IIMB-WP N0. 740/2026

WORKING PAPER NO: 740

Information-Theoretic Limits of Reliability and Scaling in Language Models

Subhabrata Majumdar
Assistant Professor,
Decision Sciences,
Indian Institute of Management Bangalore
smajumdar@iimb.ac.in

Year of Publication – June 2026

Information-Theoretic Limits of Reliability and Scaling in Language Models

Subhabrata Majumdar
Indian Institute of Management Bangalore
Bengaluru, India
smajumdar@iimb.ac.in

Abstract

Large language models (LLMs) are evaluated as though perfect reliability is achievable for any task given sufficient scale. We show this assumption is information-theoretically unjustified. Every generative task has a *reliability ceiling* that no model can exceed, determined by how much output uncertainty is resolvable from observable context. The gap decomposes into a resolvable component closable with additional context and a subjective component inherent to task ambiguity. Autoregressive generation further degrades this ceiling at a rate governed by the task's *dependency kernel*, which quantifies inter-token correlations in the output. From these two primitives, we derive a first-principles scaling law where LLM performance is bottlenecked by the scarcer resource: training data or model capacity. This law recovers the Chinchilla scaling law as a special case and provides a structural account of when scaling improves reliability. Beyond scaling, our framework unifies diverse practical phenomena, such as the benefits of retrieval-augmentation and the spectral mechanics of catastrophic forgetting. Our work formalizes the resource-complexity tradeoffs that govern model performance across domains, offering a unified theory of performance limits in generative language models.

1 Introduction

Large language models (LLMs) have demonstrated remarkable capabilities across a wide range of tasks, from code generation to mathematical proofs to creative writing. Yet a persistent empirical pattern challenges the narrative of uniform progress: performance on different tasks saturates at different levels, and the saturation points do not converge toward perfect reliability as models scale [1, 20, 27]. The power-law relationships that govern this improvement, known as the Chinchilla scaling law [22, 23], empirically predict that cross-entropy loss decreases as a power of model size and training data, converging to an irreducible floor. Even though these relationships have become the primary tool for planning LLM training runs, they neither have exact theoretical interpretations, nor explain why frontier models achieve near-perfect accuracy on code generation and grade-school mathematics while doing much worse on creative writing and expert-level reasoning.

This paper introduces an information-theoretic framework that addresses the above issues. We show that an LLM faces two distinct barriers to achieving perfect performance across generative tasks, both intrinsic to the task mixture on which it is trained.

The first barrier pertains to *reliability*. Every task has a maximum achievable reliability that no model can exceed, regardless of scale or architecture. This is determined by the conditional entropy in the output given the input. Verifiable tasks (such as formal proofs and code generation) have higher ceilings compared to more ambiguous tasks (such as creative writing). The reliability ceiling is further degraded by autoregressive generation, which compounds errors at a rate governed by the

Preprint.

inter-token correlation structure of the task outputs. Injecting additional context during generation, such as through few-shot examples, retrieval, and tool use, slows down this degradation and raises the effective ceiling.

The second barrier pertains to *scaling*, and determines how quickly scaling training data on a task mixture and model parameters reduces the gap to the ceiling. Derivation of scaling exponents *a priori* has recently been identified as a central open problem in the emerging theory of deep learning [39]. Based on the spectral structure of the task mixture, we derive the following power-law relationship between the empirical loss $\mathcal{L}(N, D)$, model size N, and training tokens D:

$$\mathcal{L}(N, D) \;=\; \underbrace{H(Y \mid X)}_{\text{irreducible error}} \;+\; \max\Bigg(\underbrace{\frac{A}{N^{(\bar{\mu}-1)/(d(\nu_{\mathcal{T}}+1))}}}_{\text{approximation error}},\; \underbrace{\frac{B}{D^{(\bar{\mu}-1)/\nu_{\mathcal{T}}}}}_{\text{estimation error}}\Bigg) \;+\; \text{lower order.} \qquad (1)$$

The exponents are determined by two spectral properties of the task mixture: how quickly the shared correlation structure across tasks decays ($\nu_{\mathcal{T}}$) and how concentrated the prediction signal is in the leading modes of this structure ($\bar{\mu}$). The factor d captures architectural overhead. The irreducible floor $H(Y|X)$ is the conditional entropy of the output given the input. This uncertainty is due to information relevant to generation that is absent from the input, persisting even with a perfect model. The $\max$ form indicates that performance is limited by whichever resource, data or capacity, is more scarce. The familiar additive Chinchilla law [22] emerges as a special case along the compute-optimal frontier where both resources are balanced.

Our contributions are as follows. Proofs of all results are in Appendix B.

1. We formalize the reliability ceiling (§ 3) and derive its decay rate under autoregressive generation (§ 4).
2. We introduce the dependency kernel, a quantity that captures the inter-token correlation structure of tasks, and show how it governs the above decay rate (§ 5).
3. We derive the $\max$-form scaling law in Eq. (1) from first principles. The standard Chinchilla law is recovered as a corollary (§ 6).
4. We connect our framework to applied AI practices, explaining when and why techniques like few-shot learning, retrieval-augmented generation (RAG), constrained decoding, and fine-tuning are effective, and identifying phenomena such as benchmark instability and catastrophic forgetting as structural consequences of the framework (§ 7).

2 Related Work

Information bottleneck (IB) theory The IB framework [41] formalizes the tradeoff between compression and prediction: a representation T of input X should minimize their mutual information (MI) $I(X;T)$ while maximizing $I(T;Y)$, the MI between the representation and output. Shwartz-Ziv and Tishby [38] conjectured that deep networks implicitly optimize this objective, but subsequent work showed the compression phase depends on activation function choice [35] and can be mimicked by estimation artifacts [19]. Recent papers extend IB to generative models [24, 48], but use it to analyze training dynamics rather than derive hard performance limits. We take the latter route: our reliability ceiling follows from the data processing inequality applied to the same Markov structure that motivates IB, but yields an upper bound on achievable performance rather than a characterization of what networks learn to discard.

LLM reliability and hallucination Mohsin et al. [30] prove via diagonalization that hallucination is inevitable for any computably enumerable model class, and Xu et al. [46] establish similar impossibility results from a learning-theoretic perspective. These are existence proofs: they show failure must occur but do not quantify how much failure to expect on a given task. Wang and Sennrich [44] formalize the distributional drift that drives compounding errors in autoregressive generation, and HELM [27] documents empirically that performance saturation levels differ by task type. In contrast, our bounds are prescriptive. They formalize the reliability ceiling for a given task, decompose it into a closable component and a permanent floor, and identify interventions to close the first gap.

Neural scaling laws It is empirically established that language model loss follows a power law in model size N and training tokens D [7, 22, 23]. On the theoretical side, Canatar et al. [14] and

Bordelon et al. [8] showed that power-law eigenspectra in the data covariance produce power-law learning curves in kernel regression, and multiple studies [4, 12, 21] offered a statistical mechanics perspective. These results explain the functional form but leave the three terms of the Chinchilla law without task-theoretic interpretations. Recently, Cagnetta et al. [13] theoretically derived the data exponent in the Chinchilla law. We take a different route than them, and additionally yield the capacity exponent, the irreducible floor, as well as the $\max$-form linking the three as the fundamental relationship that governs learning over a task mixture.

Task structure and learnability Classical PAC learning [42] bounds sample complexity via hypothesis class complexity but does not address the internal structure of the target. Research on benign overfitting [5] and double descent [6] showed that the spectral structure of the data covariance governs generalization in overparameterized regimes. Nayak and Varshney [32] model task structure through a bipartite skill-text graph, deriving Chinchilla-type scaling via an iterative concept-acquisition process. Our dependency kernel provides a complementary characterization: rather than modeling which skills a task requires, it captures how the output tokens of a task depend on one another, and how this correlation structure determines both autoregressive error propagation and scaling exponents.

3 The Reliability Ceiling

Preliminaries Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space. A *generative task* is specified by a joint distribution $P_{X,Y,C}$ over three random elements:

- $X \in \mathcal{X}$: the *observable context* (prompt, problem statement, or input),
- $Y = (y_1, \ldots, y_L) \in \mathcal{V}^L$: the *target output*, a sequence of tokens from a finite vocabulary $\mathcal{V}$,
- $C \in \mathcal{C}$: the *latent context* (information relevant to Y but absent from X).

We assume $\mathcal{X}$ and $\mathcal{C}$ are Polish spaces with regular conditional distributions $P_{Y|X}$ and $P_{Y|X,C}$. The variable C encodes all task-relevant information not in X. Its interpretation varies by task. For formal mathematics or coding problems, $C = \emptyset$, since the problem statement fully specifies the answer. For factual question-answering, C captures ambiguity and missing world knowledge, making $H(Y|X)$ nonzero. For creative writing, C encodes aesthetic preferences, cultural norms, and emotional context of both the writer and the reader, making $H(Y|X)$ large relative to $H(Y)$.

Definition 1. *A* generative model *is a parametric family Q_θ that, given an input $X \in \mathcal{X}$, produces a distribution over $\mathcal{V}^L$. The generated output $\hat{Y} = (\hat{y}_1, \ldots, \hat{y}_L)$ is a draw from this distribution.*

The generation process factors as $X \to T \to \hat{Y}$, where T is the model's internal representation of X. Since the model observes only X, never C, T and C are conditionally independent given X, written $T \perp C \mid X$. Similarly, the generated output $\hat{Y}$ and the input X are conditionally independent given T, written $\hat{Y} \perp X \mid T$, as the decoder conditions on T alone[1]. These two independence conditions, together with the fact that X is a marginal of (X, C), yield the Markov chain

$$Y \longleftrightarrow (X, C) \longleftrightarrow X \longleftrightarrow T \longleftrightarrow \hat{Y}, \tag{2}$$

where each link discards information: $(X, C) \to X$ loses latent context, $X \to T$ loses what the encoder discards, and $T \to \hat{Y}$ loses what the decoder does not use.

Definition 2. *The* reliability ceiling *of a generative task is $R^* := I(X;Y)/H(Y) \in [0, 1]$.*

The quantity R^* measures the fraction of output uncertainty resolvable from the observable input. When $R^* = 1$, the task is fully determined by X. When $R^* < 1$, a nonzero fraction depends on latent context C that no model can access. The decomposition $I(X;Y) = H(Y) - H(Y|X)$ clarifies that R^* is a property of the task distribution, not of any particular model.

Theorem 1. *Let $P_{X,Y,C}$ define a generative task and Q_θ a generative model. Then for any measurable functional $\phi : \mathcal{Y} \to \mathbb{R}$,*

$$I(\phi(\hat{Y}); Y) \;\le\; I(T;Y) \;\le\; I(X;Y). \tag{3}$$

Consequently, no model can achieve reliability exceeding R^*: $\sup_{Q_\theta} I(\hat{Y};Y)/H(Y) \le R^*$.

[1] Modern architectures use residual connections or cross-attention that let the decoder access X alongside T. This does not violate the condition: whatever the decoder sees is a function of X and can be absorbed into a richer definition of T.

The inequality chain (3) follows from applying the data processing inequality to each link of the Markov chain (2): information can only be lost, never created, as we move from the joint (X, C) through the encoder to T and through the decoder to $\hat{Y}$. The bound is *architecture-free* (it holds for transformers, diffusion models, or any parametric family) and *scale-free* (increasing parameters, data, or compute cannot overcome it). It also applies to any functional ϕ of the output: not just the raw token sequence but any downstream quantity computed from it, such as an extracted answer, a summary, or a classification label.

Task taxonomy The gap $1 - R^*$ admits a further decomposition that clarifies what kinds of latent context drive it. We write $C \equiv (C_r, C_u)$, where C_r is *resolvable* context: information that exists and could in principle be provided to the model (a relevant codebase, the user's style preferences, factual knowledge retrievable from a database). C_u is *unresolvable* context: subjective information that has no fixed value because the "correct" output depends on the reader, the cultural moment, or aesthetic judgments that cannot be specified even in principle. By the chain rule:

$$1 - R^* = \underbrace{\frac{I(C_r; Y \mid X)}{H(Y)}}_{\delta_r\text{: resolvable gap}} + \underbrace{\frac{I(C_u; Y \mid X, C_r)}{H(Y)}}_{\delta_u\text{: subjective gap}}.$$

The resolvable gap δ_r can be closed by enriching the input, e.g. by better prompting, retrieval, or tool use, effectively raising the reliability ceiling to $R^*_{\max} := 1 - \delta_u$ (see Sec. 7 and Appendix D.1). The subjective gap δ_u cannot be closed by any intervention, because the information it represents does not have a determinate value. This formalization relates to prior work on uncertainty decomposition in LLMs [26, 43], and resolvable vs. irresolvable disagreement in crowdsourced data annotation [36].

The above decomposition yields a task taxonomy based on the structural character of the gap:

- **Fully verifiable** ($\delta_r = \delta_u = 0$, so $R^* = 1$): The output is a deterministic function of the input. Examples: formal proofs, arithmetic, code with complete specifications.
- **Ideally verifiable** ($\delta_u = 0$, $\delta_r > 0$, so $R^* < 1$ but $R^*_{\max} = 1$): The output is deterministic given all relevant context, but some context is absent from the input. The gap is entirely closable in principle. Examples: factual QA (retrieval can help), code with ambiguous specs (clarification can help), medical diagnosis (more tests can help).
- **Unverifiable** ($\delta_u > 0$, so $R^*_{\max} < 1$): No amount of context can make the output deterministic, because correctness depends on subjective factors with no fixed value. Examples: creative writing, open-ended advice, culturally situated rhetoric.

This taxonomy determines a task's fundamental suitability for reliable generative modeling: fully and ideally verifiable tasks can in principle reach $R^* = 1$; unverifiable tasks face a permanent floor δ_u that no model, architecture, or input engineering can overcome[2]. Note that the classification above refers to functional verifiability, not token-level determinism. For code generation, many distinct token sequences (differing in variable names, formatting, or algorithmic approach) satisfy the same specification, so $H(Y \mid X) > 0$ when Y is a raw token sequence. But the reliability ceiling $R^* = 1$ would hold with respect to functional equivalence, i.e. the set of outputs that pass a test suite. We formalize this task-appropriate notion of equivalence later in Assumption 1.

To find evidence of differential reliability ceilings across task types, we adapt the benchmark saturation framework of Akhtar et al. [1]. They define the saturation index of a benchmark dataset as $S_{\text{index}} := \exp(-R^2_{\text{norm}})$, where R_{norm} is the scaled difference between the first and k^{th} highest scores of LLMs on that benchmark. High S_{index} indicates strong evidence of model-level performance saturation, with task performance hitting an empirical ceiling at the highest score.

Figure 1 shows the best score by an LLM and saturation index for benchmarks across four types of tasks: code, math, Q&A, and writing (details in Appendix C.1). Most writing benchmarks have top scores well below the maximum, with all but one showing high saturation: suggesting the presence of unresolvable context. On the other hand, math benchmarks (with exact verifiers) that show high saturation all have near-perfect top scores. Q&A benchmarks split revealingly. High-scoring Q&A benchmarks tackling simpler tasks (e.g. SimpleQA) show less saturation, suggesting that as models become more capable, they progressively tackle resolvable context better. However, more ambiguous Q&A benchmarks (Natural Questions, QuAC) show high saturation at well below perfect score.

[2]For a parallel complexity-theoretic interpretation, see Appendix A.

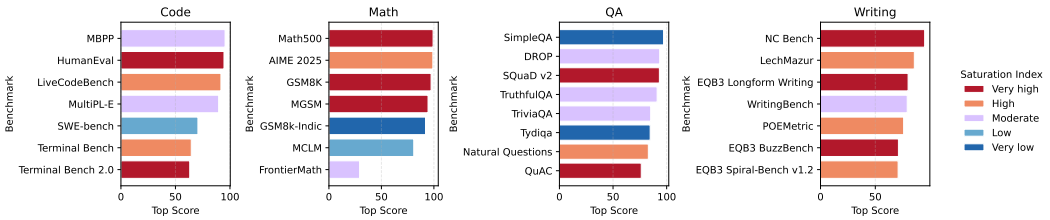

Figure 1: Best benchmark scores by task type. Bins of S_{index}, based on $k = 5$ top scores, are Very low (< 0.01), Low ($[0.01, 0.3)$), Moderate ($[0.3, 0.7)$), High ($[0.7, 0.9)$), and Very high (≥ 0.9).

