point to the right setups, observables, and ablations to use in real models. Such studies are one strand of a broader movement, sometimes referred to as the physics or mechanics of learning [69–72], which has already yielded insights into learning dynamics [73–76], scaling laws [38, 41, 77–80], in-context learning [81–86], or representation geometry [87, 88]. Our work adds the discovery of the curse of ambiguity to this list[3].

How relevant is the curse of ambiguity in practice? While the curse of ambiguity is clearly visible in the large-scale models we study (Section 5.2), the picture there is less clean than in synthetic settings, in part because of a property of natural data that our analysis sets aside: structure. Indeed, our analysis implicitly assumes that a context is learned at a rate proportional to how often it appears; structural similarities between contexts, for example synonym outputs, short-circuit this assumption, allowing rare contexts to be accurately predicted via shared, compressed mechanisms. Under ambiguity, where the memorization route is slow, such structured solutions have more time to emerge, whereas for deterministic transitions memorization is fast enough that such pressure is minimal, as per the neural race argument [75]. A corollary is that neural networks are more likely to memorize sequences with deterministic, unique transitions than stochastic ones; we leave this interesting hypothesis for future work. Our analysis reveals that scaling mitigates the curse of ambiguity, but only asymptotically. Even at the largest scale a residual fraction of the distribution is left unmodeled [77, 89], and the curse of ambiguity will be particularly relevant in this regime. Fine-tuning distributions that significantly differ from the pretraining distribution should also remain particularly exposed.

Not all neural networks are language models. While we have framed this work around large language models, our results extend well beyond them. The theoretical arguments of Sections 3 and 4 apply to any neural network that produces discrete probability distributions, and the curse of ambiguity may be an even greater concern in settings where such networks operate at much smaller scales. Beyond artificial networks, the cognitive psychology literature has long documented that humans struggle when the environment admits many plausible options. In the classical paradox of choice [90, 91], increasing the number of viable alternatives degrades the speed and the quality of decision-making, and lowers satisfaction with the eventual choice. Our results offer a complementary mechanistic perspective: representing a distribution that spreads mass over many continuations is fundamentally more demanding, in capacity and in samples, than representing one that concentrates on a single answer.

[3] In fact, we first formulated the curse of ambiguity hypothesis while playing with one-hidden-layer multilayer perceptrons trained on synthetic ambiguous distributions.

Acknowledgments

We thank Yizhou Liu, Noah Cowan and members of the Linderman lab for helpful feedback. N.Z. is supported by Postdoc.Mobility grant P500-2_235376 from the Swiss National Science Foundation. S.W.L. is supported by grants from the NIH (U01NS136507, R01NS131987, R01NS113119, RF1MH133778, R01AG097491, & R01NS130789), the NSF (2440859), and the Simons and McKnight Foundations.

The experiments were performed on the Stanford Marlowe computing cluster [92]. We thank Stanford University and the Stanford Research Computing Center for providing computational resources and support that contributed to these research results.

Code availability

The code to reproduce all experiments is available at https://github.com/NicolasZucchet/Curse-of-ambiguity.

References

[1] C E Shannon. Prediction and entropy of printed English. *Bell system technical journal*, 30(1), 1951.

[2] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of machine learning research*, 3, 2003.

[3] Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černocký, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Interspeech*, 2010.

[4] Ilya Sutskever, James Martens, and Geoffrey Hinton. Generating text with recurrent neural networks. In *International conference on machine learning*, 2011.

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

[6] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. *OpenAI technical report*, 2018.

[7] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 2019.

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

[9] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *International conference on learning representations*, 2020.

[10] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International conference on learning representations*, 2021.

[11] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in neural information processing systems*, 2022.

[12] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, and others. PaLM: scaling language modeling with pathways. *Journal of machine learning research*, 2023.

[13] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science*, 362(6419):1140–1144, 2018.

[14] Andy L. Jones. Scaling scaling laws with board games. *arXiv preprint arXiv:2104.03113*, 2021.

[15] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International conference on learning representations*, 2023.

[16] OpenAI. Learning to reason with LLMs. *OpenAI blog*, 2024.

[17] Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. Large language monkeys: scaling inference compute with repeated sampling. *arXiv preprint arXiv:2407.21787*, 2024.

[18] Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling model parameters. In *International conference on learning representations*, 2025.

[19] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in neural information processing systems*, 2022.

[20] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and others. DeepSeek-R1: incentivizing reasoning capability in LLMs via reinforcement learning. *Nature*, 645(8081), 2025.

[21] Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in LLMs beyond the base model? In *Advances in neural information processing systems*, 2025.

[22] Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Harkirat Singh Behl, Xin Wang, Sébastien Bubeck, Ronen Eldan, Adam Tauman Kalai, Yin Tat Lee, and Yuanzhi Li. Textbooks are all you need. *arXiv preprint arXiv:2306.11644*, 2023.

[23] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Aniruddh Mathur, Adam Schelten, Amy Yang, Angela Fan, and others. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[24] Brian Thompson, Mehak Preet Dhaliwal, Peter Frisch, Tobias Domhan, and Marcello Federico. A shocking amount of the web is machine translated: insights from multi-way parallelism. In *Findings of the association for computational linguistics: ACL 2024*, 2024.

[25] Weixin Liang, Yaohui Zhang, Zhengxuan Wu, Haley Lepp, Wenlong Ji, Xuandong Zhao, Hancheng Cao, Sheng Liu, Siyu He, Zhi Huang, Diyi Yang, Christopher Potts, Christopher D. Manning, and James Y. Zou. Mapping the increasing use of LLMs in scientific papers. In *Conference on language modeling*, 2024.

[26] Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Nicolas Papernot, Ross Anderson, and Yarin Gal. AI models collapse when trained on recursively generated data. *Nature*, 631(8022), 2024.

[27] Nathan Godey and Yoav Artzi. Lost in backpropagation: the LM head is a gradient bottleneck. In *Conference on language modeling*, 2026.

[28] Steven T. Piantadosi, Harry Tily, and Edward Gibson. The communicative function of ambiguity in language. *Cognition*, 122(3), 2012.

[29] F. Jelinek, R.L. Mercer, L.R. Bahl, and J.K. Baker. Perplexity—a measure of the difficulty of speech recognition tasks. *Acoustical society of america journal*, 62(S1), 1977.

[30] Daniel V. Schroeder. *An introduction to thermal physics*. Addison-Wesley, 2000.

[31] John J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8), 1982.

[32] Dmitry Krotov and John J. Hopfield. Dense associative memory for pattern recognition. In *Advances in neural information processing systems*, 2016.

[33] Hubert Ramsauer, Bernhard Schäfl, Johannes Lehner, Philipp Seidl, Michael Widrich, Thomas Adler, Lukas Gruber, Markus Holzleitner, Milena Pavlović, Geir Kjetil Sandve, Victor Greiff, David Kreil, Michael Kopp, Günter Klambauer, Johannes Brandstetter, and Sepp Hochreiter. Hopfield networks is all you need. In *International conference on learning representations*, 2021.

[34] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are RNNs: fast autoregressive transformers with linear attention. In *International conference on machine learning*, 2020.

[35] Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In *Conference on empirical methods in natural language processing*, 2021.

[36] Imanol Schlag, Kazuki Irie, and Jürgen Schmidhuber. Linear transformers are secretly fast weight programmers. In *International conference on machine learning*, 2021.

[37] Alberto Bietti, Vivien Cabannes, Diane Bouchacourt, Hervé Jegou, and Léon Bottou. Birth of a transformer: a memory viewpoint. In *Advances in neural information processing systems*, 2023.

[38] Zeyuan Allen-Zhu and Yuanzhi Li. Physics of language models: part 3.3, knowledge capacity scaling laws. In *International conference on learning representations*, 2025.

[39] Nicolas Zucchet, Jörg Bornschein, Stephanie Chan, Andrew Lampinen, Razvan Pascanu, and Soham De. How do language models learn facts? Dynamics, curricula and hallucinations. In *Conference on language modeling*, 2025.

[40] John X. Morris, Chawin Sitawarin, Chuan Guo, Narine Kokhlikyan, G. Edward Suh, Alexander M. Rush, Kamalika Chaudhuri, and Saeed Mahloujifar. How much do language models memorize? In *International conference on machine learning*, 2025.

[41] Vivien Cabannes, Elvis Dohmatob, and Alberto Bietti. Scaling laws for associative memories. In *International conference on learning representations*, 2024.

[42] Eshaan Nichani, Jason D Lee, and Alberto Bietti. Understanding factual recall in transformers via associative memories. In *International conference on learning representations*, 2025.

[43] Zi Yin and Yuanyuan Shen. On the dimensionality of word embedding. In *Advances in neural information processing systems*, 2018.

[44] Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tie-Yan Liu. Representation degeneration problem in training natural language generation models. In *International conference on learning representations*, 2019.

[45] Chaofan Tao, Qian Liu, Longxu Dou, Niklas Muennighoff, Zhongwei Wan, Ping Luo, Min Lin, and Ngai Wong. Scaling laws with vocabulary: larger models deserve larger vocabularies. In *Advances in neural information processing systems*, 2024.

[46] Jackson Petty, Sjoerd van Steenkiste, Ishita Dasgupta, Fei Sha, Dan Garrette, and Tal Linzen. The impact of depth on compositional generalization in transformer language models. In *Conference of the north american chapter of the association for computational linguistics*, 2024.

[47] Zhilin Yang, Zihang Dai, Ruslan Salakhutdinov, and William W. Cohen. Breaking the softmax bottleneck: a high-rank RNN language model. In *International conference on learning representations*, 2018.

[48] Nathan Godey, Éric de la Clergerie, and Benoît Sagot. Why do small language models underperform? Studying language model saturation via the softmax bottleneck. In *Conference on language modeling*, 2024.

[49] Haw-Shiuan Chang and Andrew McCallum. Softmax bottleneck makes language models unable to represent multi-mode word distributions. In *Annual meeting of the association for computational linguistics*, 2022.

[50] Noga Alon, Peter Frankl, and Vojtěch Rödl. Geometrical realization of set systems and probabilistic communication complexity. In *Annual symposium on foundations of computer science*, pages 277–280, 1985.

[51] Noga Alon, Shay Moran, and Amir Yehudayoff. Sign rank versus VC dimension. In *Conference on learning theory*, pages 47–80, 2016.

[52] Andreas Grivas, Nikolay Bogoychev, and Adam Lopez. Low-rank softmax can have unargmaxable classes in theory but rarely in practice. In *Annual meeting of the association for computational linguistics*, pages 6738–6758, 2022.

[53] Orion Weller, Michael Boratko, Iftekhar Naim, and Jinhyuk Lee. On the theoretical limitations of embedding-based retrieval. *arXiv preprint arXiv:2508.21038*, 2025.

[54] Kiril Bangachev, Guy Bresler, Jonathan Kogan, and Yury Polyanskiy. Is dimensionality a barrier for retrieval models? *arXiv preprint arXiv:2605.23556*, 2026.

[55] E.J. Candes, J. Romberg, and T. Tao. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE transactions on information theory*, 52(2), 2006.

[56] D.L. Donoho. Compressed sensing. *IEEE transactions on information theory*, 52(4), 2006.

[57] Simon Foucart and Holger Rauhut. *A Mathematical Introduction to Compressive Sensing*. Applied and Numerical Harmonic Analysis. Birkhäuser, New York, 2013.

[58] Branko Grünbaum. *Convex Polytopes*, volume 221 of *Graduate Texts in Mathematics*. Springer, 2nd edition, 2003. Second edition prepared by V. Kaibel, V. Klee and G. M. Ziegler.

[59] David Gale. Neighborly and cyclic polytopes. In *Proceedings of Symposia in Pure Mathematics*, volume 7. American Mathematical Society, 1963.

[60] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: convergence and generalization in neural networks. In *Advances in neural information processing systems*, 2018.

[61] Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. *Journal of machine learning research*, 19(70), 2018.

[62] Jeremy Cohen, Simran Kaur, Yuanzhi Li, J. Zico Kolter, and Ameet Talwalkar. Gradient descent on neural networks typically occurs at the edge of stability. In *International conference on learning representations*, 2021.

[63] Jeremy M. Cohen, Behrooz Ghorbani, Shankar Krishnan, Naman Agarwal, Sourabh Medapati, Michal Badura, Daniel Suo, David Cardoze, Zachary Nado, George E. Dahl, and Justin Gilmer. Adaptive gradient methods at the edge of stability. *arXiv preprint arXiv:2207.14484*, 2024.

[64] Stephan Mandt, Matthew D. Hoffman, and David M. Blei. Stochastic gradient descent as approximate bayesian inference. *Journal of machine learning research*, 18, 2017.

[65] Qianxiao Li, Cheng Tai, and Weinan E. Stochastic modified equations and adaptive stochastic gradient algorithms. In *International conference on machine learning*, 2017.

[66] Ronen Eldan and Yuanzhi Li. TinyStories: how small can language models be and still speak coherent english? *arXiv preprint arXiv:2305.07759*, 2023.

[67] Team OLMo. 2 OLMo 2 Furious. In *Conference on language modeling*, 2025.

[68] Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, Valentin Hofmann, Ananya Jha, Sachin Kumar, Li Lucy, Xinxi Lyu, Nathan Lambert, Ian Magnusson, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew Peters, Abhilasha Ravichander, Kyle Richardson, Zejiang Shen, Emma Strubell, Nishant Subramani, Oyvind Tafjord, Evan Walsh, Luke Zettlemoyer, Noah Smith, Hannaneh Hajishirzi, Iz Beltagy, Dirk Groeneveld, Jesse Dodge, and Kyle Lo. Dolma: an open corpus of three trillion tokens for language model pretraining research. In *Annual meeting of the association for computational linguistics*, 2024.

[69] Yasaman Bahri, Jonathan Kadmon, Jeffrey Pennington, Sam S Schoenholz, Jascha Sohl-Dickstein, and Surya Ganguli. Statistical mechanics of deep learning. *Annual review of condensed matter physics*, 2020.

[70] Andrew Saxe, Stephanie Nelli, and Christopher Summerfield. If deep learning is the answer, what is the question? *Nature reviews neuroscience*, 22(1), 2021.

[71] Surya Ganguli. Physics of learning and neural computation. Simons Foundation. URL https://www.simonsfoundation.org/mathematics-physical-sciences/physics-of-learning-and-neural-computation/.

[72] Jamie Simon, Daniel Kunin, Alexander Atanasov, Enric Boix-Adserà, Blake Bordelon, Jeremy Cohen, Nikhil Ghosh, Florentin Guth, Arthur Jacot, Mason Kamb, Dhruva Karkada, Eric J. Michaud, Berkan Ottlik, and Joseph Turnbull. There will be a scientific theory of deep learning. *arXiv preprint arXiv:2604.21691*, 2026.

[73] Andrew M Saxe, James L McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. In *International conference on learning representations*, 2014.

[74] Andrew M. Saxe, James L. McClelland, and Surya Ganguli. A mathematical theory of semantic development in deep neural networks. *Proceedings of the national academy of sciences*, 116(23), 2019.

[75] Andrew M. Saxe, Shagun Sodhani, and Sam Lewallen. The neural race reduction: dynamics of abstraction in gated networks. In *International conference on machine learning*, 2022.

[76] Nicolas Zucchet, Francesco d'Angelo, Andrew K. Lampinen, and Stephanie C. Y. Chan. The emergence of sparse attention: impact of data distribution and benefits of repetition. In *Advances in neural information processing systems*, 2025.

[77] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[78] Eric J. Michaud, Ziming Liu, Uzay Girit, and Max Tegmark. The quantization model of neural scaling. In *Advances in neural information processing systems*, 2023.

[79] Francesco Cagnetta, Allan Raventós, Surya Ganguli, and Matthieu Wyart. Deriving neural scaling laws from the statistics of natural language. In *International conference on machine learning*, 2026.

[80] Yizhou Liu, Ziming Liu, Cengiz Pehlevan, and Jeff Gore. Universal one-third time scaling in learning peaked distributions. In *International conference on machine learning*, 2026.

[81] Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, and others. In-context learning and induction heads. *Transformer circuits thread*, 2022.

[82] Stephanie Chan, Adam Santoro, Andrew Lampinen, Jane Wang, Aaditya Singh, Pierre Richemond, James McClelland, and Felix Hill. Data distributional properties drive emergent in-context learning in transformers. In *Advances in neural information processing systems*, 2022.

[83] Shivam Garg, Dimitris Tsipras, Percy S Liang, and Gregory Valiant. What can transformers learn in-context? a case study of simple function classes. In *Advances in neural information processing systems*, volume 35, 2022.

[84] Johannes Von Oswald, Eyvind Niklasson, Ettore Randazzo, Joao Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent. In *International conference on machine learning*, 2023.

[85] Johannes von Oswald, Maximilian Schlegel, Alexander Meulemans, Seijin Kobayashi, Eyvind Niklasson, Nicolas Zucchet, Nino Scherrer, Nolan Miller, Mark Sandler, Blaise Agüera y Arcas, Max Vladymyrov, Razvan Pascanu, and João Sacramento. Uncovering mesa-optimization algorithms in Transformers. *arXiv preprint arXiv:2309.05858*, 2023.

[86] Gautam Reddy. The mechanistic basis of data dependence and abrupt learning in an in-context classification task. In *International conference on learning representations*, 2024.

[87] Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. Toy models of superposition. *arXiv preprint arXiv:2209.10652*, 2022.

[88] Yizhou Liu, Ziming Liu, and Jeff Gore. Superposition yields robust neural scaling. In *Advances in neural information processing systems*, 2025.

[89] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Erich Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training compute-optimal large language models. In *Advances in neural information processing systems*, 2022.

[90] Sheena S. Iyengar and Mark R. Lepper. When choice is demotivating: Can one desire too much of a good thing? *Journal of personality and social psychology*, 79(6), 2000.

[91] Barry Schwartz. The paradox of choice. In *Positive psychology in practice*. John Wiley & Sons, 2015.

[92] Christopher Kapfer, Kilian Stine, Balasubramanian Narasimhan, Christine Mentzel, and Emmanuel Candès. Marlowe: Stanford's GPU-based computational instrument, 2025.

[93] Roman Vershynin. *High-dimensional probability: an introduction with applications in data science*, volume 47. Cambridge university press, 2018.

[94] Hugh E. Warren. Lower bounds for approximation by nonlinear manifolds. *Transactions of the American Mathematical Society*, 133(1), 1968.

[95] Thomas M. Cover. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE Transactions on Electronic Computers*, EC-14(3), 1965.

[96] Francis Bach and Eric Moulines. Non-asymptotic analysis of stochastic approximation algorithms for machine learning. In *Advances in neural information processing systems*, 2011.

[97] Francis Bach. Adaptivity of averaged stochastic gradient descent to local strong convexity for logistic regression. *Journal of machine learning research*, 15, 2014.

[98] Nicolas Flammarion and Francis Bach. From averaging to acceleration, there is only a step-size. In *Conference on learning theory*, 2015.

[99] Paul Erdős and Alfréd Rényi. On a classical problem of probability theory. *A magyar tudományos akadémia matematikai kutató intézetének közleményei*, 6, 1961.

[100] William Feller. *An introduction to probability theory and its applications, vol. I*. Wiley, 1968.

[101] Philippe Flajolet, Danièle Gardy, and Loïs Thimonier. Birthday paradox, coupon collectors, caching algorithms and self-organizing search. *Discrete applied mathematics*, 39(3), 1992.

[102] Luc Devroye and Gábor Lugosi. *Combinatorial methods in density estimation*. Springer, 2001.

[103] Daniel Berend and Aryeh Kontorovich. A sharp estimate of the binomial mean absolute deviation with applications. *Statistics & probability letters*, 83(4), 2013.

[104] Sudeep Kamath, Alon Orlitsky, Dheeraj Pichapati, and Ananda Theertha Suresh. On learning distributions from their samples. In *Conference on learning theory*, 2015.

[105] Yanjun Han, Jiantao Jiao, and Tsachy Weissman. Minimax estimation of discrete distributions under ℓ_1 loss. *IEEE transactions on information theory*, 61(11), 2015.

[106] Dietrich Braess and Thomas Sauer. Bernstein polynomials and learning theory. *Journal of approximation theory*, 128(2), 2004.

[107] George A. Miller. Note on the bias of information estimates. *Information theory in psychology: problems and methods*, 1955.

[108] Liam Paninski. Estimation of entropy and mutual information. *Neural computation*, 15(6), 2003.

[109] Gregory Valiant and Paul Valiant. Estimating the unseen: an $n/\log n$-sample estimator for entropy and support size, shown optimal via new CLTs. In *Symposium on theory of computing*, 2011.

[110] Jiantao Jiao, Kartik Venkat, Yanjun Han, and Tsachy Weissman. Minimax estimation of functionals of discrete distributions. *IEEE transactions on information theory*, 61(5), 2015.

[111] Yihong Wu and Pengkun Yang. Minimax rates of entropy estimation on large alphabets via best polynomial approximation. *IEEE transactions on information theory*, 62(6), 2016.

[112] Gregory Valiant and Paul Valiant. Estimating the unseen: improved estimators for entropy and other properties. *Journal of the ACM*, 64(6), 2017.

[113] Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. The Faiss library. *IEEE transactions on big data*, 12(2), 2026.

Appendix

Table of contents

Notation **17**

A Theoretical results regarding capacity **18**
A.1 Definitions of top-k and ε capacity . . . 18
A.2 Setup, Hebbian model and signal-noise decomposition of the model's logits . . . 18
A.3 Top-k capacity of the Hebbian model . . . 19
A.4 Formal statement and proof for ε-capacity . . . 21
A.5 Matching upper bound via sign-pattern counting . . . 23
A.6 Validation of theory . . . 25

B Theoretical results regarding embedding dimension **28**
B.1 Related work . . . 29
B.2 A primer on convex polytopes . . . 29
B.3 Reduction to neighborliness and proof of Theorem 7 . . . 31
B.4 A robust version of the representation result . . . 32

C Theoretical results regarding optimization **38**
C.1 One-dimensional differential equation for softmax gradient flow . . . 38
C.2 Validation of theory . . . 39

D Theoretical results regarding sampling noise **40**
D.1 The within-on-support variance controls the excess loss . . . 41
D.2 Continuous-time limit of the stochastic update . . . 42
D.3 Fluctuations live on the within-on-support contrast subspace . . . 42
D.4 Dynamics of the within-support variance and noise floor . . . 42
D.5 Remark: convergence rate is untouched at leading order . . . 43
D.6 Validation of theory . . . 43

E Connection to classical sampling results **44**
E.1 The coupon collector problem . . . 44
E.2 Maximum likelihood estimation of a categorical distribution . . . 45

F Synthetic n-gram task **46**

G Large language models experiments **53**
G.1 Experiment details . . . 53
G.2 In-context predictability explains the low-loss low-context-frequency band . . . 55
G.3 Quantifying the curse of ambiguity at fixed semantic context frequency . . . 56

Notation

Symbol	Meaning
Problem setup (Section 2)	
d	vocabulary size
k	support size of the target distribution (ambiguity level)
h	embedding dimension
n	number of contexts
$S_i \subseteq [d]$	support of the target distribution for context i
$p_i \in \Delta^{d-1}$	ground-truth distribution for context i (uniform on S_i)
$\hat{p}_i = \mathrm{softmax}(We_i) \in \Delta^{d-1}$	model's predicted distribution for context i
$e_i \in \mathbb{R}^h$	embedding of context i
$f_\theta : [d]^{t-1} \to \mathbb{R}^h$	backbone mapping a context to its embedding
$W \in \mathbb{R}^{d \times h}$	weight matrix of the linear head
$\ell_i = We_i \in \mathbb{R}^d$	logits for context i
Capacity (Appendix A)	
β	scalar factor on the Hebbian weights
$\rho = k/d$	support density
$\xi_j = e_j^\top e_i$	pairwise overlap between random embeddings
$\Xi_i = \sum_{j \neq i} z_j \, e_j^\top e_i$	cross-context noise in the logits
$\sigma^2 = (n-1)\rho(1-\rho)/h$	per-component variance of Ξ_i
$n_{\text{top-k}}$	top-k capacity
n_ε	ε capacity
Embedding dimension (Appendix B)	
τ	magnitude of an adversarial embedding perturbation
ε	KL slack in the robust representation result
Learning dynamics (Appendices C–D)	
$\Delta\ell \in \mathbb{R}$	gap between any on- and off-support logit (gradient-flow scalar)
$\ell_t^\star \in \mathbb{R}^d$	gradient-flow trajectory of the logits at time t
$\hat{p}_S^\star(t) = \sum_{j \in S} \hat{p}_j^\star(t) \in [0,1]$	on-support mass along $\ell_t^\star$
η	learning rate of stochastic gradient descent
$\delta_t = \ell_t - \ell_t^\star \in \mathbb{R}^d$	deviation of the stochastic iterates from gradient flow
$V = \frac{1}{k} \sum_{j \in S} (\ell_j - \bar{\ell}_S)^2 \in \mathbb{R}$	within-on-support variance of the logits
$H(\ell) = \mathrm{diag}(\hat{p}) - \hat{p}\hat{p}^\top \in \mathbb{R}^{d \times d}$	Hessian of the softmax cross-entropy
Empirical analysis (Section 5)	
$D_{\mathrm{KL}}(p \,\|\, \hat{p})$	Kullback–Leibler divergence
$H(p)$	Shannon entropy of p
$\exp(H(p))$	effective support size of p
n_{est}	semantic context frequency from kNN density on embeddings
λ	bandwidth of the exponential kernel used to compute n_{est}

A Theoretical results regarding capacity

A.1 Definitions of top-k and ε capacity

When $k = 1$, the output is deterministic and the notion of correct storage is clear: the model must predict the correct output. In the stochastic case, multiple criteria are possible, and we consider the following two.

