SCALING LAWS OF ENSEMBLE UNCERTAINTY IN LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Many use cases for large language models (LLMs), from AI safety to exploration and human interaction, can benefit from epistemic uncertainty estimates. While ensembles are practical candidates, multi-seed experiments show that their disagreement is small compared to individual token entropy and significantly underestimates the true epistemic uncertainty of the model. We try to shed light on these observations and broader aspects of ensembling for uncertainty estimation in LLMs. The upshot of our findings is that ensembling becomes more profitable both for uncertainty estimation and as an alternative to scaling model size at larger scales. We relate epistemic uncertainty to reducible loss in calibrated models, separating the portion visible through ensemble disagreement from an invisible portion that includes model misspecification. We show that the visible portion decreases as a mild power law with model size and changes little with additional training data or context beyond an initial stabilization range. The invisible portion decreases faster, so ensembles reveal a growing share of average epistemic uncertainty as models scale. These scaling trends also predict that ensembling becomes more competitive in terms of loss as the size of a single model increases, with matched total parameters and fixed training data. We then identify a second power law for ensembles: relative to independently trained ensembles, that relates the ensemble diversity to the shared portion of the training across ensemble members. The implication is that any late branching is highly penalized in its diversity, and this issue is even more severe when ensembling with low-rank adapters.

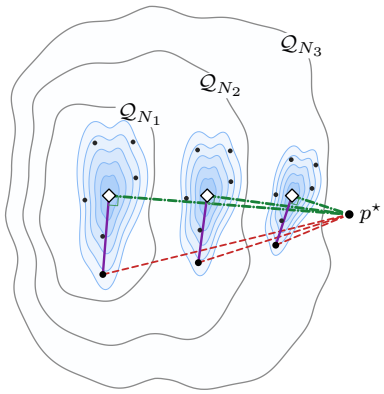

Figure 1: **Visible and invisible epistemic uncertainty.** The schematic shows increasingly expressive model families approaching the true next-token predictor $p^\star$. Blue contours idealize Bayesian posterior distributions; dots are ensemble members and diamonds their pooled predictions. **Red dashed edges** show total epistemic loss above the irreducible loss that decreases with a well-known power law with model size. **Purple solid edges** show the visible portion recovered by ensembling, which we find decreases rather slowly as a power law with model size. **Green dash–dot edges** show the invisible portion, including errors shared by the members. Their different decay rates imply a growing recovered share over the measured size regime. We investigate this phenomenon and its implications. Squared edge lengths illustrate an additive loss split.

1 INTRODUCTION

Epistemic uncertainty reflects gaps in knowledge that more data or a better model could reduce. Bayesian inference represents such uncertainty through a distribution over plausible explanations. This uncertainty can guide the selection of informative labels in active learning (Houlsby et al., 2011), experimental design (Rainforth et al., 2024), and autonomous materials discovery (Kusne et al., 2020). Recognizing knowledge gaps can also guide safe exploration and decisions about when to seek human oversight under distribution shift (Amodei et al., 2016), while uncertainty about human preferences can incentivize an agent to defer to human intervention (Hadfield-Menell et al., 2017; Russell, 2019).

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

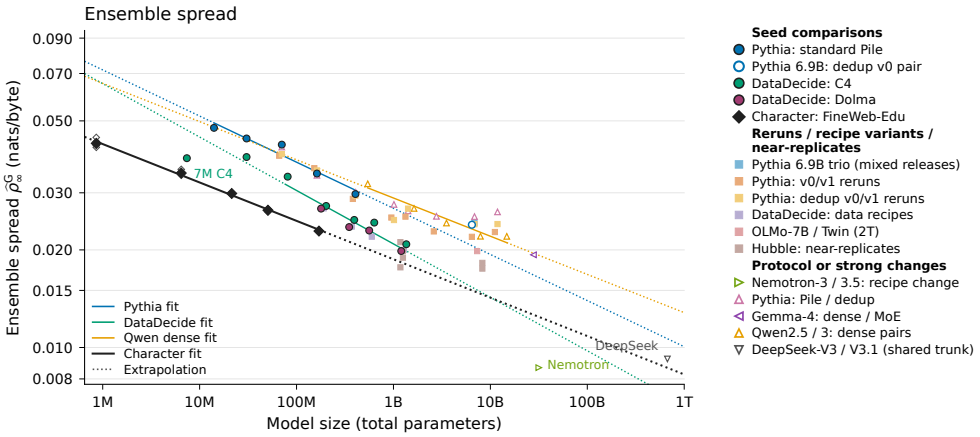

Figure 2: **Ensemble spread decreases slowly with model size.** Ensemble spread for 60 public-model comparisons and our character-level family. Axes use total parameters and nats/byte. Character evaluation uses FineWeb-Edu and 512-byte contexts; the remaining models use recent text and up to 1,024 subword tokens, so vertical offsets also reflect evaluation differences. Solid lines show fits ($\gamma = 0.118$–0.165); dotted lines show extrapolations.

Many Bayesian Deep Learning (BDL) methods, including variational inference, Monte Carlo dropout, Laplace approximations, optimization-based methods, and many variations of ensembles (Graves, 2011; Gal & Ghahramani, 2016; Daxberger et al., 2021; Maddox et al., 2019; Lakshminarayanan et al., 2017), approximate a distribution over weights within a fixed architecture. As a result, when optimal model(s) sit outside this family, their reported uncertainties are quite unreliable. The loss difference between the best prediction available in a family and the "true" predictor is the capacity loss. This loss remains *invisible* if the pooled predictor stays within the family; pooling can otherwise recover some of it. We distinguish the invisible residual from the portion accessible or *visible* to BDL methods (Figure 1). Empirical language-model loss laws include a size-dependent term (Kaplan et al., 2020; Hoffmann et al., 2022; Bahri et al., 2024) that one could loosely identify as this capacity loss or the invisible portion.[1]

Now for the visible portion, even under misspecification, Bayesian posteriors can concentrate around the best supported predictor, with posterior variance on the order of D^{-1} for D independent observations (Kleijn & van der Vaart, 2012). However, neural network symmetries and redundant parametrizations violate the regularity assumptions behind this result (Watanabe, 2013). Empirically, Rosso et al. (2025) study uncertainty scaling across data and model size, and report power-law trends in vision and weak data dependence in language-model fine-tuning. We investigate the behaviour of these visible and invisible portions of the uncertainty in large language models (LLMs) through deep ensembles, and study their implications. Concretely, we investigate:

1. Scaling laws for the ensemble disagreement, in terms of data, model and context size. We empirically report a shallow power law for ensemble spread (the visible portion) across a wide range of open-weight models and our carefully scaled family of character-level models: the disagreement scales as $N^{-\gamma}$ with $\gamma \approx 0.12$–0.16, where N denotes model parameter count; see Figures 2. Figure 5 shows that this ensemble spread decreases more slowly than the invisible portion of epistemic loss. We compare this relationship across different ways of ensembling using geometric vs arithmetic pooling, training-token budgets and context length; see Figures 3 and 14. Beyond moderate training and context budgets, the visible portion changes little (Figure 3a,c).

2. The cost-benefits of ensembling at scale. Separate power laws for the visible and invisible portions predict a growing visible fraction with model size. Their fitted sum gives projected visible shares of 13–37% at 1T parameters across the five fitted families (Figure 5). These extrapolations use separate irreducible-loss assumptions for public and character-level models and require larger-scale experiments for validation, but the prediction, which is corroborated by our limited data suggests that ensembling becomes more effective in capturing epistemic uncertainty at scale. We also corroborate this for context-dependent predictions, where we see a slowly growing correlation between ensemble

[1]The power law with model size could be attributed to either visible plus invisible (i.e., the total reducible loss), or just the invisible portion. At the scale of up to 1T parameters, these two ways of modelling the loss curves remain basically indistinguishable; see Appendix E.3 and Figure 21.

spread and the epistemic uncertainty with model size; see Figure 6. Beyond uncertainty estimation, with a fixed total number of parameters and fixed dataset sizes, ensembling becomes competitive with having a larger model at larger scales in terms of its loss; see Figure 7.

3. Scaling law for ensembling through late branching. A cost-effective approach to creating ensembles is late branching. The question is whether it confers similar benefits as "proper" ensembles that have a completely separate history. Our results suggest that later branching significantly reduces the seed spread, and does so in a predictable way; The drop in the spread relative to a proper ensemble seems to follow a power law: it roughly grows with *square root of the fraction of training tokens shared by the seeds*; see Figure 8. One implication is that ensembling techniques that produce variations around a single seed later in the training are significantly less diverse. This power-law scaling can also be used to predict the average spread for a true seed at a lower cost through late branching, before committing to a large multi-seed experiment.

2 Relation Between Loss, Epistemic Uncertainty and Ensembles

This section first establishes the relationship between loss and epistemic uncertainty and divides the epistemic part of the loss into two components based on their accessibility to an ensemble. This leads us to use geometric spread as a measure of disagreement in an ensemble.

2.1 Epistemic uncertainty as reducible loss

Let X be a context and Y its next token in a finite vocabulary $\mathcal{V}$, with true conditional distribution p_x and model prediction q_x. Write $L(q) = \mathbb{E}_{X,Y}[-\log q_X(Y)]$ for the mean cross-entropy loss. For a fully calibrated model, this loss equals mean predictive entropy. To see this, let $Q = q_X$ denote the model's prediction; full calibration means the true token distribution conditional on the prediction Q is Q itself, that is $\Pr(Y = \cdot \mid Q) = Q$. Conditioning on Q gives

$$L(q) = \mathbb{E}_Q\left[-\sum_y \Pr(Y = y \mid Q)\log Q(y)\right] \overset{\text{calibration}}{=} \mathbb{E}_X H(q_X). \tag{1}$$

A calibrated model can still miss information; for example, the true unigram prediction, which always predicts the marginal token frequencies regardless of the context, is fully calibrated. By definition of calibration, $H(Y \mid Q) = \mathbb{E}_X H(q_X)$ for a fully calibrated model. Since Q is a function of X, the conditional mutual information satisfies

$$I(Y; X \mid Q) = H(Y \mid Q) - H(Y \mid X) = \mathbb{E}_X H(q_X) - H(Y \mid X).$$

Rearranging gives the separation of mean predictive entropy into aleatoric and epistemic uncertainty (Bröcker, 2009).[2]

$$\underbrace{\mathbb{E}_X H(q_X)}_{\substack{\text{mean predictive entropy}\\ \text{total loss } L(q)}} = \underbrace{H(Y \mid X)}_{\substack{\text{aleatoric uncertainty}\\ \text{irreducible loss } E}} + \underbrace{I(Y; X \mid Q)}_{\substack{\text{epistemic uncertainty}\\ \text{reducible loss } \mathcal{E}(q)}} \tag{2}$$

This gives us a clean interpretation of the loss in LLMs; for calibrated models, the reducible loss is epistemic, in the sense that it is concerned with information between the context and the next token that our model fails to capture. Our loss-based interpretation is related to the excess-risk approach of (Lahlou et al., 2023; Kotelevskii et al., 2025).