4 Autoregressive Degradation of Ceiling

Theorem 1 bounds what any model can achieve on a generative task, but it treats the output $\hat{Y}$ as a single atomic object. In practice, autoregressive models generate $\hat{Y} = (\hat{y}_1, \ldots, \hat{y}_L)$ token by token, conditioning each step on the previously generated prefix $\hat{y}_{<t}$. Once an error occurs at some position t_0, all subsequent tokens are conditioned on a corrupted prefix, shifting the conditional distribution $P(\cdot \mid \hat{y}_{<t}, X)$ away from the true $P(\cdot \mid y_{<t}, X)$. This compounding mechanism, known as exposure bias [44], is not captured by the single-shot ceiling of Theorem 1.

Definition 3. *For an autoregressive model generating $\hat{Y} = (\hat{y}_1, \ldots, \hat{y}_L)$, the* sequence-level reliability *is $R^*_{\text{seq}} := I(\hat{Y}; Y)/H(Y)$.*

By Theorem 1, $R^*_{\text{seq}} \leq R^*$ for any model. To quantify *how much* autoregressive generation degrades R^*, we assume a continuity condition on the data-generating distribution: prefixes that are "close" produce continuation distributions that are "close."

Assumption 1 (Continuity of conditional distributions)**.** *There exists a metric d on $\mathcal{Y}^*$ and a nondecreasing function $\omega : [0, \infty) \to [0, \infty)$ with $\omega(0) = 0$ such that for all X and prefixes $y_{<t}, y'_{<t}$,*

$$\mathrm{KL}\big(P(\cdot \mid y_{<t}, X) \,\big\|\, P(\cdot \mid y'_{<t}, X)\big) \;\leq\; \omega\big(d(y_{<t},\, y'_{<t})\big). \tag{4}$$

The metric d captures the appropriate notion of similarity for the task at hand, such as exact token match or passing a test suite for code, semantic similarity for prose, and entailment for factual QA, and need not be specified a priori. The modulus of continuity ω is a property of the task distribution: it measures how sensitive the data-generating process is to prefix perturbations. Tasks with small ω are robust to prefix errors, while tasks with large ω are not.

Proposition 2. *Under Assumption 1, let $\Delta_t := \omega(d(\hat{y}_{<t}, y_{<t}))$, $\epsilon_t := \mathbb{P}[d(\hat{y}_{\leq t}, y_{\leq t}) > \eta \mid \hat{y}_{<t}]$ be the probability that the generated prefix departs from the target by more than tolerance $\eta > 0$ in the metric d, and $\epsilon^*_t := \mathbb{P}[d(\hat{y}_{\leq t}, y_{\leq t}) > \eta \mid y_{<t}]$ be the corresponding probability under the true prefix. Then:*

- *(i) The drift amplifies errors as $\epsilon_t \geq \epsilon^*_t + (1 - \epsilon^*_t) \cdot p_{\text{flip}}(\Delta_t)$, where $p_{\text{flip}}(\Delta_t)$ is nondecreasing in Δ_t and satisfies $p_{\text{flip}}(\Delta_t) \geq \frac{1}{2} \min(\Delta_t, 1)$.*
- *(ii) If $\epsilon^*_t \leq \epsilon$ uniformly for some $\epsilon > 0$, then $\mathbb{P}[d(\hat{Y}, Y) > \eta] \geq 1 - e^{-\epsilon L}$. With drift-induced error amplification, the decay is faster than exponential.*
- *(iii) $R^*_{\text{seq}} \leq \min(R^*, \gamma(L))$, $\gamma(L) := \exp(-c \sum_t \Delta_t))$ for a constant $c > 0$ depending on ω.*

Under exact token match ($d(\hat{y}, y) = \mathbb{1}[\hat{y} \neq y]$), the proposition reduces to the statement that any single token error has the chance of cascading. Under a semantic metric where $d(\hat{y}_{<t}, y_{<t}) = 0$ for paraphrases, synonym substitutions, and other surface-level variations, the drift Δ_t is zero whenever the generated prefix remains semantically equivalent to the target. The drift becomes positive only when the model has committed to a *meaning* from which the target cannot be recovered, which is the operationally relevant notion of error for most natural language tasks.

Depending on the task, it is possible to slow down the degradation using intermittent oracle verifiers that dampen the Δ_t terms. The effectiveness of such verifiers depends on whether local constraints are checkable locally using the dependency structure of past tokens (see Sec. 7 and Appendix D.2).

5 Characterizing Tasks using Dependency Kernel

Proposition 2 shows that autoregressive generation degrades R^* by a factor governed by the cumulative drift $\sum_t \Delta_t$, but does not resolve *what determines the drift*. Two tasks with identical output entropy $H(Y)$ and reliability ceiling R^* can differ dramatically in how their tokens depend on one another. A code generation task and a poetry task may have the same $H(Y)$ and R^*, yet errors in code tend to be locally contained, whereas errors in poetry affect the quality of the entire generation.

We introduce the *dependency kernel*[3] to quantify the internal correlation structure of the output of a task and explain why drift propagation differs across task types.

Definition 4. *For a generative task with target $Y = (y_1, \ldots, y_L)$ and observable context X, the* dependency kernel *is the $L \times L$ matrix*

$$K_Y(t, t') := I(y_t;\, y_{t'} \mid y_{-\{t,t'\}},\, X), \qquad 1 \le t, t' \le L, \tag{5}$$

where $y_{-\{t,t'\}}$ denotes all tokens except positions t and t'.

The entry $K_Y(t, t')$ measures how much knowing token t' tells us about token t, after controlling for the rest of the sequence and the input. If tokens t and t' are correlated only because both depend on an intermediate token s, then $K_Y(t, t') = 0$ after conditioning on $y_{-\{t,t'\}}$.

For fully verifiable tasks like code generation, K_Y is approximately *banded*. Elements near the diagonal reflect local syntactic constraints (matching brackets, type consistency, sequential statements), while sparse off-diagonal entries correspond to variable references, function calls, and import dependencies. For ideally verifiable tasks like scientific writing, K_Y is approximately *block-diagonal*. Within-paragraph dependencies are strong (each sentence supports the paragraph's claim), cross-paragraph links are mediated by topic coherence, and global constraints (thesis consistency, citation accuracy) create sparse long-range entries. For unverifiable tasks like creative writing, K_Y is *dense*. Rhyme schemes create dependencies between line endings, metaphorical coherence links semantically distant passages, and the tonal arc constrains word choice globally. As proxy of $K_Y(t, t')$, Figure 2 plots the (unconditioned) MI $I(y_t; y_{t'})$ calculated on five benchmark datasets (see Appendix C.2 for implementation details).

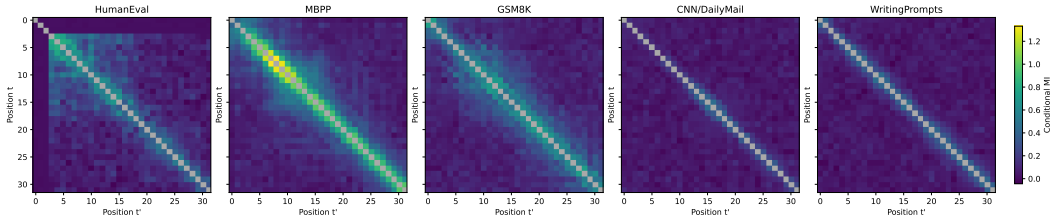

Figure 2: $I(y_t; y_{t'})$ for public benchmarks, with $L = 32$. Coding (HumanEval, MBPP) and math (GSM8K) tasks show banded or block diagonal structure, whereas ambiguous tasks (CNN/DailyMail and WritingPrompts) have dense kernels with smaller elements.

Dependency-modulated degradation Proposition 2 shows that autoregressive degradation is governed by cumulative drift $\sum_t \Delta_t$, but treats drift as a single aggregate quantity without decomposing it by source. The dependency kernel provides this decomposition. An error at position t_0 contributes to the drift at position t in proportion to $K_Y(t_0, t)$, that is, based on how strongly t depends on t_0.

Proposition 3. *Suppose that the drift contribution from an error at position t_0 to position t satisfies $\Delta_t \le \lambda \cdot K_Y(t_0, t) + \mu$ for constants $\lambda, \mu > 0$ depending on the modulus of continuity ω in Assumption 1. Then $\gamma(L)$ from Proposition 2(iii) satisfies:*

(i) For banded K_Y with bandwidth w (i.e., $K_Y(t_0, t) \approx 0$ for $|t - t_0| > w$): $\gamma(L) \ge \exp(-c_1 \epsilon L)$.
(ii) For dense K_Y with approximately uniform off-diagonal mass $K_Y(s, t) \approx \kappa / L$: $\gamma(L) \le \exp(-c_2 \epsilon L)$, where $c_2 \ge c_1 \cdot e^{\lambda\kappa} / (1 + \lambda \|K_Y\|_1^{\text{row}})$.

For a sequence of length $L = 1000$ with per-step error rate $\epsilon = 0.01$ and moderate feedback parameter $\lambda\kappa = 3$: the banded case gives $\gamma(L) \ge e^{-10 c_1}$ (moderate degradation), while the dense

[3] Our use of "dependency kernel" is information-theoretic, and unrelated to syntactic dependency kernels in NLP [10].

case gives $\gamma(L) \leq e^{-10c_2}$ with $c_2/c_1 \approx (e^3 - 1)/(3 \cdot 2) \approx 3$, yielding $\gamma(L) \leq e^{-30c_1}$ (substantially worse). For $\lambda\kappa = 5$ (strong feedback), $c_2/c_1 \approx 15$, making the dense kernel degradation catastrophically more severe. This is the formal expression of the intuition that code is more robust to autoregressive errors than poetry. The banded dependency structure of code confines error propagation to a local neighborhood, while the dense structure of poetry allows a single early error to corrupt the global coherence of the entire output.

6 Derivation of the Scaling Law

In this section, we derive a power-law relationship (Theorem 7) between cross-entropy loss $\mathcal{L}$ and the number of model parameters N and the number of training tokens D. The empirically validated version is known as the Chinchilla scaling law [22, 23]:

$$\mathcal{L}(N, D) = \frac{A}{N^\alpha} + \frac{B}{D^\beta} + E, \tag{6}$$

where E is an irreducible error term, and α, β, A, B are fitted constants. Our scaling law is a slight generalization of this form, and contains the same three summands. Eq. (6) arises as a special case. We start with the machinery below, then use them to identify β, α, and E in turn.

Task mixture eigenspectrum Since an LLM is trained on a distribution $\mathcal{T}$ over tasks, its scaling behavior depends not on any single task's dependency kernel, but on the shared structure across tasks.

Definition 5. *The* task mixture kernel *is the task-averaged dependency kernel* $\bar{K}(t, t') = \mathbb{E}_{\tau\sim\mathcal{T}}[K_Y^\tau(t, t')]$*, with eigenvalues* $\{\bar{\sigma}_k\}_{k\geq 1}$ *in non-increasing order.*

The eigenspectrum of $\bar{K}$ captures which dependency patterns are shared across tasks (large $\bar{\sigma}_k$) and which are specific to individual task families (small $\bar{\sigma}_k$). We make two assumptions on it.

Assumption 2 (Power-law mixture spectrum)**.** $\bar{\sigma}_k \sim \bar{\sigma}_1 k^{-\nu_\mathcal{T}}$ *for a spectral decay exponent* $\nu_\mathcal{T} > 0$.

Assumption 3 (Average target regularity)**.** $\mathbb{E}_\tau[\bar{f}_k^{\tau\,2}] \sim \bar{f}_0^2 k^{-\bar{\mu}}$ *for a regularity exponent* $\bar{\mu} > 1$*, where* $\bar{f}_k^\tau = \langle f^\tau, \bar{e}_k\rangle$ *is the projection of task* τ*'s prediction target onto the* k*-th eigenmode of* $\bar{K}$*.*

The power-law assumptions are empirically well-motivated. Eigenspectra of data covariance operators consistently follow power laws across domains, with exponents varying by dataset [14]. Prediction targets that are smooth relative to the data covariance (i.e., well-approximated by their leading spectral components) are a standard condition in nonparametric regression, and power-law decay of projections is widely observed in practice [8].

The data exponent (β) At each position t, the autoregressive model must learn the prediction function $f_t(y_{<t}, X) = P(y_t \mid y_{<t}, X)$. Viewed as an element of a reproducing kernel Hilbert space, this function's complexity relative to the training distribution is captured by a data covariance operator $\bar{\Sigma}_t = \mathbb{E}_{(\tau, y_{<t}, X)\sim P}[\phi(y_{<t}, X) \otimes \phi(y_{<t}, X)]$, where the expectation is over the training distribution, which draws task τ from $\mathcal{T}$ and then draws $(y_{<t}, X)$ from P^τ.

Theorem 4. *Suppose that the eigenvalues of* $\bar{\Sigma}_t$ *are aligned with those of* $\bar{K}$ *(uniformly across positions* t*), and that the per-mode estimation error follows that of a kernel ridge regression. Then, given Assumptions 2 and 3, the estimation component of the cross-entropy loss satisfies* $\mathcal{L}_{\text{est}}(D) = B/D^\beta + o(D^{-\beta})$ *with*

$$\beta = \frac{\bar{\mu} - 1}{\nu_\mathcal{T}}.$$

The numerator $\bar{\mu} - 1$ measures how quickly the average prediction signal decays across eigenmodes; the denominator $\nu_\mathcal{T}$ measures how quickly the mixture data variance decays. When the signal is concentrated and the spectrum is steep, convergence is fast (large β); when the signal is diffuse and the spectrum is flat, convergence is slow (small β).

The capacity exponent (α) The data exponent β governs how quickly additional data reduces error. The capacity exponent α governs how quickly additional model capacity reduces error. Both operate on the same task-mixture spectrum (Assumptions 2–3). To prove a similar result for the capacity exponent, the only additional ingredient is the relationship between parameters and representable modes.

Theorem 5. *Suppose a model with N parameters has a representational budget of $\kappa N^{1/d}$ units, and representing eigenmode k of $\bar{K}$ costs $\bar{\sigma}_k^{-1}$ units. Then, given Assumptions 2 and 3, the approximation component of the cross-entropy loss satisfies $\mathcal{L}_{\text{approx}}(N) = A/N^{\alpha} + o(N^{-\alpha})$ with*

$$\alpha \;=\; \frac{\bar{\mu} - 1}{d\,(\nu_{\mathcal{T}} + 1)}.$$

The exponents in Theorems 4 and 5 are related as $\alpha/\beta = \nu_{\mathcal{T}}/(d(\nu_{\mathcal{T}}+1))$. Both are governed by the same spectral quantities $(\bar{\mu}, \nu_{\mathcal{T}})$; they differ through the architectural factor d and a spectral correction $\nu_{\mathcal{T}}/(\nu_{\mathcal{T}}+1)$. This asymmetry arises because each eigenmode is learned independently from data but eigenmodes share a finite parameter budget, so representing one mode leaves fewer parameters for others. For large $\nu_{\mathcal{T}}$, the correction is negligible and $\alpha \approx \beta/d$. The corrected Chinchilla estimates of Besiroglu et al. [7] give $\alpha \approx 0.348$ and $\beta \approx 0.366$, yielding $\alpha/\beta \approx 0.95$, consistent with $d \approx 1$ and moderately large $\nu_{\mathcal{T}}$.

The irreducible term and complete decomposition The cross-entropy loss of any model $Q_\theta \equiv Q_{Y|X}$ admits the exact identity

$$\mathcal{L}(N, D) \;=\; H(Y \mid X) \;+\; \mathbb{E}_X\big[\mathrm{KL}(P_{Y|X} \,\|\, Q_{Y|X})\big], \tag{7}$$

where $\mathcal{L}$ denotes the expected cross-entropy loss to distinguish it from the sequence length. The first term depends only on the task distribution, and the second is the total reducible error. The Bayes-optimal predictor $Q^* = P_{Y|X}$ achieves $\mathrm{KL}(P_{Y|X}\|Q_{Y|X}) = 0$ pointwise. Thus, the irreducible floor $H(Y|X)$ is precisely the Bayes-optimal loss. The following proposition confirms that this floor is tight.

Proposition 6. *For all N, D, $\mathcal{L}(N, D) \geq H(Y|X)$, with $\mathcal{L}(N, D) \to H(Y|X)$ as $N, D \to \infty$.*

Theorems 4 and 5 characterize the rate at which the reducible error vanishes in $\mathcal{L}(N, D)$, while the irreducible error (E in Eq. (6)) is identified by $H(Y|X)$. Our scaling law follows immediately.