Definition 1 (top-k capacity)**.** *The top-k capacity $n_{\text{top-k}}$ is the largest n such that the top-k predictions of the model over all contexts i match the support of the true conditional distributions.*

Definition 2 (ε capacity)**.** *The ε-capacity n_ε is the largest n such that the KL divergence $\mathrm{KL}(p_i \| \hat{p}_i)$ stays below ε for every context i.*

These two capacity notions are related in the sense that the top-k capacity is always at least as large as the ε-capacity for small enough ε, since accurately matching the full distribution requires in particular identifying the correct support. The ε-capacity is thus the more demanding notion: it asks not only that the support be identified, but that the k probabilities within it be nearly equal.

A.2 Setup, Hebbian model and signal-noise decomposition of the model's logits

Let us quickly recall the setup we are considering, the Hebbian model we study as well as the notations we have introduced in the main text.

We have n input contexts, each context i having a random embedding $e_i \in \mathbb{R}^h$ sampled iid and uniformly from the unit sphere $\mathbb{S}^{h-1}$. Context i is associated with a ground truth output distribution p_i that is uniform on a random support $S_i \subseteq [d]$ of size k:

$$p_{i,j} = \begin{cases} 1/k & \text{if } j \in S_i \\ 0 & \text{otherwise.} \end{cases} \tag{14}$$

We consider a Hebbian model whose predicted output distribution for context i is $\hat{p}_i = \mathrm{softmax}(We_i)$ with weight matrix

$$W = \beta \sum_{j=1}^{n} z_j \, e_j^\top \quad \text{with} \quad z_{j,\ell} = \begin{cases} 1-\rho & \text{if } \ell \in S_j \\ -\rho & \text{otherwise,} \end{cases} \tag{15}$$

where $\rho := k/d$.

The logits of the model ℓ_i are given by

$$\ell_i = We_i = \beta z_i (e_i^\top e_i) + \beta \sum_{j \neq i} z_j (e_j^\top e_i) \tag{16}$$

Since the embeddings are normalized, $e_i^\top e_i = 1$ exactly and the signal term is simply βz_i. The cross-terms $\xi_j := e_j^\top e_i$ are mean-zero and sub-Gaussian[4] with variance $1/h$. We have additionally introduced the cross-context noise

$$\Xi_i = \sum_{j \neq i} z_j e_j^\top e_i \tag{17}$$

and so the logits are

$$\ell_i = \beta\big(z_i + \Xi_i\big). \tag{18}$$

Since context embeddings and target distributions are independent, the cross-context noise is a sum of $n-1$ independent random variables, whose first and second order moments are

$$\mathbb{E}\left[\Xi_i\right] = \sum_{j \neq i} \mathbb{E}\left[z_j\right] \mathbb{E}\left[\xi_j\right] = 0 \tag{19}$$

$$\mathrm{Cov}\left[\Xi_i\right] = \frac{(n-1)\,\rho(1-\rho)\,d}{h\,(d-1)} \left(I - \frac{1}{d}\mathbf{1}\mathbf{1}^\top\right). \tag{20}$$

[4]This is a standard result for dot products of random embeddings from the unit sphere; see Vershynin [93].

The covariance calculation uses the fact that membership in S_i of two outputs j and j' are not independent (they are drawn without replacement), inducing off-diagonal terms; we omit the details. It follows that the per-component variance of the noise is

$$\sigma^2 := \mathrm{Var}[\Xi_{i,j}] = \frac{(n-1)\,\rho(1-\rho)}{h} \tag{21}$$

A.3 Top-k capacity of the Hebbian model

In this section, we demonstrate that the top-k capacity of the Hebbian construction scales as $\frac{h}{\rho(1-\rho)}$ up to some logarithmic factor, thereby providing theoretical grounding for the scaling that we have discussed in the main text.

Theorem 1 (Top-k capacity lower bound)**.** *There exists a universal constant $C > 0$ such that the following holds. For any $\delta \in (0,1)$, if $h \geq C \log^2(hd/\delta)$ and*

$$n \leq \frac{h}{C\,\rho(1-\rho)\,\log(hd/\delta)} \tag{22}$$

then with probability at least $1-\delta$ over the draw of embeddings and supports, the top-k predictions of the Hebbian model are correct for all n contexts:

$$\forall\, i \in [n] \quad \mathrm{top}_k\big(We_i\big) = S_i. \tag{23}$$

To prove this statement, we will make use of the following concentration bound that relates the maximal magnitude of the noise Ξ (over i and j) to its variance. We will prove it after Theorem 1.

Lemma 2 (Uniform noise bound)**.** *There exists a universal constant $C > 0$ such that for any $\delta \in (0,1)$, with probability at least $1-\delta$,*

$$\max_{i\in[n],j\in[d]} |\Xi_{i,j}| \leq C\sqrt{\left(\sigma^2 + \frac{\log(nd/\delta)}{h}\right)\log(nd/\delta)}. \tag{24}$$

Proof of Theorem 1. The model correctly identifies the support S_i if and only if every on-support logit exceeds every off-support logit. Substituting $z_{i,j} = 1-\rho$ for $j \in S_i$ and $z_{i,j'} = -\rho$ otherwise, this simplifies to

$$\forall\, j \in S_i,\ j' \notin S_i \quad \Xi_{i,j'} - \Xi_{i,j} < 1. \tag{25}$$

A sufficient condition is $\max_{i,j} |\Xi_{i,j}| < 1/2$, and by Lemma 2 this holds with probability at least $1-\delta$ provided

$$\left(\sigma^2 + \frac{\log(nd/\delta)}{h}\right)\log(nd/\delta) \leq \frac{1}{4C^2}. \tag{26}$$

Splitting the two summands evenly, it holds whenever

$$\sigma^2 \log(nd/\delta) \leq \frac{1}{8C^2} \ \text{ and } \ \frac{\log^2(nd/\delta)}{h} \leq \frac{1}{8C^2}. \tag{27}$$

The hypotheses of the theorem, after possibly enlarging the universal constant C, imply both inequalities. First, under Eq. 22 we have

$$n - 1 \leq n \leq h/(C\rho(1-\rho)\log(hd/\delta)). \tag{28}$$

Combined with

$$\rho(1-\rho) \geq \frac{d-1}{d^2} \geq \frac{1}{2d} \tag{29}$$

for $1 \leq k \leq d-1$ and $d \geq 2$, this gives

$$n \leq 2hd/(C\log(hd/\delta)) \leq hd \tag{30}$$

for C large enough, and therefore $\log(nd/\delta) \leq 2\log(hd/\delta)$. The first inequality of Eq. 27 thus reduces, via $\sigma^2 = (n-1)\rho(1-\rho)/h \leq n\rho(1-\rho)/h$, to Eq. 22; the second reduces to the hypothesis $h \geq C\log^2(hd/\delta)$. Both reductions only modify the universal constant by a fixed factor. □

Proof of Lemma 2. The variance of $\Xi_{i,j}$ depends on the random supports (through the coefficients $z_{\ell,j}$), so we proceed in two steps: first control the variances uniformly via Bernstein's inequality and a union bound over the supports, then apply sub-Gaussian tail bounds conditionally on the supports.

Step 1: Uniform variance control. Without loss of generality we assume $\rho \le 1/2$ (i.e., $k \le d/2$); the complementary case follows by the symmetry of $\rho(1-\rho)$. Conditional on the supports, each $\xi_\ell = e_\ell^\top e_i$ is sub-Gaussian with variance proxy $1/h$ (a standard property of dot products of independent uniform unit-sphere vectors) and the coefficients $z_{\ell,j}$ are deterministic, so $\Xi_{i,j} = \sum_{\ell \neq i} z_{\ell,j}\xi_\ell$ is sub-Gaussian with variance proxy

$$\sigma_{i,j}^2 = \frac{1}{h}\sum_{\ell \neq i} z_{\ell,j}^2. \tag{31}$$

The values $z_{\ell,j} \in \{1-\rho, -\rho\}$ give

$$\begin{aligned}\sum_{\ell \neq i} z_{\ell,j}^2 &= M_{i,j}(1-\rho)^2 + (n-1-M_{i,j})\rho^2 \\ &= M_{i,j}(1-2\rho) + (n-1)\rho^2,\end{aligned} \tag{32}$$

where $M_{i,j} := |\{\ell \neq i \mid j \in S_\ell\}| \sim \mathrm{Binomial}(n-1,\rho)$, with mean $(n-1)\rho$ and variance at most $(n-1)\rho$. Letting $L := \log(2nd/\delta)$, Bernstein's inequality gives, with probability at least $1-\delta/(2nd)$,

$$M_{i,j} \le (n-1)\rho + \sqrt{2(n-1)\rho L} + \tfrac{2}{3}L \le 2(n-1)\rho + 2L, \tag{33}$$

where the last inequality uses $\sqrt{2(n-1)\rho L} \le (n-1)\rho + L/2$ (AM-GM). On this event, since $1-2\rho \ge 0$,

$$\begin{aligned}\sigma_{i,j}^2 &\le \frac{[2(n-1)\rho + 2L](1-2\rho) + (n-1)\rho^2}{h} \\ &\le \frac{2(n-1)\rho(1-\rho) + 2L}{h} \qquad (34) \\ &= 2\sigma^2 + \frac{2L}{h}. \qquad (35)\end{aligned}$$

A union bound over the nd pairs ensures this holds simultaneously for all (i,j) with failure probability at most $\delta/2$.

Step 2: Sub-Gaussian tail bound. Fix a pair (i,j) and work on the event from Step 1, which is measurable with respect to the supports. Condition on the supports $\{S_\ell\}$ and on the single embedding e_i; note that we cannot condition on all the embeddings, as that would leave no randomness in $\Xi_{i,j}$. Given these, $\Xi_{i,j} = \sum_{\ell \neq i} z_{\ell,j}\xi_\ell$ is a sum of independent sub-Gaussian terms with deterministic coefficients and variance proxy at most $2\sigma^2 + 2L/h$, since the e_ℓ with $\ell \neq i$ are independent of e_i. The sub-Gaussian tail bound thus gives

$$\mathbb{P}\left(|\Xi_{i,j}| > t \;\middle|\; \{S_\ell\}, e_i\right) \le 2\exp\left(-\frac{ct^2}{2\sigma^2 + 2L/h}\right). \tag{36}$$

The right-hand side does not depend on e_i, so averaging over e_i removes that conditioning and the same bound holds given the supports alone. A union bound over the nd pairs then gives

$$\mathbb{P}\left(\max_{i,j}|\Xi_{i,j}| > t \;\middle|\; \{S_\ell\}\right) \le 2nd\exp\left(-\frac{ct^2}{2\sigma^2 + 2L/h}\right). \tag{37}$$

Since this right-hand side does not depend on $\{S_\ell\}$ either, the law of total probability removes the remaining conditioning, and combining with the $\delta/2$ failure probability from Step 1 gives

$$\mathbb{P}\left(\max_{i,j}|\Xi_{i,j}| > t\right) \le \frac{\delta}{2} + 2nd\exp\left(-\frac{ct^2}{2\sigma^2 + 2L/h}\right). \tag{38}$$

Choosing $t = C\sqrt{(\sigma^2 + L/h)\log(nd/\delta)}$ with C sufficiently large makes both terms at most $\delta/2$. □

Symmetry. Since $\rho(1-\rho)$ is symmetric in k and $d-k$, predicting which k outputs are in the support is equally hard as predicting which $d-k$ are out of it.

Scaling. Since $\rho(1-\rho) = k(d-k)/d^2$, Eq. 22 gives the following scalings. Throughout, $\lesssim$ hides the universal constant C, and $\log(hd/\delta)$ is the single logarithmic factor of Eq. 22, which at fixed confidence level δ is a logarithm of the number of parameters.

- $k = 1$ **(deterministic):** $\rho(1-\rho) = (d-1)/d^2 \approx 1/d$, so $n \lesssim hd/\log(hd/\delta)$ and we recover, up to the logarithmic factor, the hd scaling (or d^2 when $h = d$) of Nichani et al. [42].
- $k \ll d$ **(sparse support):** $\rho(1-\rho) \approx k/d$, so $n \lesssim hd/(k\log(hd/\delta))$. Storing a distribution with support size k costs k times more capacity than a deterministic association; *this is the capacity-based curse of ambiguity*.
- $k = d/2$ **(maximal ambiguity):** $\rho(1-\rho) = 1/4$, so capacity reaches its minimum $n \lesssim h/\log(hd/\delta)$: the logarithmic factor is the only remaining dependence on the vocabulary size.
- k **close to** d**:** by the symmetry above, $k = d-1$ brings the bound back to $n \lesssim hd/\log(hd/\delta)$, and more generally $n \lesssim hd/((d-k)\log(hd/\delta))$ for $d-k \ll d$. The limiting case $k = d$ is excluded from the theorem and is degenerate: there is a single possible support, so all distributions coincide and nothing has to be stored.

A.4 Formal statement and proof for ε-capacity

We now turn to the ε-capacity, which requires not just correct support identification but accurate approximation of the uniform distribution on the support.

When $k = 1$ the target is deterministic and correct support identification is already enough to drive the KL to zero: once the top-1 prediction is correct, sending $\beta \to \infty$ pushes the predicted probability of the supported token to 1. The interesting regime is therefore $k \geq 2$, where the challenge is no longer merely to separate the support from its complement: one must also make the logits nearly equal within the support.

Theorem 3 (ε-capacity lower bound)**.** *There exists a universal constant $C > 0$ such that the following holds. For any $\varepsilon \in (0,1)$ and $\delta \in (0,1)$, if $2 \leq k \leq d/2$, $h \geq \frac{C\log^2(d/(k\varepsilon))\log^2(hd/\delta)}{\varepsilon}$, and*

$$n \leq \frac{h\,d\,\varepsilon}{C\,k\,\log^2\!\left(\frac{d}{k\varepsilon}\right)\,\log\!\left(\frac{hd}{\delta}\right)} \tag{39}$$

then with probability at least $1-\delta$ over the draw of embeddings and supports, every context is accurately modeled:

$$\max_{i\in[n]} \mathrm{KL}(p_i \| \hat{p}_i) \leq \varepsilon, \tag{40}$$

that is, $n_\varepsilon \geq n$.

The proof of Theorem 3 relies on the following deterministic bound on the per-context KL in terms of two elementary geometric quantities of the logit vector: its on-support variation and its margin. We state it once here, in a form that will also be invoked in the embedding-dimension appendix (Appendix B.4). We prove it after the theorem.

Lemma 4 (KL control via margin and variation)**.** *Let $S \subseteq [d]$ have size k and $\ell \in \mathbb{R}^d$ be a logit vector. Denote the* on-support variation *and* margin *of ℓ at S by*

$$\nu(S,\ell) := \max_{j,j'\in S} \left|\ell_j - \ell_{j'}\right|, \tag{41}$$

$$\mu(S,\ell) := \tfrac{1}{k}\textstyle\sum_{j\in S}\ell_j \;-\; \max_{j'\notin S}\ell_{j'}. \tag{42}$$

If $\nu(S,\ell) \leq 1$, then

$$D_{\mathrm{KL}}(p_S \,\|\, \mathrm{softmax}(\ell)) \;\leq\; \mathrm{e}\,\nu(S,\ell)^2 \;+\; \frac{d-k}{k}\,e^{-\mu(S,\ell)}. \tag{43}$$

Proof of Theorem 3. We proceed in four steps:

1. Fix β and compute the per-context variation and margin.
2. Apply the noise bound.
3. Apply the KL-control lemma.
4. Group everything together and conclude.

Step 1: Fix β and compute variation and margin. Set

$$\beta = \log \frac{8(d-k)}{k\varepsilon}. \tag{44}$$

This is a legitimate inverse temperature: $k \le d/2$ gives $d-k \ge k$ and $\varepsilon < 1$, so $\beta \ge \log(8/\varepsilon) \ge \log 8 > 0$. Fix a context i and drop the i subscript for readability. With $z_j = 1-\rho$ for $j \in S = S_i$ and $z_j = -\rho$ otherwise, the logits $\ell_j = \beta(z_j + \Xi_j)$ have on-support variation and margin

$$\nu(S,\ell) = \beta \max_{j,j' \in S} |\Xi_j - \Xi_{j'}| \;\le\; 2\beta M, \tag{45}$$

$$\mu(S,\ell) = \beta + \beta\big(\tfrac{1}{k}\textstyle\sum_{j\in S}\Xi_j - \max_{j'\notin S}\Xi_{j'}\big) \;\ge\; \beta(1-2M), \tag{46}$$

where $M := \max_{i\in[n],j\in[d]} |\Xi_{i,j}|$ is the uniform noise magnitude.

Step 2: Apply the noise bound. By Lemma 2, with probability at least $1-\delta$,

$$M \le C\sqrt{\left(\sigma^2 + \frac{\log(nd/\delta)}{h}\right)\log(nd/\delta)}. \tag{47}$$

The assumption $\nu(S,\ell) \le 1$ required by Lemma 4 is implied by $2\beta M \le 1$, that is, by $M \le 1/(2\beta)$. Squaring this requirement and inserting the bound on M above, it is enough that

$$\left(\sigma^2 + \frac{\log(nd/\delta)}{h}\right)\log(nd/\delta) \le \frac{1}{C\beta^2}, \tag{48}$$

the β^{-2} coming from squaring $M \le 1/(2\beta)$ and the universal constant C absorbing the resulting numerical factor. Since $\sigma^2 \asymp n\rho(1-\rho)/h$, the σ^2 part of Eq. 48 is implied by Eq. 39 and the $\log(nd/\delta)/h$ part by the assumption on h, both after adjusting the universal constant. Under Eq. 39 we also have $n \le hd$, so $\log(nd/\delta) \le C\log(hd/\delta)$.

Step 3: Apply the KL-control lemma. On the event from Step 2, Lemma 4 combined with Step 1 gives, for every context i,

$$\mathrm{KL}(p_i\|\hat{p}_i) \;\le\; \mathrm{e}\,(2\beta M)^2 \;+\; \frac{d-k}{k}\,e^{-\beta(1-2M)} \;=\; 4\mathrm{e}\,\beta^2M^2 \;+\; \frac{d-k}{k}\,e^{-\beta}\,e^{2\beta M}. \tag{49}$$

Step 4: Conclude. By the choice of β we have $\frac{d-k}{k}e^{-\beta} = \frac{\varepsilon}{8}$, and Eq. 48 gives $2\beta M \le 1$, hence $e^{2\beta M} \le \mathrm{e}$. The second term of Eq. 49 is therefore at most $\frac{\mathrm{e}}{8}\varepsilon \le \frac{\varepsilon}{2}$, and since Eq. 49 holds simultaneously for all contexts on the event of Step 2,

$$\max_{i\in[n]} \mathrm{KL}(p_i\|\hat{p}_i) \;\le\; 4\mathrm{e}\,\beta^2M^2 \;+\; \frac{\varepsilon}{2}. \tag{50}$$

It remains to make the first term at most $\varepsilon/2$ as well. By the noise bound from Step 2, $M^2 \le C\left(\sigma^2 + \frac{\log(hd/\delta)}{h}\right)\log(hd/\delta)$, so it suffices that

$$\beta^2\sigma^2\log\left(\frac{hd}{\delta}\right) \le \frac{\varepsilon}{C} \quad\text{and}\quad \beta^2\frac{\log^2(hd/\delta)}{h} \le \frac{\varepsilon}{C} \tag{51}$$

for a large enough universal constant C. Using $\sigma^2 \asymp n\rho(1-\rho)/h$ and $\beta^2 = O(\log^2(d/(k\varepsilon)))$, the first part of Eq. 51 becomes exactly Eq. 39, and the second part is implied by the assumption on h. The two halves then add up to ε. □

Proof of Lemma 4. Write $\bar{\ell} := \frac{1}{k}\sum_{j\in S}\ell_j$ and $\phi_j := \ell_j - \bar{\ell}$ for $j \in S$. By definition, $\sum_{j\in S}\phi_j = 0$, $|\phi_j| \le \nu(S,\ell)$, and $\bar{\ell} - \max_{j'\notin S}\ell_{j'} = \mu(S,\ell) =: \Delta$. Since p_S is uniform on S,

$$D_{\mathrm{KL}}(p_S \,\|\, \mathrm{softmax}(\ell)) = -\bar{\ell} + \log\left(\sum_{j'\in[d]} e^{\ell_{j'}}\right) - \log k \tag{52}$$

$$= \log\left(\tfrac{1}{k}\sum_{j\in S} e^{\phi_j} \;+\; \tfrac{1}{k}\sum_{j'\notin S} e^{\ell_{j'}-\bar{\ell}}\right). \tag{53}$$

For the on-support sum, use the inequality $e^u \le 1 + u + \mathrm{e}\,u^2$ valid for $|u| \le 1$ together with $\sum_{j\in S}\phi_j = 0$:

$$\tfrac{1}{k}\sum_{j\in S} e^{\phi_j} \;\le\; 1 + \tfrac{\mathrm{e}}{k}\sum_{j\in S}\phi_j^2 \;\le\; 1 + \mathrm{e}\,\nu(S,\ell)^2. \tag{54}$$

For the off-support sum, bound each term by $e^{-\Delta}$:

$$\tfrac{1}{k}\sum_{j'\notin S} e^{\ell_{j'}-\bar{\ell}} \;\le\; \frac{d-k}{k}\,e^{-\Delta}. \tag{55}$$

Combining and using $\log(1+x) \le x$ yields Eq. 43. □

A.5 Matching upper bound via sign-pattern counting

Theorem 1 shows that the Hebbian construction stores at least roughly $h/\rho(1-\rho)$ patterns, with $\rho = k/d$. We now establish a matching upper bound that applies to any linear-softmax model: with high probability over the draw of embeddings and supports, no weight matrix $W \in \mathbb{R}^{d\times h}$ can store substantially more patterns than this. Together, the two bounds pin down the top-k capacity up to logarithmic factors:

$$\text{capacity} \;\asymp\; \frac{h}{\rho(1-\rho)}. \tag{56}$$

More formally, the theorem we prove in this section is the following upper bound.