Empirical studies report good calibration of pretrained language models in several settings (Desai & Durrett, 2020; Kadavath et al., 2022; OpenAI, 2023; Spiess et al., 2025; Nakkiran et al., 2026). Our character-model tests likewise find small held-out recalibration gains (Appendix A).

2.2 The Visible and Invisible Parts of Epistemic Uncertainty in Ensembles

Geometric and arithmetic seed spreads. Let $S \geq 2$ denote the number of ensemble members $q_1, \ldots, q_S$, let $\bar{L}_S = S^{-1}\sum_s L(q_s)$ be their average loss. We consider two standard ways to pool over the members, using geometric and arithmetic pooling:

$$q^{\mathrm{A}}_{S,x}(y) = \frac{1}{S}\sum_s q_{s,x}(y), \qquad q^{\mathrm{G}}_{S,x}(y) = \frac{\exp\{S^{-1}\sum_s \log q_{s,x}(y)\}}{Z_x}. \tag{3}$$

[2]Without calibration, reducible loss also includes a calibration error: $\mathcal{E}(q) = I(Y; X \mid Q) + \mathbb{E}_Q \mathrm{KL}(c(Q)\|Q)$, where $c(Q) = \Pr(Y = \cdot \mid Q)$.

Here $Z_x = \sum_{y \in \mathcal{V}} \exp\{S^{-1} \sum_s \log q_{s,x}(y)\}$ normalizes the geometric pool. Arithmetic pooling averages probabilities, while geometric pooling averages logits or log probabilities and renormalizes. For either choice $P \in \{\mathrm{A}, \mathrm{G}\}$, the *seed spread* is the difference between the average loss of members and the ensemble loss: $\rho_S^P = \bar{L}_S - L(q_S^P)$. The epistemic loss of a seed (on average) can be decomposed as the sum of seed spread plus the reducible loss of the ensemble:

$$\underbrace{\bar{L}_S - E}_{\text{epistemic loss for a single seed}} = \underbrace{\bar{L}_S - L(q_S^P)}_{\text{seed spread } \rho_S^P \text{ (visible)}} + \underbrace{L(q_S^P) - E}_{\text{epistemic loss of the ensemble (invisible)}} . \tag{4}$$

This equality separates the visible portion from the invisible portion. Vyas et al. (2023) study a related separation between initialization variance and finite-width bias. We use $f_S^P = \rho_S^P / (\bar{L}_S - E) \in [0, 1]$ for the "fraction" of the members' epistemic uncertainty visible via the ensemble. While arithmetic pooling is used more often in practice, the geometric pooling is a closer conceptual counterpart to mean-pooling with Gaussian likelihoods in regression; with geometric pooling the seed spread equals the mean divergence from the pool to its members [3] (Wood et al., 2023):

$$\rho_S^{\mathrm{G}} = S^{-1} \sum_s \mathbb{E}_X \, \mathrm{KL}(q_{S,X}^{\mathrm{G}} \| q_{s,X}) = \mathbb{E}_X[-\log Z_X]. \tag{5}$$

This geometric spread plays the role of a variance-of-means with Gaussian ensembling: it is zero when all members agree and equals the loss recovered by the ensemble; its value for a specific context is independent of the next token and depends only on the predictions. In contrast, for arithmetic pooling, the loss reduction depends on the observed next token.

Bayesian active learning by disagreement (BALD). BALD provides a third, and most common measure of disagreement in the ensembles (Houlsby et al., 2011):

$$\underbrace{\mathbb{E}_X H(q_{S,X}^{\mathrm{A}})}_{\text{pooled predictive entropy}} = \underbrace{\frac{1}{S} \sum_s \mathbb{E}_X H(q_{s,X})}_{\text{mean member entropy}} + \underbrace{\frac{1}{S} \sum_s \mathbb{E}_X \, \mathrm{KL}(q_{s,X} \| q_{S,X}^{\mathrm{A}})}_{\text{ensemble disagreement } J_S \text{ (BALD)}} . \tag{6}$$

Relative to the average member, arithmetic pooling lowers log loss by the seed spread and raises entropy by BALD. In the above equation, mean member entropy can retain shared epistemic uncertainty, unlike the aleatoric term in Equation (2), and so BALD does not measure the true epistemic uncertainty. Given the direct connection between spreads and epistemic loss, our focus is mainly on this measure of ensemble diversity, and in particular on the geometric spread due to its simpler form and its connection to variance in ensembling for regression. However, one could also see the similarity of BALD and geometric ensembling in their KL forms in (6) and (5); for nearby predictions, BALD and geometric spread agree to leading quadratic order (Nielsen & Nock, 2014) and our evaluations generally show similar qualitative behaviour for all three measures.

Putting different ensemble sizes on the same scale. Variation around the mean of S independent, identically distributed (iid) samples, divided by S, has an expectation of $(S-1)/S$ times the population variance. For nearby predictions, both spreads and BALD have leading quadratic variance terms (Oehlert, 1992; Nielsen & Nock, 2014). We therefore multiply these measures by $S/(S-1)$ and use the subscript ∞ for the approximate population estimate. For example, $\widehat{\rho}_\infty^{\mathrm{G}} = 2\rho_2^{\mathrm{G}}$ for two members. Note that our ensembles do not generally have iid members, and the justification above does not exactly hold in practice.

3 The Empirical Observations and Their Implications

3.1 The Ensembles Used in Experiments

Our experiments rely on many existing open-weight models and a new family of character-level models that we train specifically for this purpose. Below, we organize the ensemble families by how closely their members are related.

Independent seeds. Pythia is a transformer suite for controlled comparisons (Biderman et al., 2023; van der Wal et al., 2025); we use 14M–410M models with 6–10 seeds. These models are trained on the Pile, a collection of text from 22 sources (Gao et al., 2020). DataDecide, a suite for comparing pretraining datasets (Magnusson et al., 2025), provides three-seed ensembles with

[3] For either pooling rule, simply substituting the defintion of the L terms, we see that the spread is the reduction in divergence from the true conditional distribution: $\rho_S^P = S^{-1} \sum_s \mathbb{E}_X \, \mathrm{KL}(p_X \| q_{s,X}) - \mathbb{E}_X \, \mathrm{KL}(p_X \| q_{S,X}^P)$.

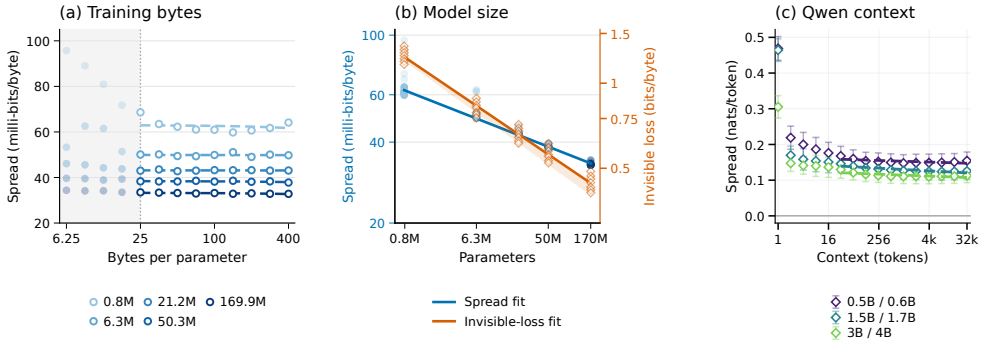

Figure 3: **Spread decreases slowly with size and changes little with more training data or context.** (a): character-model spread for five model sizes on a logarithmic x-axis. Dashed curves show linear fits in bytes per parameter from 25 onward; earlier observations are faded. (b): spread (blue, left axis) and the actual finite ensemble's invisible loss $L(q_S^{\mathrm{G}}) - E$ (orange, right axis), fitted on the same 35 cells at 50–400 bytes per parameter. Orange shading varies the irreducible loss E from 0.593341 to the Kimi-K2-Base cap 0.63764990 bits/byte; points and solid lines use the lower value. (c): spread for three Qwen2.5/Qwen3 base-model pairs on 16,384 fixed PG-19 next-token targets; points show means and $\pm$ one passage standard deviation. Dashed lines fit context budgets above 16 tokens. Appendix D gives the protocol and fitted slopes.

7M–1.28B parameters per member. It uses OLMo, an openly released transformer family (Groeneveld et al., 2024), trained on C4, cleaned web text (Raffel et al., 2020), or Dolma, a mixture of web text, books, code, and other sources (Soldaini et al., 2024).

Release reruns and small recipe changes. We combine repeated Pythia releases, different DataDecide training recipes, and runs with small changes to their training text. Their members span roughly 70M–12B total parameters. Their spread reflects procedural and data differences as well as random seeds.

Larger protocol or architecture changes. We pair models of similar size across two Qwen generations (Yang et al., 2025). The broader comparisons reach 671B parameters per member with the DeepSeek-V3/V3.1 pairing (DeepSeek-AI, 2024), and also include Nemotron-3 Nano/3.5 Lightning (NVIDIA, 2025; 2026), and Gemma-4 31B/26B-A4B (Gemma Team, 2026) pairs. We pool aligned token distributions and score loss and spread per byte. Table 3 lists the public-model pairings in Figure 2.

Our character-level model family. We train byte-level transformers on FineWeb-Edu, web text selected for educational quality (Penedo et al., 2024), with three seeds and 0.79M–170M parameters for 400 bytes per parameter. Training our own models gives every data budget its own learning-rate cooldown; this is missing in existing public checkpoints – e.g., Pythia's intermediate checkpoints have no separate cooldown for each budget. A vocabulary of 256 byte values permits sub-million-parameter models. This scale also makes late-branching experiments possible within our compute budgets. We have carefully optimized the hyperparameters to achieve a clean scaling law for this model family. Appendix B.7 gives the training protocol; Appendix E.3 compares fitted loss laws.

3.2 Scaling of Visible Epistemic Uncertainty with Model, Data and Context

We measured geometric seed spread for all 60 public-model ensembles described above on two validation sets: C4 validation, drawn from 2019 Common Crawl, and recent web text from August 2026 Common Crawl. The recent validation samples were filtered to reduce possible overlap with pretraining data when using the largest models' losses as empirical caps for the irreducible loss; see Appendix B.2 for details.