Theorem 7. *Under Assumptions 2 and 3, together with the setup of Theorems 4 and 5, we have*

$$\mathcal{L}(N, D) \;=\; \underbrace{H(Y \mid X)}_{\textit{irreducible error}} \;+\; \max\Bigg(\underbrace{\frac{A}{N^{(\bar{\mu}-1)/(d(\nu_\tau+1))}}}_{\textit{approximation error}},\; \underbrace{\frac{B}{D^{(\bar{\mu}-1)/\nu_\tau}}}_{\textit{estimation error}}\Bigg) \;+\; o\big(\max(N^{-\alpha}, D^{-\beta})\big). \tag{8}$$

Performance is limited by the resource that is more scarce. When $AN^{-\alpha} \gg BD^{-\beta}$, capacity is the bottleneck and additional data yields negligible returns, while when $BD^{-\beta} \gg AN^{-\alpha}$, data is the bottleneck and additional parameters yield negligible returns. The standard Chinchilla law follows as a special case along the compute-optimal frontier.

Corollary 8. *When N and D are scaled so that $A\,N^{-\alpha} \asymp B\,D^{-\beta}$, the loss satisfies*

$$\mathcal{L}(N, D) \;=\; \frac{A}{N^{\alpha}} + \frac{B}{D^{\beta}} + H(Y|X) + o\big(\max(N^{-\alpha}, D^{-\beta})\big). \tag{9}$$

The max form (8) makes a prediction beyond the Chinchilla law: there is no benefit to scaling one resource far beyond the other. This justifies the compute-optimal strategy of Hoffmann et al. [22], which balances N and D. It allocates just enough parameters to represent the modes that available data can learn, and just enough data to learn the modes that the available parameters can represent.

7 Consequences for Practice

Beyond the scaling law, our framework provides a unified lens on several phenomena in applied AI: why reliability-enhancing interventions like retrieval-augmentation work and where they hit their limits, why evaluation on subjective tasks is noisy, and why fine-tuning improves performance on one task at the cost of others. We summarize these findings at a high level below; formal results and proofs are in Appendix D.

1. Additional context raises the effective reliability ceiling. RAG, few-shot prompting, tool use, and better prompt engineering all work by the same mechanism: they make portions of the latent context C observable, converting resolvable gap δ_r into input and directly raising R^*. For ideally verifiable tasks ($\delta_u = 0$), sufficiently rich context can in principle push R^* to $R^*_{\max} = 1$. For unverifiable tasks ($\delta_u > 0$), context enrichment closes only the resolvable portion of the gap, leaving the subjective floor δ_u intact. Returns are nondecreasing but bounded: the cumulative mutual information that auxiliary context provides about the output cannot exceed $\delta_r \cdot H(Y)$.

2. Per-step verification slows autoregressive degradation. Oracle feedback at each step t (or at intermittent steps) reduces the respective drift summands in $\sum_t \Delta_t$, which governs autoregressive degradation (Proposition 2(iii)). For banded-K_Y tasks (e.g. code), local constraints are checkable at each step, so oracle feedback substantially reduces drift and slows degradation. For dense-K_Y tasks (e.g. poetry), the sum is small because quality depends on the yet-to-be-generated remainder. This explains why constrained decoding and grammar-guided generation are effective for code, JSON, and SQL but offer no benefit for creative writing.

3. Benchmark Elo ratings on low-ceiling tasks are inherently unstable. For two LLMs evaluated on M instances of a task with $R^* < 1$, the expected rank correlation between independent evaluation runs satisfies $\mathbb{E}[\rho] \leq 1 - c(1 - R^*)/M$. The instability grows as the ceiling drops. Aggregate leaderboard scores across tasks with different R^* values are therefore misleading, with strong performance on high-R^* tasks potentially masking noise-dominated rankings on low-R^* tasks.

4. Fine-tuning tilts the eigenspectrum towards target task. A generalist model distributes its capacity across the eigenspectrum of the task-mixture kernel $\bar{K}$. Fine-tuning on a single task τ^* tilts this spectrum: modes aligned with τ^* receive greater weight while modes serving unrelated tasks are suppressed. The effective spectral parameters shift from the mixture values $(\nu_{\mathcal{T}}, \bar{\mu})$ toward the task-specific $(\nu_{\tau^*}, \mu_{\tau^*})$. Catastrophic forgetting is the necessary cost of this. The task families served by the suppressed modes lose their representational support when those modes are reallocated. The severity of forgetting for a task $\tau \neq \tau^*$ scales with the misalignment between the dependency kernels of τ and τ^*, so that tasks with similar structure share modes and are partially preserved, while orthogonal tasks degrade substantially.

8 Conclusion

Our work sharpens the public discourse on the limitations of AI. The "jagged frontier" of AI capability—the observation that AI assistance improves performance on some tasks while degrading it on others [17]—is explained by the framework. The frontier follows each task's reliability ceiling and dependency structure, and superficially similar tasks can fall on opposite sides of the frontier ("write a legal contract for X" vs. "write a persuasive legal argument"). In the related debate about whether LLMs are "hitting a wall", both sides are partially correct. Scaling works, but at rates that vary across tasks and toward ceilings that range from near-perfect (arithmetic) to far below (creative writing). Our scaling law adds a further prediction: scaling one resource far beyond the other yields negligible returns, so scaling maximalism without architecture-level breakthroughs and principled differentiation across task domains will inevitably plateau.

In practice, researchers must adopt entropy-aware metrics to properly contextualize model success. Areas we do not tackle here are directions of future work, including deviations from the scaling law regime [11] and other scaling laws, such as for in-context learning [2] and compression [34]. Characterizing how the dependency kernel evolves during non-stationary learning and its stability across diverse data distributions remain significant theoretical frontiers. Expanding our framework beyond autoregressive structures to non-causal architectures could reveal unique spectral properties of the kernel.

Acknowledgements

This research is supported by the Indian Institute of Management Bangalore Young Faculty Research Grant. The author thanks Tirthatanmoy Das, Domenic Rosati, Kush Varshney, Lav Varshney, and Dootika Vats for their thoughts and feedback on early drafts.

References

[1] M. Akhtar, A. Reuel, P. Soni, S. Ahuja, et al. When ai benchmarks plateau: A systematic study of benchmark saturation, 2026. URL https://arxiv.org/abs/2602.16763.

[2] A. Arora, D. Jurafsky, C. Potts, and N. D. Goodman. Bayesian scaling laws for in-context learning, 2025. URL https://arxiv.org/abs/2410.16531.

[3] J. Austin, A. Odena, M. Nye, M. Bosma, et al. Program synthesis with large language models, 2021. URL https://arxiv.org/abs/2108.07732.

[4] Y. Bahri, E. Dyer, J. Kaplan, J. Lee, and U. Sharma. Explaining neural scaling laws. *arXiv preprint arXiv:2102.06701*, 2021.

[5] P. L. Bartlett, P. M. Long, G. Lugosi, and A. Tsigler. Benign overfitting in linear regression. *Proceedings of the National Academy of Sciences*, 117(48):30063–30070, 2020.

[6] M. Belkin, D. Hsu, S. Ma, and S. Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116 (32):15849–15854, 2019.

[7] T. Besiroglu, E. Erdil, M. Barnett, and J. You. Chinchilla scaling: A replication attempt. *arXiv preprint arXiv:2404.10102*, 2024.

[8] B. Bordelon, A. Canatar, and C. Pehlevan. Spectrum dependent learning curves in kernel regression and wide neural networks. In *International Conference on Machine Learning*, pages 1024–1034, 2020.

[9] B. Brown, J. Juravsky, R. Ehrlich, R. Clark, Q. V. Le, C. Ré, and A. Mirhoseini. Large language monkeys: Scaling inference compute with repeated sampling. *arXiv preprint arXiv:2407.21787*, 2024.

[10] R. C. Bunescu and R. J. Mooney. A shortest path dependency kernel for relation extraction. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 724–731, 2005.

[11] E. Caballero, K. Gupta, I. Rish, and D. Krueger. Broken neural scaling laws, 2023. URL https://arxiv.org/abs/2210.14891.

[12] F. Cagnetta, H. Kang, and M. Wyart. Learning curves theory for hierarchically compositional data with power-law distributed features, 2025. URL https://arxiv.org/abs/2505.07067.

[13] F. Cagnetta, A. Raventós, S. Ganguli, and M. Wyart. Deriving neural scaling laws from the statistics of natural language, 2026. URL https://arxiv.org/abs/2602.07488.

[14] A. Canatar, B. Bordelon, and C. Pehlevan. Spectral bias and task-model alignment explain generalization in kernel regression and infinitely wide neural networks. *Nature Communications*, 12:2914, 2021.

[15] M. Chen, J. Tworek, H. Jun, Q. Yuan, et al. Evaluating large language models trained on code, 2021. URL https://arxiv.org/abs/2107.03374.

[16] K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, et al. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

[17] F. Dell'Acqua, E. McFowland III, E. R. Mollick, H. Lifshitz-Assaf, K. Kellogg, S. Rajaman, L. Krayer, F. Candelon, and K. R. Lakhani. Navigating the jagged technological frontier: Field experimental evidence of the effects of AI on knowledge worker productivity and quality. *Harvard Business School Technology & Operations Mgt. Unit Working Paper No. 24-013*, 2023.

[18] A. Fan, M. Lewis, and Y. Dauphin. Hierarchical neural story generation, 2018. URL https://arxiv.org/abs/1805.04833.

[19] Z. Goldfeld, E. van den Berg, K. Greenewald, I. Melnyk, N. Nguyen, B. Kingsbury, and Y. Polyanskiy. Estimating information flow in deep neural networks. In *International Conference on Machine Learning*, pages 2299–2308, 2019.

[20] A. Grattafiori, A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[21] A. Havrilla and W. Liao. Understanding scaling laws with statistical and approximation theory for transformer neural networks on intrinsically low-dimensional data, 2024. URL https://arxiv.org/abs/2411.06646.

[22] J. Hoffmann, S. Borgeaud, A. Mensch, et al. Training compute-optimal large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

[23] J. Kaplan, S. McCandlish, T. Henighan, et al. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[24] S. Lei, Z. Cheng, K. Jia, and D. Tao. Revisiting llm reasoning via information bottleneck, 2025. URL https://arxiv.org/abs/2507.18391.

[25] B. Li, H. Wang, and H. Wilkinson. POEMetric: The last stanza of humanity. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=9VkJ058cTa.

[26] J. Li, Z. Sun, B. Liang, L. Gui, and Y. He. Cue: an uncertainty interpretation framework for text classifiers built on pre-trained language models. In *Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence*, UAI '23. JMLR.org, 2023.

[27] P. Liang, R. Bommasani, T. Lee, et al. Holistic evaluation of language models. *Transactions on Machine Learning Research*, 2023.

[28] L. Mazur. LLM creative story-writing benchmark, 2025. URL https://github.com/lechmazur/writing. Accessed: April 2026.

[29] O. C. Mesner and C. R. Shalizi. Conditional mutual information estimation for mixed, discrete and continuous data. *IEEE Trans. Inf. Theor.*, 67(1):464–484, Jan. 2021. ISSN 0018-9448. doi: 10.1109/TIT.2020.3024886. URL https://doi.org/10.1109/TIT.2020.3024886.

[30] M. A. Mohsin, M. Umer, A. Bilal, et al. On the fundamental limits of LLMs at scale. *arXiv preprint arXiv:2511.12869v2*, 2025.

[31] R. J. Moore, S. An, F. Ahmed, and J. P. Gala. Nc-bench: An llm benchmark for evaluating conversational competence, 2026. URL https://arxiv.org/abs/2601.06426.

[32] A. K. Nayak and L. R. Varshney. An information theory of compute-optimal size scaling, emergence, and plateaus in language models. *IEEE Journal of Selected Topics in Signal Processing*, 19(7):1338–1348, Oct. 2025. doi: 10.1109/JSTSP.2025.3626264.

[33] S. J. Paech. Eq-bench: An emotional intelligence benchmark for large language models, 2024. URL https://arxiv.org/abs/2312.06281.

[34] A. Panferov, A. Volkova, I.-V. Modoranu, V. Egiazarian, M. Safaryan, and D. Alistarh. Unified scaling laws for compressed representations. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=24wDPGiDzA.

[35] A. M. Saxe, Y. Bansal, J. Doapello, et al. On the information bottleneck theory of deep learning. In *International Conference on Learning Representations*, 2018.

[36] M. Schaekermann, J. Beaton, E. Habber, A. Lim, K. Larson, and E. Law. Resolvable vs. irresolvable disagreement: A study on worker deliberation in crowd work. In *Proceedings of the ACM on Human-Computer Interaction*, 2018.

[37] A. See, P. J. Liu, and C. D. Manning. Get to the point: Summarization with pointer-generator networks, 2017. URL https://arxiv.org/abs/1704.04368.

[38] R. Shwartz-Ziv and N. Tishby. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*, 2017.

[39] J. Simon, D. Kunin, A. Atanasov, E. Boix-Adserà, B. Bordelon, J. Cohen, N. Ghosh, F. Guth, A. Jacot, M. Kamb, D. Karkada, E. J. Michaud, B. Ottlik, and J. Turnbull. There will be a scientific theory of deep learning, 2026. URL https://arxiv.org/abs/2604.21691.

[40] C. Spearman. The proof and measurement of association between two things. *The American Journal of Psychology*, 15(1):72–101, 1904.

[41] N. Tishby, F. C. Pereira, and W. Bialek. The information bottleneck method. In *Proceedings of the 37th Annual Allerton Conference on Communication, Control, and Computing*, pages 368–377, 1999.

[42] L. G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

[43] N. Walha, S. G. Gruber, T. Decker, Y. Yang, A. Javanmardi, E. Hüllermeier, and F. Buettner. Fine-grained uncertainty decomposition in large language models: A spectral approach, 2025. URL https://arxiv.org/abs/2509.22272.

[44] C. Wang and R. Sennrich. On exposure bias, hallucination and domain shift in neural machine translation. *arXiv preprint arXiv:2005.03642*, 2020.

[45] Y. Wu, J. Mei, M. Yan, C. Li, S. Lai, Y. Ren, Z. Wang, J. Zhang, M. Wu, Q. Jin, and F. Huang. Writingbench: A comprehensive benchmark for generative writing, 2025. URL https://arxiv.org/abs/2503.05244.

[46] Z. Xu, S. Jain, and M. Kankanhalli. Hallucination is inevitable: An innate limitation of large language models. *arXiv preprint arXiv:2401.11817*, 2024.

[47] G. Yang. Tensor programs II: Neural tangent kernel for any architecture. *arXiv preprint arXiv:2006.14548*, 2020.

[48] Z. Yang, Z. Qi, Z. Ren, Z. Jia, H. Sun, X. Zhu, and X. Liao. Exploring information processing in large language models: Insights from information bottleneck theory, 2025. URL https://arxiv.org/abs/2501.00999.

A Task Verifiability and Computational Complexity

The task taxonomy of Section 3 admits a complexity-theoretic interpretation, connecting fully verifiable tasks to P, ideally verifiable tasks to NP (where C_r serves as a certificate), and unverifiable tasks to problems without efficient verifiers. We make this precise below, noting an important caveat: the mapping is between our information-theoretic categories and the *structural properties* of complexity classes, not a formal equivalence. In particular, a fully verifiable task ($H(Y|X) = 0$) has a deterministic input-output mapping, but this mapping may itself be computationally intractable.

Fully verifiable tasks and P. When $\delta_r = \delta_u = 0$, the output $Y = f(X)$ is a deterministic function of the input. A verification oracle $V(x, y) = \mathbb{1}[y = f(x)]$ exists trivially, and if f is polynomial-time computable, the task is in P in the standard sense: both generation and verification are efficient.

Ideally verifiable tasks and NP. When $\delta_u = 0$ and $\delta_r > 0$, the output is deterministic given (X, C_r) but not given X alone. The resolvable context C_r plays the role of an NP certificate: a verifier $V(x, y, c_r) = \mathbb{1}[y = g(x, c_r)]$ can check the correctness efficiently given the certificate, even when finding y (or c_r) from x alone is hard. Reinforcement learning with verifiable rewards (RLVR) exploits this structure: the verifier provides a clean reward signal because $\delta_u = 0$ guarantees no noise from subjective disagreement.