Theorem 5 (Top-k capacity upper bound)**.** *There exists a universal constant $C > 0$ such that, for any $\delta \in (0,1)$ and any $k < d$, if*

$$n \ge \frac{C\,(hd\log(hd) + \log(1/\delta))}{\log\binom{d}{k}}, \tag{57}$$

then with probability at least $1-\delta$ over the embeddings $\{e_i\}$ and supports $\{S_i\}$, no weight matrix $W \in \mathbb{R}^{d\times h}$ achieves $\mathrm{top}_k(We_i) = S_i$ for all $i \in [n]$.

High-level idea. The proof of the theorem relies on a counting argument. We compare two quantities:

- The number of possible target labelings $(S_1,\dots,S_n)$, which is $\binom{d}{k}^n$ since each S_i is drawn uniformly from the size-k subsets of $[d]$.
- The number of realizable labelings, that is, the number of distinct tuples $(\mathrm{top}_k(We_1),\dots,\mathrm{top}_k(We_n))$ that can be produced as W ranges over $\mathbb{R}^{d\times h}$.

The map $W \mapsto \mathrm{top}_k(We_i)$ depends on W only through the signs of a few linear functions of W, so a classical sign-pattern bound, due to Warren [94], controls the number of realizable labelings by the number of free parameters in W. Once hd is too small for the realizable labelings to cover all the possible target labelings, with high probability the random targets fall outside the realizable set and no W can fit them.

Hyperplane arrangements and regions. Given m affine functions $f_1, \dots, f_m$ on $\mathbb{R}^P$, the corresponding hyperplane arrangement $\mathcal{A}$ is the union of their zero sets, and the connected components of the complement $\mathbb{R}^P \setminus \mathcal{A}$ are called the open regions of $\mathcal{A}$. In $\mathbb{R}^2$, for instance, three lines in general position partition the plane into seven such regions. Each region corresponds to one of the sign patterns $(\operatorname{sign} f_1(x), \dots, \operatorname{sign} f_m(x)) \in \{+, -\}^m$ that can be realized as x varies in $\mathbb{R}^P$: two points lie in the same region if and only if they yield the same sign pattern. The total number of sign patterns is at most 2^m, but for m much larger than P the number of realizable patterns is dramatically smaller because the ambient space $\mathbb{R}^P$ is too low-dimensional to interleave sign changes in too many ways. The intuition is already visible in $\mathbb{R}^1$: m points cut the line into only $m + 1$ intervals, that is, $m + 1$ realizable sign patterns out of the 2^m a priori options, because once a real number lies to the left of all m points its sign pattern is fully determined and moving rightward can only flip one sign at a time. Warren's theorem makes this precise in arbitrary dimensions.

Lemma 6 (Region count, linear case of Warren [94])**.** *Let $f_1, \dots, f_m$ be affine functions on $\mathbb{R}^P$, with $m \geq P$. The number of open regions of the corresponding hyperplane arrangement is at most $(2\mathrm{e}m/P)^P$, where e is Euler's number. In particular, the number of distinct sign patterns realizable by $(f_1(x), \dots, f_m(x))$ as x varies in $\mathbb{R}^P$ is also at most $(2\mathrm{e}m/P)^P$.*

This is the linear specialization of Warren's bound on sign patterns of polynomials; it is also a consequence of Cover [95]'s counting formula. The bound holds uniformly in the choice of the f_j: it does not require the hyperplanes to be in general position. We can now turn to the proof of our upper bound on capacity.

Proof of Theorem 5. The proof proceeds in three steps.

1. We first express the labeling $(\operatorname{top}_k(We_1), \dots, \operatorname{top}_k(We_n))$ as a sign pattern in W.
2. We then apply Warren's lemma to bound the number of distinct realizable labelings.
3. Finally, we compare the number of realizable labelings to the number of possible random target labelings.

Step 1: Realized labelings are determined by sign patterns in W. Writing $w_j \in \mathbb{R}^h$ for the j-th row of W, the j-th output logit for context i is $\ell_{ij} = w_j^\top e_i$. The set $\operatorname{top}_k(We_i) \subseteq [d]$ is the set of j for which ℓ_{ij} is among the k largest entries of $(\ell_{i1}, \dots, \ell_{id})$, and is in particular determined by the comparisons $\ell_{ij} > \ell_{ij'}$ across the $\binom{d}{2}$ unordered pairs $\{j, j'\}$. Note that this fixes the full sort of the logits, of which the top-k set is a coarsening: distinct sign patterns of these comparisons may yield the same top-k set, but this overcount can only inflate the count of realizable labelings, so an upper bound on sign patterns gives an upper bound on labelings. Each such comparison is the sign of the affine function

$$f_{i,j,j'}(W) := \ell_{ij} - \ell_{ij'} = (w_j - w_{j'})^\top e_i, \qquad i \in [n],\ 1 \leq j < j' \leq d, \tag{58}$$

which is linear in W, viewed as a vector in $\mathbb{R}^{hd}$, with the embedding e_i playing the role of fixed coefficient. The realized labeling

$$\mathcal{L}(W) := (\operatorname{top}_k(We_i))_{i \in [n]} \tag{59}$$

is therefore a function of W only through the sign pattern of the $m := \binom{d}{2} n$ affine functions $\{f_{i,j,j'}\}_{i,j<j'}$ on $\mathbb{R}^{hd}$, in the sense that two weight matrices W, W' that yield the same sign pattern also yield the same labeling. Ties, where some $f_{i,j,j'}(W) = 0$ and the sign pattern is undefined, are harmless: storing a given labeling requires the strict inequalities $\min_{j \in S_i} \ell_{ij} > \max_{j' \notin S_i} \ell_{ij'}$, so the set of W that store it is open, and if nonempty it has positive Lebesgue measure and therefore contains a point off the measure-zero union of the hyperplanes $\{f_{i,j,j'} = 0\}$; it thus suffices to count the labelings realized by tie-free W, which is what Warren's lemma does.

Step 2: There are not too many realizable labelings. By Step 1, the number of distinct labelings realizable by some $W \in \mathbb{R}^{d \times h}$ is at most the number of sign patterns of the $m = \binom{d}{2} n$ affine functions $\{f_{i,j,j'}\}_{i,j<j'}$ on $\mathbb{R}^{hd}$. Applying Lemma 6 with $P = hd$ and $m = \binom{d}{2} n$ gives, for some universal $C_0 > 0$,

$$|\{\mathcal{L}(W) : W \in \mathbb{R}^{d \times h}\}| \leq \left(\frac{C_0 n d^2}{hd}\right)^{hd} = \left(\frac{C_0 n d}{h}\right)^{hd}, \tag{60}$$

provided $\binom{d}{2}n \geq hd$, which holds in the regime of interest as $n \geq 2h/(d-1)$. Crucially, the right-hand side does not depend on the specific embeddings $\{e_i\}$: Lemma 6 counts regions of any arrangement of m hyperplanes in $\mathbb{R}^{hd}$, regardless of the orientation of those hyperplanes, hence regardless of how the embeddings happen to be drawn.

Step 3: Random targets typically miss the realizable set. Condition on the embeddings $\{e_i\}$ and view the targets $S_1, \ldots, S_n$ as drawn independently and uniformly from $\binom{[d]}{k}$. Some weight matrix W stores all n patterns if and only if the random target tuple $(S_1, \ldots, S_n)$ lies in the realizable set $\{\mathcal{L}(W) : W \in \mathbb{R}^{d\times h}\} \subseteq \binom{[d]}{k}^n$. Since the targets are uniform on the $\binom{d}{k}^n$ possible labelings,

$$\mathbb{P}\Big[\exists W : \operatorname{top}_k(We_i) = S_i \ \forall i \,\Big|\, \{e_i\}\Big] \leq \frac{|\{\mathcal{L}(W) : W \in \mathbb{R}^{d\times h}\}|}{\binom{d}{k}^n} \tag{61}$$

$$\leq \binom{d}{k}^{-n} \left(\frac{C_0 nd}{h}\right)^{hd} \quad \text{by Eq. 60.} \tag{62}$$

The right-hand side does not depend on $\{e_i\}$, so the bound also holds unconditionally on the embeddings. Taking logarithms, that right-hand side is at most δ if and only if n satisfies the implicit condition

$$n \log \binom{d}{k} \geq hd \log\left(\frac{C_0 nd}{h}\right) + \log(1/\delta). \tag{63}$$

We now show that the explicit hypothesis Eq. 57 implies the implicit condition Eq. 63 for a large enough universal C. The only n-dependent term on the RHS of Eq. 63 is $hd \log n$, and since $\log n$ grows much more slowly than n it cannot keep up with the linear LHS $n \log \binom{d}{k}$. We make this quantitative via the tangent-line bound $\log n \leq \alpha n + \log \frac{1}{\alpha}$, valid for any $n \geq 1$ and $\alpha > 0$, which we apply with $\alpha = \frac{\log \binom{d}{k}}{2hd}$ so that the αn term consumes exactly half of the LHS:

$$hd \log n \leq \frac{n \log \binom{d}{k}}{2} + hd \log \frac{2hd}{\log \binom{d}{k}} \leq \frac{n \log \binom{d}{k}}{2} + 3hd \log(hd), \tag{64}$$

where the last step uses $\log \binom{d}{k} \geq \log 2$ for $1 \leq k < d$, so that $\log \frac{2hd}{\log \binom{d}{k}} \leq \log(3hd) \leq 3 \log(hd)$ whenever $hd \geq 2$. Plugging into the RHS of Eq. 63 and using $\log\big(\frac{C_0 d}{h}\big) \leq \log C_0 + \log(hd)$,

$$hd \log\left(\frac{C_0 nd}{h}\right) + \log(1/\delta) = hd \log\left(\frac{C_0 d}{h}\right) + hd \log n + \log(1/\delta) \tag{65}$$

$$\leq \frac{n \log \binom{d}{k}}{2} + C_1 hd \log(hd) + \log(1/\delta) \tag{66}$$

for a universal $C_1 > 0$ absorbing $\log C_0$ and the constant from Eq. 64. Rearranging, Eq. 63 holds whenever

$$n \log \binom{d}{k} \geq 2C_1 hd \log(hd) + 2 \log(1/\delta), \tag{67}$$

which is implied by the explicit hypothesis Eq. 57 for $C \geq 2 \max(C_1, 1)$. □

Behavior across k. The Stirling approximation gives $\log \binom{d}{k} \approx dH(\rho)$ with $\rho = k/d$, where $H(\rho) = -\rho \log \rho - (1-\rho)\log(1-\rho)$ is the binary entropy. Dropping the explicit constants and absorbing the $\log(hd)$ term into the $\lesssim$ notation, Eq. 57 reads $n \gtrsim h/H(\rho)$ up to logarithmic factors. This matches the lower bound $n \lesssim h/(\rho(1-\rho))$ of Theorem 1 up to logarithmic factors, since $H(\rho)$ and $\rho(1-\rho)$ agree up to a factor of $\log(1/\rho)$ in the sparse regime $\rho \ll 1$ and up to constants near $\rho = 1/2$.

A.6 Validation of theory

Theorem 1 only provides a lower bound on the capacity of the linear-softmax model: the Hebbian construction is one specific weight configuration, and a fully optimized model could in principle achieve a higher capacity and a qualitatively different scaling with k. We empirically verify that,

while the trained model achieves a higher capacity than the Hebbian construction, its capacity differs from the one of the construction by a roughly constant factor (between 7 and 15 over the range of k we test), and thus scales similarly with k; see Figure 6. Figure 7 additionally verifies that the capacity of the trained head grows proportionally to the number of parameters hd, with the $1/k$ dependence predicted by the theory.

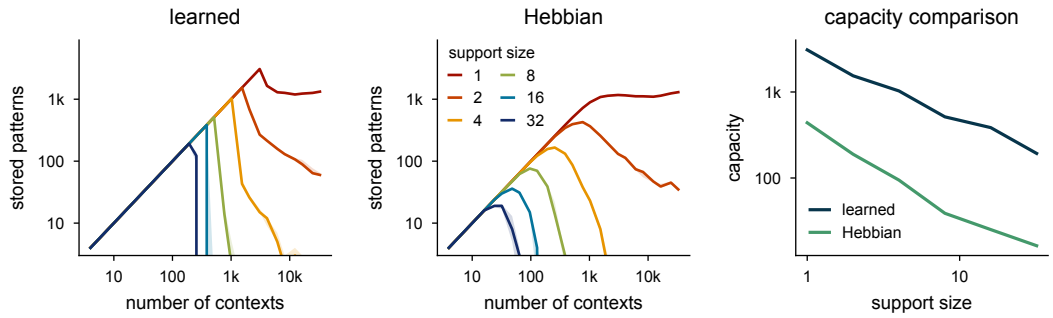

Figure 6: **The Hebbian model captures the same dependency on ambiguity as learned linear heads.** The left and middle panels show the number of contexts whose top-k predictions exactly match the ground-truth support, as a function of the dataset size n, for a linear head trained with Adam (left) and the closed-form Hebbian construction of Section 3.1 (middle). The right panel shows the corresponding capacity n^* as a function of the support size k, where n^* is the largest n for which the median top-k accuracy stays above 0.99. Both models exhibit similar dependencies on ambiguity and only approximately differ by a constant factor.

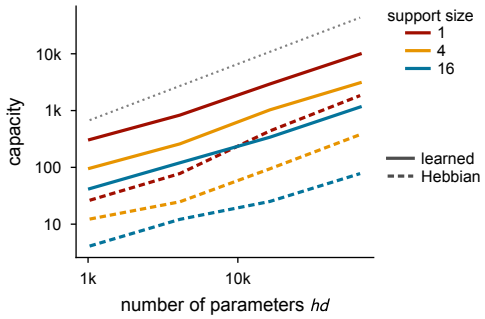

Figure 7: **The capacity of a trained linear head grows proportionally to the number of parameters hd, with the $1/k$ offset predicted by Theorem 1.** Same capacity definition and learning-rate selection as Figure 6, up to the reductions described in the experimental setup below, with the capacity n^* reported as a function of the number of parameters hd for $k \in \{1, 4, 16\}$, sweeping $h = d \in \{32, 64, 128, 256\}$. The gray dotted line indicates a slope of 1, that is, a capacity directly proportional to hd.

Experimental setup. We train a linear head with $h = d = 128$ using Adam on the exact KL divergence to uniform size-k target distributions; the swept configurations are summarized in Table 1. Embeddings are sampled iid from the unit sphere in h dimensions. The reported metric is the top-k accuracy, that is, the fraction of patterns whose top-k predictions match the support of the target data distribution. For each (k, n) we first select the learning rate that maximizes the mean top-k accuracy across seeds, breaking ties by the lowest mean KL, which picks the best-converged run among the top-k-optimal ones; we then define the capacity n^* as the largest n for which the median top-k accuracy across seeds stays above 0.99, interpolating linearly in $\log n$ between the two swept values that bracket the crossing. Reading n^* off the mean instead of the median across seeds changes it by less than 0.5% at every k, and leaves the log-log slope in k unchanged. The Hebbian construction does not require training; we evaluate it on the same metric. As k ranges from 1 to 32, the learned capacity exceeds the Hebbian one by a factor between 7 and 15, and the log-log slopes of n^* against k are -0.77 (learned) and -0.97 (Hebbian).

Figure 7 uses the same capacity definition, the same learning-rate selection rule and the same target distributions, with three reductions that make the parameter sweep tractable locally: full-batch Adam rather than minibatches of $2{,}048$ contexts (the loss is the exact distributional KL in both cases), $15{,}000$ steps, and the reduced learning-rate grid $\{3{\cdot}10^{-3},\ 10^{-2},\ 3{\cdot}10^{-2},\ 10^{-1},\ 3{\cdot}10^{-1}\}$, which brackets the optima for $k \in \{4, 16\}$. This protocol reproduces the sweep's capacities at $h = d = 128$ within 12% ($n^* = 1{,}030$ against $1{,}028$ for $k = 4$, and 340 against 385 for $k = 16$).

Table 1: Sweep configuration for the capacity validation experiment of Figure 6. $\{\cdots\}$ indicate the values that we sweep over.

Architecture	linear head, $h = d = 128$ ($hd = 16{,}384$ parameters)
Loss	exact KL divergence to the target distributions
Optimizer	Adam, batch size $2{,}048$, constant learning rate, $150{,}000$ steps
Support size k	$\{1,\ 2,\ 4,\ 8,\ 16,\ 32\}$
Number of contexts n	21 values, log-spaced between 32 and $32{,}768$
Learning rate	$\{10^{-3},\ 3{\cdot}10^{-3},\ 10^{-2},\ 3{\cdot}10^{-2},\ 10^{-1},\ 3{\cdot}10^{-1},\ 1\}$
Seeds	5

B Theoretical results regarding embedding dimension

In this appendix, we study the minimum embedding dimension h needed for the linear-softmax model to exactly represent all target distributions, for a fixed support size k. In contrast with the rest of our theoretical analysis, we first assume that the neural network backbone is powerful enough to produce any desired embedding (Appendices B.2 and B.3). This means that we can treat both the weights W and the embeddings $\{e_i\}$ as free parameters that we can pick as works best. Since the ability of the backbone to produce any embedding is related to its capacity, we are implicitly assuming that the backbone has infinite capacity. Appendix B.4 will consider the robustness of that result when we relax this assumption.

This analysis studies how much h affects what the linear-softmax model can represent. Formally, we say that it can *represent* a family of target distributions $\{p_i\}$ if there exist weights W and embeddings $\{e_i\}$, both independent of the inverse temperature $\beta > 0$, such that the predictions $\hat{p}_i = \mathrm{softmax}(\beta W e_i)$ satisfy $\hat{p}_i \to p_i$ as $\beta \to \infty$ for every i. This limit is needed because the softmax only puts exactly zero mass on off-support tokens when the logits diverge; the temperature β is the only quantity we let vary along it, so that representing a family means exhibiting a single configuration of weights and embeddings that becomes exact when sharpened.[5] We use throughout this appendix the same target distributions as in the rest of our theoretical analysis, namely uniform distributions on random size-k supports $S_i \subseteq [d]$ (Eq. 3).

Lower bound on the embedding dimension. Our first main result characterizes the minimum embedding dimension needed for such a representation to exist.

Theorem 7 (Minimum embedding dimension)**.** *Let $d \geq 2k + 2$ and $d \geq h + 1$. The linear-softmax model can represent every family of uniform distributions on size-k supports of $[d]$ if and only if $h \geq 2k$.*

From the matrix factorization argument of Yang et al. [47], we know that, in general, the embedding dimension needed to represent all possible probability distributions should be at least the vocabulary size d; hence the $d \geq h + 1$ assumption of the Theorem is not restrictive. This is what the authors called the softmax bottleneck. However, this is a general result that is agnostic to the type of distributions being modeled. Theorem 7 refines this answer for the specific type of distributions we consider, parametrized by support size k. Closer to our setting, Chang and McCallum [49] show through a geometric argument that a single hidden state cannot place high probability on several words whose embeddings surround interfering word embeddings, identifying such multi-modal distributions as a failure case of the softmax layer; our results sharpen this observation into a precise linear scaling with ambiguity. We prove Theorem 7 by turning this statistical question into a geometric one: in Section B.2 we review the classical results from the theory of convex polytopes that we later invoke, and in Section B.3 we relate the problem of the model representing these distributions to the problem of neighborliness in convex polytopes to prove Theorem 7.

Extension to more realistic noisy embeddings. An important assumption behind Theorem 7 is that each context embedding e_i can be tuned arbitrarily finely to the support it encodes. In practice, the embeddings produced by the backbone are constrained, and small deviations should not destroy the outputs. Our second main result, Theorem 8 below, shows that such robustness is recoverable at a modest cost: with random weights and an embedding dimension larger than $2k$ by a factor of order $\log(d/k)$, the representation survives bounded adversarial perturbations of the embedding at the price of a small KL slack. Appendix B.4 formalizes and proves this result; the informal statement below gives the high-level takeaway.

Theorem 8 (Robust representation via random weights, informal version of Theorem 8)**.** *With an embedding dimension $h = \Omega\big(k \log(d/k)\big)$ and randomly-drawn output weights, the linear-softmax model admits shared-per-support embeddings such that, for every size-k support S and every bounded adversarial context-dependent perturbation of the corresponding embedding, the predicted*

[5] Letting the whole configuration vary with β would not change the threshold of Theorem 7. Indeed, $\hat{p}_i \to p_i$ forces the logit-level quantities of Lemma 4 to satisfy $\nu(S_i, \ell_i) \to 0$ and $\mu(S_i, \ell_i) \to \infty$, since logit differences are log-ratios of predicted probabilities. Fixing any point of such a sequence at which $\mu > 2\nu$, every $A \subseteq [d]$ with $|A| \leq k$ is then strictly separated from its complement by some vector: directly when $|A| = k$, and otherwise by $e_S + e_{S'}$ for two size-k supports with $S \cap S' = A$, whose on-A values exceed all its other values by at least $\mu - 2\nu > 0$. Radon's theorem applied to $h + 2$ of the rows of that fixed W, whose smaller side has at most $\lfloor (h+2)/2 \rfloor \leq k$ elements when $h \leq 2k - 1$, then contradicts this separation exactly as in the proof of Proposition 9.

distribution is within ε of the target uniform distribution on S in terms of KL divergence, for a small enough perturbation radius.

B.1 Related work

The question of which output patterns a low-dimensional linear layer can produce has been studied from several angles, which we review here.

Softmax bottleneck. Beyond the general rank argument of Yang et al. [47], Godey et al. [48] relate the late-training saturation of small language models to a mismatch between their hidden dimension and the high rank of the target contextual distribution, with models of fewer than about 1000 hidden dimensions adopting degenerate representations late in pretraining. Our analysis points to the contexts that should suffer most from this mismatch: those whose next-token distribution spreads over many tokens. Chang and McCallum [49] show that a single hidden state cannot place high probability on several words when the embeddings of other words lie between them, so that multi-mode distributions can be unrepresentable regardless of model size. Theorem 7 makes this failure mode quantitative for uniform distributions on k tokens: for a suitable choice of output weights, all of them become representable exactly when $h \geq 2k$. Complementarily, Godey and Artzi [27] show that the output layer is also an optimization bottleneck, as it compresses the gradients backpropagated to the rest of the network.

Argmaxability and the geometry of the output layer. The geometric viewpoint we adopt, which treats the rows of W as points in $\mathbb{R}^h$ and asks which subsets of them can be singled out by a linear functional, was developed by Grivas et al. [52] for softmax layers. In the bias-free case, a token can only be predicted as the most likely one if its weight vector is a vertex of the convex hull of all weight vectors, the $k = 1$ instance of the face condition we use in Appendix B.3; they show that such unargmaxable tokens exist in some trained language models, but that they are rare. Our analysis asks the analogous question for sets of k tokens, which must jointly form a face of this convex hull.

Sign rank. Theorem 7 can also be read through the lens of sign rank. Since the softmax is invariant to a common shift of all logits, a size-k subset S is the top-k set of We if and only if the rows $\{w_j\}_{j\in S}$ are strictly separated from the others by an affine hyperplane of $\mathbb{R}^h$. Singling out every size-k subset therefore requires the sign matrix whose rows indicate all size-k subsets of $[d]$ to have sign rank at most $h + 1$. Alon et al. [50] show, through a moment-curve construction, that any sign matrix with at most k positive entries per row has sign rank at most $2k + 1$, and Alon et al. [51] show that this is tight for the matrix of all size-k subsets as soon as $d \geq 3k + 1$. The resulting threshold, $h = 2k$, is the one of Theorem 7. Our statement differs in that representing uniform distributions requires the on-support logits to be exactly equal rather than merely larger than the off-support ones, which is why we phrase it in terms of k-neighborliness, and our lower bound only requires $d \geq 2k + 2$.