Scaling with model size. Fitting power laws $\widehat{\rho}_{\infty}^{\mathrm{G}} \propto N^{-\gamma}$ separately within Pythia, DataDecide, and dense Qwen gives geometric-spread exponents between 0.118 and 0.165 on recent text (Appendix Tables 4 and 8). Figure 2 fits the combined DataDecide reference independently of Pythia. See Appendix Figure 13 for C4 validation. Arithmetic spread and BALD show similar exponents in these families, approximately 0.12–

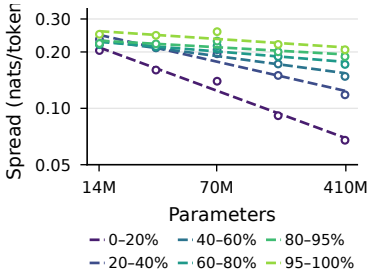

Figure 4: **Spread by difficulty.** Pythia measurements (points) and power fits (dashes) in fixed entropy bins.

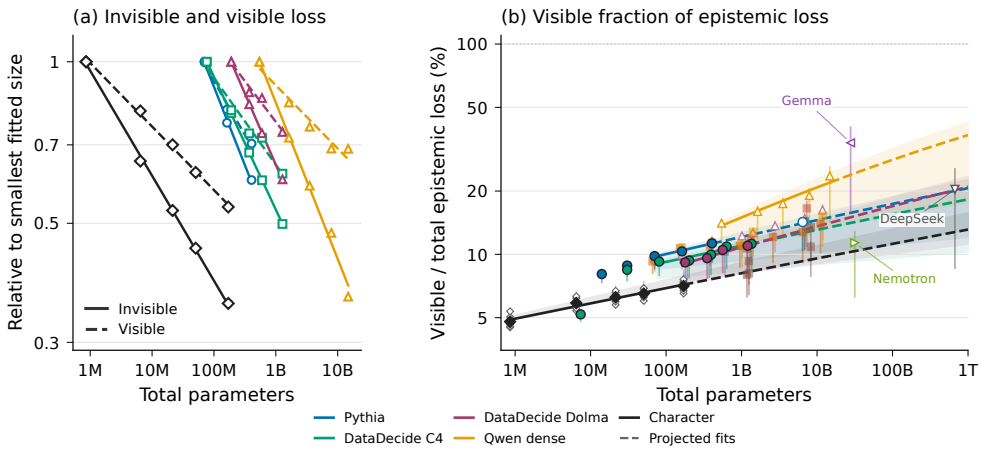

Figure 5: **A larger portion of the epistemic uncertainty is visible at larger scales.** (a): invisible and visible loss normalized at each family's smallest fitted size. Public fits use 17 recent-text cells and a shared floor $E = 0.32436$ nats/byte (profile $[0.26147, 0.33373]$); the character family uses 35 cells and its own $E = 0.593341$ bits/byte, varied to 0.637650. (b): fitted visible fractions, 60 public comparisons, and 35 character cells. Character symbols in (a) also show geometric means. Bands and intervals reflect the respective floor ranges; dashed curves in (b) extrapolate to 1T. Appendix E.1 gives details.

0.15 and 0.12–0.16, respectively; Appendix C.2 gives their figures and family-by-family fits. Our character-level models show the same shallow size trend on held-out FineWeb-Edu text, with exponent 0.118 on the total-parameter axis of Figure 2. Figure 3b uses the analytic parameter count of our training protocol and compares its geometric spread with the invisible portion, $L(q_S^{\mathrm{G}}) - E$, on the same model sizes and data budgets. Under that convention, spread decreases as $N^{-0.117}$, while the invisible portion decreases more steeply, as $N^{-0.191}$ at the reference irreducible loss.

Intuitively, the spread should depend on the difficulty in predicting the next token. We evaluate this hypothesis by binning tokens based on their difficulty level (prediction entropy) according to some model. Figure 4 shows the trend for Pythia seed ensembles: we see a power law trend where the decay exponent depends on the difficulty bin , ranging from 0.31–0.36 for easy tokens to 0.04–0.06 for difficult tokens as the model size grows. Difficult positions account for a growing share of the spread, and their slow resolution with model size can intuitively explain the mild scaling of the spread; Appendix C has the details.

Scaling with data. Since for multi-seed open-weight model families we do not have clean checkpoints after cooldown and the learning rate varies during the training due to cosine scheduling, we cannot reliably measure spread with number of tokens in those model families. However, for our character-level family we could do so, and the fitted net spread changes from 25 to 400 bytes per parameter are below 2%, despite larger nonmonotonic variation (Figure 3a). Appendix B.7 has the details.

Scaling with context. Context length is another axis along which language-model loss can decrease approximately as a power law (Kaplan et al., 2020; Xiong et al., 2024). Since our character-level model has a relatively short context length, we consider Pythia and Qwen families. Figure 3c shows geometric spread for three Qwen2.5/Qwen3 base-model pairs on fixed targets as context grows to 32,768 tokens. After a substantial initial contraction, spread depends only weakly on additional context. Pythia seed ensembles show the same qualitative behaviour through 2k tokens (Figure 18). Appendix D gives the evaluation protocol and additional results, including both types of spread and BALD for Pythia and Qwen families.

3.3 A Larger Share of Epistemic Uncertainty is Visible at Scale

A direct power law for total epistemic loss, $\bar{L}_S - E \simeq KN^{-\eta}$, combined with spread $\rho_\infty^P \simeq c_P N^{-\gamma_P}$ predicts a visible fraction which is their ratio $f_\infty^P \simeq (c_P/K)N^{\eta-\gamma_P}$. When $\eta > \gamma_P$, this extrapolation eventually exceeds 100%. Alternatively, separate power laws for the visible (V) and invisible (I) portions give

$$V_\infty(N) = cN^{-\gamma}, \qquad I_\infty(N) = AN^{-\alpha}, \qquad f_\infty(N) = \frac{V_\infty(N)}{I_\infty(N) + V_\infty(N)}. \tag{7}$$

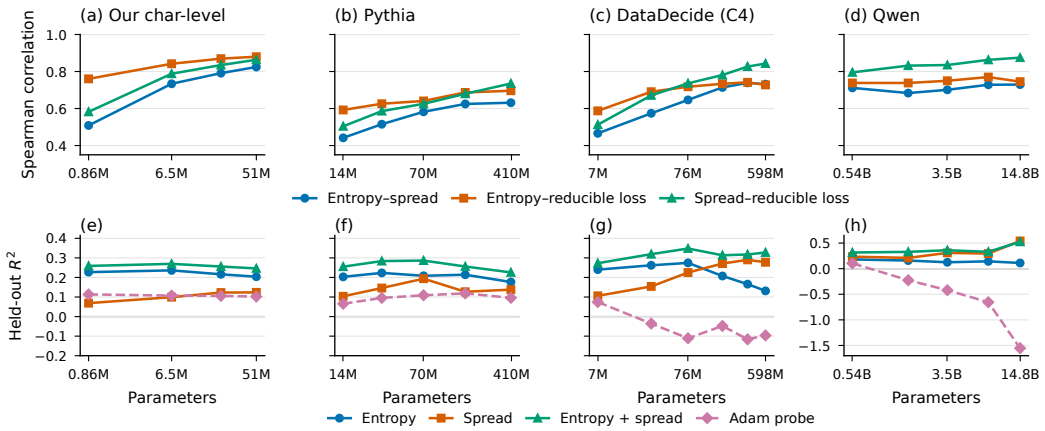

Figure 6: **Uncertainty across contexts.** (a–d): correlations among single-model entropy, spread, and reference-based reducible loss. (e–h): held-out R^2 for predicting this loss from entropy, spread, their combination, or a hidden-state probe. Qwen uses Qwen2.5/Qwen3 base-model pairs, Qwen2.5 entropy and hidden states, and geometric mean parameter counts. Panel (h) has a separate vertical scale. Spread outperforms entropy for Qwen; their combination often predicts best across families.

Here, the fraction stays below 100%, but the sum need not itself be a power law. The question becomes which one of these two alternatives better explains the data. Comparisons on our character-level model, Pythia, and Qwen give similar fit quality for the two approaches, with no consistent advantage (Appendix E.3). Identifying the correct model requires larger families, where predictions for the visible fraction clearly diverge because the second model saturates. Figure 5 shows the fit with the second approach, where irreducible loss drives the uncertainty in (b). Appendix E.1 has the details, limitations and additional results.

These results suggest that at larger scales, the ensemble covers a larger fraction of the total epistemic uncertainty and ensembling becomes more effective in estimating the uncertainty. Absolute disagreement can decrease (Kirsch, 2025), while its recovered fraction grows if reducible loss falls faster.

Context-dependent Epistemic Uncertainty While the results so far suggest ensembles can identify a larger portion of the epistemic uncertainty at scale, here we briefly consider its context-dependent prediction at existing scales. Figure 6a–d shows pairwise correlations among single-seed, ensemble spread (visible epistemic), and reference-based reducible loss (epistemic) and entropy (epistemic + aleatoric) in four model families. Since the true next-token distribution is unknown, we approximate the total reducible loss using the mean forward KL from a stronger reference model. The references are BLT for our character-level models, MPT-30B for Pythia, OLMo-7B for DataDecide, and Qwen2.5-72B for Qwen. We see all three values are positively correlated, and the correlations generally increase with model size. Since both ensemble spread and entropy can predict epistemic uncertainty, we then assess their accuracy for this "prediction" task. To this end, we fit affine predictors of the epistemic uncertainty estimates using entropy, geometric spread, or both, and compare them with a linear probe of the same seed's hidden states, adapting the protocol of Ahdritz et al. (2024) (Appendix F.3.1).

Figure 6e–h reports held-out R^2, which shows that entropy and ensemble spread often outperform a linear probe. The figures also suggest that spread becomes more correlated with total epistemic uncertainty and a better predictor as the models scale. For the Qwen family, which has the largest models, spread correlates more strongly with epistemic uncertainty and consistently outperforms entropy. This is consistent with the slow growth of the visible fraction we saw earlier. See Appendix F.3 for details and additional results.