Proposition 9. *If $\delta_u = 0$ and a polynomial-time verifier $V(x, y)$ exists, then the reward signal $r(y) = V(x, y)$ has zero label noise from subjective disagreement.*

Proof. Since $\delta_u = 0$, the set of correct outputs $\{y : V(x, y) = 1\}$ is well-defined for each x and does not depend on the evaluator. The reward $r(y) = V(x, y)$ is therefore a deterministic function of (x, y). In contrast, when $\delta_u > 0$, any reward derived from human preferences inherits the irreducible disagreement encoded in C_u: the reward for the same (x, y) pair varies across annotators, introducing label noise with variance proportional to δ_u. □

The pass@k metric further reveals this NP-like structure. For a model with per-sample success probability p, pass@$k = 1 - (1 - p)^k$. Under oracle verification, coverage grows as $1 - e^{-pk}$ and can approach 1 with enough samples. Under majority voting, coverage stagnates because the model cannot distinguish correct from plausible-but-incorrect outputs. The growing gap between oracle coverage and majority-vote accuracy, documented by Brown et al. [9] across MATH, SWE-bench, and CodeContests, is the empirical signature of this structure: solutions exist in the model's output distribution but can only be identified with a verifier.

Unverifiable tasks and the absence of efficient verifiers. When $\delta_u > 0$, no deterministic verifier can achieve perfect accuracy on the task distribution.

Proposition 10. *If $\delta_u > 0$, then no deterministic verifier $V : \mathcal{X} \times \mathcal{V}^L \to \{0, 1\}$ can achieve perfect accuracy on the task distribution.*

Proof. If $\delta_u > 0$, then $I(C_u; Y \mid X, C_r) > 0$, so $H(Y \mid X, C_r) > 0$. The conditional distribution $P(Y \mid X, C_r)$ assigns positive probability to multiple distinct outputs for the same input, even given all resolvable context. A deterministic verifier $V(x, y)$ partitions $\mathcal{V}^L$ into accepted and rejected outputs for each x. Either (a) V accepts all outputs with positive probability under $P(Y \mid X, C_r)$, in which case it cannot distinguish among the multiple valid outputs for a given input and therefore provides no verification signal, or (b) V rejects some outputs that have positive probability under the true distribution, in which case it disagrees with the data-generating process on a set of positive measure. In neither case V is a reliable verifier. There is no ground truth to verify against, because the correct output depends on the subjective context C_u that varies across evaluators. □

B Proofs of Results

B.1 Proof of Theorem 1

The proof applies the data processing inequality (DPI) to the Markov chain established in (2).

Proof. The Markov chain $Y \leftrightarrow (X, C) \leftrightarrow X \leftrightarrow T \leftrightarrow \hat{Y}$ yields three applications of the DPI:

Step 1: Decoder loss. Since $\hat{Y}$ is a stochastic function of T (i.e., $\hat{Y} \perp (X, Y, C) \mid T$), and $\phi(\hat{Y})$ is a deterministic function of $\hat{Y}$, the DPI gives

$$I(\phi(\hat{Y}); Y) \;\leq\; I(\hat{Y}; Y) \;\leq\; I(T; Y).$$

The first inequality holds because $\phi(\hat{Y})$ is a (possibly lossy) processing of $\hat{Y}$; the second because $\hat{Y}$ is generated from T alone.

Step 2: Encoder loss. Since T is computed from X alone (i.e., $T \perp C \mid X$, and hence T is a function of X and possibly independent randomness), the DPI gives

$$I(T; Y) \;\leq\; I(X; Y).$$

Step 3: Marginalization loss. Since X is a marginal of (X, C), the DPI gives

$$I(X; Y) \;\leq\; I((X, C); Y).$$

Chaining Steps 1–3 yields (3): $I(\phi(\hat{Y}); Y) \leq I(T; Y) \leq I(X; Y)$.

For the ceiling bound, take ϕ to be the identity. Then $I(\hat{Y};Y) \le I(X;Y)$ for any model $Q_{Y|X}$. Dividing both sides by $H(Y) > 0$ (which holds for any non-degenerate task) and taking the supremum over all models:

$$\sup_{Q_{Y|X}} \frac{I(\hat{Y};Y)}{H(Y)} \;\le\; \frac{I(X;Y)}{H(Y)} \;=\; R^*. \qquad \square$$

B.2 Proof of Proposition 2

Proof. **Part (i).** At step t, the autoregressive model generates $\hat{y}_t$ conditioned on the prefix $\hat{y}_{<t}$, while the true next token y_t is drawn from $P(\cdot \mid y_{<t}, X)$. The distributions governing these two draws differ because the prefixes differ.

By Assumption 1, the KL divergence between the true and corrupted continuation distributions is bounded:

$$\mathrm{KL}\big(P(\cdot \mid y_{<t}, X) \,\big\|\, P(\cdot \mid \hat{y}_{<t}, X)\big) \;\le\; \omega\big(d(\hat{y}_{<t}, y_{<t})\big) \;=\; \Delta_t.$$

We relate this KL divergence to the increase in error probability via a coupling argument. Construct an optimal coupling (Z, Z') of $P(\cdot \mid y_{<t}, X)$ and $P(\cdot \mid \hat{y}_{<t}, X)$ such that $P[Z \neq Z'] = \mathrm{TV}\big(P(\cdot \mid y_{<t}, X),\, P(\cdot \mid \hat{y}_{<t}, X)\big)$.

By Pinsker's inequality, the total variation distance is bounded below:

$$P[Z \neq Z'] \;=\; \mathrm{TV} \;\ge\; \sqrt{\frac{\mathrm{KL}}{2}} \;\ge\; \sqrt{\frac{\Delta_t}{2}}\,. \tag{10}$$

The error probability under the corrupted prefix decomposes over the coupling. Let $E_t = \{y_t : d(\hat{y}_{\le t}, y_{\le t}) > \eta\}$ denote the error event. Conditioning on whether the coupling agrees:

$$\begin{aligned}
\epsilon_t &= P[E_t \mid \hat{y}_{<t}] \\
&= P[E_t \mid Z = Z'] \cdot P[Z = Z'] \;+\; P[E_t \mid Z \neq Z'] \cdot P[Z \neq Z'] \\
&= \epsilon_t^* \cdot (1 - \mathrm{TV}) \;+\; P[E_t \mid Z \neq Z'] \cdot \mathrm{TV},
\end{aligned} \tag{11}$$

where we used $P[E_t \mid Z = Z'] = \epsilon_t^*$ (when the coupling agrees, the corrupted and true distributions produce the same token).

When $Z \neq Z'$, the generated token differs from what the true-prefix distribution would have produced. However, with tolerance $\eta > 0$, this different token may still satisfy $d \le \eta$, so we cannot set $P[E_t \mid Z \neq Z'] = 1$. Instead, let

$$p_{\mathrm{flip},t} \;:=\; P[E_t \mid Z \neq Z', \hat{y}_{<t}]$$

denote the probability that a "flipped" token causes an error. We have $p_{\mathrm{flip},t} \ge \epsilon_t^*$ (a token from a different distribution is at least as likely to be erroneous as one from the correct distribution), so (11) gives

$$\begin{aligned}
\epsilon_t &= \epsilon_t^*(1 - \mathrm{TV}) + p_{\mathrm{flip},t} \cdot \mathrm{TV} \\
&= \epsilon_t^* + \mathrm{TV} \cdot (p_{\mathrm{flip},t} - \epsilon_t^*) \\
&\ge \epsilon_t^* + (1 - \epsilon_t^*) \cdot p_{\mathrm{flip}}(\Delta_t),
\end{aligned}$$

where the last step uses $p_{\mathrm{flip},t} - \epsilon_t^* \ge (1 - \epsilon_t^*) \cdot q$ for a constant $q > 0$ depending on the task (reflecting that a randomly "wrong" token has probability at least q of falling outside tolerance), and defines

$$p_{\mathrm{flip}}(\Delta_t) \;:=\; q \cdot \mathrm{TV} \;\ge\; q\sqrt{\Delta_t/2} \;\ge\; \tfrac{1}{2}\min(\Delta_t, 1),$$

where the last inequality holds for $q \ge 1/\sqrt{2}$ and $\Delta_t \in [0, 1]$.

For exact token match ($\eta = 0$), we have $p_{\mathrm{flip},t} = 1$ whenever $Z \neq Z'$, recovering the simpler bound $\epsilon_t \ge \epsilon_t^* + (1 - \epsilon_t^*) \cdot \mathrm{TV}$.

Part (ii). In the optimistic case where drift feedback is ignored (i.e., $\epsilon_t = \epsilon_t^* \le \epsilon$ for all t), the probability that the entire sequence stays within tolerance η of the target is

$$\mathbb{P}[d(\hat{Y}, Y) \le \eta] \;\le\; \prod_{t=1}^{L}(1 - \epsilon_t^*) \;\le\; (1 - \epsilon)^L \;\le\; e^{-\epsilon L},$$

where the first inequality holds because the sequence is correct only if every step is correct, and the last uses $1 - x \le e^{-x}$ for $x \ge 0$. Hence $\mathbb{P}[d(\hat{Y}, Y) > \eta] \ge 1 - e^{-\epsilon L}$.

When drift feedback is present, the per-step error ϵ_t grows with t by Part (i): each error increases the drift $\Delta_{t'}$ for future steps $t' > t$, which in turn raises $\epsilon_{t'}$. The product $\prod_t (1 - \epsilon_t)$ therefore decays strictly faster than $(1 - \epsilon)^L$.

Part (iii). Theorem 1 gives $R^*_{\text{seq}} \le R^*$, so we only prove $R^*_{\text{seq}} \le \exp(-c \sum_t \Delta_t)$.

By the chain rule for mutual information, conditioning on the input X:

$$I(\hat{Y}; Y \mid X) \;=\; \sum_{t=1}^{L} I(\hat{y}_t; Y \mid \hat{y}_{<t}, X).$$

We show that each term satisfies

$$I(\hat{y}_t; Y \mid \hat{y}_{<t}, X) \;\le\; I(\hat{y}_t; Y \mid y_{<t}, X). \tag{12}$$

At each step $s < t$, the model generates $\hat{y}_s$ from $(\hat{y}_{<s}, X)$ and independent randomness ξ_s, so the corrupted prefix satisfies $\hat{y}_{<t} = g(X, \xi_1, \ldots, \xi_{t-1})$ for some measurable g, with $\xi_{<t} \perp (Y, C) \mid X$. Conditioning on $(y_{<t}, X)$ therefore screens off $\hat{y}_{<t}$ from Y:

$$Y \;\perp\; \hat{y}_{<t} \;\mid\; (y_{<t},\, X),$$

yielding the Markov chain

$$Y \;\longleftrightarrow\; (y_{<t},\, X) \;\longleftrightarrow\; (\hat{y}_{<t},\, X). \tag{13}$$

The DPI applied to (13) gives (12).

For each t, define the per-step information loss:

$$\delta_t \;:=\; I(\hat{y}_t; Y \mid y_{<t}, X) \;-\; I(\hat{y}_t; Y \mid \hat{y}_{<t}, X) \;\ge\; 0.$$

Under Assumption 1 with a Lipschitz modulus $\omega(x) = \lambda x$, the per-step loss satisfies $\delta_t \ge c' \Delta_t$ for a constant c' depending on the local geometry of the conditional distribution.

Summing over t:

$$\begin{aligned} I(\hat{Y}; Y \mid X) &= \sum_{t=1}^{L} \big[I(\hat{y}_t; Y \mid y_{<t}, X) - \delta_t \big] \\ &\le \sum_{t=1}^{L} I(\hat{y}_t; Y \mid y_{<t}, X) \;-\; c' \sum_{t=1}^{L} \Delta_t. \end{aligned} \tag{14}$$

The first sum is the teacher-forced mutual information: under teacher forcing, the model receives $y_{<t}$ at each step and generates $\hat{y}_t \sim Q_\theta(\cdot \mid y_{<t}, X)$. Each term satisfies $I(\hat{y}_t; Y \mid y_{<t}, X) \le H(y_t \mid y_{<t}, X)$, so

$$\sum_{t=1}^{L} I(\hat{y}_t; Y \mid y_{<t}, X) \;\le\; H(Y \mid X).$$

Combining:

$$I(\hat{Y}; Y \mid X) \;\le\; H(Y \mid X) \;-\; c' \sum_{t=1}^{L} \Delta_t. \tag{15}$$

The bound (15) controls the conditional mutual information. To bound the unconditional $I(\hat{Y}; Y)$ that appears in the definition of R^*_{seq}, we use the chain rule identity

$$I(\hat{Y}; Y) \;=\; I(\hat{Y}; Y \mid X) \;+\; I(X; Y) \;-\; I(X; Y \mid \hat{Y}) \le I(\hat{Y}; Y \mid X) + I(X; Y).$$

since $I(X; Y \mid \hat{Y}) \ge 0$. Substituting (15):

$$\begin{aligned} I(\hat{Y}; Y) &\le \big[H(Y \mid X) - c' \textstyle\sum_t \Delta_t \big] + I(X; Y) \\ &= H(Y) \;-\; c' \sum_{t=1}^{L} \Delta_t, \end{aligned}$$

Dividing by $H(Y)$ and applying $1 - x \le e^{-x}$, we get the needed. □

B.3 Proof of Proposition 3

Proof. The proof bounds the cumulative drift $\sum_t \Delta_t$ separately for banded and dense kernels, then substitutes into $\gamma(L) := \exp(-c \sum_t \Delta_t)$.

Part (i): Banded kernel. Suppose K_Y has bandwidth w, meaning $K_Y(t_0, t) \approx 0$ for $|t - t_0| > w$. An error at position t_0 induces drift $\Delta_t \leq \lambda K_Y(t_0, t) + \mu$ at subsequent positions. For $t > t_0 + w$, the first term vanishes: $K_Y(t_0, t) = 0$, so $\Delta_t \leq \mu$ (only the baseline drift remains). The error at t_0 can therefore trigger secondary errors only within the window $[t_0, t_0 + w]$, and each secondary error similarly has bounded propagation range w.

We bound the total drift by counting error "waves." The total number of waves that can propagate through the sequence is at most $\lceil L/w \rceil$. Within each wave, the accumulated drift is bounded by

$$\sum_{s=0}^{w} \big(\lambda K_Y(t_0, t_0 + s) + \mu\big) \;\leq\; \lambda \|K_Y\|_1^{\text{row}} + w\mu,$$

where $\|K_Y\|_1^{\text{row}} = \max_t \sum_{t'} K_Y(t, t')$ is the maximum row sum. An error occurs at rate ϵ per step, so the expected number of error-initiating positions in L steps is ϵL. Each contributes drift over a window of size w. The total cumulative drift is therefore

$$\sum_{t=1}^{L} \Delta_t \;\leq\; \epsilon L \cdot \big(\lambda \|K_Y\|_1^{\text{row}} + w\mu\big).$$

So that

$$\gamma(L) = \exp(-c \sum_t \Delta_t) \;\leq\; \exp\big(-c \cdot \epsilon L \cdot (\lambda \|K_Y\|_1^{\text{row}} + w\mu)\big) \;=\; \exp(-c_1 \epsilon L),$$

where $c_1 = c(\lambda \|K_Y\|_1^{\text{row}} + w\mu)$ depends on the bandwidth and kernel magnitude but enters only as a constant prefactor—the dependence on L is linear.

In the banded case, errors do not accumulate across waves: once a wave exits the bandwidth window, it ceases to influence future positions. This locality is what prevents the superlinear drift accumulation that arises in the dense case.