Embedding-based retrieval. These questions have recently resurfaced in embedding-based retrieval, where a query embedding must score relevant documents above irrelevant ones. Weller et al. [53] lower bound the embedding dimension needed to realize all top-k subsets of a document collection and introduce a simple dataset on which state-of-the-art embedding models fail. In concurrent work, Bangachev et al. [54] study the margin with which such patterns can be realized: for k-sparse relevance patterns over n documents, a near-optimal margin, of order $1/\sqrt{k}$, is achievable in dimension $O(k \log n)$, and dimension $\Omega(k \log(n/k))$ is necessary to reach margins of this order. This mirrors our robust result (Theorem 8), in which random weights with $h = \Omega(k \log(d/k))$ yield a margin of order $1/\sqrt{k}$ that tolerates perturbations of the embedding, and their lower bound indicates that, at least for normalized embeddings and weights, a logarithmic overhead over the exact threshold $2k$ is the price of robustness.

B.2 A primer on convex polytopes

Our analysis of the embedding-dimension root of the curse of ambiguity in the ideal case relies on classical results from the theory of convex polytopes, and in particular on the notion of neighborliness. This subsection provides a self-contained primer on the relevant concepts and states the seminal

results that we later invoke, namely the classical upper bound on neighborliness and the cyclic polytope construction of Gale [59].

Convex polytopes and their faces. A *convex polytope* $P \subset \mathbb{R}^h$ is the convex hull of a finite set of vertices $v_1, \dots, v_d \in \mathbb{R}^h$. A subset $F \subseteq \{v_1, \dots, v_d\}$ is a *face* of P if there exist a vector $e \in \mathbb{R}^h$ and a scalar $a \in \mathbb{R}$ such that

$$\begin{aligned} e^\top v_i &= a \ \text{ for } v_i \in F, \\ e^\top v_j &< a \ \text{ for } v_j \notin F. \end{aligned} \tag{68}$$

Geometrically, F is the set of vertices that are maximally and equally aligned with e, and the affine hyperplane $\{x : e^\top x = a\}$ is a supporting hyperplane of P touching P exactly at F. The vector e is typically called the witness vector of the face.

Neighborliness. A polytope P is called *m-neighborly* if every subset of at most m of its vertices is a face. Being 2-neighborly means that every pair of vertices is connected by an edge, which is the case for the tetrahedron but not for the cube, see Figure 8. Being 3-neighborly additionally means that every triple of vertices bounds a triangular face, and so on. Neighborliness is thus a measure of how many vertices belong to the faces of the polytope.

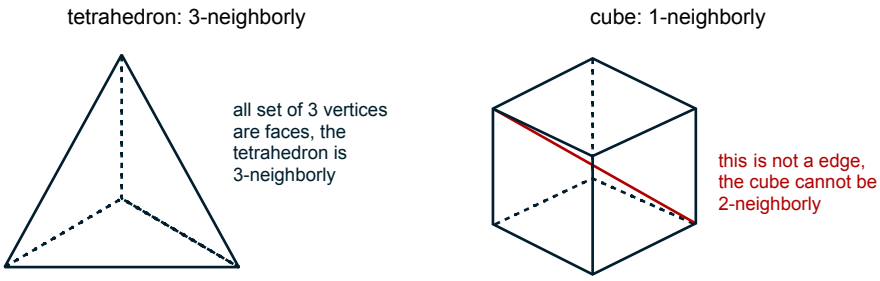

Figure 8: The tetrahedron is 3-neighborly: every subset of its four vertices is a face. The cube is only 1-neighborly: antipodal vertices are not connected by an edge, so not every pair of vertices forms a face.

How neighborly can a polytope be? A natural question to ask is how neighborly we can expect a polytope, and in particular how this depends on the number of vertices. The answer is given by the following classical threshold, a consequence of Radon's theorem.

Proposition 9 (Neighborliness threshold, see e.g. Grünbaum [58])**.** *Any convex polytope in $\mathbb{R}^h$ with strictly more than $h+1$ vertices is at most $\lfloor h/2 \rfloor$-neighborly.*

In $\mathbb{R}^3$, this means that any polytope with at least 5 vertices is at most 1-neighborly, that is, no polytope with five or more vertices can have every pair of vertices joined by an edge. The result does not apply to the tetrahedron example of Figure 8, which is 3-neighborly, as it has only 4 vertices. The $\lfloor h/2 \rfloor$ threshold comes from Radon's theorem: any set of $h+2$ points in $\mathbb{R}^h$ can be partitioned into two disjoint subsets whose convex hulls intersect. Once there are enough points, some of them are thus forced to lie inside the convex combinations of the others. The smaller side A of such a partition has at most $\lfloor (h+2)/2 \rfloor$ elements and cannot be a face: a supporting hyperplane exact on A would have to contain a point of $\mathrm{conv}(A)$ that is simultaneously a convex combination of vertices lying strictly below it.

Cyclic polytopes and the polynomial construction. This threshold is tight: for every h and $d \geq h+1$, there exist polytopes achieving $\lfloor h/2 \rfloor$-neighborliness. The canonical construction is the *cyclic polytope* $C(d,h)$, introduced by Gale [59]. It is obtained by placing d vertices along the *moment curve*

$$\phi(t) = (t, t^2, \dots, t^h), \tag{69}$$

namely $v_i = \phi(t_i)$ for $t_1 < \cdots < t_d$.

Proposition 10 (Neighborliness of the cyclic polytope, Gale [59])**.** *For every h and $d \geq h+1$, the cyclic polytope $C(d,h)$ is $\lfloor h/2 \rfloor$-neighborly.*

The power of this construction is that it reduces the problem of neighborliness to a polynomial interpretation. The dot product between a vector $e \in \mathbb{R}^h$ and the vertex $v_i = \phi(t_i)$ gives

$$e^\top v_i = e_1 t_i + e_2 t_i^2 + \cdots + e_h t_i^h =: q(t_i), \tag{70}$$

where q is a polynomial of degree at most h. The face condition of Eq. 68 then translates to q taking specific values on the different vertices. Indeed, a subset F is a face of $C(d, h)$ if and only if there exists a polynomial q of degree at most h that equals a common value a on $\{t_i : i \in F\}$ and is strictly less than a on $\{t_i : i \notin F\}$. Equivalently, denoting $r(t) := a - q(t)$, F is a face if and only if there is a polynomial r of degree at most h that vanishes on $\{t_y : y \in F\}$ and is strictly positive elsewhere. For any subset F of size $k \leq h/2$, such a polynomial can easily be constructed explicitly:

$$r(t) \;=\; \prod_{y \in F} (t - t_y)^2. \tag{71}$$

This polynomial has degree $2k \leq h$, vanishes on the points of F, and is strictly positive everywhere else, which establishes Proposition 10 in the regime $k \leq h/2$; we refer to Gale [59] for the general statement.

B.3 Reduction to neighborliness and proof of Theorem 7

The key insight behind our analysis is to relate the problem of the linear-softmax model representing the target distributions to the problem of neighborliness in convex polytopes. We first characterize the structure that W and $\{e_i\}$ must have to achieve this, then show that this structure is exactly a k-neighborliness condition on the polytope defined by the rows of W, and finally combine this reduction with the results of the previous subsection to prove Theorem 7.

Required structure of the logits. Denote by $w_j \in \mathbb{R}^h$ the j-th row of the weight matrix, so that the j-th logit for context i reads $\ell_{ij} = \beta\, w_j^\top e_i$ with β the inverse temperature that goes to infinity. Since the configuration itself does not depend on β, sharpening the softmax simply concentrates the prediction on the largest logits: for any fixed vector $v \in \mathbb{R}^d$,

$$\mathrm{softmax}(\beta v)_j \;=\; \frac{e^{\beta v_j}}{\sum_{l \in [d]} e^{\beta v_l}} \;\xrightarrow[\beta \to \infty]{}\; \begin{cases} 1/|\arg\max v| & \text{if } j \in \arg\max v, \\ 0 & \text{otherwise,} \end{cases} \tag{72}$$

that is, $\mathrm{softmax}(\beta v)$ converges to the uniform distribution on the set of maximizers of v. Applied to $v = We_i$, whose target is the uniform distribution on S_i, this shows that the model represents the family if and only if, for every context i,

$$S_i \;=\; \arg\max_{j \in [d]} \; w_j^\top e_i. \tag{73}$$

Writing $a_i := \max_j w_j^\top e_i$ for the common maximal value, Eq. 73 amounts to two conditions: $w_j^\top e_i = a_i$ for every $j \in S_i$, which makes all on-support outputs receive the same probability mass in the limit, and $w_j^\top e_i < a_i$ for every $j \notin S_i$, which makes the off-support mass vanish.

Reduction to a k-neighborliness problem. These two conditions are precisely the definition (Eq. 68) of the vectors $\{w_j\}_{j \in S_i}$ being a face of the polytope with vertices $\{w_j\}_{j \in [d]}$, with e_i acting as the witness vector and a_i as its value. That is, the output weights define the polytope, and each embedding selects one of its faces. Since we want these conditions satisfied for all supports S_i of size k, we want all subsets of size k to be faces of the polytope, that is, the polytope to be k-neighborly. Each row is then automatically a vertex of that polytope, which therefore does have d of them, as required to later invoke Proposition 9. Indeed, a row that is not a vertex is a convex combination of the others, $w_j = \sum_{l \neq j} \alpha_l w_l$ with α in the simplex. Pick $l_0 \neq j$ with $\alpha_{l_0} > 0$ and a support S_i that contains j but not l_0, which exists as $d \geq k + 1$. By Eq. 73, $w_l^\top e_i \leq a_i$ for every l, with strict inequality at l_0, so that $w_j^\top e_i = \sum_{l \neq j} \alpha_l\, w_l^\top e_i < a_i$, contradicting $w_j^\top e_i = a_i$. In particular the rows are pairwise distinct. We formalize this reduction in the following lemma.

Lemma 11 (Reduction to neighborliness)**.** *The linear-softmax model can represent every family of uniform distributions on size-k supports of $[d]$ if and only if there exists a convex polytope in $\mathbb{R}^h$ with d vertices that is k-neighborly.*

Proof. ($\Rightarrow$) Suppose the model can represent every such family. Instantiate this hypothesis at the maximal family, the one whose contexts enumerate all $\binom{d}{k}$ size-k supports of $[d]$; this is the step that forces a single polytope to serve every support, as different families may otherwise be represented by different weights. Let W and $\{e_S\}$ be parameters realizing that representation. By Eq. 73, for every size-k subset $S \subseteq [d]$ the vectors $\{w_j\}_{j\in S}$ form a face of $\mathrm{conv}(\{w_j\}_{j\in[d]})$ with witness e_S and value a_S, and each of the d rows is a vertex as observed above. The polytope $\mathrm{conv}(\{w_j\}_{j\in[d]}) \subset \mathbb{R}^h$ therefore has d vertices and every size-k subset of them as a face, that is, it is k-neighborly.

($\Leftarrow$) Conversely, let $P \subset \mathbb{R}^h$ be a k-neighborly polytope with vertices $\{w_j\}_{j\in[d]}$. For every size-k subset S, pick $(e_S, a_S) \in \mathbb{R}^h \times \mathbb{R}$ witnessing that $\{w_j\}_{j\in S}$ is a face, that is, satisfying Eq. 68. Take as weight matrix the W whose rows are the $\{w_j\}$, associate to each context of the family the embedding e_S of its support, and let $\beta \to \infty$. The logits $\beta\, w_j^\top e_S$ take the common value βa_S on S and strictly smaller values elsewhere, so the softmax output converges to the uniform distribution on S. The model therefore represents the family. □

We can now prove our main theorem by combining Lemma 11 with the two polytope results of Section B.2.

Proof of Theorem 7. ($\Leftarrow$) Suppose $h \geq 2k$. Since $d \geq h+1$, the cyclic polytope $C(d,h)$ is well defined, and Proposition 10 gives that it is $\lfloor h/2 \rfloor$-neighborly, and therefore k-neighborly since $k \leq h/2$. Lemma 11 then implies that the model represents every family of uniform distributions on size-k supports of $[d]$.

($\Rightarrow$) Suppose $h < 2k$, that is, $k > \lfloor h/2 \rfloor$. Since $d \geq 2k+2 > h+1$, Proposition 9 tells us that no polytope in $\mathbb{R}^h$ with d vertices is k-neighborly. By Lemma 11, the model cannot represent every family of uniform distributions on size-k supports of $[d]$. □

B.4 A robust version of the representation result

One of the main assumptions behind Theorem 7 is that there is no constraint on the choice of context embeddings: we implicitly assume that the backbone can produce any desired embedding, so that each context can be tuned arbitrarily finely to the support it encodes. In practice this does not hold, and small deviations of the backbone output from the ideal embedding should not destroy the ability of the model to reconstruct the target output distribution.

In this subsection, we complement Theorem 7 by considering adversarial perturbations, of bounded norm τ, of a shared embedding per support. Since the representation can no longer be exact as the logits on support will generally differ under any nonzero perturbation, we also relax the quality criterion and allow for a small KL slack ε. The embeddings are taken of unit norm and the rows of the output weight matrix of norm at most one, which makes τ a scale-free measure of the precision required of the backbone.

The construction behind Theorem 7 does not survive this relaxation: it turns every support into a face of a single polytope only by making those faces nearly degenerate, as the witnessing polynomial of Eq. 71 has a double root at each on-support vertex and therefore separates a support from its complement by a margin that shrinks very quickly as the support size grows. In the exact setting this costs nothing, since an arbitrarily large inverse temperature compensates for any positive margin, but under perturbations the inverse temperature can no longer be taken arbitrarily large: past a certain point it amplifies the perturbation enough to spread the on-support logits apart. Random output weights instead give margins of order $1/\sqrt{k}$ uniformly over all supports; we use them to build a robust representation and quantify the embedding dimension required to do so.

Proof strategy. The argument proceeds in four steps.

1. We introduce the notion of (ε, τ)-robust representation (Definition 3), which asks a single embedding per support to stay within ε of the target distribution in KL divergence, uniformly over perturbations of norm at most τ.
2. We relate the KL divergence, the quantity we ultimately care about, to two geometric properties of the polytope formed by the rows of W: the margin separating the on-support rows from the off-support ones, and the variation of the logits within the support (Eq. 76).

Those are the same quantities we used in the capacity argument of Section A and this will enable us to reuse some of the results from that section. If the margin is large enough and the on-support variation small enough, then, after adjusting the temperature β, the predicted distribution looks like the target one.

3. We exhibit weights and embeddings meeting these geometric conditions: we draw the output weights at random and define the embedding of each support as the solution of a minimax program tailored to that draw (Eq. 77). Lemma 12 shows, through duality, that the null-space property of compressed sensing implies zero on-support variation and a margin of order $1/\sqrt{k}$, uniformly over all $\binom{d}{k}$ supports, and Proposition 13 recalls that random weights satisfy this property with high probability as soon as $h \gtrsim k \log(d/k)$.
4. We tie everything back together in the proof of Theorem 8, where the temperature β is traded off against the perturbation radius τ.

Robust representation. Let us first formally introduce the notion of robust representation, which is one generalization of the notion of representation that we have been dealing with so far.

Definition 3 ((ε, τ)-robust representation)**.** *A family of target distributions $\{p_S\}$, with S ranging over size-k supports of $[d]$, admits an* (ε, τ)-robust representation *in $\mathbb{R}^h$ if there exist weights $W \in \mathbb{R}^{d\times h}$ with $\|w_j\| \le 1$ for every j, an inverse temperature $\beta > 0$, and unit-norm centers $\{e_S \in \mathbb{R}^h\}$, $\|e_S\| = 1$, such that, for every size-k subset S and every $\xi \in \mathbb{R}^h$ with $\|\xi\| \le \tau$,*

$$D_{\mathrm{KL}}\big(p_S \,\big\|\, \mathrm{softmax}\big(\beta\, W(e_S + \xi)\big)\big) \le \varepsilon. \tag{74}$$

In contrast with the limiting setup of Theorem 7, the inverse temperature β needs to be finite: allowing $\beta \to \infty$ together with a nonzero perturbation of the embedding would blow up the KL. Both normalizations, $\|w_j\| \le 1$ and $\|e_S\| = 1$, are needed for the definition to have content. Indeed, the reparametrization $(\beta, e_S, \xi) \mapsto (\beta/\lambda,\ \lambda e_S,\ \lambda\xi)$ leaves every logit unchanged, so without a constraint on the centers one could inflate them and make any fixed radius τ negligible in comparison; every exact representation in the sense of Theorem 7 would then be (ε, τ)-robust for every τ, and the notion would be vacuous. Fixing $\|e_S\| = 1$ quotients out this scale ambiguity and makes τ a relative precision: it is the fraction of the embedding norm that the backbone is allowed to get wrong. The normalization $\|w_j\| \le 1$ plays the analogous role on the weight side, and gives that a perturbation of magnitude τ in embedding space translates to a logit perturbation of magnitude at most $\beta\tau$ by Cauchy–Schwarz.

From margin and variation to KL. Our first step is to reduce the robust-representation condition to two geometric quantities on the weight polytope. For a support S of size k, a weight matrix W, and an embedding e, define the *geometric margin* and *geometric on-support variation*

$$\mu(S, e) := \min_{j\in S,\, j'\notin S} \big(w_j^\top e - w_{j'}^\top e\big), \qquad \nu(S, e) := \max_{j,j'\in S} \big|w_j^\top e - w_{j'}^\top e\big|. \tag{75}$$

An exact witness to S being a face corresponds to $\mu(S, e) > 0$ and $\nu(S, e) = 0$. Note that we overload notation with the logit-level quantities of Lemma 4: for $\ell = \beta W e$, one has $\nu(S, \ell) = \beta\, \nu(S, e)$ (exactly) and $\mu(S, \ell) \ge \beta\, \mu(S, e)$, the inequality coming from the fact that the geometric margin of Eq. 75 is defined with a minimum over the support whereas the logit-level margin of Eq. 42 is defined with the on-support mean, which is at least that minimum. Substituting into Eq. 43 gives, for any e such that $\beta\, \nu(S, e) \le 1$,

$$D_{\mathrm{KL}}(p_S \,\|\, \mathrm{softmax}(\beta W e)) \ \le\ \mathrm{e}\,\big(\beta\, \nu(S, e)\big)^2 \ +\ \frac{d-k}{k}\, e^{-\beta\, \mu(S,e)}. \tag{76}$$

Eq. 76 thus reduces the robust-representation problem to a quantitative-neighborliness question: we need a polytope in which every size-k subset of vertices is a face with a margin that is large compared to the perturbation radius.

A random-weight construction with uniformly positive margin. We draw $g_1, \dots, g_d$ iid from $\mathcal{N}(0, I_h/h)$, collect them as the rows of $G \in \mathbb{R}^{d\times h}$, and set $W := G/\max_j \|g_j\|$ so that $\|w_j\| \le 1$ as Definition 3 requires. For each size-k subset $S \subseteq [d]$, we use as center the normalized *regularized minimax witness* $e_S/\|e_S\|$, where

$$e_S \in \arg\min_{e\in\mathbb{R}^h} \Big\{ \max_{j'\notin S} \big|w_{j'}^\top e\big| \ +\ \tfrac{\kappa}{\sqrt{k}}\|e\| \Big\} \ \text{ s.t. } \ w_j^\top e = \tfrac{1}{\sqrt{k}} \text{ for every } j \in S, \tag{77}$$

where $\kappa > 0$ is a parameter of our construction, which we will set in Lemma 12 below as a function of the geometry of the draw of W. Without the penalty, this program simply maximizes the separation between the support and its complement; the penalty is what will additionally give us control on $\|e_S\|$, which we need because normalizing a center to unit norm divides its margin by $\|e_S\|$. The on-support target $\frac{1}{\sqrt{k}}$ is the natural level for a unit-norm center, since $e = \sum_{j\in S} w_j / \|\sum_{j\in S} w_j\|$ already achieves $w_j^\top e \approx \frac{1}{\sqrt{k}}$ on the support for near-orthogonal weights. By construction, all on-support logits are equal to $\frac{1}{\sqrt{k}}$, so $\nu(S, e_S) = 0$, and $\mu(S, e_S) \geq \frac{1}{\sqrt{k}} - \eta_S$ where

$$\eta_S := \min_{e\in\mathbb{R}^h} \left\{ \max_{j'\notin S} |w_{j'}^\top e| + \tfrac{\kappa}{\sqrt{k}}\|e\| \right\} \quad \text{s.t.} \quad w_j^\top e = \tfrac{1}{\sqrt{k}} \text{ for every } j \in S \tag{78}$$

is the optimal value of the program, whose dependence on the fixed κ we leave implicit. Since the two terms of the objective are nonnegative, a bound on η_S controls the off-support alignment and the norm of the witness simultaneously,

$$\max_{j'\notin S} |w_{j'}^\top e_S| \;\leq\; \eta_S, \qquad \|e_S\| \;\leq\; \frac{\sqrt{k}}{\kappa}\,\eta_S, \tag{79}$$

which is precisely why the two requirements can be handled by a single argument. A naive pointwise union bound on η_S over the $\binom{d}{k}$ supports is wasteful: the e_S corresponding to overlapping supports are tightly coupled through W. The null-space property exploits this coupling by treating the kernel of $W^\top$ as a single object.

Definition 4 (Robust null-space property)**.** *The matrix W satisfies the ℓ_2-*robust null-space property *at sparsity k with constants $\gamma \in (0, 1)$ and $\chi > 0$ if, for every $v \in \mathbb{R}^d$ and every size-k or smaller subset $S \subseteq [d]$,*

$$\|v_S\|_2 \;\leq\; \frac{\gamma}{\sqrt{k}}\,\|v_{S^c}\|_1 \;+\; \chi\,\|W^\top v\|_2. \tag{80}$$

The following deterministic lemma is the bridge from the null-space property, a statement about the kernel of $W^\top$, to our geometric quantities on the row polytope of W. It goes through the dual of the minimax-witness program of Eq. 78. Recall that this program, the *primal*, is a minimization over embeddings $e \in \mathbb{R}^h$ subject to the k on-support constraints, and that its *dual* is a maximization over one multiplier per such constraint, $\lambda \in \mathbb{R}^k$, whose value never exceeds the primal one; *strong duality*, which holds here since the constraints are affine, means that the two optimal values coincide, so that η_S can be computed on whichever side is more convenient. The dual side is the convenient one here: solving the primal directly would require handling the $\binom{d}{k}$ supports separately, whereas the dual expresses η_S in terms of vectors that almost lie in $\ker(W^\top)$, and a single property of that kernel then covers all supports at once.

Lemma 12 (Null-space property implies uniform margin and bounded witness)**.** *Assume W satisfies the ℓ_2-robust null-space property at sparsity k with constants $\gamma < 1$ and $\chi > 0$, and set $\kappa := \frac{1-\gamma}{2\chi}$ in Eq. 77. Then, for every size-k subset $S \subseteq [d]$, the regularized minimax witness e_S exists, satisfies $\|e_S\| \leq \frac{(1+\gamma)\chi}{1-\gamma}$, and its normalization $\hat{e}_S := e_S/\|e_S\|$ satisfies*

$$\nu(S, \hat{e}_S) = 0, \qquad \mu(S, \hat{e}_S) \;\geq\; \frac{c_{\gamma,\chi}}{\sqrt{k}} \quad \textit{with} \quad c_{\gamma,\chi} := \frac{(1-\gamma)^2}{2\,(1+\gamma)\,\chi}. \tag{81}$$

Proof. Fix S. The proof proceeds in three steps: we first check that the witness is well defined, then reduce the claim to a bound on the optimal value η_S, and finally obtain that bound by dualizing the program.