3.4 Ensembling as an Alternative to a Larger Model

A fixed total parameter budget can be allocated to one large model or an ensemble of smaller models. The power-law scaling of spread suggests ensembling using S members can become more competitive at scale – specifically, when the loss reduction captured by its spread exceeds the improvement from the larger model:

$$\rho_S^P > \bar{L}_S - L(q_{\text{big}}) \tag{8}$$

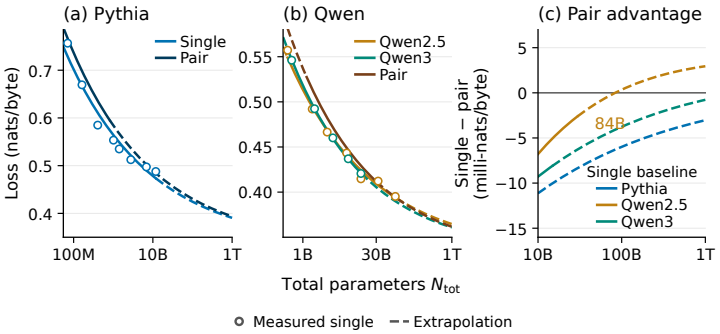

Figure 7: **Pair losses and extrapolations.** (a,b): measured losses and fitted single/pair curves versus total parameters. (c): conditional projections at matched total parameters.

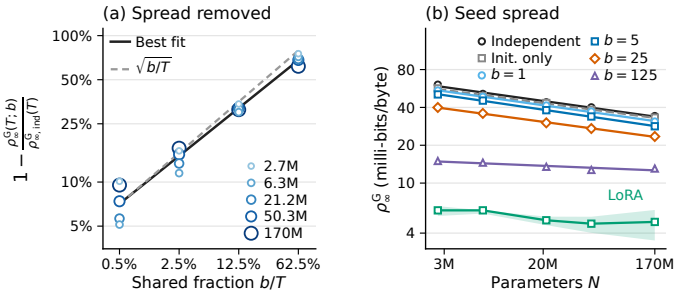

Figure 8: **Shared training reduces ensemble spread.** (a) Fraction of independent-seed spread lost versus the shared fraction of training, b/T, where b is the branching point and T the final training budget. The loss of spread follows an approximate square-root law. (b) Geometric spread versus model size for independent seeds, late branching, and LoRA ensembles.

We compare an ensemble of S size-N models with one size-SN model, each trained on D tokens. These assumptions match total parameters and leading-order dense inference and training compute.[4] Now suppose loss follows $L(N, D) = E + aN^{-\alpha} + bD^{-\beta}$, and population spread $\widehat{\rho}^P_\infty = \rho^P_S/(1 - S^{-1})$ follows $c_P N^{-\gamma_P}$. The common terms $E + bD^{-\beta}$ on the rhs of Equation (8) cancel, and we have an advantage to ensembling when

$$(1 - S^{-1})c_P N^{-\gamma_P} > a[N^{-\alpha} - (SN)^{-\alpha}] \quad \Longleftrightarrow \quad c_P N^{\alpha-\gamma_P} > a\frac{1 - S^{-\alpha}}{1 - S^{-1}}. \tag{9}$$

When $\alpha > \gamma_P$, the equation above predicts a regime of increasing competitiveness for ensembling with scale. Figure 7 corroborates this prediction: smaller pair ensembles generally lose to larger single models, while our largest ensemble outperforms the larger model: geometric pooling of Qwen2.5-14B and Qwen3-14B-Base achieves 0.40665 nats/byte, below Qwen2.5-32B's 0.41223, using 29.54B total parameters versus 32.76B. Neither member alone beats this reference; this particular comparison using Qwen mixes training recipes, so it does not fully reflect the theoretical prediction, but it conveys the trend; see Appendix Table 23. The fitted curves extend the comparison to matched total parameter budgets, showing where ensemble spread overtakes the benefit of increasing model size—the tradeoff expressed by Equation (9). These projected crossovers depend on the assumed irreducible loss. Appendix G gives the detailed comparisons and their limitations.

These findings complement Kim et al. (2026), who show that ensembling can outperform larger single language models under extensive data reuse and tuned regularization. Our analysis derives a crossover criterion from the relative scaling of loss and ensemble spread, with conditional projections based on existing open-weight models. The criterion assumes the stated scaling laws and budget constraints, without requiring a particular regularization scheme. Related parameter-budget tradeoffs in image classification were studied by Lobacheva et al. (2020), who also find regimes where several medium-size networks outperform one larger network at the same total parameter count.

3.5 Ensemble Spread with Late Branching and LoRA Ensembles

Late branching reduces ensemble cost by starting the process late in training. Ensemble members will have the same initialization and early updates. We first study this type of ensemble with full-parameter continuations, then with LoRA ensembles, where training continues on low-rank adaptors.

[4] However, for a typical compute-optimal regime of $D \propto N$, training cost scales as N^2 and so S size-N runs match a single size-$\sqrt{S}N$ run trained on $\sqrt{S}D$ tokens and both its size and data benefits should be included.

The spread follows an approximate square-root law. Using the FineWeb-Edu byte models, we branch after $b \in \{1, 5, 25, 125\}$ bytes per parameter and evaluate at a final budget $T = 200$. The parent and its continuation preserve their shared training state, then see the same remaining data in different orders; all parameters remain trainable. Independent references for this section change both initialization and data order.[5] Appendix H gives the details and additional results. We define the retained share of spread as $R(N, t, b) = \frac{\rho^{\mathrm{G}}_{\infty,\mathrm{branch}}(N,t,b)}{\rho^{\mathrm{G}}_{\infty,\mathrm{ind}}(N,t)}$. where $\rho_{\infty,\mathrm{branch}}$ and $\rho_{\infty,\mathrm{ind}}$ denote the late branching and reference ensemble spreads. The fitted law shown in Figure 8 suggests a clear power-law scaling of the share of spread not retained:

$$1 - R(N, T, b) \simeq 0.85 \left(\frac{b}{T}\right)^{0.47}. \tag{10}$$

For example, branching halfway through training is predicted to lose approximately 61% of the independent-seed spread; extrapolating to branching with only 10% of the run remaining gives a loss of approximately 81%.

These results show that late branching substantially reduces ensemble diversity, highlighting a cost of sharing training history to construct ensembles efficiently (Wasay et al., 2020). This result is also consistent with the greater functional variation of independent restarts relative to local weight sampling as reported by (Fort et al., 2019), and suggests a potential limitation for BDL methods that approximate the posterior using the final portions of the training trajectory (Maddox et al., 2019) .

LoRA ensembles. LoRA ensembles (Wang et al., 2023) and Bayesian adapters (Yang et al., 2024; Lin et al., 2026; Rahmati et al., 2025) also share a pretrained backbone, so we measure spread from independently trained adapters on a late checkpoint. Here, we compare the spread of LoRA in late branching to full seed, and see that LoRA pairs branched at $b = 125$ retain about 9–18% of independent-pair spread, a further 2.1–3.7-fold reduction relative to full-parameter branches (Figure 8b). Higher rank improves individual-model loss and increases spread at the three largest widths, but even rank $d/2$ retains less than half of full-branch spread (Appendix H.5; Figure 22). Mühlematter et al. (2026) report favourable diversity and calibration for LoRA ensembles in classification. However, their shared-backbone fine-tuning baseline differs from our independent-pretraining reference.

4 Discussion and limitations

Here we discuss the theoretical and empirical upshot from our study. It is fair to say that many of its fragments are already present in the past literature, yet an aggregate picture and its implications, to our knowledge, have been missing. On the theoretical side, we argue that reducible loss is the correct average measure of epistemic uncertainty for calibrated LLMs – this calibration condition is important because in (2), the equality of the mutual information term that has a clear epistemic interpretation to the reducible loss depends on it. Moreover, the spread, rather than BALD, and in particular the geometric spread, is the suitable way to assess ensemble disagreement due to its direct connection to the epistemic loss. In this context, the existence of two clear scaling laws with distinct ranges for their exponents has important implications. As models grow, a growing fraction of true epistemic uncertainty for a given context becomes visible through ensemble diversity, and ensembling becomes competitive even for better prediction, independent of its use for uncertainty estimation. Finally, a third scaling law suggests that cheap approximations through late branching and LoRA significantly reduce the spread.

The most important limitation of our results and predictions is uncertainty in the irreducible loss. While some of the best open weight models, including DeepSeek-V4-Pro-Base (DeepSeek-AI, 2026) and Kimi-K2-Base (Moonshot AI, 2025) were used to get empirical upper caps for the floor, its plausible range is still rather wide (Figures 5 and 19). These and other limitations are further elaborated in the appendix and referenced in different sections.

[5] Compared to ensembles that differ only in initialization, changing the data order increases the spread slightly, on average by about 4%.

AI USE STATEMENT

We used LLMs to assist with implementing our new character-level model, evaluating existing models, and analyzing the results. They have also been used for literature retrieval and citation checking, drafting, polishing, and figure generation.

REFERENCES

Gustaf Ahdritz, Tian Qin, Nikhil Vyas, Boaz Barak, and Benjamin L. Edelman. Distinguishing the Knowable from the Unknowable with Language Models, 2024. URL https://proceedings.mlr.press/v235/ahdritz24a.html. ICML.

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in AI safety, 2016. URL https://arxiv.org/abs/1606.06565. arXiv:1606.06565.

Yasaman Bahri, Ethan Dyer, Jared Kaplan, Jaehoon Lee, and Utkarsh Sharma. Explaining Neural Scaling Laws, 2024. URL https://arxiv.org/abs/2102.06701. PNAS; preprint first posted 2021.

Yavuz Faruk Bakman, Sungmin Kang, Zhiqi Huang, Duygu Nur Yaldiz, Catarina Belém, Chenyang Zhu, Anoop Kumar, Alfy Samuel, Daben Liu, Salman Avestimehr, and Sai Karimireddy. Uncertainty as Feature Gaps: Epistemic Uncertainty Quantification of LLMs in Contextual Question-Answering. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/b904bc78eadc7969d04613d903dcb046-Abstract-Conference.html.

Freddie Bickford Smith, Jannik Kossen, Eleanor Trollope, Mark van der Wilk, Adam Foster, and Tom Rainforth. Rethinking Aleatoric and Epistemic Uncertainty. In *International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 4345–4359, 2025. URL https://proceedings.mlr.press/v267/bickford-smith25a.html.

Stella Biderman et al. Pythia: A Suite for Analyzing Large Language Models Across Training and Scaling, 2023. URL https://arxiv.org/abs/2304.01373. ICML.

Sid Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, Michael Pieler, USVSN Sai Prashanth, Shivanshu Purohit, Laria Reynolds, Jonathan Tow, Ben Wang, and Samuel Weinbach. GPT-NeoX-20B: An Open-Source Autoregressive Language Model, 2022. URL https://arxiv.org/abs/2204.06745. arXiv:2204.06745.