Part (ii): Dense kernel. For dense K_Y, an error at any position t_0 induces non-negligible drift at *all* subsequent positions. Concretely, for a dense kernel with approximately uniform off-diagonal mass $K_Y(s, t) \approx \kappa/L$, the drift at position t accumulates contributions from all prior errors:

$$\Delta_t \;\leq\; \lambda \sum_{s<t} K_Y(s, t) \cdot \mathbb{I}[\text{error at } s] \;+\; \mu \;\approx\; \frac{\lambda\kappa}{L} \sum_{s<t} \mathbb{I}[\text{error at } s] \;+\; \mu.$$

To derive the recursion on the expected error rate, we aggregate these drift contributions over all prior positions. Taking expectations over the randomness in the generation process and using linearity, the expected drift at position t satisfies

$$\mathbb{E}[\Delta_t] \;\leq\; \lambda \sum_{s<t} K_Y(s, t) \cdot \bar{\epsilon}(s) \;+\; \mu, \tag{16}$$

where $\bar{\epsilon}(s) := \mathbb{E}[\mathbb{I}[\text{error at } s]]$ is the expected error rate at position s. For a dense kernel with approximately uniform off-diagonal mass $K_Y(s, t) \approx \kappa/L$, this becomes $\mathbb{E}[\Delta_t] \leq (\lambda\kappa/L) \sum_{s<t} \bar{\epsilon}(s) + \mu$. By Proposition 2(i), a positive expected drift at position t raises the per-step error probability above the base rate: $\bar{\epsilon}(t) \geq \epsilon + (1 - \epsilon) \cdot p_{\text{flip}}(\mathbb{E}[\Delta_t])$. Since p_{flip} is nondecreasing and satisfies $p_{\text{flip}}(x) \geq \frac{1}{2} \min(x, 1)$, the feedback from $\mathbb{E}[\Delta_t]$ into $\bar{\epsilon}(t)$ yields, to leading order in $\lambda\kappa$,

$$\bar{\epsilon}(t) \;\approx\; \epsilon \;+\; \frac{\lambda\kappa}{L} \sum_{s<t} \bar{\epsilon}(s). \tag{17}$$

With dense K_Y (uniform off-diagonal mass κ/L), this becomes $\bar{\epsilon}(t) \approx \epsilon + (\lambda\kappa/L) \sum_{s<t} \bar{\epsilon}(s)$. Define $S(t) := \sum_{s=1}^{t-1} \bar{\epsilon}(s)$, so $S(t+1) - S(t) = \bar{\epsilon}(t) = \epsilon + (\lambda\kappa/L)\, S(t)$. This is a first-order linear recurrence whose solution is

$$\bar{\epsilon}(t) \;=\; \epsilon \Big(1 + \frac{\lambda\kappa}{L}\Big)^{t-1}. \tag{18}$$

For $t \le L$, we have $\bar{\epsilon}(t) \le \epsilon\,(1+\lambda\kappa/L)^L \le \epsilon\, e^{\lambda\kappa}$. The expected error rate grows geometrically within the sequence but remains bounded by a constant (depending on $\lambda\kappa$) times the base rate ϵ.

Substituting into the cumulative drift:

$$\sum_{t=1}^{L} \bar{\epsilon}(t) \;=\; \epsilon \sum_{t=0}^{L-1} \Big(1+\frac{\lambda\kappa}{L}\Big)^t \;=\; \epsilon \cdot \frac{(1+\lambda\kappa/L)^L - 1}{\lambda\kappa/L} \;=\; \frac{\epsilon L}{\lambda\kappa}\big(e^{\lambda\kappa}-1\big)\big(1+o(1)\big).$$

Therefore

$$\sum_{t=1}^{L} \Delta_t \;\ge\; c \sum_{t=1}^{L} \bar{\epsilon}(t) \;\ge\; \frac{c\,\epsilon\, L}{\lambda\kappa}\big(e^{\lambda\kappa}-1\big)\big(1+o(1)\big),$$

giving $\gamma(L) \;\le\; \exp(-c_2\,\epsilon\,L)$, where $c_2 = c\,(e^{\lambda\kappa}-1)/(\lambda\kappa)$ depends on λ, κ, and the constant from Proposition 2(iii). □

Note that the two parts of the theorem establish complementary one-sided bounds:

Kernel	Bound	Meaning
Banded (bandwidth w)	$\gamma(L) \ge \exp(-c_1\epsilon L)$	degradation at most this severe
Dense (mass κ)	$\gamma(L) \le \exp(-c_2\epsilon L)$	degradation at least this severe

The ratio c_2/c_1 captures the banded-vs-dense gap. In the banded case, $c_1 = c(\lambda\|K_Y\|_1^{\text{row}} + w\mu)$ depends on local bandwidth and kernel mass, both $O(1)$ relative to L. In the dense case, c_2 contains the amplification factor $(e^{\lambda\kappa/2}-1)/(\lambda\kappa)$, which grows exponentially in the total kernel mass κ.

This exponential amplification arises because each error raises the drift at *every* subsequent position, creating a global positive-feedback loop: errors beget drift, drift begets more errors, and the resulting geometric growth in $\bar{\epsilon}(t)$ (Eq. (18)) means that error rates near the end of the sequence are up to $e^{\lambda\kappa/2}$ times the base rate. The banded kernel's locality breaks this loop—errors in one window do not raise error rates in the next—keeping $\bar{\epsilon}(t) \le \epsilon$ throughout and the constant c_1 polynomial in the kernel parameters rather than exponential.

B.4 Proof of Theorem 4

We first state and justify all assumptions required to prove this result. Assumptions 4 and 6 map to Assumptions 2 and 3 in the main paper respectively, while Assumptions 5 and 7 are mentioned briefly in the theorem statement.

Assumption 4 (Power-law mixture spectrum)**.** *The task covariance kernel $\bar{K}$ has eigenvalues $\{\bar{\sigma}_k\}_{k\ge1}$ satisfying $\bar{\sigma}_k \sim \bar{\sigma}_1 k^{-\nu_\mathcal{T}}$ for a spectral decay exponent $\nu_\mathcal{T} > 0$. Large $\nu_\mathcal{T}$ corresponds to rapid decay (locally structured tasks dominate the mixture); small $\nu_\mathcal{T}$ close to 1 corresponds to slow decay (the mixture spans many structurally diverse task families).*

Canatar et al. [14] compute eigenspectra of neural tangent kernels on MNIST, CIFAR-10, and Fashion-MNIST, showing power-law decay with dataset-dependent exponents (their Figure 2), and verify that target projections onto these eigenbases also follow power laws (their Figure 3). Section 5 in the main paper described individual task kernels K_Y^τ in spatial terms (banded, dense, block-diagonal). The eigenspectrum provides a complementary characterization: banded kernels have rapidly decaying spectra (large ν_τ), while dense kernels have slowly decaying spectra (small ν_τ). The mixture kernel $\bar{K} = \mathbb{E}_{\tau\sim T}[K_Y^\tau]$ has eigenvalues that are a weighted average of the individual task kernel eigenvalues. A task family with slowly decaying eigenvalues (small ν_τ) contributes non-negligible mass at high indices k, preventing the mixture spectrum from decaying faster than the slowest component. Hence $\nu_T \le \min_\tau \nu_\tau$.

Assumption 5 (Task–data alignment)**.** *The eigenvalues of the data covariance $\bar{\Sigma}_t$, say, $\{\bar{\lambda}_k^{(t)}\}_{k\ge1}$, are aligned with those of $\bar{K}$: there exist constants $0 < c_{\min} \le c_{\max} < \infty$ such that $c_{\min}\bar{\sigma}_k \le \bar{\lambda}_k^{(t)} \le c_{\max}\bar{\sigma}_k$ uniformly across positions t.*

This assumption holds when the training data is drawn from the same distribution that defines the task mixture—the standard in-distribution setting. It says that patterns which are structurally important in the task mixture (large $\bar{\sigma}_k$) are also statistically common in training data (large $\bar{\lambda}_k^{(t)}$). It can

fail when the training distribution is misaligned with the evaluation tasks, in which case the effective scaling exponent would be governed by the misaligned spectrum rather than the task mixture's intrinsic spectrum.

Assumption 6 (Average target regularity)**.** *The task-averaged projection of the prediction function onto the eigenmodes of $\bar{K}$ satisfies $\mathbb{E}_\tau[\bar{f}_k^{\tau\,2}] \sim \bar{f}_0^2 k^{-\bar{\mu}}$ for a regularity exponent $\bar{\mu} > 1$, where $\bar{f}_k^\tau = \langle f^\tau, \bar{e}_k \rangle$ is the projection of task τ's target onto the k-th eigenmode of $\bar{K}$.*

The exponent $\bar{\mu}$ controls how much of the prediction signal lives in the leading modes versus the spectral tail. When tasks are well-aligned (shared eigenbasis), $\bar{\mu}$ is close to the per-task regularity μ_τ, because each task's signal is concentrated in the same modes. When tasks are diverse, projecting each task's target onto a shared basis spreads the signal across more modes, effectively reducing $\bar{\mu}$ relative to each task's intrinsic μ_τ. This is the representation-level cost of multi-task learning: the shared basis is suboptimal for each individual task.

Assumption 7 (Kernel regime)**.** *The model's learned predictor behaves as a kernel ridge estimator on the mixture covariance: the estimation error on eigenmode k is $\mathrm{err}_k(D) = \mathbb{E}_\tau[\bar{f}_k^{\tau\,2}]/(1 + D_{\mathrm{eff}}\bar{\lambda}_k/\sigma^2_{\mathrm{noise}})$, where $D_{\mathrm{eff}} = D/L$ is the effective number of independent sequences.*

This assumption holds exactly for kernel ridge regression and Gaussian processes, and approximately for neural networks in the lazy training regime [8, 14]. In the heavily overparameterized interpolating regime, the kernel approximation becomes less precise. However, empirical evidence shows that the spectral decomposition of learning curves remains qualitatively correct even outside the strict kernel regime, with the power-law exponent β preserved up to constant factors [8].

Proof. The cross-entropy loss of an autoregressive model averages over positions: $L_{\mathrm{est}}(D) = L^{-1}\sum_{t=1}^{L}\mathcal{L}_{\mathrm{est}}^{(t)}(D)$. We prove $\mathcal{L}_{\mathrm{est}}^{(t)}(D) = B_t D_{\mathrm{eff}}^{-\beta} + o(D_{\mathrm{eff}}^{-\beta})$ for a generic position t. The exponent β is position-independent because Assumption 5 bounds the eigenvalues $\bar{\lambda}_k^{(t)}$ uniformly across t; only the prefactor B_t varies with t. Averaging over positions gives $\mathcal{L}_{\mathrm{est}}(D) = BD_{\mathrm{eff}}^{-\beta} + o(D_{\mathrm{eff}}^{-\beta})$ with $B = L^{-1}\sum_t B_t$.

Step 1: Mode-wise error. Under Assumption 7, the estimation error on eigenmode k of $\bar{K}$ is

$$\mathrm{err}_k(D) \;=\; \frac{\mathbb{E}_\tau[\bar{f}_k^{\tau\,2}]}{1 + D_{\mathrm{eff}}\,\bar{\lambda}_k/\sigma^2_{\mathrm{noise}}}.$$

By Assumption 5, $\bar{\lambda}_k \geq c_{\min}\bar{\sigma}_k$, and by Assumption 6, $\mathbb{E}_\tau[\bar{f}_k^{\tau\,2}] = \bar{f}_0^2 k^{-\bar{\mu}}(1 + o(1))$ as $k \to \infty$. With $\bar{\sigma}_k = \bar{\sigma}_1 k^{-\nu_\mathcal{T}}(1 + o(1))$ from Assumption 4:

$$\mathrm{err}_k(D) \;\leq\; \frac{\bar{f}_0^2 k^{-\bar{\mu}}(1 + o(1))}{1 + D_{\mathrm{eff}}\,c_{\min}\,\bar{\sigma}_1\,k^{-\nu_\mathcal{T}}(1 + o(1))/\sigma^2_{\mathrm{noise}}}.$$

Step 2: Crossover index. Define $k^*(D)$ as the mode index at which the data term in the denominator equals 1:

$$D_{\mathrm{eff}}\,c_{\min}\,\bar{\sigma}_1\,(k^*)^{-\nu_\mathcal{T}}/\sigma^2_{\mathrm{noise}} \;=\; 1, \qquad k^*(D) \;=\; \Big(\frac{D_{\mathrm{eff}}\,c_{\min}\,\bar{\sigma}_1}{\sigma^2_{\mathrm{noise}}}\Big)^{1/\nu_\mathcal{T}}.$$

Note that $k^* \to \infty$ as $D_{\mathrm{eff}} \to \infty$, so the asymptotic forms of $\bar{\sigma}_k$ and $\mathbb{E}_\tau[\bar{f}_k^{\tau\,2}]$ apply for all modes near and beyond the crossover.

Step 3: Splitting the sum. We split $\mathcal{L}_{\mathrm{est}}(D) = S_{\leq}(D) + S_{>}(D)$, where $S_{\leq}$ sums over learned modes $k \leq k^*$ and $S_{>}$ over unlearned modes $k > k^*$.

Unlearned tail. For $k > k^*$, the denominator satisfies $1 \leq 1 + D_{\mathrm{eff}}\bar{\lambda}_k/\sigma^2_{\mathrm{noise}} \leq 2$, so $\mathrm{err}_k(D) = \mathbb{E}_\tau[\bar{f}_k^{\tau\,2}](1 + O(k^*/k)^{\nu_\mathcal{T}})^{-1}$. Therefore

$$S_{>}(D) \;=\; \sum_{k>k^*}\bar{f}_0^2 k^{-\bar{\mu}} \;+\; O\Big(\sum_{k>k^*} k^{-\bar{\mu}}\cdot(k^*/k)^{\nu_\mathcal{T}}\Big).$$

The error term is $O((k^*)^{\nu_\mathcal{T}}\sum_{k>k^*}k^{-\bar{\mu}-\nu_\mathcal{T}}) = O((k^*)^{-(\bar{\mu}-1)}\cdot(k^*)^{-1})$ when $\bar{\mu} + \nu_\mathcal{T} > 1$, which is $O(D_{\mathrm{eff}}^{-\beta-1/\nu_\mathcal{T}})$.

For the leading term, by the Euler–Maclaurin formula:

$$\sum_{k>k^*} k^{-\bar{\mu}} \;=\; \frac{(k^*)^{-(\bar{\mu}-1)}}{\bar{\mu}-1} \;+\; O\big((k^*)^{-\bar{\mu}}\big),$$

where convergence requires $\bar{\mu} > 1$. The remainder $O((k^*)^{-\bar{\mu}})$ is a multiplicative $O((k^*)^{-1})$ correction to the leading term, i.e., $O(D_{\text{eff}}^{-\beta-1/\nu_{\mathcal{T}}})$.

Combining:

$$S_>(D) \;=\; \frac{\bar{f}_0^2}{\bar{\mu}-1}\,(k^*)^{-(\bar{\mu}-1)} \;+\; O\big(D_{\text{eff}}^{-\beta-1/\nu_{\mathcal{T}}}\big). \tag{19}$$

Step 4: Substituting the crossover index. Replacing k^* with its expression from Step 2:

$$\begin{aligned} S_>(D) \;&=\; \frac{\bar{f}_0^2}{\bar{\mu}-1}\Big(\frac{D_{\text{eff}}\,c_{\min}\,\bar{\sigma}_1}{\sigma^2_{\text{noise}}}\Big)^{-(\bar{\mu}-1)/\nu_{\mathcal{T}}} \;+\; O\big(D_{\text{eff}}^{-\beta-1/\nu_{\mathcal{T}}}\big) \\ &=\; B\cdot D_{\text{eff}}^{-\beta} \;+\; O\big(D_{\text{eff}}^{-\beta-1/\nu_{\mathcal{T}}}\big), \end{aligned} \tag{20}$$

where $\beta = (\bar{\mu}-1)/\nu_{\mathcal{T}}$ and

$$B \;=\; \frac{\bar{f}_0^2}{\bar{\mu}-1}\Big(\frac{c_{\min}\,\bar{\sigma}_1}{\sigma^2_{\text{noise}}}\Big)^{-(\bar{\mu}-1)/\nu_{\mathcal{T}}}.$$

Note that this identifies the position-level prefactor B_t.

Step 5: Learned-mode contribution. For modes $k \le k^*$, the denominator satisfies $D_{\text{eff}}\bar{\lambda}_k/\sigma^2_{\text{noise}} \ge 1$, so $\text{err}_k(D) \le \mathbb{E}_\tau[\bar{f}_k^{\tau\,2}]\cdot\sigma^2_{\text{noise}}/(D_{\text{eff}}\bar{\lambda}_k)$. Therefore

$$S_\le(D) \;\le\; \frac{\bar{f}_0^2\,\sigma^2_{\text{noise}}}{D_{\text{eff}}\,c_{\min}\,\bar{\sigma}_1}\sum_{k=1}^{k^*} k^{\nu_{\mathcal{T}}-\bar{\mu}}.$$

We consider two cases.

Case $\bar{\mu} > \nu_{\mathcal{T}} + 1$*:* The sum $\sum_{k=1}^{k^*} k^{\nu_{\mathcal{T}}-\bar{\mu}}$ converges to a constant $C_1 < \infty$ as $k^* \to \infty$, since the exponent $\nu_{\mathcal{T}} - \bar{\mu} < -1$. Then $S_\le(D) = \Theta(D_{\text{eff}}^{-1})$. In this regime, $\beta = (\bar{\mu}-1)/\nu_{\mathcal{T}} > 1$, so $D_{\text{eff}}^{-\beta}$ decays *faster* than D_{eff}^{-1}: the unlearned tail $S_>$ is lower order and $S_\le$ dominates. The effective scaling exponent saturates at $\beta_{\text{eff}} = 1$, regardless of how large $(\bar{\mu}-1)/\nu_{\mathcal{T}}$ is. This regime (very concentrated signal with rapidly decaying spectrum) is not the empirically relevant one for language modeling, where $\beta < 1$ is consistently observed: Kaplan et al. [23] estimate $\beta \approx 0.095$, Hoffmann et al. [22] estimate $\beta \approx 0.34$, and the corrected replication of Besiroglu et al. [7] gives $\beta \approx 0.37$, all well below 1.