Step 1: the witness exists. Let $v \in \mathbb{R}^d$ be supported on S and such that $W^\top v = \sum_{j\in S} v_j w_j = 0$. Applying Eq. 80 to this v, whose two right-hand-side terms both vanish, gives $\|v_S\|_2 \leq 0$, hence $v = 0$: no nontrivial linear combination of the on-support rows vanishes, that is, $\{w_j\}_{j\in S}$ are linearly independent. The k constraints of Eq. 77 are therefore feasible. The objective is moreover continuous and diverges as $\|e\| \to \infty$, because of the penalty term, so the infimum is attained on the (closed) feasible set and e_S is well defined.

Step 2: reduction to a bound on η_S. The constraints force all on-support inner products to equal $\frac{1}{\sqrt{k}}$, so $\nu(S, e_S) = 0$ exactly, and $\mu(S, e_S) \geq \frac{1}{\sqrt{k}} - \eta_S$ as noted above. Both μ and ν are 1-homogeneous

in e, so normalizing does not change the first equality, $\nu(S, \hat{e}_S) = 0$, and turns the second bound into $\mu(S, \hat{e}_S) = \mu(S, e_S)/\|e_S\|$. Establishing Eq. 81 thus only requires a lower bound on $\mu(S, e_S)$ and an upper bound on $\|e_S\|$, and by Eq. 79 both follow from the single bound $\eta_S \le \frac{1+\gamma}{2\sqrt{k}}$ that we prove next: it gives $\mu(S, e_S) \ge \frac{1-\gamma}{2\sqrt{k}}$ and $\|e_S\| \le \frac{1+\gamma}{2\kappa} = \frac{(1+\gamma)\chi}{1-\gamma}$, whose ratio is $c_{\gamma,\chi}/\sqrt{k}$.

Step 3: bounding η_S by duality. Write the objective of Eq. 77 as $f(e) := \max_{j' \notin S} |w_{j'}^\top e| + \frac{\kappa}{\sqrt{k}}\|e\|$ and its constraints as $W_S e = \frac{1}{\sqrt{k}}\mathbf{1}$, where W_S collects the rows indexed by S. Dualizing the equality constraints with multipliers $\lambda \in \mathbb{R}^k$ gives

$$\eta_S = \max_{\lambda} \left[\frac{1}{\sqrt{k}} \mathbf{1}^\top \lambda + \min_{e} \left(f(e) - (W_S^\top \lambda)^\top e \right) \right], \tag{82}$$

where strong duality holds because the constraints are affine and the primal is feasible with finite value. As f is convex and positively homogeneous, meaning that $f(te) = tf(e)$ for every $t \ge 0$, the inner minimum is 0 if $W_S^\top \lambda$ belongs to the subdifferential of f at the origin, $\partial f(0) := \{u \in \mathbb{R}^h : u^\top e \le f(e) \text{ for every } e\}$, and $-\infty$ otherwise: the dual is thus a maximization of the linear objective $\frac{1}{\sqrt{k}}\mathbf{1}^\top\lambda$ over the multipliers whose image $W_S^\top \lambda$ falls in that set. Here $\partial f(0) = \{\sum_{j' \notin S} a_{j'} w_{j'} + z : \|a\|_1 \le 1,\ \|z\|_2 \le \frac{\kappa}{\sqrt{k}}\}$ is the sum of the convex hull of $\{\pm w_{j'}\}_{j' \notin S}$, which comes from the maximum in f, and of the ball of radius $\frac{\kappa}{\sqrt{k}}$, which comes from the penalty. An optimal λ therefore comes with such an (a, z), and gluing the two together into the single vector $v \in \mathbb{R}^d$ defined by $v_S := \lambda$ and $v_{S^c} := -a$ makes $W^\top v = z$ small,

$$\eta_S \;=\; \tfrac{1}{\sqrt{k}}\,\mathbf{1}^\top v_S, \qquad \|v_{S^c}\|_1 \le 1, \qquad \|W^\top v\|_2 \le \tfrac{\kappa}{\sqrt{k}}. \tag{83}$$

In other words, the optimal value is attained by a vector that is nearly, but not exactly, in $\ker(W^\top)$; penalizing $\|e\|$ in the primal is exactly what relaxes $v \in \ker(W^\top)$ into this near-kernel condition, and this is the reason we need the robust version of the null-space property rather than its classical exact-kernel form. Eq. 83 is now precisely of the shape that Eq. 80 controls: bounding $\mathbf{1}^\top v_S \le \sqrt{k}\|v_S\|_2$ by Cauchy–Schwarz and applying the null-space property to v yields

$$\eta_S \;\le\; \|v_S\|_2 \;\le\; \frac{\gamma}{\sqrt{k}} + \frac{\chi\kappa}{\sqrt{k}} \;=\; \frac{1+\gamma}{2\sqrt{k}} \tag{84}$$

by definition of κ, which is the bound used in Step 2 and concludes the proof. □

It remains to check that our random weights satisfy the robust null-space property with a constant γ bounded away from one and a constant χ bounded above. This is a standard result in compressed sensing, and we obtain it by chaining two textbook results applied to $G^\top \in \mathbb{R}^{h \times d}$, viewed as a measurement matrix taking h linear measurements of vectors in $\mathbb{R}^d$: subgaussian matrices satisfy the restricted isometry property as soon as the number of measurements exceeds $k \log(\mathrm{e}d/k)$ up to a constant, and a small restricted isometry constant implies the robust null-space property.

Proposition 13 (Random weights satisfy the null-space property)**.** *There exist universal constants $C > 0$, $\gamma_0 \in (0, 1)$ and $\chi_0 > 0$ such that the following holds. For any $\delta \in (0, 1)$, if*

$$h \;\ge\; C\,k\,\log(\mathrm{e}d/k) + C\log(1/\delta), \tag{85}$$

then the matrix W constructed above satisfies the ℓ_2-robust null-space property of Definition 4 at sparsity k with constants $\gamma = \gamma_0$ and $\chi = \chi_0$, with probability at least $1 - \delta$.

Proof. The matrix $G^\top$ is a subgaussian random matrix with h rows and d columns, normalized as in Foucart and Rauhut [57, Definition 9.1]. Under Eq. 85 with C large enough, Foucart and Rauhut [57, Theorem 9.2] applied at sparsity $2k$ gives that its restricted isometry constant satisfies $\delta_{2k} \le 3/8 < 4/\sqrt{41}$ with probability at least $1 - \delta/2$. On that event, Foucart and Rauhut [57, Theorem 6.13] yields the ℓ_2-robust null-space property of order k for G, with constants $\gamma_0 < 1$ and χ_1 that depend only on the numerical value $3/8$ bounding δ_{2k} and are therefore universal. Rescaling G into $W = G/\zeta$ with $\zeta := \max_j \|g_j\|$ leaves γ_0 and $\ker(W^\top) = \ker(G^\top)$ unchanged and only replaces χ_1 by $\zeta\,\chi_1$ in Eq. 80. The factor ζ is random, but Eq. 80 only becomes weaker as its constant grows, so it suffices to bound ζ on a high-probability event. Each $\|g_j\|$ concentrates around 1, and,

since $k\log(\mathrm{e}d/k) \geq \log(\mathrm{e}d)$ makes Eq. 85 imply $h \geq C\log(d/\delta)$, Gaussian norm concentration and a union bound over the d rows give $\zeta \leq 2$ with probability at least $1-\delta/2$ for C large enough. On the intersection of the two events, of probability at least $1-\delta$, the property therefore holds with the universal constants γ_0 and $\chi_0 := 2\chi_1$. □

Putting everything together. We now have all the ingredients to prove the main result of this subsection: Eq. 76 turns the KL divergence we want to bound into a requirement on the margin and on-support variation of the embeddings, Proposition 13 gives that random output weights satisfy the null-space property once $h \gtrsim k\log(d/k)$, and Lemma 12 converts that property into the uniform margin our embeddings need. The proof below combines the three, the only remaining work being the choice of the temperature β, which must be large enough for the off-support mass to be negligible yet small enough for the perturbation not to spread the on-support logits.

Theorem 8 (Robust representation via random weights)**.** *There exist universal constants $C, c > 0$ such that the following holds. For every k, d with $d \geq 2k+2$, every $\varepsilon, \delta \in (0,1)$, and every*

$$\tau \;\leq\; \frac{c\,\sqrt{\varepsilon}}{\sqrt{k}\,\log\!\big(d/(k\varepsilon)\big)}, \tag{86}$$

the family of uniform distributions on size-k supports of $[d]$ admits an (ε,τ)-robust representation in $\mathbb{R}^h$ with probability at least $1-\delta$ over the draw of the random output weights, provided

$$h \;\geq\; C\,k\,\log(ed/k) + C\log(1/\delta). \tag{87}$$

Proof. Take W and the normalized witnesses $\hat{e}_S$ of the construction above as the weights and centers of Definition 3. Under Eq. 87 with a large enough constant, Proposition 13 guarantees that W satisfies the ℓ_2-robust null-space property at sparsity k with constants $\gamma = \gamma_0$ and $\chi = \chi_0$ with probability at least $1-\delta$. On this event, Eq. 81 of Lemma 12 yields $\nu(S,\hat{e}_S) = 0$ and $\mu(S,\hat{e}_S) \geq c_0/\sqrt{k}$ uniformly over all size-k subsets S, where $c_0 := c_{\gamma_0,\chi_0}$ is universal.

Fix a size-k subset S and a perturbation ξ with $\|\xi\| \leq \tau$, and set $e := \hat{e}_S + \xi$. Since $|w_j^\top \xi| \leq \tau$ for every j by Cauchy–Schwarz, such a perturbation moves each of the two geometric quantities by at most 2τ,

$$\mu(S,e) \geq \mu(S,\hat{e}_S) - 2\tau \geq \frac{c_0}{\sqrt{k}} - 2\tau, \qquad \nu(S,e) \leq \nu(S,\hat{e}_S) + 2\tau = 2\tau. \tag{88}$$

Since $d \geq 2k+2$ and $\varepsilon < 1$, we have $\log\big(d/(k\varepsilon)\big) \geq \log 2$, so Eq. 86 together with $\sqrt{\varepsilon} \leq 1$ gives $\tau \leq c/(\sqrt{k}\log 2) \leq c_0/(4\sqrt{k})$ as soon as $c \leq c_0\log(2)/4$; we may therefore assume $\tau \leq c_0/(4\sqrt{k})$, so that $\mu(S,e) \geq c_0/(2\sqrt{k})$. Choose $\beta = \frac{2\sqrt{k}}{c_0}\,\log\big(2(d-k)/(k\varepsilon)\big)$, so that

$$\frac{d-k}{k}\,e^{-\beta\mu(S,e)} \;\leq\; \frac{d-k}{k}\,e^{-\beta c_0/(2\sqrt{k})} \;\leq\; \frac{\varepsilon}{2}. \tag{89}$$

The condition $\beta\,\nu(S,e) \leq 1$ required by Eq. 76 is implied by $\frac{4}{c_0}\sqrt{k}\,\tau\log\big(2d/(k\varepsilon)\big) \leq 1$, which holds for τ satisfying Eq. 86 with c small enough, using $\log\big(2d/(k\varepsilon)\big) \leq 2\log\big(d/(k\varepsilon)\big)$ as $d/(k\varepsilon) \geq 2$. Applying Eq. 76,

$$D_{\mathrm{KL}} \;\leq\; \mathrm{e}\,(2\beta\tau)^2 + \frac{\varepsilon}{2}. \tag{90}$$

Finally, $\mathrm{e}\,(2\beta\tau)^2 \;\leq\; \frac{16\mathrm{e}}{c_0^2}\,k\,\tau^2\log^2\!\big(2d/(k\varepsilon)\big) \;\leq\; \varepsilon/2$ by Eq. 86 for c small enough, which yields $D_{\mathrm{KL}} \leq \varepsilon$. □

We make three observations about this result.

- **Exact versus robust regime.** Theorem 7 guarantees exact representation as soon as $h \geq 2k$. At the cost of an additional $\log(d/k)$ factor on the embedding dimension, Theorem 8 shows that the representation survives bounded adversarial perturbations of the context embedding and a D_{KL} slack ε, whereas the witness behind Theorem 7 tolerates only perturbations as small as the margin it leaves, which shrinks very quickly with the support size. Random weights break this pathology, in line with the compressed-sensing intuition that random projections yield quantitatively neighborly polytopes [55, 56].

– **Curse of ambiguity in the tolerable perturbation.** The tolerable radius in Eq. 86 shrinks as $1/\sqrt{k}$ and, the centers being of unit norm, it is directly a relative precision. This reflects the fact that coordinating k on-support vertices on a face of the polytope becomes geometrically more demanding as k grows: larger ambiguity costs not only more embedding dimension, but also more precision on the backbone embedding.

– **Connection to sparse recovery.** The null-space property of Definition 4 is, in its unrobust form, the standard sufficient and necessary condition for ℓ_1-minimization to recover every k-sparse vector from linear measurements [55–57]. Lemma 12 repurposes it to control the row polytope of W instead: a small null-space overlap with any k-subset translates into a strict separation between the k on-support rows and the remaining vertices, and the robust term additionally bounds the norm of the witness that realizes this separation.

C Theoretical results regarding optimization

C.1 One-dimensional differential equation for softmax gradient flow

In the main text, we have mentioned that the gradient flow dynamics of the linear model can be reduced in some regimes (precisely far below the capacity threshold) to a one-dimensional differential equation. We here justify how we can transform gradient flow on a single logistic regression problem over d parameters to a one-dimensional differential equation. The reduction itself is exact; the asymptotic expansions that follow are all taken in the regime $k \ll d$, in which the number of admissible next tokens is small compared to the vocabulary size, as is the case in practice. In that regime we systematically use $d - k \approx d$, and keep only the leading order in k/d.

Let $\ell \in \mathbb{R}^d$ be the model's logits and initialize them to 0 and $\hat{p} = \mathrm{softmax}(\ell)$ its corresponding probability distribution. For a single context with unit-norm embedding e, gradient flow on the weights W induces exactly gradient flow on the logits $\ell = We$, so we work with the logits directly. We take the target distribution p to be the one from Equation 3, that is uniform over k out of the d classes:

$$p_y = \begin{cases} \frac{1}{k} & \text{if } y \in S \\ 0 & \text{otherwise.} \end{cases} \tag{91}$$

It is worthwhile to note that in this analysis, we do not consider any context (or equivalently we fix it) and therefore have dropped x subscripts compared to the main text.

The gradient flow dynamics on the logits of a single context on the expected cross entropy loss, that is, on the KL divergence between p and $\hat{p}$, are

$$\dot{\ell} = \nabla_\ell \mathbb{E}_y[\log(\hat{p}_y)] \tag{92}$$
$$= -\nabla_\ell D_{\mathrm{KL}}(p\|\hat{p}) \tag{93}$$
$$= p - \hat{p}. \tag{94}$$

Since all on-support logits ($y \in S$) start equal and receive the same gradient, they remain equal for all time. Likewise for the off-support logits. Additionally, the sum of the logits does not affect the probability distribution $\hat{p}$ so we can reduce the model to one variable, $\Delta\ell$, that is the difference between on- and off-support logits. It follows that

$$\dot{\Delta\ell} = (p_{\mathrm{on}} - \hat{p}_{\mathrm{on}}) - (p_{\mathrm{off}} - \hat{p}_{\mathrm{off}}) \tag{95}$$
$$= \frac{1}{k} - \frac{1 - (d-k)\hat{p}_{\mathrm{off}}}{k} + \hat{p}_{\mathrm{off}} \tag{96}$$
$$= \frac{d}{k} \frac{1}{d - k + k\exp(\Delta\ell)} \tag{97}$$

with $\Delta\ell(0) = 0$ and $\hat{p}_{\mathrm{on}}$ (resp. $\hat{p}_{\mathrm{off}}$) the probability of on- (resp. off-) support outputs.

The differential equation is separable: $\frac{k}{d}(k\exp(\Delta\ell) + d - k)\,\mathrm{d}(\Delta\ell) = \mathrm{d}t$. Integrating both sides yields

$$\frac{k^2}{d}\left(\exp(\Delta\ell) - 1\right) + \frac{k(d-k)}{d}\Delta\ell = t. \tag{98}$$

Asymptotics. Let us now look at the early and late evolution of the KL divergence. First, it can be written as a function of $\Delta\ell$:

$$D_{\mathrm{KL}}(p\|\hat{p}) = \sum_{y\in S} \frac{1}{k} \log\left(\frac{1}{k\hat{p}_{\mathrm{on}}}\right) \tag{99}$$
$$= \log\left(\frac{1}{1 - (d-k)\hat{p}_{\mathrm{off}}}\right) \tag{100}$$
$$= \log\left(\frac{k\exp(\Delta\ell) + d - k}{k\exp(\Delta\ell)}\right) \tag{101}$$
$$= \log\left(1 + \frac{d-k}{k\exp(\Delta\ell)}\right). \tag{102}$$

Around $t = 0$, $\Delta\ell$ remains close to 0, which means that up to a first order approximation

$$D_{\mathrm{KL}}(p\|\hat{p}) \approx D_{\mathrm{KL}}(p\|\hat{p})|_{t=0} + \left.\frac{\mathrm{d}D_{\mathrm{KL}}(p\|\hat{p})}{\mathrm{d}\Delta\ell}\right|_{t=0} \Delta\ell \tag{103}$$

$$= \log\left(\frac{d}{k}\right) - \frac{d-k}{d}\Delta\ell. \tag{104}$$

In this regime we can linearize the exponential in the implicit equation given by Equation 98, $\exp(\Delta\ell) - 1 \approx \Delta\ell$, whose left-hand side then collapses to $\frac{k^2}{d}\Delta\ell + \frac{k(d-k)}{d}\Delta\ell = k\Delta\ell$, so that

$$\Delta\ell \approx \frac{d}{k(d-k)}t \approx \frac{t}{k}, \tag{105}$$

where the first expression is what one obtains by dropping the exponential term altogether instead of linearizing it; the two agree at leading order in k/d, which is the accuracy we work at here. Combining Equation 105 with $\frac{d-k}{d} \approx 1$, we get that D_{KL} is approximately

$$D_{\mathrm{KL}}(p\|\hat{p}) \approx \log\left(\frac{d}{k}\right) - \frac{t}{k} \tag{106}$$

at the beginning of the dynamics, showing that learning a distribution with k different possible targets takes k times longer than the deterministic case early on in the dynamics. Now moving to late-stage dynamics, the exponential term $\exp(\Delta\ell)$ dominates as $\Delta\ell \to +\infty$, so Equation 98 gives

$$\exp(\Delta\ell) \approx \frac{td}{k^2}. \tag{107}$$

Plugging this into the KL divergence expression, we get

$$D_{\mathrm{KL}}(p\|\hat{p}) \approx \log\left(1 + \frac{k(d-k)}{td}\right) \tag{108}$$

$$\approx \frac{k(d-k)}{td} \approx \frac{k}{t}, \tag{109}$$

where the last step again uses $d - k \approx d$. This highlights that, asymptotically, the convergence rate $1/t$ is the same for all k values, but the loss increases linearly with k when $k \ll d$.

C.2 Validation of theory

The gradient-flow analysis above is derived for a single fixed target distribution and a d-dimensional logits vector that is updated directly. The actual toy setting we are interested in operates on a linear head with shared parameters across many contexts of varying ambiguity. We empirically check that the simplified theory still qualitatively captures the learning dynamics in this more realistic regime, and where it breaks down. We find that, below the capacity threshold, the per-k KL curves match the theoretical picture: ambiguity slows learning early (Eq. 98) and the late-time decay tends to a slope of -1 on a log-log scale, as predicted by the asymptotics of Eq. 10. As we add more contexts, the overall pace of learning slows, even after normalizing time by the number of contexts, presumably because gradient updates that are useful for one context interfere with others. Close to and beyond the capacity threshold, gradient flow is no longer the right lens: the network's finite capacity becomes the dominant constraint, and the less ambiguous distributions converge first because they are cheaper to store, as predicted by the capacity analysis of Appendix A.

Experimental setup. The setup mirrors the capacity validation experiment of Appendix A.6, with two differences. First, we train with plain stochastic gradient descent instead of Adam, so as to stay close to the gradient-flow regime that the theory describes. Second, each context is associated with one of six target distributions of varying support size $k \in \{1, 2, 4, 8, 16, 32\}$, with the six ambiguity levels mixed in equal proportion within every batch: a single trained model thus sees the full range of ambiguities at the same time. This lets us probe whether the per-k gradient-flow predictions still hold when distributions of different difficulty compete for the same parameters, as is expected to be the case in practice. The full sweep configuration is summarized in Table 2, where n is the number of contexts per ambiguity level so that the total dataset contains $6n$ context-distribution pairs; we report the mean of the per-k KL across 5 seeds.

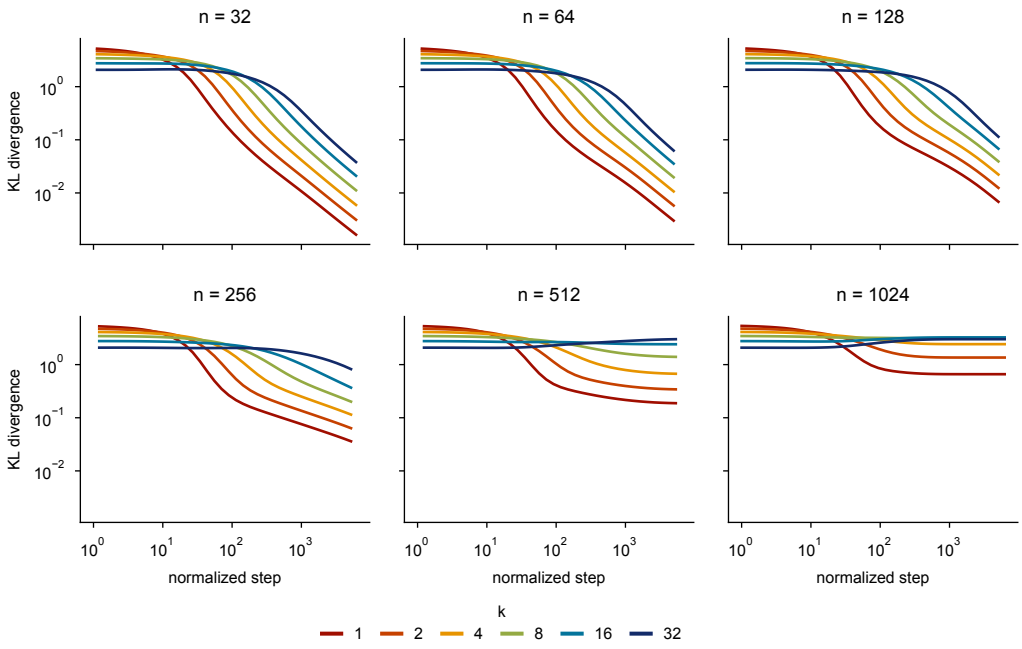

Figure 9: **Gradient-flow theory qualitatively captures learning below the capacity threshold, even with mixed ambiguity within the data.** Each panel shows the mean KL divergence on the size-k subset of contexts, against normalized step step/n, for one value of the number of contexts n. Below the capacity threshold (top row, $n \in \{32, 64, 128\}$), the curves stack in increasing order of k and decay at similar rate, mirroring the theoretical prediction that more ambiguous distributions are harder to learn but share the same asymptotic slope; see Figure 2. As we approach and cross the capacity threshold (bottom row, $n \in \{256, 512, 1024\}$), the dynamics flatten and the simplified theory no longer applies: ambiguous distributions plateau at high loss while less ambiguous ones are still learned, consistent with the capacity-based picture of Appendix A.

Table 2: Sweep configuration for the gradient-flow validation experiment of Figures 9 and 10. Each ambiguity level forms a group of n contexts, for a total of $6n$ context-distribution pairs per run.