Mark Braverman, Xinyi Chen, Sham Kakade, Karthik Narasimhan, Cyril Zhang, and Yi Zhang. Calibration, entropy rates, and memory in language models. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 1089–1099, 2020. URL https://proceedings.mlr.press/v119/braverman20a.html.

Jochen Bröcker. Reliability, sufficiency, and the decomposition of proper scores. *Quarterly Journal of the Royal Meteorological Society*, 135(643):1512–1519, 2009. doi: 10.1002/qj.456. URL https://arxiv.org/abs/0806.0813.

Steven Cao, Gregory Valiant, and Percy Liang. On the entropy calibration of language models. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/7ab0b27a78dcc27755b55d1bc8274918-Abstract-Conference.html.

Erik Daxberger, Agustinus Kristiadi, Alexander Immer, Runa Eschenhagen, Matthias Bauer, and Philipp Hennig. Laplace Redux – Effortless Bayesian Deep Learning, 2021. URL https://arxiv.org/abs/2106.14806. NeurIPS.

DeepSeek-AI. DeepSeek-V3 Technical Report, 2024. URL https://arxiv.org/abs/2412.19437. arXiv:2412.19437.

DeepSeek-AI. DeepSeek-V4-Pro-Base. Hugging Face model release, 2026. URL https://huggingface.co/deepseek-ai/DeepSeek-V4-Pro-Base. Checkpoint revision 98730c030fbdbaca4950788280a35c4642b208a9.

Shrey Desai and Greg Durrett. Calibration of pre-trained transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 295–302, 2020. doi: 10.18653/v1/2020.emnlp-main.21. URL https://aclanthology.org/2020.emnlp-main.21/.

EvaByte. EvaByte Model Card, 2025. URL https://huggingface.co/EvaByte/EvaByte. Base model release.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. Detecting hallucinations in large language models using semantic entropy, 2024. URL https://www.nature.com/articles/s41586-024-07421-0. Nature 630.

Stanislav Fort, Huiyi Hu, and Balaji Lakshminarayanan. Deep ensembles: A loss landscape perspective, 2019. URL https://arxiv.org/abs/1912.02757. arXiv preprint.

Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning, 2016. URL https://proceedings.mlr.press/v48/gal16.html. ICML.

Leo Gao et al. The Pile: An 800GB Dataset of Diverse Text for Language Modeling, 2020. URL https://arxiv.org/abs/2101.00027. arXiv:2101.00027.

Gemma Team. Gemma 4 Technical Report, 2026. URL https://arxiv.org/abs/2607.02770. arXiv:2607.02770.

Alex Graves. Practical Variational Inference for Neural Networks. In *Advances in Neural Information Processing Systems*, volume 24, pp. 2348–2356. Curran Associates, Inc., 2011. URL https://proceedings.neurips.cc/paper/2011/hash/7eb3c8be3d411e8ebfab08eba5f49632-Abstract.html.

Dirk Groeneveld et al. OLMo: Accelerating the Science of Language Models, 2024. URL https://arxiv.org/abs/2402.00838. arXiv:2402.00838.

Sebastian Gruber and Florian Buettner. Uncertainty estimates of predictions via a general bias-variance decomposition. In *Proceedings of the 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pp. 11331–11354, 2023. URL https://proceedings.mlr.press/v206/gruber23a.html.

Dylan Hadfield-Menell, Anca Dragan, Pieter Abbeel, and Stuart Russell. The off-switch game. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, pp. 220–227, 2017. doi: 10.24963/ijcai.2017/32. URL https://www.ijcai.org/proceedings/2017/32.

Jordan Hoffmann et al. Training Compute-Optimal Large Language Models, 2022. URL https://arxiv.org/abs/2203.15556. NeurIPS.

Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian active learning for classification and preference learning, 2011. URL https://arxiv.org/abs/1112.5745. arXiv:1112.5745.

Gao Huang, Yixuan Li, Geoff Pleiss, Zhuang Liu, John E. Hopcroft, and Kilian Q. Weinberger. Snapshot ensembles: Train 1, get M for free. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=BJYwwY9ll.

Saurav Kadavath et al. Language Models (Mostly) Know What They Know, 2022. URL https://arxiv.org/abs/2207.05221. arXiv preprint.

Jared Kaplan et al. Scaling Laws for Neural Language Models, 2020. URL https://arxiv.org/abs/2001.08361. arXiv paper.

Konwoo Kim, Suhas Kotha, Percy Liang, and Tatsunori Hashimoto. Pre-training under infinite compute, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/79682ef54677a2cb3dfb8383b372037f-Abstract-Conference.html. ICLR; preprint first posted 18 September 2025.

Andreas Kirsch. (Implicit) Ensembles of Ensembles: Epistemic Uncertainty Collapse in Large Models, 2025. URL https://arxiv.org/abs/2409.02628v2. TMLR; preprint first posted 2024.

B. J. K. Kleijn and A. W. van der Vaart. The Bernstein–von Mises theorem under misspecification. *Electronic Journal of Statistics*, 6:354–381, 2012. doi: 10.1214/12-EJS675. URL https://doi.org/10.1214/12-EJS675.

Nikita Kotelevskii, Vladimir Kondratyev, Martin Takáč, Eric Moulines, and Maxim Panov. From risk to uncertainty: Generating predictive uncertainty measures via Bayesian estimation. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/e6f32e64b9c27d153b46c94f0fe22b56-Abstract-Conference.html.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic Uncertainty: Linguistic Invariances for Uncertainty Estimation in Natural Language Generation, 2023. URL https://arxiv.org/abs/2302.09664. ICLR.

A. Gilad Kusne, Heshan Yu, Changming Wu, Huairuo Zhang, Jason Hattrick-Simpers, Brian DeCost, Suchismita Sarker, Corey Oses, Cormac Toher, Stefano Curtarolo, Albert V. Davydov, Ritesh Agarwal, Leonid A. Bendersky, Mo Li, Apurva Mehta, and Ichiro Takeuchi. On-the-fly closed-loop materials discovery via Bayesian active learning. *Nature Communications*, 11:5966, 2020. doi: 10.1038/s41467-020-19597-w. URL https://www.nature.com/articles/s41467-020-19597-w.

Salem Lahlou, Moksh Jain, Hadi Nekoei, Victor Ion Butoi, Paul Bertin, Jarrid Rector-Brooks, Maksym Korablyov, and Yoshua Bengio. DEUP: Direct epistemic uncertainty prediction. *Transactions on Machine Learning Research*, 2023. URL https://openreview.net/forum?id=eGLdVRvvfQ.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles, 2017. URL https://arxiv.org/abs/1612.01474. NeurIPS.

Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, et al. DataComp-LM: In Search of the Next Generation of Training Sets for Language Models, 2024. URL https://arxiv.org/abs/2406.11794. arXiv:2406.11794.

Moule Lin, Shuhao Guan, Andrea Patane, David Gregg, and Goetz Botterweck. Bayesian-LoRA: Probabilistic Low-Rank Adaptation of Large Language Models. In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2601.21003.

Ekaterina Lobacheva, Nadezhda Chirkova, Maxim Kodryan, and Dmitry Vetrov. On Power Laws in Deep Ensembles, 2020. URL https://arxiv.org/abs/2007.08483. NeurIPS.

Wesley J. Maddox, Pavel Izmailov, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. A Simple Baseline for Bayesian Uncertainty in Deep Learning, 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/hash/118921efba23fc329e6560b27861f0c2-Abstract.html. NeurIPS.

Ian Magnusson, Nguyen Tai, Ben Bogin, David Heineman, Jena D. Hwang, Luca Soldaini, Akshita Bhagia, Jiacheng Liu, Dirk Groeneveld, Oyvind Tafjord, Noah A. Smith, Pang Wei Koh, and Jesse Dodge. DataDecide: How to Predict Best Pretraining Data with Small Experiments, 2025. URL https://arxiv.org/abs/2504.11393. arXiv paper.

Andrey Malinin and Mark Gales. Uncertainty Estimation in Autoregressive Structured Prediction, 2021. URL https://arxiv.org/abs/2002.07650. ICLR.

Moonshot AI. Kimi-K2-Base. Hugging Face model release, 2025. URL https://huggingface.co/moonshotai/Kimi-K2-Base. Checkpoint revision ce72df012259dcc55d945e890f815fe7ef69159c.

Niklas Muennighoff, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Jacob Morrison, et al. OLMoE: Open Mixture-of-Experts Language Models, 2024. URL https://arxiv.org/abs/2409.02060. arXiv:2409.02060.

Dominik J. Mühlematter, Michelle Halbheer, Alexander Becker, Dominik Narnhofer, Helge Aasen, Konrad Schindler, and Mehmet Ozgur Turkoglu. LoRA-Ensemble: Efficient uncertainty modelling for self-attention networks. *Transactions on Machine Learning Research*, 2026. URL https://openreview.net/forum?id=yhXXmOMpSQ.

Preetum Nakkiran, Arwen Bradley, Adam Golinski, Eugene Ndiaye, Michael Kirchhof, and Sinead Williamson. Trained on Tokens, Calibrated on Concepts: The Emergence of Semantic Calibration in LLMs. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/39b228b1a83232b14a1c81b0ed5a7a05-Abstract-Conference.html.

Frank Nielsen and Richard Nock. On the chi square and higher-order chi distances for approximating f-divergences. *IEEE Signal Processing Letters*, 21(1):10–13, 2014. doi: 10.1109/LSP.2013.2288355. URL https://arxiv.org/abs/1309.3029.

NVIDIA. NVIDIA Nemotron 3 Nano 30B-A3B Base BF16: Model card. https://huggingface.co/nvidia/NVIDIA-Nemotron-3-Nano-30B-A3B-Base-BF16, 2025.

NVIDIA. NVIDIA Nemotron 3.5 Lightning 30B-A3B Base BF16: Model card. https://huggingface.co/nvidia/NVIDIA-Nemotron-3.5-Lightning-30B-A3B-Base-BF16, 2026.

Gary W. Oehlert. A note on the delta method. *The American Statistician*, 46(1):27–29, 1992. doi: 10.1080/00031305.1992.10475842. URL https://doi.org/10.1080/00031305.1992.10475842.

OpenAI. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. URL https://arxiv.org/abs/2303.08774.

Artidoro Pagnoni, Ram Pasunuru, Pedro Rodriguez, John Nguyen, Benjamin Muller, Margaret Li, Chunting Zhou, Lili Yu, Jason Weston, Luke Zettlemoyer, Gargi Ghosh, Mike Lewis, Ari Holtzman, and Srinivasan Iyer. Byte Latent Transformer: Patches Scale Better Than Tokens, 2024. URL https://arxiv.org/abs/2412.09871. arXiv:2412.09871.