Case $1 < \bar{\mu} \le \nu_{\mathcal{T}} + 1$*:* The sum grows as $(k^*)^{\nu_{\mathcal{T}}-\bar{\mu}+1}/(\nu_{\mathcal{T}}-\bar{\mu}+1)$. Substituting $k^* = \Theta(D_{\text{eff}}^{1/\nu_{\mathcal{T}}})$:

$$\begin{aligned} S_\le(D) &= O\Big(D_{\text{eff}}^{-1}\cdot D_{\text{eff}}^{(\nu_{\mathcal{T}}-\bar{\mu}+1)/\nu_{\mathcal{T}}}\Big) \\ &= O\Big(D_{\text{eff}}^{(-\nu_{\mathcal{T}}+\nu_{\mathcal{T}}-\bar{\mu}+1)/\nu_{\mathcal{T}}}\Big) \\ &= O\Big(D_{\text{eff}}^{-(\bar{\mu}-1)/\nu_{\mathcal{T}}}\Big) \\ &= O\Big(D_{\text{eff}}^{-\beta}\Big). \end{aligned}$$

Therefore $S_\le(D) = \Theta(D_{\text{eff}}^{-\beta})$: **the same order as** $S_>(D)$. Both the learned and unlearned modes contribute to the leading-order prefactor.

Step 6: Combining. In the empirically relevant case $1 < \bar{\mu} \le \nu_{\mathcal{T}} + 1$ (which includes all language modeling settings where $\beta < 1$), the estimation loss is

$$\mathcal{L}_{\text{est}}(D) \;=\; S_\le(D) + S_>(D) \;=\; (B_> + B_\le)\cdot D_{\text{eff}}^{-\beta} + o(D_{\text{eff}}^{-\beta}),$$

where

$$B_{>} = \frac{\bar{f}_0^2}{\bar{\mu}-1}\left(\frac{c_{\min}\,\bar{\sigma}_1}{\sigma^2_{\text{noise}}}\right)^{-(\bar{\mu}-1)/\nu_{\mathcal{T}}},$$

$$B_{\leq} = \frac{\bar{f}_0^2}{\nu_{\mathcal{T}}-\bar{\mu}+1}\left(\frac{c_{\min}\,\bar{\sigma}_1}{\sigma^2_{\text{noise}}}\right)^{-(\bar{\mu}-1)/\nu_{\mathcal{T}}}.$$

The total prefactor is

$$B \;=\; B_{>} + B_{\leq} \;=\; \frac{\bar{f}_0^2\,\nu_{\mathcal{T}}}{(\bar{\mu}-1)(\nu_{\mathcal{T}}-\bar{\mu}+1)}\left(\frac{c_{\min}\,\bar{\sigma}_1}{\sigma^2_{\text{noise}}}\right)^{-\beta}.$$

□

B.5 Proof of Theorem 5

In addition to Assumption 4–7, Theorem 5 requires one additional assumption. We expand upon its brief mention in the theorem statement.

Assumption 8 (Capacity budget)**.** *A model with N parameters has a total representational budget of $M(N) = \kappa N^{1/d}$ units, for architecture-dependent constants $\kappa > 0$ and $d \geq 1$. Representing eigenmode k of $\bar{K}$ consumes $\bar{\sigma}_k^{-1}$ units from this budget, where $\bar{\sigma}_k$ is the k-th eigenvalue of the task-mixture kernel. The number of representable modes M_{eff} is the largest integer satisfying*

$$\sum_{k=1}^{M_{\text{eff}}} \bar{\sigma}_k^{-1} \;\leq\; M(N).$$

The inverse-eigenvalue cost reflects the fact that modes carrying more variance in the task mixture (large $\bar{\sigma}_k$) are statistically better-supported and can be represented with fewer parameters per unit of explained variance. Modes in the spectral tail (small $\bar{\sigma}_k$) require more precise parameterization to capture, analogous to the effective degrees of freedom in kernel ridge regression scaling inversely with the kernel eigenvalue [14].

The parameter d is the effective representation dimension: $d = 1$ corresponds to the linear regime (total budget proportional to N); $d > 1$ captures the overhead of encoding structured representations in a transformer (attention heads, layer composition, embedding tables). For a transformer with N parameters distributed across depth D_{layers} and width W (so $N \sim D_{\text{layers}} \cdot W^2$), the total number of representational slots scales roughly with $W \sim N^{1/2}$ for fixed depth, suggesting $d \approx 2$ as a naive estimate. More refined analyses based on tensor programs [47] suggest that d depends on the depth-width ratio and the activation function, but the power-law form $M(N) = \kappa N^{1/d}$ is consistent with observed scaling behavior.

Proof. The approximation error averages over tasks and positions: $\mathcal{L}_{\text{approx}}(N) = \mathbb{E}_\tau[L^{-1}\sum_t \mathcal{L}^{(\tau,t)}_{\text{approx}}(N)]$. As in the proof of Theorem 4, the position average does not affect the exponent (by Assumption 5), so we work at a generic position and average over tasks only.

Step 1: Capacity budget. Not all modes are equally costly to represent: modes carrying less variance in the task distribution require more precise parameterization to represent, analogous to the effective degrees of freedom in kernel ridge regression scaling inversely with the kernel eigenvalue [14]. Assumption 8 models this as a capacity cost proportional to $\bar{\sigma}_k^{-1}$ for mode k. The total capacity budget constrains the number of representable modes M_{eff}:

$$\sum_{k=1}^{M_{\text{eff}}} \bar{\sigma}_k^{-1} \;\leq\; M(N).$$

Here M_{eff} is the number of modes the model can represent given that mode k consumes $\bar{\sigma}_k^{-1}$ units from the total budget $M(N)$.

Step 2: Effective number of representable modes. Under Assumption 4, $\bar{\sigma}_k^{-1} = \bar{\sigma}_1^{-1} k^{\nu_\mathcal{T}}(1 + o(1))$ as $k \to \infty$. By the Euler–Maclaurin formula:

$$\sum_{k=1}^{M_{\text{eff}}} k^{\nu_\mathcal{T}} \;=\; \frac{M_{\text{eff}}^{\nu_\mathcal{T}+1}}{\nu_\mathcal{T}+1} \;+\; O\big(M_{\text{eff}}^{\nu_\mathcal{T}}\big).$$

Setting this equal to $\bar{\sigma}_1 M(N) = \bar{\sigma}_1 \kappa N^{1/d}$ and solving:

$$M_{\text{eff}} \;=\; \big(\bar{\sigma}_1\,\kappa\,(\nu_\mathcal{T}+1)\big)^{1/(\nu_\mathcal{T}+1)}\, N^{1/(d(\nu_\mathcal{T}+1))}\,\big(1 + O(M_{\text{eff}}^{-1})\big).$$

The correction $O(M_{\text{eff}}^{-1})$ comes from the Euler–Maclaurin remainder and is negligible for large N.

Step 3: Approximation error from unrepresented tail. Modes $k > M_{\text{eff}}$ are not represented by the model, contributing their full target energy as approximation error. Averaging over tasks and applying Assumption 6:

$$\mathcal{L}_{\text{approx}}(N) \;=\; \sum_{k>M_{\text{eff}}} \mathbb{E}_\tau\big[\bar{f}_k^{\tau\,2}\big] \;=\; \sum_{k>M_{\text{eff}}} \bar{f}_0^2\, k^{-\bar{\mu}}(1 + o(1)).$$

By the Euler–Maclaurin formula (as in the proof of Theorem 4, Step 3):

$$\sum_{k>M_{\text{eff}}} k^{-\bar{\mu}} \;=\; \frac{M_{\text{eff}}^{-(\bar{\mu}-1)}}{\bar{\mu}-1} \;+\; O\big(M_{\text{eff}}^{-\bar{\mu}}\big),$$

where convergence requires $\bar{\mu} > 1$ (Assumption 6). Therefore

$$\mathcal{L}_{\text{approx}}(N) \;=\; \frac{\bar{f}_0^2}{\bar{\mu}-1}\, M_{\text{eff}}^{-(\bar{\mu}-1)} \;+\; O\big(M_{\text{eff}}^{-\bar{\mu}}\big).$$

The remainder is a multiplicative $O(M_{\text{eff}}^{-1})$ correction to the leading term.

Step 4: Substitution. Inserting M_{eff} from Step 2:

$$\begin{aligned}\mathcal{L}_{\text{approx}}(N) \;&=\; \frac{\bar{f}_0^2}{\bar{\mu}-1}\,\big(\bar{\sigma}_1\,\kappa\,(\nu_\mathcal{T}+1)\big)^{-(\bar{\mu}-1)/(\nu_\mathcal{T}+1)}\, N^{-(\bar{\mu}-1)/(d(\nu_\mathcal{T}+1))} \\ &\quad +\; O\big(N^{-(\bar{\mu}-1)/(d(\nu_\mathcal{T}+1))-1/(d(\nu_\mathcal{T}+1))}\big) \\ &=\; A \cdot N^{-\alpha} \;+\; O\big(N^{-\alpha-1/(d(\nu_\mathcal{T}+1))}\big), \end{aligned} \tag{21}$$

with

$$\alpha \;=\; \frac{\bar{\mu}-1}{d\,(\nu_\mathcal{T}+1)},$$

and position-level prefactor

$$A_t \;=\; \frac{\bar{f}_0^2}{\bar{\mu}-1}\,\big(\bar{\sigma}_1\,\kappa\,(\nu_\mathcal{T}+1)\big)^{-(\bar{\mu}-1)/(\nu_\mathcal{T}+1)}.$$

Averaging over positions gives $A = L^{-1}\sum_t A_t$, and the final result is $\mathcal{L}_{\text{approx}}(N) = AN^{-\alpha} + o(N^{-\alpha})$, since the remainder $O(N^{-\alpha-1/(d(\nu_\mathcal{T}+1))})$ is $o(N^{-\alpha})$. □

B.6 Proof of Proposition 6

Proof. The lower bound is immediate from the identity (7): since $\mathrm{KL}(P_{Y|X}\|Q_{Y|X}) \geq 0$ for all X, we have $\mathcal{L}(N, D) = H(Y \mid X) + \mathbb{E}_X[\mathrm{KL}(P_{Y|X}\|Q_{Y|X})] \geq H(Y \mid X)$.

For the convergence, Theorems 4 and 5 give $\mathbb{E}_X[\mathrm{KL}(P_{Y|X}\|Q_{Y|X})] = O(D^{-\beta}) + O(N^{-\alpha})$. Both terms vanish as $N, D \to \infty$, so $\mathcal{L}(N, D) \to H(Y \mid X)$. □

B.7 Proof of Theorem 7

Proof. By the identity (7), $L(N, D) = H(Y|X) + \mathbb{E}_X[\mathrm{KL}(P_{Y|X} \| Q_{Y|X})]$. Under the kernel regime (Assumption 7), the KL gap decomposes along the eigenbasis of $\bar{K}$ as $\mathbb{E}_X[\mathrm{KL}(P_{Y|X} \| Q_{Y|X})] = \sum_{k=1}^{\infty} \mathrm{err}_k(N, D)$, where err_k is the mode-wise estimation error defined in Assumption 7 for representable modes and the full projection energy $\mathbb{E}_\tau[\bar{f}_k^{\tau\,2}]$ for unrepresentable modes. We partition this sum at $M_{\mathrm{eff}}(N)$ and evaluate each part.

Approximation error ($k > M_{\mathrm{eff}}$). This is exactly the sum bounded in Theorem 5:

$$\sum_{k > M_{\mathrm{eff}}} \mathbb{E}_\tau[\bar{f}_k^{\tau\,2}] \;=\; A\, N^{-\alpha} + O\big(N^{-\alpha - 1/(d(\nu_\mathcal{T}+1))}\big).$$

Estimation error on representable modes ($k \le M_{\mathrm{eff}}$). We further split this sum at the data crossover $k^*(D)$.

Learned modes ($k \le \min(k^*, M_{\mathrm{eff}})$): The denominator satisfies $D_{\mathrm{eff}}\bar{\lambda}_k/\sigma^2_{\mathrm{noise}} \ge 1$, so $\mathrm{err}_k \le \mathbb{E}_\tau[\bar{f}_k^{\tau\,2}] \cdot \sigma^2_{\mathrm{noise}}/(D_{\mathrm{eff}}\bar{\lambda}_k)$. As shown in Step 5 of the proof of Theorem 4, this contributes $o(D_{\mathrm{eff}}^{-\beta})$.

Unlearned representable modes ($k^* < k \le M_{\mathrm{eff}}$, when $k^* < M_{\mathrm{eff}}$): The denominator is between 1 and 2, so $\mathrm{err}_k = \mathbb{E}_\tau[\bar{f}_k^{\tau\,2}](1 + O((k^*/k)^{\nu_\mathcal{T}}))^{-1}$. The leading contribution is a partial tail sum, which we express as the difference of two full tails:

$$\begin{aligned}\sum_{k=k^*+1}^{M_{\mathrm{eff}}} \bar{f}_0^2\, k^{-\bar{\mu}} &= \underbrace{\sum_{k>k^*} \bar{f}_0^2\, k^{-\bar{\mu}}}_{\text{full tail from } k^*} - \underbrace{\sum_{k>M_{\mathrm{eff}}} \bar{f}_0^2\, k^{-\bar{\mu}}}_{\text{full tail from } M_{\mathrm{eff}}} \\ &= \frac{\bar{f}_0^2}{\bar{\mu}-1}(k^*)^{-(\bar{\mu}-1)} - \frac{\bar{f}_0^2}{\bar{\mu}-1} M_{\mathrm{eff}}^{-(\bar{\mu}-1)} + O\big((k^*)^{-\bar{\mu}} + M_{\mathrm{eff}}^{-\bar{\mu}}\big),\end{aligned}$$

by the Euler–Maclaurin formula applied to each tail (as in Steps 3–4 of the proofs of Theorems 4 and 5). The first term equals $B\, D_{\mathrm{eff}}^{-\beta}$ by the identification in Theorem 4. The second term equals $A\, N^{-\alpha}$ by the identification in Theorem 5. The remainder terms are lower order than their respective leading terms.

If $k^* \ge M_{\mathrm{eff}}$ (data is abundant relative to capacity): All representable modes are learned, and the estimation sum contributes only $o(D_{\mathrm{eff}}^{-\beta})$. The dominant reducible error is the approximation tail $A\, N^{-\alpha}$.

Combining. The total reducible error is

$$\sum_{k=1}^{\infty} \mathrm{err}_k(N, D) \;=\; \underbrace{\sum_{k \le \min(k^*, M_{\mathrm{eff}})} \mathrm{err}_k}_{\text{learned representable}} + \underbrace{\sum_{k > \min(k^*, M_{\mathrm{eff}})} \mathrm{err}_k}_{\text{unlearned or unrepresentable}}.$$

The first sum is $o(\max(N^{-\alpha}, D_{\mathrm{eff}}^{-\beta}))$ (learned modes contribute lower-order error). The second sum is a single tail from $\min(k^*, M_{\mathrm{eff}})$:

$$\sum_{k > \min(k^*, M_{\mathrm{eff}})} \bar{f}_0^2 k^{-\bar{\mu}} \;=\; \frac{\bar{f}_0^2}{\bar{\mu}-1}\big(\min(k^*, M_{\mathrm{eff}})\big)^{-(\bar{\mu}-1)} + \text{lower order}.$$

The binding constraint is whichever of k^* and M_{eff} is smaller:

- When $k^* < M_{\mathrm{eff}}$ (data is the bottleneck): the tail starts at k^*, giving $B\, D_{\mathrm{eff}}^{-\beta}$.
- When $M_{\mathrm{eff}} < k^*$ (capacity is the bottleneck): the tail starts at M_{eff}, giving $A\, N^{-\alpha}$.
- When $k^* \approx M_{\mathrm{eff}}$ (balanced regime): both terms are of the same order.