Architecture	linear head, $h = d = 256$ ($hd = 65{,}536$ parameters)
Optimizer	stochastic gradient descent, batch size $1{,}024$, constant learning rate 1, $6.4{\cdot}10^6$ steps
Support size k	$\{1,\ 2,\ 4,\ 8,\ 16,\ 32\}$, mixed in equal proportion
Contexts per ambiguity level n	$\{32,\ 64,\ 128,\ 256,\ 512,\ 1024\}$
Seeds	5

D Theoretical results regarding sampling noise

In this appendix we analyze the effect of label sampling noise on the learning dynamics of the logistic regression model of Appendix C. The setup is identical, except that we now train with sampled labels: at each step we draw $y \sim p$ and apply the stochastic gradient descent update on the cross-entropy loss, that is

$$\ell_j \leftarrow \ell_j + \eta\,(\mathbf{1}[y = j] - \hat{p}_j)\,, \tag{110}$$

with $\eta > 0$ the learning rate. We measure time in units of the learning rate, $t = \eta \times (\text{number of steps})$, so that the drift of Eq. 110 matches the gradient flow $\dot{\ell} = p - \hat{p}$ of Appendix C and the two trajectories can be compared at equal t. The main text states the results below at $\eta = 1$ for readability; we keep η explicit throughout this appendix because the continuous-time approximation we rely on is a small-η statement, and the noise floor we derive is proportional to η.

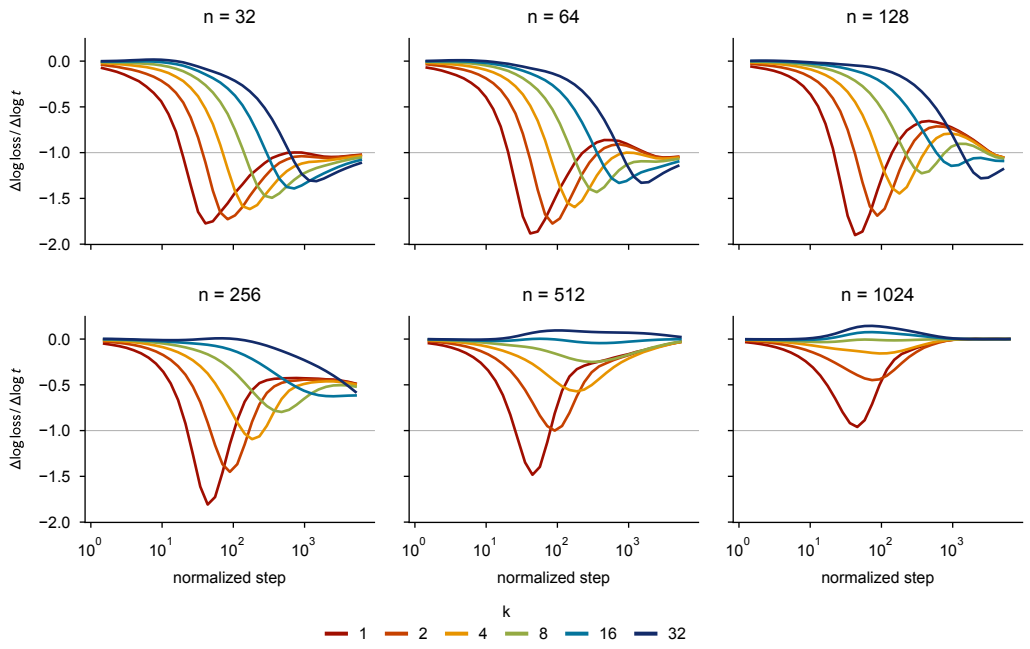

Figure 10: **Local log-log slope $\Delta \log \text{loss}/\Delta \log t$ for each panel of Figure 9.** Below the capacity threshold, all k converge to slopes close to the dashed reference at -1, matching the gradient-flow prediction $D_{\text{KL}} \sim k(d-k)/(td) \approx k/t$ for small k values.

The update has mean $\eta\,(p_j - \hat{p}_j)$, that is, the gradient-flow drift, and a zero-mean per-step fluctuation whose consequences we now characterize. Recall that $\ell_t^\star$ denotes the gradient-flow trajectory of Appendix C, $\hat{p}^\star(t) = \text{softmax}(\ell_t^\star)$ its softmax, and $\hat{p}_S^\star(t) := \sum_{j\in S} \hat{p}_j^\star(t)$ the on-support probability mass along that trajectory.

The central object of the analysis is the within-on-support logit variance

$$V := \frac{1}{k}\sum_{j\in S}(\ell_j - \bar{\ell}_S)^2, \text{ with } \bar{\ell}_S := \frac{1}{k}\sum_{j\in S}\ell_j. \tag{111}$$

Our analysis proceeds in four steps:

1. Show that, at leading order in the fluctuations, the excess KL loss is entirely controlled by $\mathbb{E}V$, so that the whole problem reduces to understanding how this scalar evolves.
2. Take the small-learning-rate continuous-time limit of the update in Eq. 110, which is tracked by a stochastic differential equation driven by the per-step gradient covariance.
3. Identify the subspace on which the fluctuations live.
4. Apply Itô's lemma to V to derive its ordinary differential equation, and combine everything to obtain the noise floor.

D.1 The within-on-support variance controls the excess loss

We first show why V is the natural quantity to track. Writing $\ell = \ell^\star + \delta$ and expanding the KL divergence to second order in δ,

$$D_{\text{KL}}(p\|\hat{p}) = D_{\text{KL}}(p\|\hat{p}^\star) + (\hat{p}^\star - p)^\top \delta + \tfrac{1}{2}\,\delta^\top H(\ell^\star)\,\delta + O(\|\delta\|^3), \tag{112}$$

where $H(\ell) = \text{diag}(\hat{p}) - \hat{p}\hat{p}^\top$ is the Hessian of the softmax cross-entropy loss. We will establish in the next subsections that, along the gradient-flow trajectory, δ has zero mean (so that the linear term vanishes in expectation) and lives in the $(k-1)$-dimensional within-on-support contrast subspace

$$\mathcal{E} := \Big\{u \in \mathbb{R}^d \,:\, u_j = 0 \text{ for } j \notin S,\ \sum_{j\in S} u_j = 0\Big\}, \tag{113}$$

on which H acts as $Hu = (\hat{p}_S^\star(t)/k)\, u$. Under these facts, the quadratic form in Eq. 112 reduces to $(\hat{p}_S^\star(t)/k)\, \|\delta\|^2$, and using $\|\delta\|^2 = k\, V$,

$$\mathbb{E}[D_{\mathrm{KL}}(p\|\hat{p})] - D_{\mathrm{KL}}(p\|\hat{p}^\star) \approx \frac{\hat{p}_S^\star(t)}{2}\, \mathbb{E}V. \tag{114}$$

The main effect of label sampling noise on the loss is therefore captured by the scalar $\mathbb{E}V$, and it remains to compute its dynamic, which we will do in the next three subsections.

D.2 Continuous-time limit of the stochastic update

We briefly recall the standard continuous-time approximation of stochastic gradient descent, following Mandt et al. [64], Li et al. [65]. For a general loss $L(\theta)$ minimized with stochastic gradient descent with learning rate η on parameter θ, and per-step gradient noise covariance $\Sigma(\theta)$, the iterates are tracked in the small-learning-rate limit, on the time scale $t = \eta \times$ (number of steps), by the stochastic differential equation

$$\mathrm{d}\theta_t = -\nabla L(\theta_t)\, \mathrm{d}t + \sqrt{\eta\, \Sigma(\theta_t)}\, \mathrm{d}W_t, \tag{115}$$

with W_t a standard Brownian motion. The drift is η-independent in this time parametrization, whereas the diffusion carries a $\sqrt{\eta}$: this is the usual statement that the learning rate controls how far the iterates wander from gradient flow, not how fast they travel along it. Writing $\theta_t = \theta_t^\star + \delta_t$, the deviation δ_t obeys, to leading order, the Ornstein–Uhlenbeck equation

$$\mathrm{d}\delta_t = -H(\theta_t^\star)\, \delta_t\, \mathrm{d}t + \sqrt{\eta\, \Sigma(\theta_t^\star)}\, \mathrm{d}W_t. \tag{116}$$

For the softmax cross-entropy loss and the update in Eq. 110,

$$\Sigma(\ell) = \mathrm{Cov}_{y\sim p}(\mathbf{1}[y = \cdot]) = \mathrm{diag}(p) - pp^\top, \qquad H(\ell) = \mathrm{diag}(\hat{p}) - \hat{p}\hat{p}^\top. \tag{117}$$

Note that Σ depends only on the target p, not on the current logits, and that $\mathrm{Tr}(\Sigma) = 1 - \|p\|_2^2 = (k-1)/k$.

D.3 Fluctuations live on the within-on-support contrast subspace

Along the symmetric gradient-flow trajectory of Appendix C, all on-support logits are equal and likewise for off-support logits. For any $u \in \mathcal{E}$, direct computation from Eq. 117 gives

$$\Sigma\, u = \frac{1}{k}\, u, \qquad H\, u = \frac{\hat{p}_S^\star(t)}{k}\, u, \tag{118}$$

because $p^\top u = 0$ (the entries of p are constant on S and u sums to zero on S) and likewise $\hat{p}^{\star\top} u = 0$. Outside $\mathcal{E}$, the covariance Σ acts as zero: the symmetric on-support direction $\mathbf{1}_S$ is annihilated since $\Sigma\, \mathbf{1}_S = p - p\,(p^\top \mathbf{1}_S) = p - p = 0$, and off-support coordinates are untouched by sampling since $p_j = 0$ for $j \notin S$. Thus all of the gradient noise feeds into the within-on-support contrast subspace $\mathcal{E}$, and δ_t has zero expectation outside it. Restricting to the projection $u_t := P_{\mathcal{E}}\delta_t$, we obtain the scalar Ornstein–Uhlenbeck equation

$$\mathrm{d}u_t = -\frac{\hat{p}_S^\star(t)}{k}\, u_t\, \mathrm{d}t + \sqrt{\frac{\eta}{k}}\, \mathrm{d}\tilde{W}_t, \tag{119}$$

where $\tilde{W}_t$ is a $(k-1)$-dimensional standard Brownian motion on $\mathcal{E}$. Note that the damping rate is set by the loss curvature and does not involve η, while the noise injected per unit time is proportional to η.

D.4 Dynamics of the within-support variance and noise floor

Since $\bar{\ell}_S$ is fixed under $P_{\mathcal{E}}$ (the symmetric direction is not driven by noise), we have $V = \|u_t\|^2/k$ along the gradient-flow trajectory. Itô's lemma applied to $\|u_t\|^2$ with Eq. 119 gives

$$\mathrm{d}\|u_t\|^2 = 2\, u_t^\top \mathrm{d}u_t + \eta\, \mathrm{Tr}\big(\Sigma\big|_{\mathcal{E}}\big)\, \mathrm{d}t \tag{120}$$

$$= -\frac{2\hat{p}_S^\star(t)}{k}\, \|u_t\|^2\, \mathrm{d}t + 2\sqrt{\frac{\eta}{k}}\, u_t^\top \mathrm{d}\tilde{W}_t + \eta\, \frac{k-1}{k}\, \mathrm{d}t, \tag{121}$$

where we used

$$\mathrm{Tr}(\Sigma|_{\mathcal{E}}) = \mathrm{Tr}(\Sigma) = \frac{k-1}{k}. \tag{122}$$

Taking expectations kills the martingale term $\mathrm{d}\tilde{W}_t$, and dividing by k yields the ordinary differential equation for $\mathbb{E}V$:

$$\frac{\mathrm{d}\,\mathbb{E}V}{\mathrm{d}t} = -\frac{2\,\hat{p}_S^\star(t)}{k}\,\mathbb{E}V + \eta\,\frac{k-1}{k^2}, \tag{123}$$

which is the equation announced in the main text (Eq. 12) at $\eta = 1$. As a result, $\mathbb{E}V$ converges to

$$\mathbb{E}V_\infty = \eta\,\frac{k-1}{2k} \tag{124}$$

as $\hat{p}_S^\star(t)$ converges to 1. Plugging Eq. 124 into Eq. 114 gives the asymptotic noise floor

$$\lim_{t\to\infty}\left(\mathbb{E}[D_{\mathrm{KL}}^{\mathrm{SGD}}(t)] - D_{\mathrm{KL}}^{\mathrm{GF}}(t)\right) = \frac{1}{2}\cdot\eta\,\frac{k-1}{2k} = \eta\,\frac{k-1}{4k}, \tag{125}$$

which is the formula reported in the main text as Eq. 13, there stated at $\eta = 1$. It is exactly zero for $k = 1$ (no ambiguity, no noise), monotonically increasing in k, and asymptotes to $\eta/4$ as $k \to \infty$. The learning rate only sets the overall scale of the floor: the ambiguity dependence, which is what we are after, is the η-independent factor $(k-1)/(4k)$.

D.5 Remark: convergence rate is untouched at leading order

The derivation above is a leading-order statement in the learning rate: the mean trajectory $\mathbb{E}\ell_t$ coincides with gradient flow up to higher-order corrections in η, so the convergence rates of Appendix C carry over unchanged. Sampling only adds zero-mean fluctuations whose variance saturates at the floor in Eq. 125. This contrasts with discrete stochastic gradient descent at finite step size, where Jensen-style corrections to the drift do modify the effective rate; classical analyses on logistic regression [96, 97] and quadratic losses [98, 64] give step-size slowdowns and floor values consistent with Eq. 125 as the leading term.

D.6 Validation of theory

The motivation here mirrors that of Appendix C.2: the analysis of Appendix D is derived for a single fixed target distribution and a logits vector updated directly, and we want to check that its qualitative predictions carry over to a linear head with shared parameters across many contexts of varying ambiguity. The key difference is that we now study the impact of output sampling noise rather than the underlying gradient-flow dynamics. The numerics are demanding because the quantity of interest is the small gap between the sampled and the population trajectory; as a reference, the theoretical curves of Figure 2 required averaging over $1{,}024$ Monte-Carlo trajectories to be smooth, which is already costly for the simple linear models we consider here. As a result, we cannot resolve the precise early-time onset of the sampling gap, but Figure 11 shows that the late-time excess loss increases with k, consistent with the prediction of Equation 13.

Experimental setup. The setup is identical to the gradient-flow validation of Appendix C.2 (Table 2), except that targets are now sampled at every step, so the model is trained on the cross-entropy loss instead of the full KL divergence (which is, up to the entropy of the target, the population objective used in the gradient-flow validation). To resolve the small sampling gap we restrict to $n \in \{32, 256\}$, shorten training to $1.6{\cdot}10^6$ steps, and average over 32 seeds for both the sampling sweep and the matched distributional reference used to compute the gap. These runs use $\eta = 1$ and mini-batches spread over all contexts, so the effective per-context learning rate differs from the single-context setting of Eq. 110; this rescales the absolute height of the plateau, and we therefore only compare its k dependence to the theory.

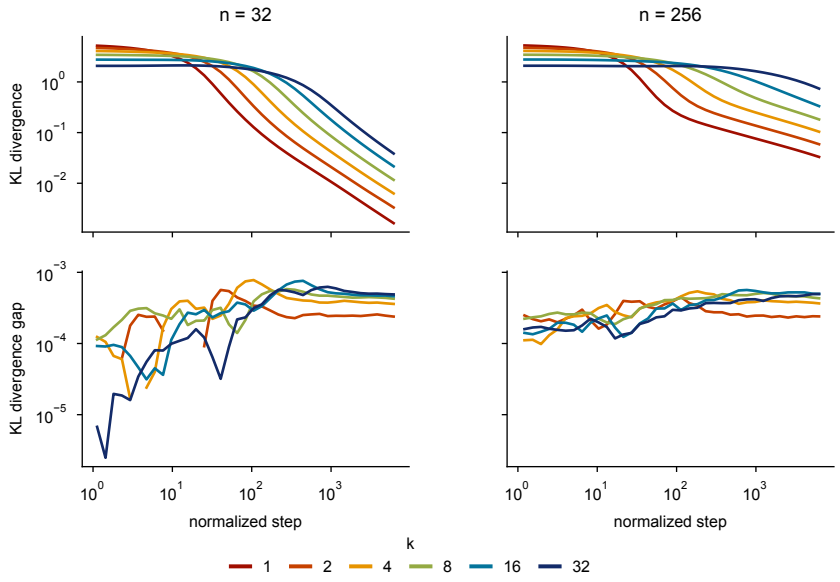

Figure 11: **The empirical sampling gap grows with k asymptotically, in line with the theoretical prediction of Section 4.2.** Top row: evolution of the mean per-k KL divergence under stochastic gradient descent on the cross-entropy loss (with sampled targets), against the training step normalized by the number of contexts per ambiguity level n, for $n \in \{32, 256\}$. Bottom row: gap to the matched distributional-target reference (full-batch gradient descent on the same task); $k = 1$ is omitted as its gap is below the seed-noise floor. The late-time gap increases with k, matching the asymptote of Equation 13.

E Connection to classical sampling results

The analyses of Appendices C–D look at the curse of ambiguity through the lens of learning; that is, how a parametric model trained by gradient descent on a finite set of contexts fits a target distribution. We complement that view here with two classical, non-parametric results on sampling from a single discrete distribution: the coupon collector problem and the maximum likelihood estimator of a categorical distribution. The curse of ambiguity already appears in these simple statistical problems: the sample complexity grows with k even in this idealized setting where no representation or parameters have to be learned.

E.1 The coupon collector problem

For a neural network to model p accurately, it must, at the very least, have observed every token in the support S at least once during training: a token that never appears in the data is a token that the network cannot meaningfully tell apart from any other off-support token. We can therefore lower bound the sample complexity of learning p by the time it takes for all k tokens of S to appear in the sample.

This problem is known as the coupon collector problem [99, 100]: how many independent draws from p are needed to observe all of its k values? For the uniform distribution we consider, the expected collection time T_k satisfies

$$\mathbb{E}[T_k] = k \sum_{i=1}^{k} \frac{1}{i} = k \log k + \gamma k + \frac{1}{2} + o(1), \tag{126}$$

with $\gamma \approx 0.577$ the Euler-Mascheroni constant. Generalizations of Equation 126 to non-uniform p are known but less clean, with the expected collection time depending on the full distribution rather than on a single summary statistic; see Flajolet et al. [101] for an extensive treatment. The qualitative picture is unchanged: $\mathbb{E}[T] \geq 1/p_{\min}$ at minimum, and the logarithmic overhead in the effective number of tokens persists.

Two consequences of these results are worth highlighting. First, the cost scales as $k \log k$ so the "naive" budget of one sample per support element is not enough, and even seeing each token a constant number of times requires an additional logarithmic factor in k. Second, this $k \log k$ floor holds regardless of the model, the optimizer, or the loss. As a result, any learner will inherit from this sampling-based curse of ambiguity.

E.2 Maximum likelihood estimation of a categorical distribution

Once all k tokens of the support have been observed, a learner can start estimating the actual probabilities on top of identifying the support. The standard non-parametric estimator for this problem is the maximum likelihood estimator

$$\hat{p}_j = \frac{N_j}{m}, \quad \text{where } N_j = \sum_{t=1}^{m} \mathbf{1}[y_t = j], \tag{127}$$

with m the number of drawn samples.

Total variation. The total-variation rate of the frequency estimator has been studied extensively [102–105]. For a uniform target on a support of size k and at least as many samples as support elements, $m \geq k$, we have

$$\mathbb{E} \, \|\hat{p} - p\|_1 \asymp \sqrt{\frac{k}{m}}, \tag{128}$$

which matches the minimax lower bound over all distributions on k symbols [105]. The restriction to $m \geq k$ is only there to exclude the degenerate regime $m \ll k$, in which the ℓ_1 error saturates at its maximal value instead of following Eq. 128. Reaching a fixed total-variation error ε thus requires $m \asymp k/\varepsilon^2$ samples, which is a manifestation of the curse of ambiguity, this time at the level of pure statistics.

KL divergence. Until all k support tokens have been observed, $\hat{p}$ has some $\hat{p}_j = 0$ on the support of p and $D_{\mathrm{KL}}(p \,\|\, \hat{p}) = +\infty$. Since that happens with positive probability at every finite sample size, the unconditional expected KL divergence of the frequency estimator is infinite for every m: collecting all coupons is not just a lower bound for the support-identification problem, it is a strict prerequisite for the KL divergence of the maximum likelihood estimator to be finite at all. We therefore state the rate conditionally on the coupon-collection event of Appendix E.1, which by Eq. 126 has high probability once $m \gg k \log k$. In that regime the frequency estimator attains the minimax rate for this problem [106, 105],

$$\mathbb{E} \left[D_{\mathrm{KL}}(p \,\|\, \hat{p}) \mid \text{all } k \text{ tokens observed} \right] \asymp \frac{k}{m}, \tag{129}$$

a rate that the same references show is achieved unconditionally, at every $m \geq k$, by the add-constant smoothed variants of $\hat{p}$ that are standard precisely because they avoid the infinite-KL issue above. The numerator now scales linearly in k. This matches qualitatively the late-time gradient flow loss $D_{\mathrm{KL}} \approx k(d-k)/(td) \approx k/t$ of Section 4.1, confirming that the k dependence we observe in our learning dynamics analysis reflects the intrinsic statistical difficulty of the problem.

Estimation of functionals. Scalar functionals of p, most notably its entropy or support size, admit faster estimators than p itself: plug-in estimators incur an $O(k/m)$ bias [107, 108], while polynomial-approximation estimators [109–112] reach the minimax rate $m \asymp k/\log k$, strictly faster than the $\Theta(k)$ rate of full-distribution estimation. Language models, however, are trained to model the full next-token distribution rather than scalar summaries of it, so our analysis inherits the harder $\Theta(k)$ scaling of Equations 128 and 129 rather than this milder $k/\log k$ rate.

F Synthetic n-gram task

Dataset construction. We start from the TinyStories corpus [66], lowercase each story, strip non-alphabetic characters, and split on whitespace. We then take as vocabulary the $V = 2{,}000$ most frequent words, all of which occur more than $10{,}000$ times in the corpus, and count 3-gram occurrences over the corpus, dropping 3-grams that contain any out-of-vocabulary word. Normalizing these counts yields the ground-truth conditional distributions p_x for approximately 1.1 million unique contexts x (pairs of in-vocabulary words). For each context, we compute the effective support size $\exp(H(p_x))$ directly from p_x. We also report results for the raw support size, defined as the number of next tokens with $p_{x,j} \geq 1/(10V)$; this threshold removes tokens whose empirical probability sits at numerical-noise level relative to the uniform distribution.

Ambiguity measured by raw support size. The main text quantifies ambiguity via the effective support size. To check that our conclusions do not hinge on this choice, Figure 12 reproduces Figure 3 with the raw support size on the y-axis, and shows the same qualitative trends.

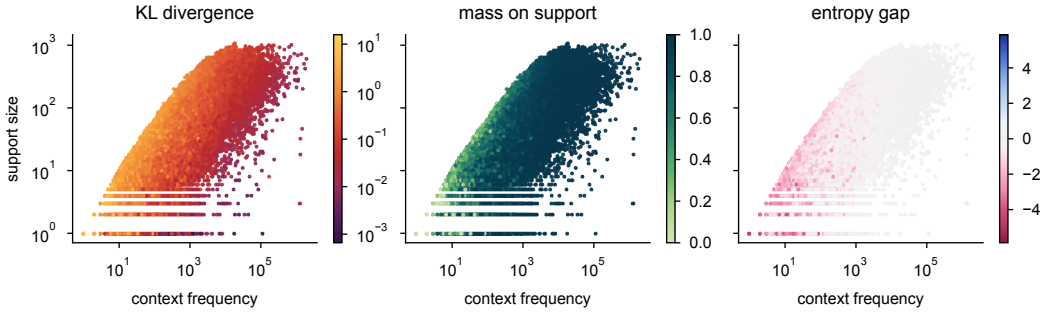

Figure 12: Same as Figure 3, but using the raw support size of the ground-truth distribution as the measure of ambiguity instead of the effective support size.