Guilherme Penedo et al. The FineWeb Datasets: Decanting the Web for the Finest Text Data at Scale, 2024. URL https://arxiv.org/abs/2406.17557. arXiv:2406.17557.

Qwen Team. Qwen2.5-72B: Base model card. https://huggingface.co/Qwen/Qwen2.5-72B, 2024. Final pretrained release; accessed September 14, 2026.

Qwen Team. Qwen3-14B-Base: Model card. https://huggingface.co/Qwen/Qwen3-14B-Base, 2025. Final pretrained release; accessed September 14, 2026.

Jack W. Rae, Anna Potapenko, Siddhant M. Jayakumar, and Timothy P. Lillicrap. Compressive transformers for long-range sequence modelling, 2019. URL https://arxiv.org/abs/1911.05507.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL https://www.jmlr.org/papers/v21/20-074.html.

Amir Hossein Rahmati, Sanket Jantre, Weifeng Zhang, Yucheng Wang, Byung-Jun Yoon, Nathan Urban, and Xiaoning Qian. C-LoRA: Contextual low-rank adaptation for uncertainty estimation in large language models. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/615e48e9f5200d8f6d31ae08a348a497-Abstract-Conference.html.

Tom Rainforth, Adam Foster, Desi R. Ivanova, and Freddie Bickford Smith. Modern Bayesian experimental design. *Statistical Science*, 39(1):100–114, 2024. doi: 10.1214/23-STS915. URL https://doi.org/10.1214/23-STS915.

Mattia Rosso, Simone Rossi, Giulio Franzese, Markus Heinonen, and Maurizio Filippone. Scaling Laws for Uncertainty in Deep Learning, 2025. URL https://arxiv.org/abs/2506.09648v2. arXiv preprint; v2 revised 9 February 2026.

Benjamin Samuel Ruben, William Lingxiao Tong, Hamza Tahir Chaudhry, and Cengiz Pehlevan. No free lunch from random feature ensembles: Scaling laws and near-optimality conditions. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 52198–52224, 2025. URL https://proceedings.mlr.press/v267/ruben25a.html.

Stuart Russell. *Human Compatible: Artificial Intelligence and the Problem of Control*. Viking, 2019. URL https://www.penguinrandomhouse.com/books/566677/human-compatible-by-stuart-russell/.

Luca Soldaini et al. Dolma: An Open Corpus of Three Trillion Tokens for Language Model Pretraining Research, 2024. URL https://arxiv.org/abs/2402.00159. arXiv:2402.00159.

Claudio Spiess, David Gros, Kunal Suresh Pai, Michael Pradel, Md Rafiqul Islam Rabin, Amin Alipour, Susmit Jha, Prem Devanbu, and Toufique Ahmed. Calibration and correctness of language models for code. In *2025 IEEE/ACM 47th International Conference on Software Engineering (ICSE)*, pp. 540–552, 2025. doi: 10.1109/ICSE55347.2025.00040. URL https://www.software-lab.org/publications/icse2025_calibration.pdf.

The Databricks AI Research Team. MPT-30B: Raising the Bar for Open-Source Foundation Models, 2023. URL https://www.databricks.com/blog/mpt-30b. Model release, June 22.

Oskar van der Wal, Pietro Lesci, Max Müller-Eberstein, Naomi Saphra, Hailey Schoelkopf, Willem Zuidema, and Stella Biderman. PolyPythias: Stability and outliers across fifty language model pre-training runs. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/d611d06e3207330555fbc10810e70163-Abstract-Conference.html.

Nikhil Vyas, Alexander Atanasov, Blake Bordelon, Depen Morwani, Sabarish Sainathan, and Cengiz Pehlevan. Feature-learning networks are consistent across widths at realistic scales. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/03600ae6c3392fd65ad7c3a90c6f7ce8-Abstract-Conference.html.

Xi Wang, Laurence Aitchison, and Maja Rudolph. LoRA ensembles for large language model fine-tuning, 2023. URL https://arxiv.org/abs/2310.00035. arXiv:2310.00035.

Yibin Wang, Haizhou Shi, Ligong Han, Dimitris Metaxas, and Hao Wang. BLoB: Bayesian Low-Rank Adaptation by Backpropagation for Large Language Models, 2024. URL https://arxiv.org/abs/2406.11675. NeurIPS.

Abdul Wasay, Brian Hentschel, Yuze Liao, Sanyuan Chen, and Stratos Idreos. MotherNets: Rapid Deep Ensemble Learning. In *Proceedings of Machine Learning and Systems*, 2020. URL https://proceedings.mlsys.org/paper_files/paper/2020/file/4a420924d20bc025ebb37849169e6ebd-Paper.pdf.

Sumio Watanabe. A widely applicable Bayesian information criterion. *Journal of Machine Learning Research*, 14(27):867–897, 2013. URL https://jmlr.org/papers/v14/watanabe13a.html.

Danny Wood, Tingting Mu, Andrew M. Webb, Henry W. J. Reeve, Mikel Luján, and Gavin Brown. A unified theory of diversity in ensemble learning. *Journal of Machine Learning Research*, 24 (359):1–49, 2023. URL https://jmlr.org/papers/v24/23-0041.html.

Wenhan Xiong, Jingyu Liu, Igor Molybog, Hejia Zhang, Prajjwal Bhargava, Rui Hou, Louis Martin, Rashi Rungta, Karthik Abinav Sankararaman, Barlas Oguz, Madian Khabsa, Han Fang, Yashar Mehdad, Sharan Narang, Kshitiz Malik, Angela Fan, Shruti Bhosale, Sergey Edunov, Mike Lewis, Sinong Wang, and Hao Ma. Effective Long-Context Scaling of Foundation Models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 4643–4663. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.naacl-long.260. URL https://aclanthology.org/2024.naacl-long.260/.

Adam X. Yang, Maxime Robeyns, Xi Wang, and Laurence Aitchison. Bayesian Low-rank Adaptation for Large Language Models, 2024. URL https://arxiv.org/abs/2308.13111. ICLR.

An Yang et al. Qwen3 Technical Report, 2025. URL https://arxiv.org/abs/2505.09388. arXiv:2505.09388.

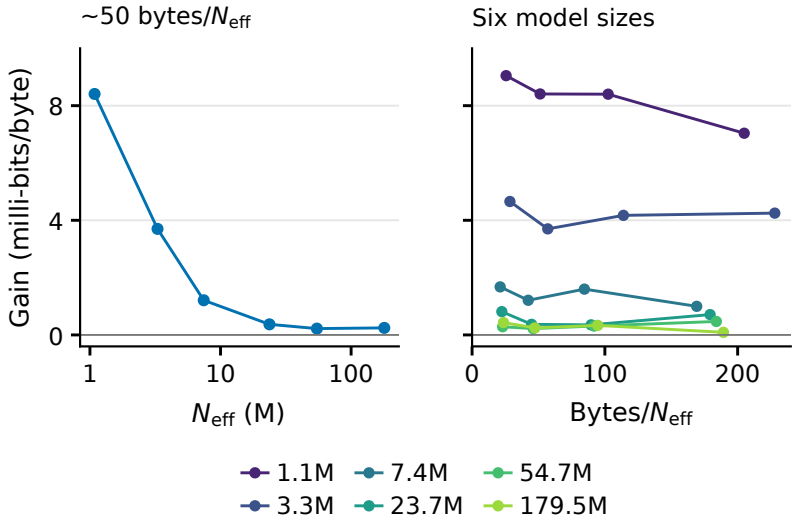

Figure 9: **Recalibration across model and data size.** Geometric-ensemble loss recovered by an MLP recalibrator on 2,097,152 held-out bytes. Left: dependence on FLOP-equivalent size N_{eff} at the available budget nearest 50 bytes/N_{eff}. Right: all six sizes and four training budgets. These are achieved corrections, which estimate ideal full-forecast calibration error from below.

A Calibration

We test forecast-only recalibration in the byte models and Pythia, and confidence calibration in Pythia and Qwen. These tests detect particular calibration errors; they do not establish full forecast calibration.

A.1 What recalibration measures

For the forecast $Q = q_X$, let $c(Q) = \Pr(Y = \cdot \mid Q)$ and $C_{\text{cal}}(q) = \mathbb{E}\,\mathrm{KL}(c(Q)\|Q)$. A map a that uses only Q has population loss gain

$$L(q) - L(a \circ q) = C_{\text{cal}}(q) - \mathbb{E}\,\mathrm{KL}(c(Q)\|a(Q)) \le C_{\text{cal}}(q).$$

An achieved gain therefore estimates calibration error from below. Small gains can reflect either calibration or a limited correction class. Held-out gains also have sampling error and can be negative.

A.2 Character-level models

We evaluate 24 settings: widths $128, 192, 256, 384, 512, 768$, each at the available budgets nearest in $\log(D/N_{\text{eff}})$ to $25, 50, 100, 200$ bytes/N_{eff}. Here N_{eff} is the checkpoint's FLOP-equivalent count, which includes the output projection and attention computation; it is not an exact non-embedding parameter count or the analytic count in Appendix B.7.1. Values span 1.085M–179.503M and actual budgets 21.1–228.0 bytes/N_{eff}. Width 192 retains seeds 0 and 1; other widths use seeds 0–2. This population differs from the primary spread fit, which omits width 192 and seed 1 at width 128.

Temperature scaling, per-class scale and bias, and a residual MLP use respectively 1, 512, and 131,584 parameters. The MLP takes the 256 log probabilities as input, uses 256 hidden units, and adds a logit correction. Training uses 25 AdamW epochs, batch size 8,192, learning rate 10^{-3} with cosine decay, weight decay 10^{-2}, and seed 0. Vector scaling uses L-BFGS for at most 150 iterations.

We score 4,096 held-out sequences of 512 bytes. Even and odd sequences form fitting and scoring folds of 1,048,576 bytes each; we exchange their roles and average scoring results. Members and their finite geometric pool are recalibrated separately, without a population correction. Figure 9 and Table 1 report the results.

Geometric-pool MLP gains range from 0.091 to 9.047 milli-bits/byte. At $E = 0.593341$ bits/byte, they recover at most 0.677% of reducible loss; at the higher floor 0.63764990, the maximum is 0.702%. At $N_{\text{eff}} \ge 23.7$M, gains are at most 0.812 milli-bits/byte. The gain generally falls more with model size than with training budget, although some budget curves are nonmonotonic.