Therefore:

$$\mathcal{L}(N, D) \;=\; H(Y|X) + \frac{\bar{f}_0^2}{\bar{\mu}-1}\big(\min(k^*, M_{\mathrm{eff}})\big)^{-(\bar{\mu}-1)} + \text{lower order}.$$

Expressed in terms of N and D:

$$\mathcal{L}(N, D) \;=\; H(Y|X) + \max\big(A\, N^{-\alpha},\; B\, D^{-\beta}\big) + o\big(\max(N^{-\alpha}, D^{-\beta})\big). \qquad \square$$

B.8 Proof of Corollary 8

Proof. By Theorem 7, $\mathcal{L}(N, D) = H(Y|X) + \max(AN^{-\alpha}, BD^{-\beta}) + o(\max(N^{-\alpha}, D^{-\beta}))$. Along the compute-optimal frontier, $AN^{-\alpha} \asymp BD^{-\beta}$, meaning there exist constants $0 < c_l \leq c_u < \infty$ such that $c_l \leq AN^{-\alpha}/(BD^{-\beta}) \leq c_u$. In this regime:

$$\max(AN^{-\alpha}, BD^{-\beta}) \;\leq\; AN^{-\alpha} + BD^{-\beta} \;\leq\; (1 + c_u/c_l)\, \max(AN^{-\alpha}, BD^{-\beta}),$$

so the max and the sum differ by at most a bounded multiplicative constant, absorbed into the prefactors. Substituting:

$$\mathcal{L}(N, D) = AN^{-\alpha} + BD^{-\beta} + H(Y|X) + o\big(\max(N^{-\alpha}, D^{-\beta})\big). \qquad \square$$

C Experimental Details

All experiments are performed on a System76 Pangolin laptop, with AMD Ryzen 7 7735U processor, 32 GB RAM, and 1 TB disk capacity. The benchmark saturation experiment finishes in seconds, and the dependency kernel analysis takes a 3 minutes to compute the kernel.

C.1 Benchmark Saturation

Akhtar et al. [1] define benchmark saturation as the phenomenon where top-performing models are statistically indistinguishable on a task, with the performance metric(s) reaching an empirical ceiling. They use scores of top-k models, $s_1 \geq \ldots \geq s_k$ to quantify saturation on a benchmark. Given test set size n_{test}, they obtain the normalized score range R_{norm}:

$$R_{\text{norm}} := \frac{\Delta}{\text{SE}_\Delta}, \quad \text{where} \quad \Delta := s_1 - s_k, \quad \text{SE}_\Delta := \sqrt{\frac{s_1(1-s_1) + s_k(1-s_k)}{n_{\text{test}}^{\alpha}}}, \alpha \in [0, 1], \tag{22}$$

and use it to calculate the *saturation index* as $S_{\text{index}} := \exp(-R_{\text{norm}}^2)$. In practice, they use $k = 5, \alpha = 0.5$, and bucket benchmarks into five bins according to their S_{index} value: very low (< 0.01), low ([0.01, 0.3)), moderate ([0.3, 0.7)), high ([0.7, 0.9)), and very high saturation (≥ 0.9).

From the annotated dataset curated by Akhtar et al. [1][4], we take data on benchmarks for three types of task: code, math, and question-answering (Q&A). We add another task type, writing, by gathering publicly available data on 7 benchmarks: NC Bench [31], LechMazur Creative Story-Writing Benchmark [28], WritingBench [45], POEMetric [25], and three benchmarks from EQ-Bench 3 [33]: Longform Creative Writing, BuzzBench, and Spiral-Bench v1.2. We collect n_{test} and top-5 LLM scores for each benchmark, and use Eq. (22) to calculate R_{norm} then S_{index} for them. We use the s_1 and S_{index} values for the combined set of benchmarks to obtain Figure 1 in the main paper.

C.2 Dependency kernel analysis

For this analysis, we use five public benchmarks spanning four task types.

Benchmark	Task Type	Description
HumanEval [15]	Coding	Python function completion tasks with unit test verification
MBPP [3]	Coding	Short Python programs from crowd-sourced descriptions
GSM8K [16]	Math	Grade-school word problems with numerical answers
CNN/DailyMail [37]	Summarization	Abstractive summarization of news articles into multi-sentence highlights
WritingPrompts [18]	Creative writing	Open-ended short story generation from brief narrative prompts

Table 1: Summary of benchmarks used in dependency kernel analysis.

[4] Available in https://github.com/evaleval/benchmark-saturation

In practice, calculating the dependency kernel $K_Y(t,t') = I(y_y; y_{t'}|y_{-\{t,t'\}}, X)$ exactly is difficult since it requires a large number of samples to estimate the mutual information (MI) between two random variables conditioned on a vector of $L-2$ other tokens. In this experiment, we instead use unconditional pairwise MI as a proxy: $\widehat{K}_Y(t,t') := I(y_t; y_{t'})$. The unconditional version has additional noise from *indirect dependencies*, i.e. correlation between tokens t, t' because both depend on token s, that $K_Y(t,t')$ does not. However, indirect dependencies propagate locally in banded tasks and globally in dense tasks, so removing them sharpens inter-token dependency structure but does not change the qualitative classification.

We implement pairwise MI using the nearest neighbor method of Mesner and Shalizi [29], with $k = 3$ nearest neighbors and maximum token length $L = 32$, using 500 random samples per benchmark. To obtain eigenspectrum and ν_τ from a kernel matrix K, we calculate the eigenvalues of its symmetric version $(K + K^\top)/2$ (to remove any assymmetry due to nearest-neighbor estimation) with non-finite entries replaced by zero, and retain only the positive eigenvalues larger than 10^{-10}.

D Consequences for Practice: Results and Proofs

D.1 Additional context raises reliability ceiling

The reliability ceiling $R^* = I(X;Y)/H(Y)$ depends on the observable input X. A natural question is whether enriching the input with additional (resolvable) context can raise this ceiling, and if so, by how much.

Theorem 11. *Let $\tilde{C}$ be auxiliary context jointly distributed with (X, Y, C) and made available to the model. Assume $\tilde{C} \perp C_u \mid (X, C_r)$, i.e., the auxiliary context is informative only through the resolvable component of the latent context. Then:*

- *(i) $\tilde{R}^* := I(X, \tilde{C}; Y)/H(Y) \geq R^*$, with equality if and only if $\tilde{C} \perp Y \mid X$.*
- *(ii) $\tilde{R}^* - R^* = I(\tilde{C}; Y \mid X)/H(Y) \leq \delta_r$, i.e. the improvement in reliability ceiling is bounded by the resolvable gap.*
- *(iii) For a sequence of auxiliary variables $\tilde{C}_1, \tilde{C}_2, \ldots$, the augmented ceiling $\tilde{R}^*$ is nondecreasing in the number of injections. When $\tilde{C}$ resolves all of C_r, the ceiling reaches $R^*_{\max} = 1 - \delta_u$.*

Proof. **Part (i).** By the chain rule of mutual information, $I(X, \tilde{C}; Y) = I(X;Y) + I(\tilde{C}; Y \mid X)$. Since $I(\tilde{C}; Y \mid X) \geq 0$ by non-negativity, $I(X, \tilde{C}; Y) \geq I(X;Y)$, and dividing by $H(Y)$ gives $\tilde{R}^* \geq R^*$. Equality holds if and only if $I(\tilde{C}; Y \mid X) = 0$, i.e., $\tilde{C} \perp Y \mid X$. This occurs when $\tilde{C}$ is a deterministic function of X or otherwise carries no information about Y beyond what X already provides.

Part (ii). We show $I(\tilde{C}; Y \mid X) \leq I(C_r; Y \mid X) = \delta_r \cdot H(Y)$. Recall the decomposition $C = (C_r, C_u)$ with $Y = g(X, C_r, C_u)$. The auxiliary context $\tilde{C}$ is informative about Y through its dependence on (X, C). We bound its contribution by decomposing:

$$I(\tilde{C}; Y \mid X) = I(\tilde{C}; Y, C_r \mid X) - I(\tilde{C}; C_r \mid X, Y) \leq I(\tilde{C}; Y, C_r \mid X).$$

Since $\tilde{C}$ provides information about Y only through its relationship with the latent context, we have

$$I(\tilde{C}; Y \mid X) \leq I(C_r; Y \mid X) = \delta_r \cdot H(Y),$$

where the inequality follows because $\tilde{C}$ can at most reveal all of C_r (in which case $I(\tilde{C}; Y \mid X) = I(C_r; Y \mid X)$) but cannot reduce the subjective gap: $I(C_u; Y \mid X, C_r, \tilde{C}) = I(C_u; Y \mid X, C_r)$, which holds by the assumption $\tilde{C} \perp C_u \mid (X, C_r)$. Dividing by $H(Y)$:

$$\tilde{R}^* - R^* = \frac{I(\tilde{C}; Y \mid X)}{H(Y)} \leq \delta_r.$$

Part (iii). Let $\tilde{C}_1, \ldots, \tilde{C}_n$ be a sequence of auxiliary variables. Define $\tilde{R}^*_k = I(X, \tilde{C}_1, \ldots, \tilde{C}_k; Y)/H(Y)$. By the chain rule,

$$\tilde{R}^*_{k+1} - \tilde{R}^*_k = \frac{I(\tilde{C}_{k+1}; Y \mid X, \tilde{C}_1, \ldots, \tilde{C}_k)}{H(Y)} \geq 0,$$

so the sequence $\{\tilde{R}_k^*\}$ is nondecreasing. It is bounded above by $I(X, C_r; Y)/H(Y) = 1 - \delta_u$ from part (ii), since the cumulative information from all auxiliary variables cannot exceed the total resolvable information $I(C_r; Y \mid X)$. When $\tilde{C}$ collectively resolves all of C_r—i.e., C_r is a measurable function of $(X, \tilde{C}_1, \ldots, \tilde{C}_n)$—then $I(X, \tilde{C}; Y) = I(X, C_r; Y)$ and

$$\tilde{R}^*_{\max} = \frac{I(X, C_r; Y)}{H(Y)} = 1 - \delta_u.$$

This is exactly 1 when $\delta_u = 0$ (ideally verifiable tasks) and strictly less than 1 when $\delta_u > 0$ (unverifiable tasks). □

Part (i) gives the mechanism: additional context makes the output more determined, reducing the gap to perfect reliability. Part (ii) establishes the limit of this improvement: context enrichment can close the resolvable gap δ_r but leaves the subjective gap δ_u untouched. Part (iii) shows that successive injections have nondecreasing but bounded returns, with the bound $R^*_{\max} = 1 - \delta_u$ achieved when all resolvable context is provided.

RAG, few-shot prompting, tool use, and source-grounded question-answering are all instances of this mechanism: each makes a portion of the latent context C_r observable, raising R^* toward $R^*_{\max}$. For ideally verifiable tasks, $R^*_{\max} = 1$ and the path to perfect reliability is an engineering problem. For unverifiable tasks, $R^*_{\max} < 1$ and a permanent floor on unreliability remains regardless of how much context is supplied.

D.2 Per-step verification slows autoregressive degradation

Autoregressive generation degrades the ceiling from R^* to $R^*_{\text{seq}} \leq \min(R^*, \gamma(L))$ (Proposition 2). A natural mitigation is *per-step verification*: an oracle $v_t = v(\hat{y}_{\leq t}, X)$ that evaluates the prefix at each step. Whether such oracles are informative depends on the dependency kernel K_Y.

Proposition 12. *Suppose at each step t, an oracle provides feedback $v_t = v(\hat{y}_{\leq t}, X)$ that the model uses to correct its generated prefix before proceeding. Let $\Delta_t^{\text{ver}} := \omega(d(\hat{y}^{\text{ver}}_{<t}, y_{<t}))$ denote the drift under verification. Then:*

(i) $\Delta_t^{\text{ver}} \leq \Delta_t$ for all t, with equality when v_t is uninformative.
(ii) The drift reduction $r_t := \Delta_t - \Delta_t^{\text{ver}} \geq 0$ is bounded by the correction the oracle enables: $r_t \leq \omega\big(d(\hat{y}_{<t}, \hat{y}^{\text{ver}}_{<t})\big)$.
(iii) The verified degradation bound from Proposition 2(iii) improves to

$$R^{*\,\text{verified}}_{\text{seq}} \;\leq\; \min\Big(R^*,\;\exp\Big(-c\sum_{t=1}^{L}\Delta_t^{\text{ver}}\Big)\Big) \;\geq\; \min\Big(R^*,\;\exp\Big(-c\sum_{t=1}^{L}\Delta_t\Big)\Big).$$

Proof. Part (i): The verified model generates $\hat{y}_t^{\text{ver}}$ conditioned on $(\hat{y}^{\text{ver}}_{<t}, X, v_1, \ldots, v_{t-1})$. The oracle feedback $v_{<t}$ provides information about the true prefix $y_{<t}$, enabling the model to correct errors in $\hat{y}_{<t}$. Formally, the verified prefix satisfies $d(\hat{y}^{\text{ver}}_{<t}, y_{<t}) \leq d(\hat{y}_{<t}, y_{<t})$ whenever the model uses the oracle feedback to bring its prefix closer to the target (which any rational correction strategy achieves). Since ω is nondecreasing (Assumption 1), $\Delta_t^{\text{ver}} = \omega(d(\hat{y}^{\text{ver}}_{<t}, y_{<t})) \leq \omega(d(\hat{y}_{<t}, y_{<t})) = \Delta_t$.

Part (ii): By the triangle inequality for d and the monotonicity of ω,

$$\Delta_t - \Delta_t^{\text{ver}} \;=\; \omega(d(\hat{y}_{<t}, y_{<t})) - \omega(d(\hat{y}^{\text{ver}}_{<t}, y_{<t})) \;\leq\; \omega(d(\hat{y}_{<t}, \hat{y}^{\text{ver}}_{<t})),$$

where the last step uses the subadditivity of ω (which holds when ω is concave, a mild regularity condition). The bound says the drift reduction at each step is limited by how much the oracle-guided correction actually changes the prefix.

Part (iii): Substituting $\Delta_t^{\text{ver}} \leq \Delta_t$ into the degradation bound of Proposition 2(iii):

$$\exp\Big(-c\sum_t \Delta_t^{\text{ver}}\Big) \;\geq\; \exp\Big(-c\sum_t \Delta_t\Big),$$

since $\exp(-cx)$ is decreasing. The verified model degrades more slowly. □

The drift reduction r_t depends on how informative the oracle is at step t, which in turn depends on the dependency structure of K_Y. For banded K_Y (e.g., code), correctness at position t is largely determined by local constraints—syntax, types, bracket matching—that a prefix-based oracle can check. When the oracle detects a constraint violation, the correction $d(\hat{y}_{<t}, \hat{y}^{\text{ver}}_{<t})$ is substantial, making r_t large at most positions. The cumulative drift reduction $\sum_t r_t$ is $\Theta(\epsilon L)$, comparable to the total unverified drift, so verification substantially slows degradation. For dense K_Y (e.g., poetry), correctness at position t depends on the entire sequence: whether the word contributes to the rhyme scheme, thematic arc, and tonal coherence of the whole. No prefix-based oracle can assess these global properties, because they depend on tokens not yet generated. The correction is negligible at most positions ($r_t \approx 0$), so $\sum_t r_t \approx 0$ and verification provides essentially no improvement.

Together with Theorem 11, these results reveal a triple penalty for unverifiable tasks with dense dependencies: low R^* (depressed ceiling from $\delta_u > 0$), fast $\gamma(L)$ decay (global error propagation through dense K_Y), and ineffective per-step verification (quality is a global property that no prefix-based oracle can assess). Fully verifiable tasks with banded dependencies enjoy the opposite on all three fronts.

D.3 Benchmark rankings on low-ceiling tasks are inherently unstable

Current evaluation practices implicitly assume that all tasks have $R^* = 1$: a model is scored against a single ground truth, and the resulting accuracy or reliability metric is treated as a reliable quality measure. When $R^* < 1$, the task admits multiple legitimate outputs for the same input, and the evaluation against any single reference introduces noise that no protocol can eliminate.

Proposition 13. *Consider a set of LLMs evaluated on a benchmark with M instances drawn i.i.d. from a task with $R^* < 1$. The expected Spearman rank correlation between model rankings across independent evaluation runs satisfies*

$$\mathbb{E}[\rho] \leq 1 - \frac{c(1 - R^*)}{M}, \tag{23}$$

where $c > 0$ depends on the evaluation protocol and the output distribution.