Sequence sampling. Training sequences are sampled autoregressively from the empirical 3-gram distribution. Each sequence is initialized with two beginning-of-sequence tokens, and at each subsequent position t we draw

$$y_t \sim p(\cdot \mid y_{t-2},\, y_{t-1}), \tag{130}$$

where $p(\cdot \mid y_{t-2}, y_{t-1})$ is the empirical conditional. Two edge cases arise. First, contexts containing one or two beginning-of-sequence tokens use the corresponding empirical unigram and bigram marginals. Second, the autoregressive procedure can produce a pair of in-vocabulary words that never co-occurred in the corpus, in which case the empirical conditional is undefined; we fall back to the uniform distribution over the vocabulary for these unseen contexts. Both edge cases account for a small fraction of the sampled tokens and are excluded from the per-context evaluation, which is restricted to contexts x for which p_x is well defined.

Architecture and training. The models are decoder-only transformers with 4 layers, 4 attention heads, rotary position embeddings, and an MLP expansion factor of 4; at the default width $d_{\text{model}} = 256$ this amounts to 4M parameters. We train on sequences of 256 tokens with batches of 128 sequences, so the default run of $3{,}000$ steps sees 98M tokens, using AdamW with weight decay 0.01 and a cosine schedule. We use one data seed per configuration and tune the learning rate per configuration, as summarized in Table 3.

We separate two width parameters. The residual stream, and with it the whole backbone, has width d_{model}; the linear head that produces the logits acts on a d_{embed}-dimensional vector, obtained from the last-layer residual stream by an additional linear projection when $d_{\text{embed}} \neq d_{\text{model}}$ (no projection is inserted when they are equal). It is d_{embed} that plays the role of the embedding dimension h of our theory, which is why the embedding-dimension ablation varies it at fixed backbone. One caveat is worth stating: since the projection is applied to a d_{model}-dimensional vector, the embeddings entering the head span a subspace of dimension at most $\min(d_{\text{embed}}, d_{\text{model}})$, so the two largest values of the ablation, $d_{\text{embed}} \in \{512, 1024\}$ at $d_{\text{model}} = 256$, cannot buy representational room beyond 256

dimensions. The informative part of that sweep is therefore its lower half, where the head is genuinely bottlenecked.

Per-context evaluation. At the end of training we sample $262{,}144$ fresh sequences from the same 3-gram process, run the model on them, and average each per-context metric over the positions at which the context occurs. Each dot of Figure 3 is one context, and its context frequency is its count in the training corpus rather than in this evaluation sample.

Iso-loss pipeline. We exploit the empirical observation that $\log D_{\mathrm{KL}}$ is well approximated by a linear function of the log effective support size and the log context frequency $\log f$. For each configuration, we fit

$$\log D_{\mathrm{KL}}(x) \approx a \log k(x) + b \log f(x) + c \tag{131}$$

to the per-context losses by weighted least squares, where $k(x)$ denotes the effective support size of context x. Since the bulk of contexts sits in the dense low-k, low-frequency region, we bin the $(\log k, \log f)$ plane into a 50×50 grid and weight each context by the inverse of the number of contexts in its bin; this prevents the dense region from dominating the fit and effectively targets the support of the (k, f)-distribution rather than its mode. We then report the level set on which the fitted D_{KL} equals a representative mid-range value of 0.5 nats, chosen because it sits in the middle of the per-context loss range observed in Figure 3. Figure 13 summarizes this pipeline when applied on Figure 3 (left).

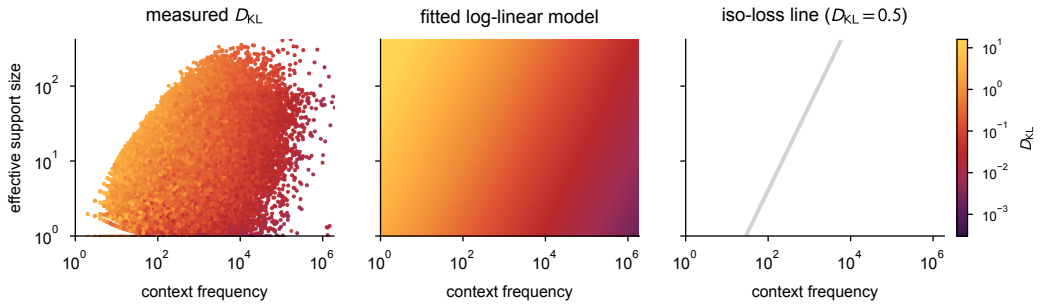

Figure 13: **Visual representation of how we summarize the per-context loss landscape of Figure 3 (left) into the iso-loss line we use in Figure 4.**

Extensive ablation. Figure 4 summarizes each ablation configuration by a single iso-loss contour fitted to its per-context losses. For completeness, we show in Figures 14–17 the underlying per-context data from which those contours are derived: each row is one configuration of the corresponding ablation, plotted as a scatter of (context frequency, effective support size) points colored by the same three metrics as Figure 3 (KL divergence, mass on support, entropy gap). Within each figure, color scales are shared across rows so that configurations can be compared directly. These cloud-point views show that the qualitative trends summarized by the iso-loss contours are visible at the per-context level and not artifacts of the log-linear fit of Equation 131.

	Default	Capacity	Embedding	Optimization	Sampling
Architecture	4-layer transformer, 4 attention heads				
Vocabulary	2,000 most frequent words of TinyStories				
Data	256-token sequences, batch size 128, one seed				
Optimizer	AdamW, cosine schedule, weight decay 0.01				
Learning rate	$\{10^{-4}, 3\times10^{-4}, 10^{-3}, 3\times10^{-3}\}$	$\{3\times10^{-4}, 10^{-3}, 3\times10^{-3}\}$			
d_{model}	256	$\{\mathbf{32}, \mathbf{64}, \mathbf{128}, \mathbf{256}, \mathbf{512}, \mathbf{1024}\}$	256	256	256
d_{embed}	256	256	$\{\mathbf{32}, \mathbf{64}, \mathbf{128}, \mathbf{256}, \mathbf{512}, \mathbf{1024}\}$	256	256
Training steps	3000	3000	3000	$\{\mathbf{356}, \mathbf{725}, \mathbf{1500}, \mathbf{3000}, \mathbf{6000}, \mathbf{12000}\}$	3000
Distributional	no	no	no	no	{**yes**, **no**}
Runs	4	18	18	18	6

Table 3: **Configuration of ablation experiments on the synthetic 3-gram task.** Each column corresponds to a sweep that varies a single parameter (highlighted in bold) while holding all others at the default value. Learning rate is tuned for every configuration in every sweep: the number of runs is the number of swept values times the size of the learning-rate grid, and the reported curves use the best learning rate (3×10^{-3} for the default configuration).

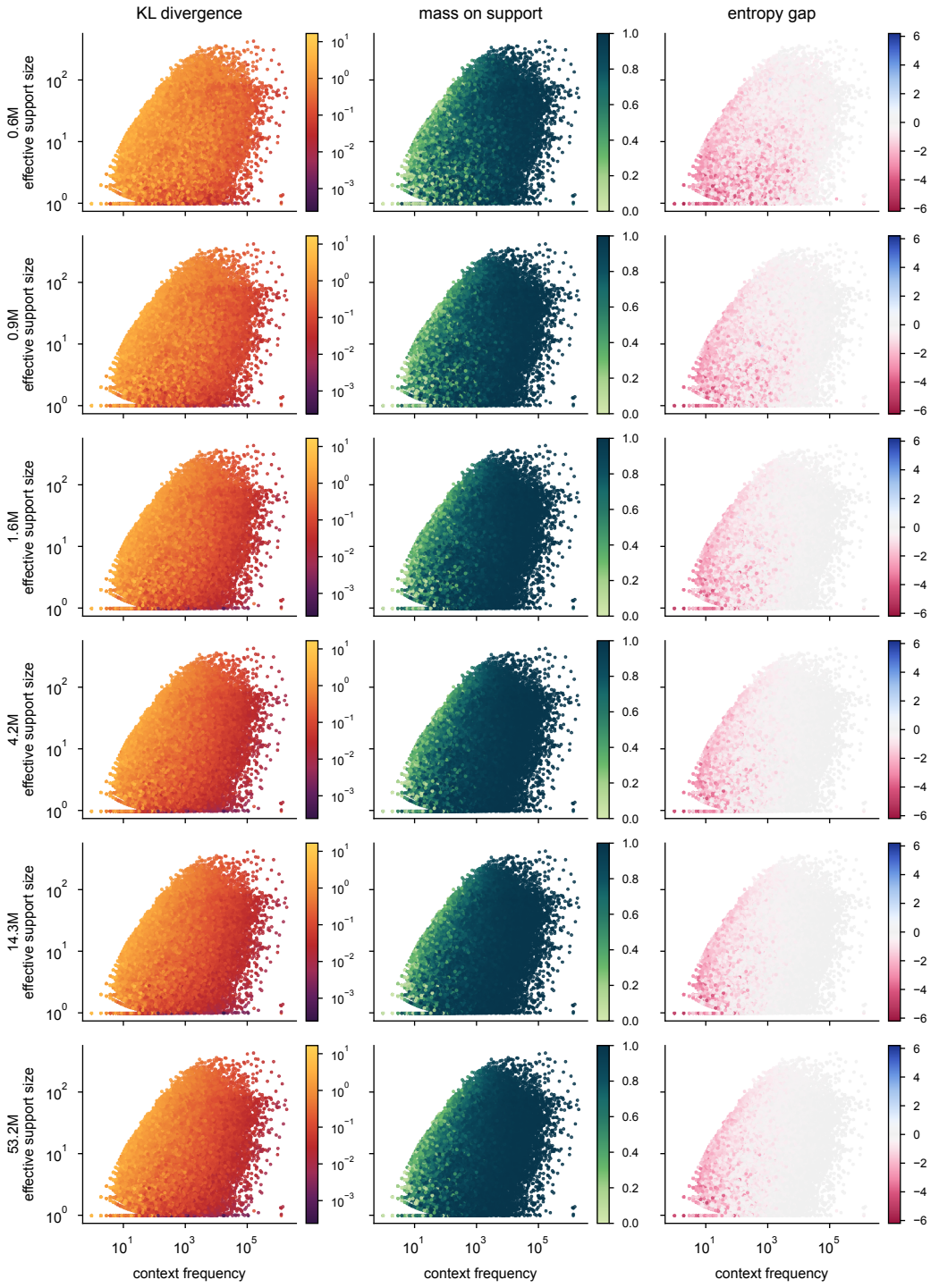

Figure 14: **Per-context losses across the capacity ablation.** Each row corresponds to one value of $d_{\text{model}} \in \{32, 64, 128, 256, 512, 1024\}$ (annotated as parameter count in millions). All other parameters are held at the defaults of Table 3: embedding dimension 256, 3000 training steps, sampled cross-entropy loss, with learning rate selected per configuration as the best of $\{3 \times 10^{-4}, 10^{-3}, 3 \times 10^{-3}\}$.

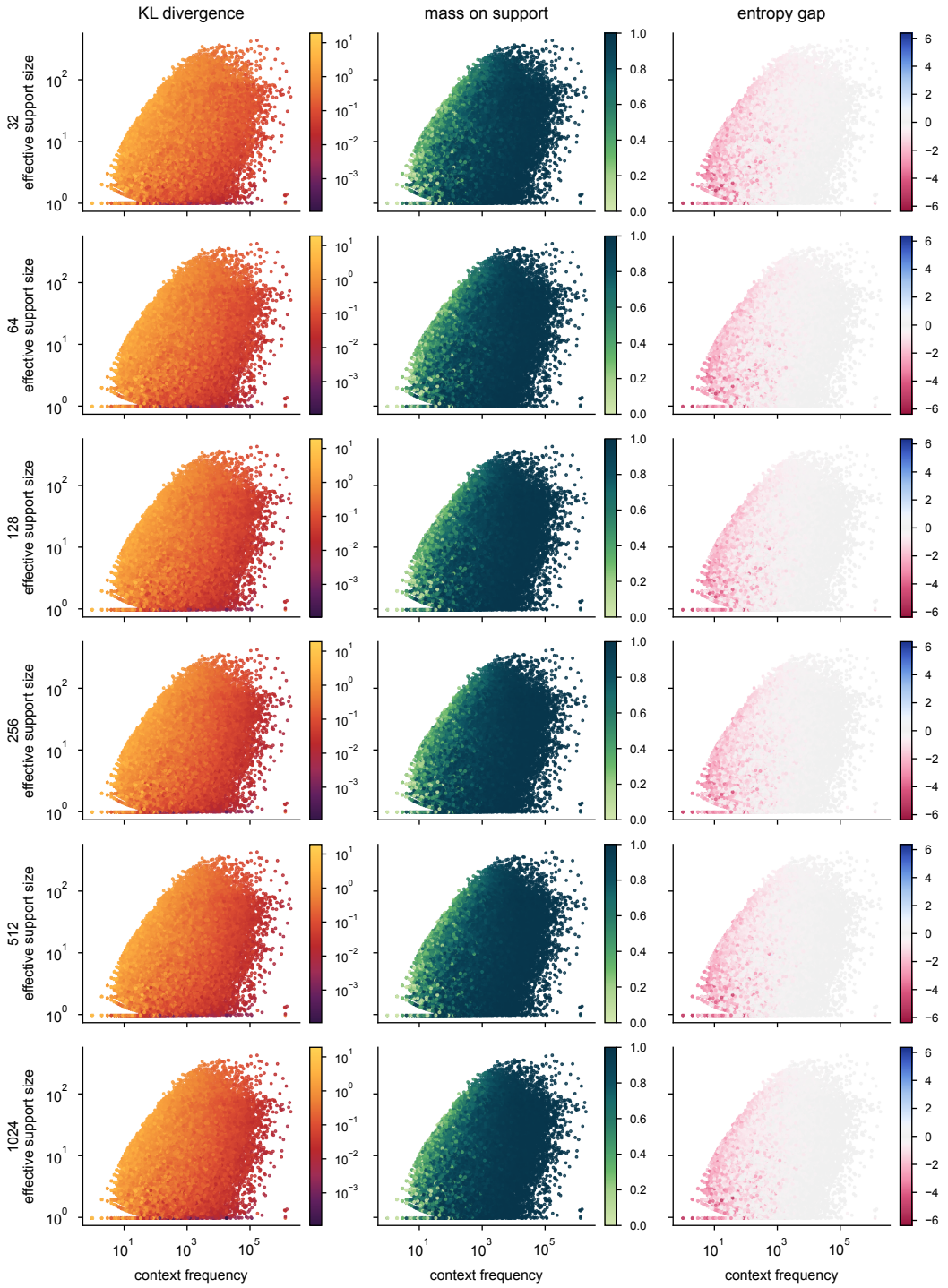

Figure 15: **Per-context losses across the embedding-dimension ablation.** Each row corresponds to one value of $d_{\text{embed}} \in \{32, 64, 128, 256, 512, 1024\}$. All other parameters are held at the defaults of Table 3: $d_{\text{model}} = 256$, 3000 training steps, sampled cross-entropy loss, with learning rate selected per configuration as the best of $\{3 \times 10^{-4}, 10^{-3}, 3 \times 10^{-3}\}$.

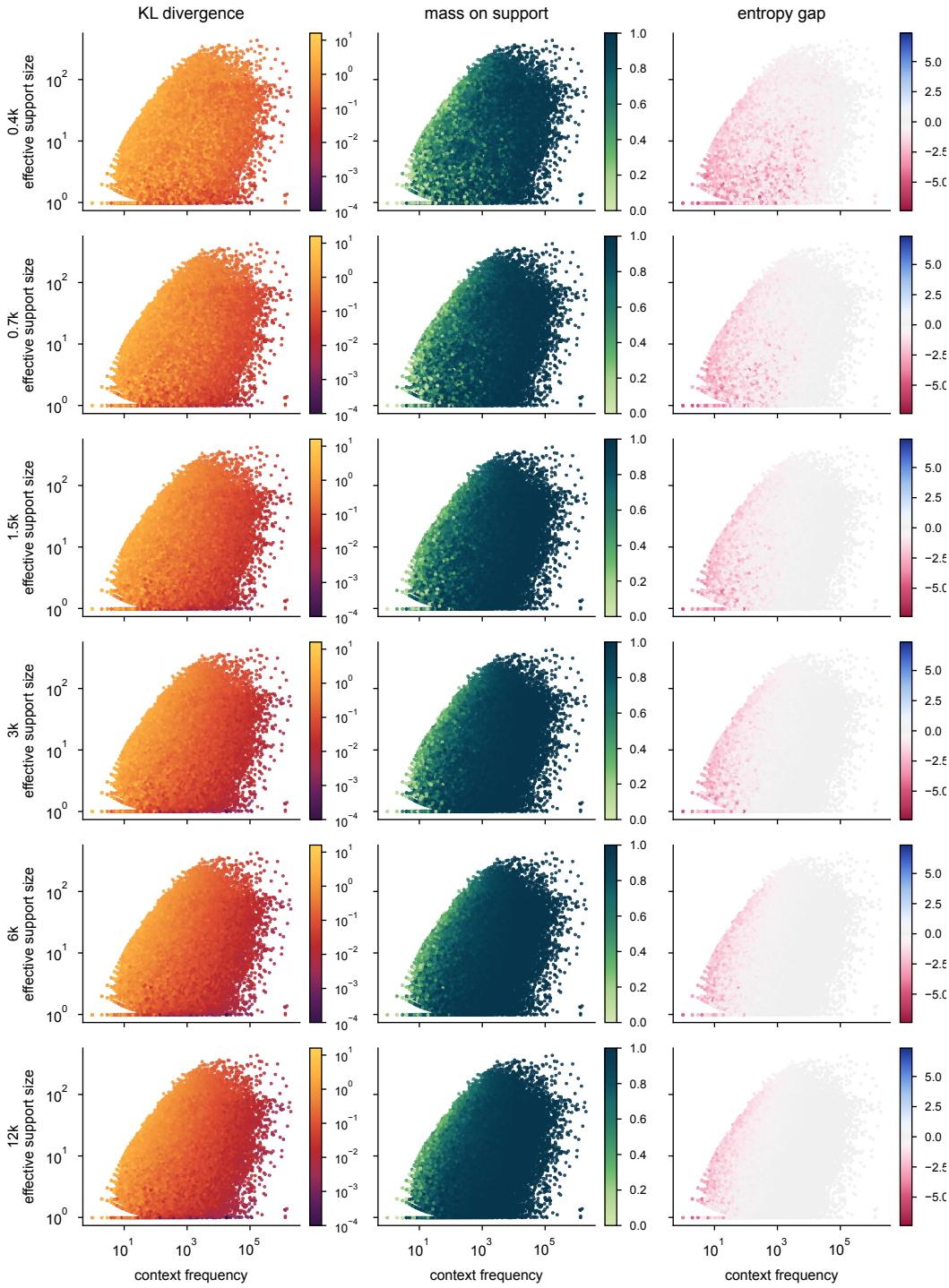

Figure 16: **Per-context losses across the training-duration ablation.** Each row corresponds to one number of training steps in $\{356, 725, 1500, 3000, 6000, 12000\}$. All other parameters are held at the defaults of Table 3: $d_{\text{model}} = d_{\text{embed}} = 256$, sampled cross-entropy loss, with learning rate selected per configuration as the best of $\{3 \times 10^{-4}, 10^{-3}, 3 \times 10^{-3}\}$.

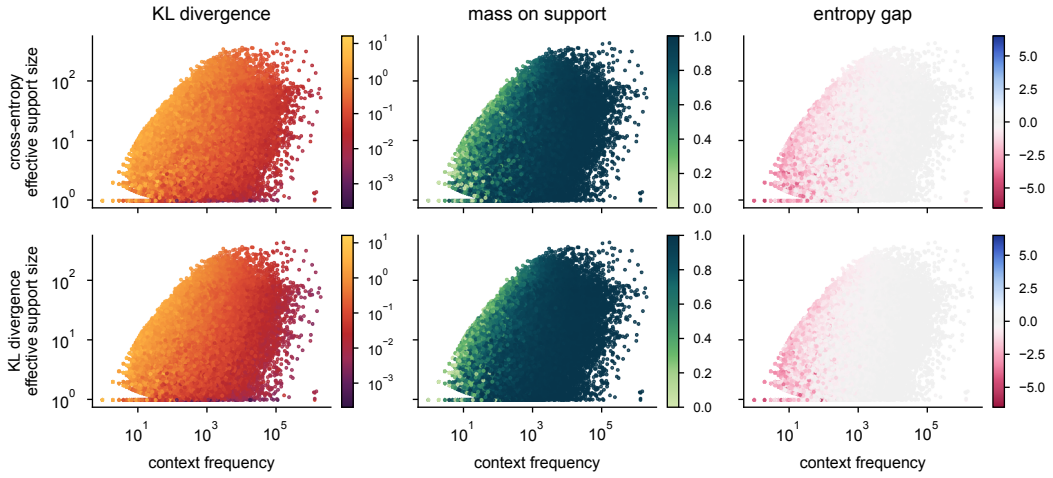

Figure 17: **Per-context losses across the loss ablation.** The two rows compare optimizing the full KL divergence $D_{\mathrm{KL}}(p_x \,\|\, \hat{p}_x)$ (top) against the sampled cross-entropy with one label drawn per step (bottom). All other parameters are held at the defaults of Table 3: $d_{\mathrm{model}} = d_{\mathrm{embed}} = 256$, 3000 training steps, with learning rate selected per configuration as the best of $\{3\times10^{-4}, 10^{-3}, 3\times10^{-3}\}$.

G Large language models experiments

G.1 Experiment details

Models and data. We use OLMo-2 13B [67] as a proxy for the ground-truth next-token distribution and OLMo-2 1B as the main model evaluated against this proxy. Both checkpoints are taken before any post-training. We sample $100{,}000$ sequences of length 512 from the Dolma v1.7 corpus [68] on which OLMo-2 was pretrained, tokenize them with the OLMo-2 tokenizer, and run a single forward pass of each model on these sequences in bfloat16. This yields $100{,}000 \times 512 = 5.12 \times 10^7$ context-target pairs, one per (sequence, position).

Per-context metrics. For each context, we compute the effective support size $\exp(H(p_{13\mathrm{B}}))$ of the proxy ground-truth distribution; we also report results for the raw support size, defined as the number of tokens with predicted probability $\geq 1/(10V)$ under OLMo-2 13B, where V is the OLMo-2 vocabulary size. Both procedures match those used for the 3-gram task in Appendix F. We then evaluate OLMo-2 1B against the same proxy ground truth via the per-context KL divergence $D_{\mathrm{KL}}(p_{13\mathrm{B}} \,\|\, p_{1\mathrm{B}})$, the mass that OLMo-2 1B assigns to the support of OLMo-2 13B, and the entropy gap between the two predicted distributions, exactly as in Figure 3.

Semantic context frequency. Since natural sequences rarely repeat past a few tokens, the context frequency that drives our 3-gram analysis cannot be read off from the corpus directly. We instead estimate it by performing density estimation on the cloud of OLMo-2 13B embeddings of all evaluated contexts, leveraging the fact that semantically similar contexts produce similar embeddings. Concretely, we

i) extract the last-layer hidden state of OLMo-2 13B for every context, which provides a 5120-dimensional embedding,

ii) PCA-reduce the resulting cloud to dimension 512, fitting the projection on a uniformly subsampled set of 5×10^5 embeddings,

iii) build a FAISS index [113] on the projected cloud,

iv) for each context, query its $2{,}000$ nearest neighbors using the index, excluding the query itself.