Table 1: Calibration evaluation of our character-level models on 2,097,152 held-out bytes. N_{eff} is the FLOP-equivalent count, including attention computation. Each entry is a signed held-out loss reduction in milli-bits/byte. Columns T, V, and M denote temperature, vector, and MLP recalibration of the geometric ensemble; the member column averages separate MLP recalibrations. The final column divides the ensemble MLP gain by its reducible loss at $E = 0.593341$ bits/byte. All 24 preselected cells are retained.

N_{eff} (M)	Bytes/N_{eff}	Ens. T	Ens. V	Ens. M	Member M	Gain / reducible loss
1.09	25.6	0.154	5.452	9.047	8.677	0.671%
1.09	51.2	0.465	5.253	8.409	8.232	0.652%
1.09	102.5	0.251	5.370	8.401	8.029	0.677%
1.09	204.9	0.280	4.197	7.038	7.174	0.588%
3.29	28.5	0.466	3.358	4.656	4.335	0.425%
3.29	57.0	0.288	2.466	3.700	3.468	0.365%
3.29	114.0	0.161	2.973	4.172	4.148	0.432%
3.29	228.0	0.074	2.837	4.250	4.379	0.454%
7.44	21.1	0.212	1.522	1.677	1.420	0.179%
7.44	42.3	0.371	1.603	1.210	0.990	0.138%
7.44	84.6	0.159	1.539	1.598	1.401	0.192%
7.44	169.2	0.346	1.374	0.998	0.830	0.126%
23.69	22.4	0.140	0.879	0.812	0.641	0.107%
23.69	44.8	0.234	0.676	0.370	0.272	0.052%
23.69	89.6	0.154	0.603	0.352	0.195	0.052%
23.69	179.3	0.105	0.850	0.709	0.579	0.111%
54.72	23.0	0.174	0.581	0.285	0.178	0.044%
54.72	46.0	0.193	0.510	0.221	0.124	0.037%
54.72	92.0	0.128	0.475	0.310	0.220	0.055%
54.72	184.0	0.075	0.564	0.470	0.381	0.087%
179.50	23.7	0.083	0.544	0.431	0.379	0.083%
179.50	47.3	0.128	0.434	0.246	0.175	0.051%
179.50	94.6	0.072	0.421	0.333	0.279	0.074%
179.50	189.3	0.121	0.276	0.091	0.052	0.022%

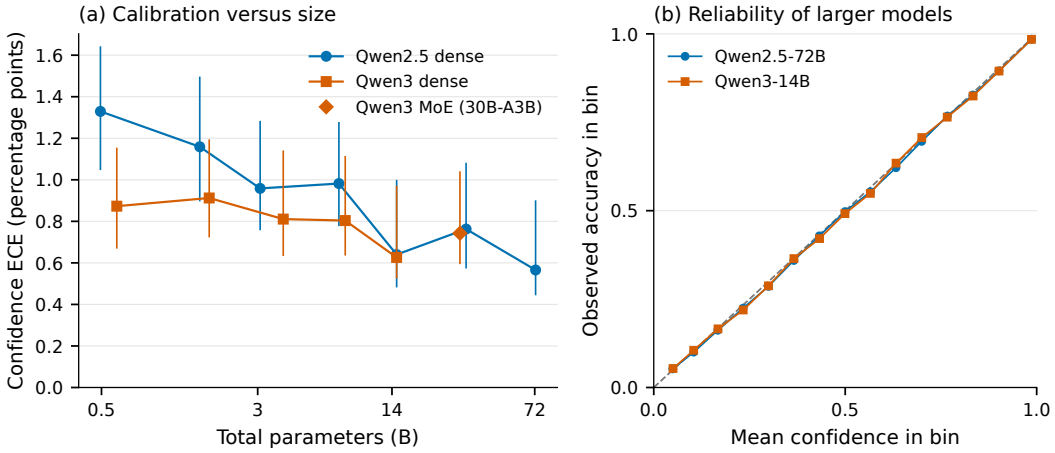

Figure 10: **Confidence calibration of final Qwen base models.** Left: ECE versus total parameters, with 95% source-text bootstrap ranges. Circles and squares distinguish dense Qwen2.5 and Qwen3 models; the diamond marks the Qwen3 MoE and is not joined to the dense series. Right: bin-average confidence and observed accuracy for the largest dense model in each series; the diagonal represents perfect confidence calibration. All results use matching Qwen prediction events.

A.3 Pythia and Qwen confidence calibration

We use final released base models: ten Pythia sizes from 14M to 12B (step 143000, default seed), seven Qwen2.5 sizes from 0.5B to 72B, five dense Qwen3 sizes from 0.6B to 14B, and Qwen3-30B-A3B (Qwen Team, 2024; 2025). Pythia has a common 300B-token training budget. Qwen generations and the MoE are separate series; plotted sizes count total parameters.

The same 96 recent-text documents supply up to 1,024 tokens per model, with the first 64 as context. Tokenizers give 92,160 Pythia predictions over 432,878 bytes and 89,485 Qwen predictions over 429,055 bytes. Within each family, positions match across sizes. Confidence is the maximum predicted probability. Expected calibration error (ECE) is the bin-frequency-weighted absolute difference between confidence and accuracy, using 15 equal-width bins; we also check 30 and 60 bins. Paired 95% ranges use 1,000 bootstrap resamples of the 96 source pages. They condition on these checkpoints and text; absolute-value ECE has positive finite-sample bias.

Archived FP32 evaluations reconstruct correctness by comparing target and maximum probabilities with relative tolerance 10^{-6} and absolute tolerance 10^{-9}. Direct argmax checks through Pythia-1.4B disagree at most once per 92,160 predictions; saved Qwen vectors agree at every unique maximum. Increasing the relative tolerance to 10^{-5} changes at most two Qwen decisions. The earlier BF16 Qwen2.5-32B/72B estimates are excluded because rounded logits can tie. We rerun both in FP32 with TF32 disabled and record token IDs directly, using PyTorch 2.14.0, Transformers 5.14.1, and Accelerate 1.14.0. Weight hashes are checked for these reruns; the older evaluations did not record them.

Pythia ECE is nonmonotonic: 1.493 percentage points at 14M, 0.436 at 160M, and 1.642 at 12B. The endpoint change is $+0.149$ with paired interval $[-0.437, 0.733]$. The largest model has confidence 53.920% and accuracy 52.279% (Figure 11a).

Qwen ECE ranges from 0.566 to 1.329 percentage points. Qwen2.5 changes by -0.764 from 0.5B to 72B, with interval $[-0.948, -0.396]$; its 14B-to-72B change is -0.074 with interval $[-0.294, 0.157]$. Qwen3's 0.6B-to-14B change is -0.247 with interval $[-0.420, 0.107]$. Neither series is strictly monotonic (Figure 10). These confidence tests do not measure C_{cal} for the full vocabulary distribution.

A.4 Pythia full-distribution recalibration

For seven Pythia sizes through 1.4B, we save all 50,304 forecast coordinates on the same 96 windows. Inference is FP32; float16 storage followed by FP32 renormalization changes checked forecasts by less than 2.4×10^{-6} nats/token in KL. Baseline and corrected losses use the same stored forecasts. Swapping even and odd windows gives folds of 46,080 tokens.

The residual MLP uses the character recipe, with a zero-initialized output layer, effective batch 8,192 accumulated in chunks of 512, and input log probabilities clipped below at -30 and scaled by 0.1.

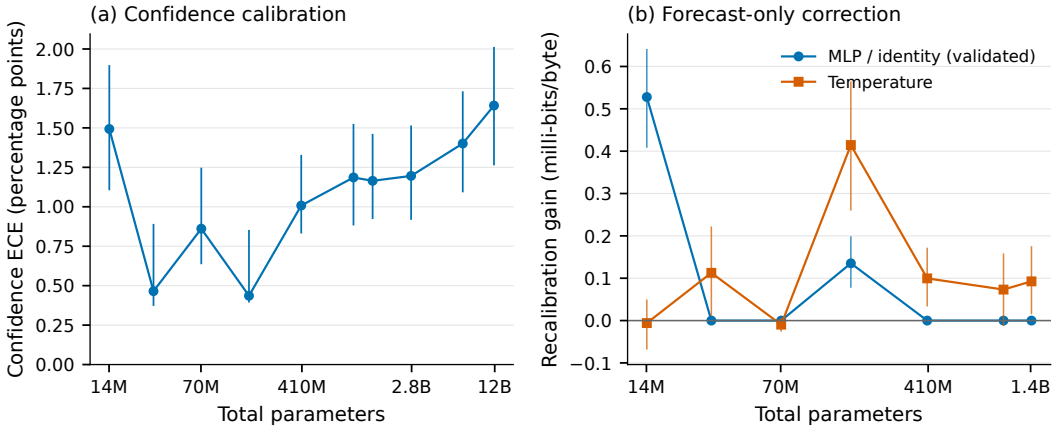

Figure 11: **Calibration trends depend on the test and model family.** Final step-143000 Pythia checkpoints on matching recent text. Left: ordinary confidence ECE for ten sizes, with 95% text-bootstrap ranges. Right: forecast-only held-out loss corrections for the seven sizes from 14M to 1.4B. An inner validation split selects an MLP learning rate or the identity map, which leaves the forecast unchanged. Zero gains can therefore mean that validation selected no correction. These are achieved corrections, not measurements establishing zero full-forecast calibration error. Correction error bars condition on the fitted maps.

Table 2: Forecast-only corrections for Pythia. Gains are signed held-out loss reductions in milli-bits/byte. The MLP recipe used for our character-level models is retained even when it worsens loss. Nested validation chooses identity or a learning rate separately within each fitting fold; outer scoring text never chooses the correction. The last columns give that choice for each fold.

N (M)	Temperature	Original MLP	Validated MLP	Fold 0	Fold 1
14.1	-0.006	+0.528	+0.528	10^{-3}	10^{-3}
30.5	+0.113	-238.249	+0.000	Identity	Identity
70.4	-0.010	-245.615	+0.000	Identity	Identity
162.3	+0.414	+0.361	+0.135	Identity	10^{-3}
405.3	+0.100	+0.203	+0.000	Identity	Identity
1011.8	+0.073	-191.980	+0.000	Identity	Identity
1414.6	+0.093	-90.873	+0.000	Identity	Identity

Its 25,806,208 parameters greatly exceed the fitting sample. Temperature is fitted by bounded scalar minimization over $[e^{-2}, e^{2}]$. The unchanged MLP recipe overfits: at 31M, fitting loss improves by 0.316 nats/token while held-out loss worsens by 0.776 (Table 2).