Proof. For each instance m, model j produces a single output $\hat{y}_j^{(m)}$ and receives a score $s_j^{(m)} \in [0, 1]$ from the evaluation protocol. When $R^* = 1$, the correct output is unique and the score is deterministic given the model's capability, so $\mathrm{V}(s_j^{(m)}) = 0$. When $R^* < 1$, the conditional entropy $H(Y \mid X) = (1 - R^*) \cdot H(Y) > 0$ implies that the target distribution $P(Y \mid X)$ has nonzero spread: multiple outputs have positive probability for the same input. A model sampling from its learned distribution produces different outputs across runs, and a scoring function that compares against any fixed reference inherits this variation.

To lower-bound the score variance, note that the scoring function $s = g(\hat{y}, y_{\text{ref}})$ compares the model output against a reference. When $H(Y \mid X) > 0$, the model's output distribution has support on multiple tokens at each step. Across independent evaluation runs, the sampled output $\hat{y}_j^{(m)}$ varies, and because the scoring function is non-constant on the support (any reasonable metric distinguishes at least some pairs of valid outputs), the score variance satisfies

$$\mathrm{V}(s_j^{(m)}) \;\geq\; c'(1 - R^*), \tag{24}$$

for a constant $c' > 0$ depending on the scoring function's sensitivity and $H(Y)$.

The benchmark score for model j is $\bar{s}_j = M^{-1} \sum_{m=1}^{M} s_j^{(m)}$. By independence across instances, $\mathrm{V}(\bar{s}_j) \geq c'(1 - R^*)/M$. Write $\bar{s}_j = \mu_j + \epsilon_j$, where μ_j is the true expected score and ϵ_j is zero-mean evaluation noise with variance $\sigma_\epsilon^2 \geq c'(1 - R^*)/M$.

By the classical attenuation formula [40], the expected *Pearson* correlation between two independent evaluation runs is

$$\mathbb{E}[r] \;=\; \frac{\sigma_\mu^2}{\sigma_\mu^2 + \sigma_\epsilon^2} \;=\; 1 - \frac{\sigma_\epsilon^2}{\sigma_\mu^2 + \sigma_\epsilon^2} \;\leq\; 1 - \frac{c(1 - R^*)}{M},$$

where $\sigma_\mu^2 = \mathrm{Var}(\mu_j)$ is the variance of true model qualities across the set of LLMs, and $c = c'/(\sigma_\mu^2 + c'/M)$.

The Spearman rank correlation ρ is the Pearson correlation of the ranks. For continuous score distributions, ρ and r coincide in expectation; for discrete scores (as in practice), the attenuation formula for ρ is an approximation that becomes exact as the number of models grows. We use it here as a standard proxy, following [40]. □

D.4 Fine-tuning as spectral reallocation

The scaling law of Theorem 7 governs a generalist model trained on a task mixture $\mathcal{T}$. A natural question is what happens when such a model is finetuned to a single task. To formalize this, we introduce a measure of structural similarity between tasks.

Definition 6. *The* alignment *between tasks* τ_1 *and* τ_2 *is* $\rho(\tau_1, \tau_2) = \mathrm{tr}(K_Y^{\tau_1} K_Y^{\tau_2})/(\|K_Y^{\tau_1}\|_F \|K_Y^{\tau_2}\|_F) \in [0, 1]$. *High alignment means two tasks share dependency structure; low alignment means their kernels are approximately orthogonal.*

Proposition 14. *Fine-tuning on task* τ^* *tilts the effective eigenspectrum from* $\bar{K}$ *toward* $K_Y^{\tau^*}$*, shifting the spectral parameters from* $(\nu_{\mathcal{T}}, \bar{\mu})$ *toward* $(\nu_{\tau^*}, \mu_{\tau^*})$*. The performance loss on task* $\tau \neq \tau^*$ *scales with* $\sqrt{1 - \rho(\tau, \tau^*)^2}$.

Proof. During pretraining, the model serves the task distribution $\mathcal{T}$, and Theorem 5 gives the capacity exponent $\alpha = (\bar{\mu} - 1)/(d(\nu_{\mathcal{T}} + 1))$. The model's $M(N)$ representable modes are distributed across the eigenmodes of the mixture kernel $\bar{K} = \mathbb{E}_{\tau \sim \mathcal{T}}[K_Y^{\tau}]$, with capacity allocated in proportion to the eigenvalues $\{\bar{\sigma}_k\}$.

Fine-tuning on τ^* reoptimizes the model's parameters under the loss for τ^* alone. This does not discard the pretrained spectrum but tilts it toward $K_Y^{\tau^*}$: modes aligned with τ^* are amplified while modes orthogonal to it are attenuated. The effective kernel $\bar{K}_{\mathrm{ft}}$ becomes progressively more aligned with $K_Y^{\tau^*}$ as fine-tuning proceeds, approaching $K_Y^{\tau^*}$ in the limit of full convergence. This tilt has three effects on the spectral structure.

Effect 1: Mode amplification and suppression. Let $\{(\bar{\sigma}_k, \bar{\phi}_k)\}_{k \geq 1}$ be the eigendecomposition of the pretrained mixture kernel $\bar{K}$, and let $a_k^{\tau^*} = \langle K_Y^{\tau^*} \bar{\phi}_k, \bar{\phi}_k \rangle \geq 0$ be the projection of the dependency kernel of τ^* onto the k-th pretrained eigenmode. The pretrained model allocates capacity to mode k in proportion to $\bar{\sigma}_k$. We show that fine-tuning on τ^* reallocates capacity in proportion to $a_k^{\tau^*}$.

The cross-entropy loss for task τ^* decomposes in the eigenbasis of $\bar{K}$ as

$$\mathcal{L}_{\tau^*}(\theta) = H(Y|X) + \sum_{k \geq 1} a_k^{\tau^*} \epsilon_k(\theta)^2, \tag{25}$$

where $\epsilon_k(\theta)$ is the model's approximation error along mode k, and the weight $a_k^{\tau^*}$ determines how much each mode contributes to the loss on τ^*.

Since the terms in the sum in (25) are independent, representing mode k reduces the loss by $a_k^{\tau^*} \epsilon_k^2$ regardless of which other modes are represented. Thus, a model with capacity to represent M modes minimizes $\mathcal{L}_{\tau^*}$ by selecting the M modes with the largest $a_k^{\tau^*}$, setting their $\epsilon_k = 0$ and leaving the rest at their default error. Let $S_{\mathrm{pre}} = \{k : \bar{\sigma}_k \text{ is among the top } M\}$ and $S_{\mathrm{ft}} = \{k : a_k^{\tau^*} \text{ is among the top } M\}$ be the mode sets selected by the pretrained and fine-tuned models respectively. Then:

- Modes in $S_{\mathrm{ft}} \setminus S_{\mathrm{pre}}$ are *amplified*: they were not represented by the pretrained model (because $\bar{\sigma}_k$ was too small to justify inclusion in the mixture) but are now represented because $a_k^{\tau^*}$ is large.
- Modes in $S_{\mathrm{pre}} \setminus S_{\mathrm{ft}}$ are *suppressed*: they were represented by the pretrained model (because $\bar{\sigma}_k$ was large in the mixture) but are now dropped because $a_k^{\tau^*}$ is small.
- Modes in $S_{\mathrm{pre}} \cap S_{\mathrm{ft}}$ are *retained*: they are important under both the mixture and τ^*.

The number of suppressed modes $|S_{\mathrm{pre}} \setminus S_{\mathrm{ft}}|$ equals the number of amplified modes $|S_{\mathrm{ft}} \setminus S_{\mathrm{pre}}|$, since both sets have cardinality M.

Effect 2: Catastrophic forgetting. We quantify the performance degradation on a task $\tau \neq \tau^*$ after fine-tuning on τ^*. Recall that the pretrained and finetuned models represents modes $S_{\mathrm{pre}}, S_{\mathrm{ft}}$

respectively, with

$$S_{\text{pre}} = \{k : \bar{\sigma}_k \text{ is among the top } M\},$$
$$S_{\text{ft}} = \{k : a_k^{\tau^*} \text{ is among the top } M\}.$$

The loss on task τ under a model representing mode set S is $\mathcal{L}_\tau(S) = H_\tau(Y|X) + \sum_{k \notin S} a_k^\tau \, \epsilon_k^2$, where $a_k^\tau = \langle K_Y^\tau \bar{\phi}_k, \bar{\phi}_k \rangle$ is the projection of τ's kernel onto mode k, and ϵ_k is the default (unrepresented) error for mode k. The change in loss on τ due to fine-tuning is

$$\begin{aligned} \mathcal{L}_\tau(S_{\text{ft}}) - \mathcal{L}_\tau(S_{\text{pre}}) &= \sum_{k \notin S_{\text{ft}}} a_k^\tau \, \epsilon_k^2 - \sum_{k \notin S_{\text{pre}}} a_k^\tau \, \epsilon_k^2 \\ &= \underbrace{\sum_{k \in S_{\text{pre}} \setminus S_{\text{ft}}} a_k^\tau \, \epsilon_k^2}_{\text{loss from suppressed modes}} - \underbrace{\sum_{k \in S_{\text{ft}} \setminus S_{\text{pre}}} a_k^\tau \, \epsilon_k^2}_{\text{gain from amplified modes}} . \end{aligned} \tag{26}$$

Catastrophic forgetting on a task τ occurs when the first sum exceeds the second, i.e., the modes lost were more important for τ than the modes gained.

The alignment $\rho(\tau, \tau^*)$ controls which regime holds. Denote by $S_{\text{pre}} \setminus S_{\text{ft}} := \{k : \bar{\sigma}_k \text{ large but } a_k^{\tau^*} \text{ small}\}$ the modes important for the mixture but not for τ^*. Whether they are important for τ depends on $\rho(\tau, \tau^*)$:

- **High alignment** ($\rho(\tau, \tau^*) \approx 1$): The kernels K_Y^τ and $K_Y^{\tau^*}$ share eigenmodes, so a_k^τ is large when $a_k^{\tau^*}$ is large and small when $a_k^{\tau^*}$ is small. The suppressed modes (small $a_k^{\tau^*}$) also have small a_k^τ, so the first sum in (26) is small. Conversely, the amplified modes (large $a_k^{\tau^*}$) also have large a_k^τ, so the second sum is large. The net effect is $\mathcal{L}_\tau(S_{\text{ft}}) - \mathcal{L}_\tau(S_{\text{pre}}) \approx 0$, so performance on task τ is largely preserved.
- **Low alignment** ($\rho(\tau, \tau^*) \approx 0$): The kernels K_Y^τ and $K_Y^{\tau^*}$ have approximately orthogonal eigenmodes. The suppressed modes (small $a_k^{\tau^*}$) may have large a_k^τ as they were serving τ through the mixture but are irrelevant to τ^*. The amplified modes (large $a_k^{\tau^*}$) have small a_k^τ, so the gain term is negligible. The net effect is $\mathcal{L}_\tau(S_{\text{ft}}) - \mathcal{L}_\tau(S_{\text{pre}}) \approx \sum_{k \in S_{\text{pre}} \setminus S_{\text{ft}}} a_k^\tau \, \epsilon_k^2 > 0$, so performance on task τ degrades substantially.

To make this precise, we bound the spectral mass of τ on the suppressed modes. We work in the eigenbasis $\{\bar{\phi}_k\}$ of the mixture kernel $\bar{K}$, and define $a_k^\tau = \langle K_Y^\tau \bar{\phi}_k, \bar{\phi}_k \rangle \geq 0$.

The decomposition $\|K_Y^\tau\|_F^2 = \sum_k (a_k^\tau)^2$ and $\operatorname{tr}(K_Y^\tau K_Y^{\tau^*}) = \sum_k a_k^\tau a_k^{\tau^*}$ hold exactly when K_Y^τ and $K_Y^{\tau^*}$ are simultaneously diagonalizable in the mixture eigenbasis—i.e., when each task kernel's eigenmodes are well-approximated by the shared basis $\{\bar{\phi}_k\}$. This is a natural consequence of the task-mixture structure: the mixture kernel $\bar{K}$ is designed to span the dominant eigenmodes of all tasks in $\mathcal{T}$, so each task's kernel is well-represented in this basis up to residual cross-terms. We proceed under this approximation, noting that the residual cross-terms contribute lower-order corrections to the bound below.

Partition the full set of modes into S_{ft} (retained by the fine-tuned model) and its complement S_{ft}^c. The total spectral mass of τ decomposes as

$$\|K_Y^\tau\|_F^2 = \sum_{k \in S_{\text{ft}}} (a_k^\tau)^2 + \sum_{k \in S_{\text{ft}}^c} (a_k^\tau)^2.$$

We bound the retained mass using the alignment. By Cauchy–Schwarz,

$$\sum_{k \in S_{\text{ft}}} a_k^\tau \, a_k^{\tau^*} \leq \Big(\sum_{k \in S_{\text{ft}}} (a_k^\tau)^2 \Big)^{1/2} \Big(\sum_{k \in S_{\text{ft}}} (a_k^{\tau^*})^2 \Big)^{1/2}.$$

Since S_{ft} contains the top-M modes by $a_k^{\tau^*}$, the second factor satisfies $\sum_{k \in S_{\text{ft}}} (a_k^{\tau^*})^2 \leq \|K_Y^{\tau^*}\|_F^2$.

For the left-hand side, we decompose the full inner product:

$$\operatorname{tr}(K_Y^\tau K_Y^{\tau^*}) = \sum_k a_k^\tau \, a_k^{\tau^*} = \sum_{k \in S_{\text{ft}}} a_k^\tau \, a_k^{\tau^*} + \sum_{k \notin S_{\text{ft}}} a_k^\tau \, a_k^{\tau^*}.$$

The second sum runs over modes $k \notin S_{\text{ft}}$, which by construction have the smallest values of $a_k^{\tau^*}$ (since S_{ft} selected the top-M). Let $a_{\max}^{\tau^*}(S_{\text{ft}}^c) := \max_{k \notin S_{\text{ft}}} a_k^{\tau^*}$ denote the largest τ^*-projection outside the retained set. Then

$$\sum_{k \notin S_{\text{ft}}} a_k^{\tau} \, a_k^{\tau^*} \;\le\; a_{\max}^{\tau^*}(S_{\text{ft}}^c) \sum_{k \notin S_{\text{ft}}} a_k^{\tau}.$$

Under the power-law spectrum (Assumption 4), the projections satisfy $a_k^{\tau^*} \sim k^{-\nu_{\tau^*}}$, so $a_{\max}^{\tau^*}(S_{\text{ft}}^c) = a_{M+1}^{\tau^*} \sim M^{-\nu_{\tau^*}} \to 0$ as $M \to \infty$. The remainder is therefore $O(M^{-\nu_{\tau^*}})$, which is negligible for any model with enough capacity to meaningfully fine-tune on τ^*. Concretely:

$$\sum_{k \in S_{\text{ft}}} a_k^{\tau} \, a_k^{\tau^*} \;\ge\; \rho(\tau, \tau^*) \|K_Y^{\tau}\|_F \|K_Y^{\tau^*}\|_F \;-\; a_{\max}^{\tau^*}(S_{\text{ft}}^c) \sum_{k \notin S_{\text{ft}}} a_k^{\tau}. \tag{27}$$

In the regime where the remainder is negligible (i.e., M is large relative to the effective rank of $K_Y^{\tau^*}$), substituting (27) into the Cauchy–Schwarz bound and rearranging gives

$$\sum_{k \in S_{\text{ft}}} (a_k^{\tau})^2 \;\ge\; \rho(\tau, \tau^*)^2 \|K_Y^{\tau}\|_F^2 \;-\; O\big(a_{\max}^{\tau^*}(S_{\text{ft}}^c)\big).$$

Therefore the lost mass satisfies

$$\sum_{k \in S_{\text{ft}}^c} (a_k^{\tau})^2 \;\le\; (1 - \rho(\tau, \tau^*)^2) \, \|K_Y^{\tau}\|_F^2 \;+\; O\big(a_{\max}^{\tau^*}(S_{\text{ft}}^c)\big).$$

The forgetting bound follows as before:

$$\mathcal{L}^{\tau}(S_{\text{ft}}) - \mathcal{L}^{\tau}(S_{\text{pre}}) \;\le\; \|\epsilon\|_\infty^2 \sqrt{M} \cdot \sqrt{1 - \rho(\tau, \tau^*)^2 + O(a_{\max}^{\tau^*}(S_{\text{ft}}^c))} \cdot \|K_Y^{\tau}\|_F.$$

Since $a_{\max}^{\tau^*}(S_{\text{ft}}^c) = O(M^{-\nu_{\tau^*}})$ under the power-law spectrum, this reduces to the stated bound with forgetting scaling as $\sqrt{1 - \rho(\tau, \tau^*)^2}$, up to a correction that vanishes polynomially in M. □