We then compute $n_{\mathrm{est}} = \sum_i \mathrm{e}^{-d_i/\lambda}$ over these nearest neighbors, where d_i is the distance to the i-th neighbor. This kernel density estimate counts how many contexts in the data are similar to the query, and is what we call the semantic context frequency. It depends on two choices, the bandwidth λ and the number of neighbors retrieved. We use $\lambda = 10$ and $2{,}000$ neighbors; the latter caps n_{est} at $2{,}000$, and the largest value we observe at $\lambda = 10$ is around 10^3, so the cap only compresses the extreme right tail of the cloud. Varying λ over $\{5, 10, 20\}$ yields qualitatively similar plots, as shown in Figure 18. We visualize the joint distribution of n_{est} and the estimated effective support size, together with the average position in the sequence on which each context occurs, in Figure 19.

Validation against true context frequency. On natural data, we have no direct access to the true context frequency, so the proxy above cannot be checked end-to-end. To validate that it carries the right information, we apply the same density pipeline on the synthetic 3-gram task of Section 5, where the true context frequency is known. We extract the last-layer hidden state of the 4-layer transformer of Appendix F at every position of a held-out evaluation set, and run the same FAISS-based density estimation, up to two immaterial differences: the kernel weights are averaged over the retrieved neighbors rather than summed, which only rescales the proxy by a constant, and the bandwidth is set to $\lambda = 2$, the scale of the distances in this smaller embedding space. This yields 2.5×10^7 evaluated positions, spanning 5.2×10^5 distinct contexts. Figure 20 compares the resulting proxy to the true context frequency: the two are strongly correlated, with a Pearson correlation of 0.86 between the log frequencies over all evaluated positions and a Spearman correlation of 0.92 over 2×10^6 positions subsampled uniformly among them (both correlations vary by less than 0.03 across the bandwidths $\lambda \in \{1, 2, 5\}$ we ran on this task). Weighting positions equally is the relevant convention here, since this is also how contexts enter the per-context evaluation of Figure 5, each context contributing in proportion to how often it occurs. It does mean that the agreement is dominated by frequent contexts: computed over distinct contexts instead, so that a context seen once weighs as much as one seen

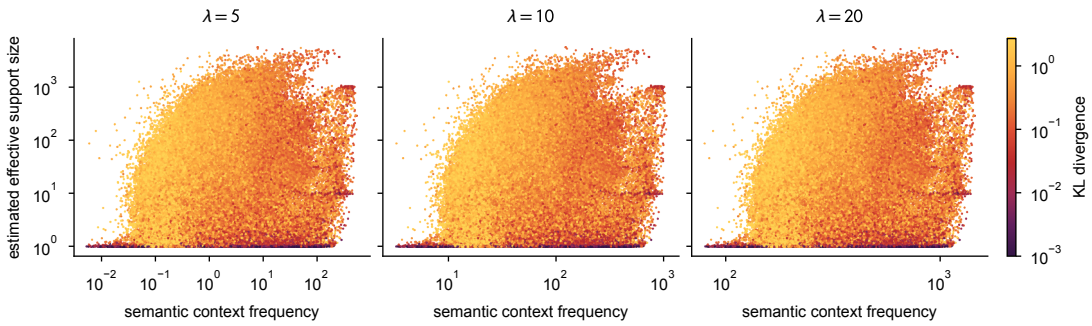

Figure 18: **Effect of changing the kernel bandwidth λ on the results of Figure 5.** Same setup as Figure 5, varying the bandwidth λ of the exponential kernel $\exp(-d_i/\lambda)$ used to compute n_{est} from kNN distances. Larger λ shifts the x-axis to the right (more neighbors are weighted in), but the cloud shape and the per-context KL coloring are essentially invariant under λ.

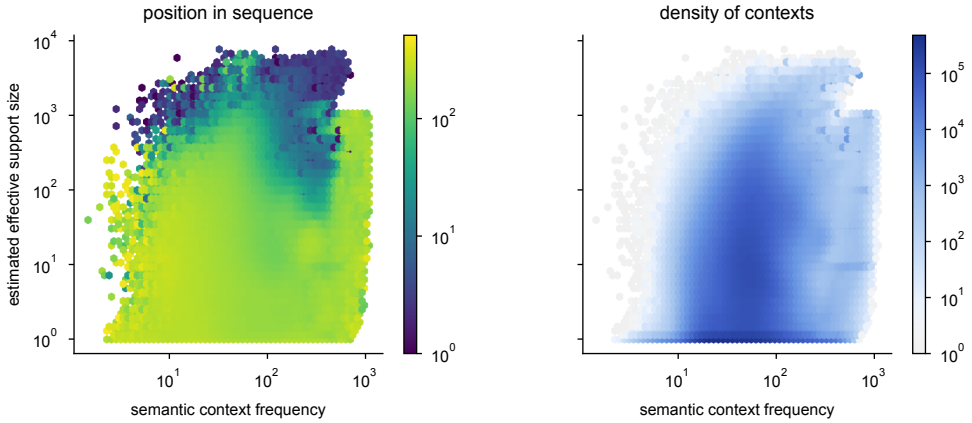

Figure 19: **Joint distribution of semantic context frequency and estimated effective support size, alongside the average position in the sequence at which each context occurs.** Both panels share axes with Figure 5. *Left:* each hex is colored by the geometric-mean position (in tokens) of the contexts that fall in it, with opacity modulated by the log of the per-hex count so sparse cells fade out. Low-frequency, low-effective-support cells are dominated by early-sequence positions, where local context is short and the kNN density estimate is least reliable. *Right:* log-density of contexts in the same plane. Most of the mass concentrates in a narrow band where n_{est} and effective support size are positively correlated.

a million times, the same correlations drop to 0.30 and 0.24. The proxy therefore ranks frequent contexts reliably but is noisy on the rare tail, consistent with the caveat we make in Figure 19 about the low-n_{est} region.

Additional ablations. For completeness, we report two further ablations. First, we replace effective support size by the raw support size of the OLMo-2 13B distribution as the ambiguity measure (Figure 21). The qualitative picture is unchanged. Second, we vary the size of the probe model from 1B to 7B parameters (Figure 22). The 7B probe tracks the reference more closely overall, with the largest gains concentrated at low effective support size and high context frequency, that is, in the bottom-right region of the cloud. We have two hypotheses for where this improvement takes place:

(i) The 7B probe has more capacity than the 1B probe, allowing it to fit medium-ambiguity contexts that exceeded the 1B model's capacity. This could explain the phase-transition-like improvement at the bottom of the cloud.

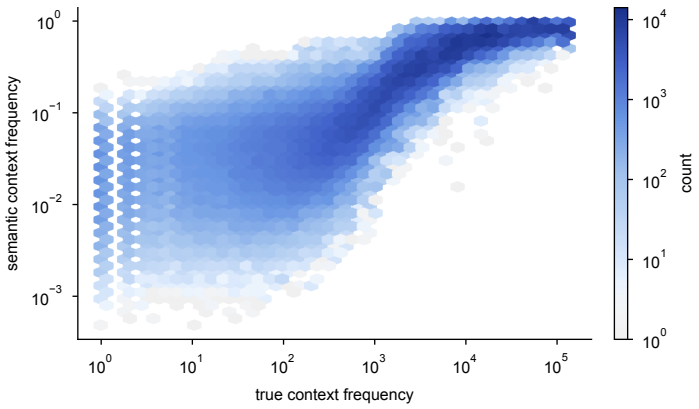

Figure 20: **The kNN density proxy correlates with the true context frequency on the 3-gram task.** Log-density (hexbin, 10^6 positions subsampled uniformly from the held-out evaluation set) of the true frequency of each context in the training corpus (x-axis) against its kernel density $\mathrm{e}^{-d_i/\lambda}$ averaged over its retrieved nearest neighbors in the last-layer embedding space of the trained 4-layer transformer (y-axis), at $\lambda = 2$. On this task we plot the average rather than the sum of Appendix G.1, which is why the y-axis lies below one; the two differ by the constant number of retrieved neighbors, so both correlations below are identical for either choice. The two quantities are strongly rank-correlated across positions (Spearman correlation of 0.92), supporting the use of n_{est} as a context-frequency proxy on natural data, where the true count is not observable.

(ii) On the right side of the cloud, where contexts are common enough to escape the capacity bottleneck, the gain could be explained by the gradual late-time decay of Equation 10 from the optimization analysis of Section 4.1, in which the convergence rate no longer depends on the support size and all contexts improve in parallel.

These remain speculations and verifying them empirically is left to future work.

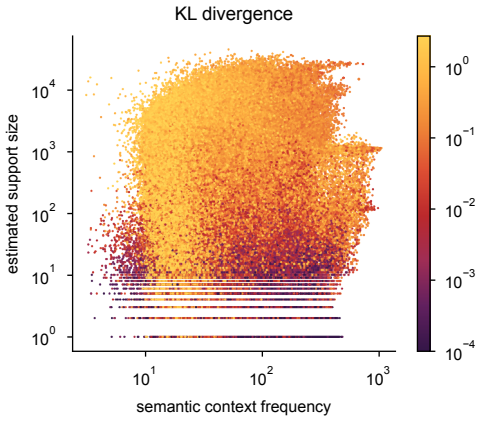

Figure 21: **The curse of ambiguity persists when ambiguity is measured by raw support size instead of effective support size.** Same setup as Figure 5, except that the y-axis now shows the raw support size of the OLMo-2 13B reference distribution. The qualitative picture is unchanged.

G.2 In-context predictability explains the low-loss low-context-frequency band

Figure 5 reveals a band of tokens with low semantic context frequency ($n_{\text{est}} < 10$) on which the small probe model and the large target model nevertheless agree very tightly. This is surprising at first sight: how can a model accurately predict the next token given a context that has rarely been seen during training?

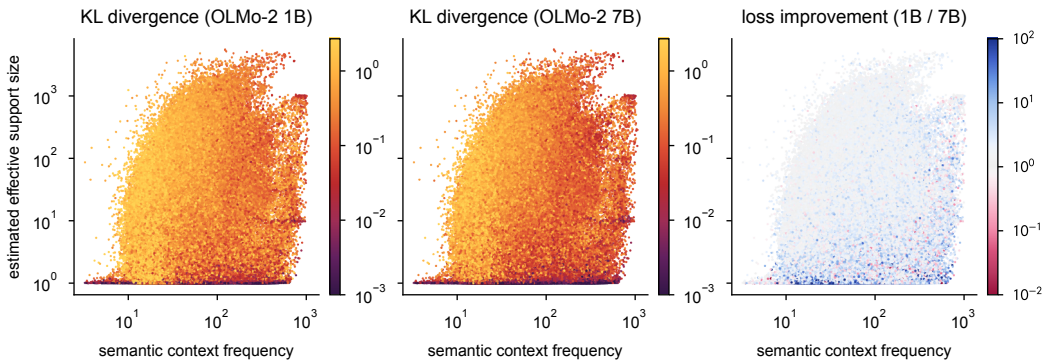

Figure 22: **Effect of varying the size of the evaluated model on the results of Figure 5.** Same setup as Figure 5, with OLMo-2 1B (left) and OLMo-2 7B (middle) as the model evaluated against the OLMo-2 13B reference. The 7B model tracks the reference more closely (KL is lower across the cloud), mirroring the improvement observed when increasing model size in the synthetic 3-gram task; cf. Figures 4 and 14.

We investigate this band by inspecting the underlying contexts (Table 4). In each reported example, the last few tokens preceding the prediction already appear earlier in the same sequence, followed there by the same continuation. A generic in-context copying mechanism, such as an induction head, would suffice to make these predictions. To verify that this pattern extends beyond the subsampled examples of Table 4, we measure, for each token in the band, whether the one- or two-token prefix preceding it appears at any earlier position of the same sequence and whether the actual next token coincides with one of the in-context continuations. We find that 61% of in-band predictions are explained by an in-context bigram completion (one-token prefix), versus 11.2% on the rest of the corpus; for trigram completions (two-token prefix), the corresponding rates are 33% in the band and 3.4% elsewhere. The prefixes themselves also recur in the same sequence far more often in the band (74% for one-token prefixes, 36% for two-token ones) than in the rest of the corpus (47% and 11%, respectively).

The low-loss band of Figure 5 therefore primarily corresponds to context-dependent predictions. This is a concrete example of how the structure of the data can affect our analysis: the simple fact that text repeats itself at short range within a document enables fast prediction and generalization on contexts that have been barely seen, or not seen at all, during training. Precisely measuring the curse of ambiguity in practice would require controlling for this kind of structure, which we leave to future work.

G.3 Quantifying the curse of ambiguity at fixed semantic context frequency

Figure 5 provides a qualitative picture confirming that large language models suffer from the curse of ambiguity. We take a step towards quantifying it by computing the correlation between the per-context loss and the estimated ambiguity, at fixed semantic context frequency. We do this for both ambiguity proxies, effective support size (Figure 23) and raw support size (Figure 24). In both cases, the per-context loss is positively correlated with ambiguity in log-log space, quantitatively confirming the curse of ambiguity. We report the squared Pearson correlation R^2, which reaches 0.37 for effective support size and 0.68 for raw support size, that is, correlations of $r \approx 0.61$ and $r \approx 0.83$. The values are higher for raw support size partially because R^2 measures linear correlation while the binned plots for effective support size look more nonlinear. The correlation is also highest in regimes where the overall loss is not too large, in particular it drops in the middle band of semantic context frequency.

population (1994 est.) Net migration rate: -12.05 migrant(s)/1,000 population (1994 est.) Infant mortality rate: 18.5 deaths/1,
RESSIDA. I will not, uncle. I have forgot my father; I know no touch of consanguinity, No kin, no love, no blood, no soul so near me As the sweet Troilus. O you gods divine, Make Cressid's name the very crown of falsehood, If ever she leave Tro*il*
from the men of Shechem, and from the house of Millo, and devour Abimelech. 9:21 And Jotham ran away, and fled, and went to Beer, and dwelt there, for fear of Abimelech his brother. 9:22 When Abime*le*
:014:017 And I, behold, I will harden the hearts of the Egyptians, and they shall follow them: and I will get me honour upon Pharaoh, and upon all his host, upon his chariots, and upon his horsemen. 02:014:018 And the Egyptians shall know that I am the LORD, when I have gotten me honour upon *Ph*
/1,000 population (1994 est.) Net migration rate: -1.78 migrant(s)/1,000 population (1994 est.) Infant mortality rate: 8.9 deaths/1,
God, that I may be at once avenged of the Philistines for my two eyes. 16:29 And Samson took hold of the two middle pillars upon which the house stood, and on which it was borne up, of the one with his right hand, and of the other with his left. 16:30 And *Sam*
the tone of fairy tales, and it is certainly not lawlessness or even liberty, though men under a mean modern tyranny may think it liberty by comparison. People out of Portland Gaol might think Fleet Street free; but closer study will prove that both fairies and journalists are the slaves of duty. Fairy godmothers seem at least as strict as other god*mo*
creditors by purchasing a sizable amount of the delinquent commercial debt in the secondary market at a substantial discount. The government had paid 100% of remaining official debt arrears to the US, Germany, France, and Spain. All commercial debt ar*re*
: and Abimelech king of Gerar sent, and took Sarah. 20:3 But God came to Abime*le*
prepare to meet and attack him. Wright, with his division of Sedgwick's corps, was ordered, by any road he could find, to join on to Warren's right, and Getty with his division, also of Sed*g*

Table 4: **Examples of low-loss tokens from the band highlighted in Figure 5** ($n_{\text{est}} < 10$). The token to predict is shown in *italics* and the matching n-gram earlier in the same sequence is shown in dark red (a trigram when a trigram match exists, a bigram otherwise).

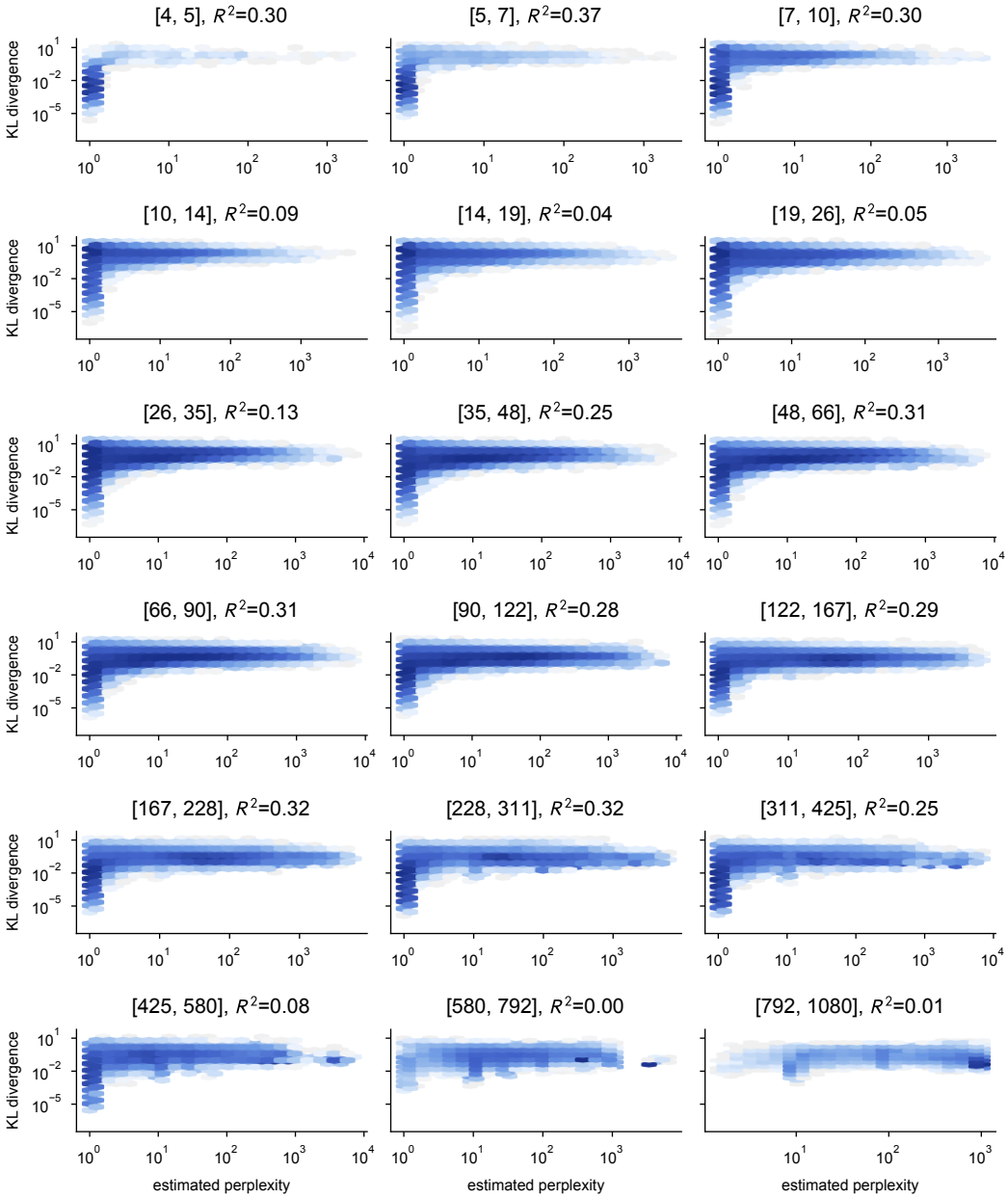

Figure 23: **Per-context loss and effective support size are correlated at fixed semantic context frequency.** Each panel slices the data of Figure 5 by a log-spaced bin of semantic context frequency, indicated in brackets in the title, and shows the 2D log-density of the KL divergence between the predictions of the 13B reference model and the 1B probe against the estimated effective support size; the value next to each title is the squared Pearson correlation R^2 between $\log D_{\mathrm{KL}}$ and $\log$ effective support size within the bin. The correlation is strongest for both small (rare contexts, $n_{\mathrm{est}} \lesssim 10$) and large (common contexts, $n_{\mathrm{est}} \gtrsim 100$) semantic context frequencies. It drops for intermediate values of n_{est}, which is partially explained by the high overall loss in that regime: when most contexts already sit near the maximum KL, effective support size has little remaining variance to predict.

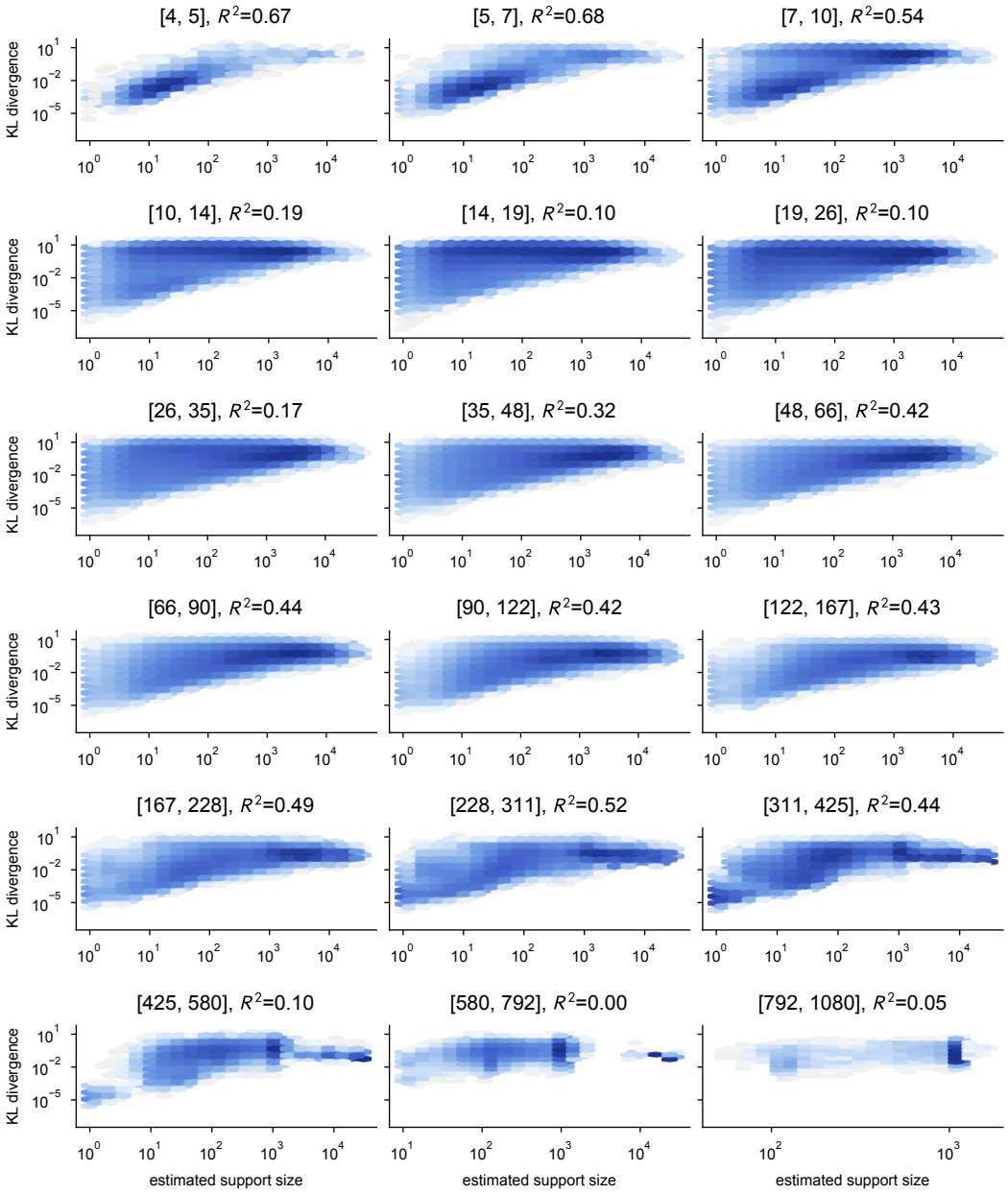

Figure 24: **Per-context loss vs. raw support size at fixed semantic context frequency.** Same setup as Figure 23, with raw support size in place of effective support size on the x-axis. R^2 is higher than in the effective-support-size version, partly because the relationship is more linear in this plane.