After observing this failure, we add exploratory nested validation. Within each 48-window fitting fold, 40 windows train the MLP and every sixth window selects among learning rates $10^{-3}, 10^{-4}, 10^{-5}$ and identity. The selected map is refitted on the fitting fold and evaluated on the untouched outer fold. Identity wins 11 of 14 selections. Mean gains are 0.528 milli-bits/byte at 14M, 0.135 at 160M, and zero elsewhere. These zeros reflect rejected corrections, not established calibration. Figure 11b resamples scoring windows with maps fixed, excluding fitting and selection uncertainty. The large-vocabulary experiment remains sample-limited; Qwen has no corresponding full-vector test.

B MODELS, EVALUATION, AND SIZE–DATA SCALING

This appendix gives the model populations, evaluation text, selection controls, and size–data fits supporting Sections 3.1 and 3.2. Unless stated otherwise, pretrained model sizes count all parameters, and losses and spreads use natural logarithms divided by scored bytes. A seed identifies a run's initialization and training randomness. We multiply finite-ensemble spreads and BALD by $S/(S-1)$. Recovered-share fits and ensemble comparisons appear separately in Appendices E and G.

B.1 PRETRAINED SEED POPULATIONS

Primary pretrained seeds. The main comparison contains five Pythia suites trained on the standard Pile text corpus (14M, 31M, 70M, 160M, and 410M), seven DataDecide-C4 seed sets, and four DataDecide-Dolma seed sets (Biderman et al., 2023; Magnusson et al., 2025). DataDecide compares training-data choices using the OLMo architecture; C4 (Colossal Clean Crawled Corpus) and Dolma identify its training corpora. M, B, and T denote 10^6, 10^9, and 10^{12}, respectively. DataDecide uses three seeds per set; its nominal 4M–1B ladder spans 6.96M–1.28B total parameters including embeddings. The Pythia suites come from the public multi-seed release studied by van der Wal et al. (2025): the original seed 1234 and seeds 1–9, before exclusions. After the loss-based filter described below, the five Pythia sizes retain 6, 9, 10, 9, and 8 members, respectively. These suites use the standard, non-deduplicated Pile and a fixed 300B-token budget.

The small-suite filter removes a member when its loss exceeds the suite median by more than 0.01 nats/byte. The filter uses the recent-text evaluation distribution. At 14M and 31M, only pairwise pooled evaluations are available: corrected spread is averaged over retained pairs and member loss over retained models. Consequently, these two sizes have no measured loss for the full retained ensemble. This filter can truncate observed seed variability. Appendix B.5 reports the effects of member filtering and size selection using a consistent pair-based estimator.

The separate public 6.9B pair. The larger Pythia comparison uses two public EleutherAI checkpoints:

- `EleutherAI/pythia-6.9b-deduped-v0`;
- `EleutherAI/pythia-6.9b-deduped-v0-seed42`.

Each has 6,857,302,016 parameters. This is a public seed-labelled pair within the older deduplicated-Pile release, not a matching extension of the standard-Pile suites. The seed-42 repository has no model card; compatible inference configurations and a seed label do not independently verify every training setting. We therefore retain its measured losses and spread as a separate comparison, exclude it from primary size, difficulty, and recovered-share fits, and avoid transferring the small suites' sampling errors to it. It is an open circle in Figures 2 and 5; the pooling and irreducible-loss control figures use open diamonds for this pair. The links above identify the evaluated releases.

The scalar evaluations did not record revision hashes for this pair, which limits exact checkpoint reproducibility. Comparisons between public release versions, including a three-model ensemble combining this pair with another release, are classified separately.

B.2 EVALUATION DATA AND UNITS

Two evaluation corpora. The main pretrained comparison in Figure 2 uses recent web text; an earlier evaluation uses 96 segments from the C4 validation split, drawn from 2019 Common Crawl. Figure 13 reports all 60 comparison definitions on both corpora. Here "DataDecide-C4" and "DataDecide-Dolma" name the models' training corpora, not the evaluation text: both model ladders are evaluated on each corpus. Membership in C4's validation split does not establish that every externally trained model excluded that text from pretraining.

Recent-text construction. We use Common Crawl collection `CC-MAIN-2026-34`, collected in August 2026, and require a June–August 2026 date expression in each cleaned document. C4-like cleaning retains lines with at least five words and terminal punctuation, drops pages containing braces or "lorem ipsum" and lines mentioning "javascript", and requires at least five sentence boundaries and 300 words. An English stop-word fraction of at least 0.22 provides a language heuristic. Documents must contain at least 5,500 characters and 1,025 GPT-NeoX tokens. To reduce publisher differences from C4, we retain only domain groups represented in C4 validation or six sampled C4 training shards, with one document per group; a group is the final two hostname labels. The 188 retained

documents are shuffled with random seed 0, and the first 96 supply the evaluation segments. Their recorded crawl dates span August 7–20, 2026.

The date filter was introduced to reduce the possibility that strong reference-model losses merely reflect memorized evaluation text. It is applied to the document before truncation, so the matching date need not occur in the scored prefix. A recent crawl and an in-document date are evidence of recency, not verified publication dates or a passage-level audit of training overlap: older quoted material and repeated boilerplate can remain. Matching publishers also does not equate the two text distributions. Accordingly, differences between the corpora do not isolate memorization, and the recent-text evaluation does not certify absence of contamination or identify the true irreducible loss.

Scoring and units. The recent documents are truncated to their first 1,025 GPT-NeoX tokens to form source segments. On each corpus, every model retokenizes the decoded segment and processes up to 1,024 tokens; the first 64 provide context, and scoring starts with the next token. This gives approximately 92,000 scored tokens, with exact totals depending on tokenization and segment length. The matched-position pooling comparison evaluates 56 ensembles on 89,485–92,160 matched positions per ensemble (Appendix C). Pretrained loss and geometric spread are normalized by scored bytes for comparisons across tokenizers. Ensembling averages distributions over the same token identities, so it requires aligned vocabularies; cross-tokenizer comparisons concern separately evaluated losses and can cover different prefixes of a source segment. Per-position difficulty results are in nats/token within compatible token spaces. Achieved reference-model losses on both corpora appear in Table 15; their interpretation is discussed in Appendix E.2.

B.3 Broader public-model comparisons

The 60 public-model comparisons in Figures 2 and 12a are grouped by the source of variation. Color identifies model series, and marker shape identifies one of three classes:

- **Seed comparisons (17; circles).** These comprise the 16 primary Pythia and DataDecide suites and the separately labelled public 6.9B Pythia pair, shown as an open circle.
- **Release reruns and small recipe changes (27; squares).** These include Pythia release-version pairs on standard or deduplicated Pile, the 6.9B mixed-release trio, four DataDecide ensembles that combine five training recipes, six Hubble sets, and OLMo-7B and its separately trained Twin counterpart at a matched two-trillion-token budget. Hubble denotes the model collection with deliberately altered training text; its variants change about 0.02–0.08% of training tokens, so these runs are near-replicates rather than pure seed comparisons.
- **Larger protocol or architecture changes (16; triangles).** These include Pythia standard-Pile/deduplicated-Pile pairs, dense Qwen2.5/3, Gemma dense/mixture-of-experts (MoE), Nemotron-3/3.5, and DeepSeek-V3/V3.1. The two DeepSeek models share an initial period of training.

The latter two classes are descriptive comparisons, whose variation includes changes beyond the random seed. For these comparisons, multiplying the finite spread by $S/(S-1)$ is a descriptive rescaling only; it has no exchangeable-seed population interpretation. The measured finite spread is recovered by multiplying the plotted value by $(S-1)/S$. It compares a pool with the mean member, not with its best member; Table 23 reports both for selected Pythia and Qwen comparisons.

Public Nemotron training recipes. NVIDIA publishes both base-model cards. Nemotron-3 Nano is pretrained for next-token prediction, while Nemotron-3.5 Lightning documents a changed data cutoff, an NVFP4 pretraining recipe, and additional pretraining for multi-token prediction (NVIDIA, 2025; 2026). We therefore classify this pair as a protocol change. The evaluated checkpoints are the base releases in bfloat16 (BF16) precision; NVFP4 is NVIDIA's four-bit floating-point format used during pretraining.

A Pythia 2.8B standard-Pile/deduplicated-Pile pair with nearly identical weights is excluded. The related deduplicated v0/v1 pair is retained among data changes in the broad spread plot; neither comparison enters the seed-only fits.

In Figure 2, model size is the total parameter count, or the geometric mean of member counts when they differ; for sparse models, this count does not measure the floating-point operations performed per prediction. To separate overlapping symbols, displayed sizes are multiplied by 0.94 for Pythia version reruns, the public 6.9B pair, and DataDecide-Dolma; and by 1.06 for DataDecide-C4 and OLMo twins. All fits use the unshifted counts.

Table 3: **Public-model ensemble inventory for Figure 2.** Sizes count total parameters, including embeddings; unequal-member pools use the geometric mean of member counts, not their sum. For mixture-of-experts models these are total, not active, parameter counts. Pythia's 14M and 31M entries average corrected pair spreads over the retained seeds; larger seed suites use their full selected pools. The three groups contain 17, 27, and 16 comparisons, respectively.

Model family	Ensemble construction	Plotted size N
Seed comparisons		
Pythia, standard Pile	6–10 retained seeds per size	14.1–405M
Pythia, deduplicated v0	Separate public seed-labelled pair	6.86B
DataDecide-C4	Three seeds per size	6.96M–1.28B
DataDecide-Dolma	Three seeds per size	191M–1.28B
Release reruns, recipe variants, and near-replicates		
Pythia release reruns	v0/v1 pairs, standard or deduplicated Pile	70.4M–11.85B
Pythia mixed releases	Three members at one size	6.86B
DataDecide recipes	Five training recipes per size	191M–1.28B
Hubble	2–12 near-replicates per set	1.18–8.27B
OLMo-7B / Twin	Pair at a matched 2T-token budget	6.89B
Larger protocol or architecture changes		
Pythia Pile / deduplicated	Pairs changing training data	70.4M–11.85B
Qwen2.5 / Qwen3, dense	Five cross-generation pairs	543M–14.77B
Gemma-4	Dense/MoE pair	27.83B
Nemotron-3 / 3.5	Pair changing training protocol	31.58B
DeepSeek-V3 / V3.1	Pair with shared early training	671.03B

B.4 Spread fits and their uncertainty

We report regression standard errors (SE), the coefficient of determination R^2, and root mean squared error (RMSE); log-RMSE uses residuals after taking natural logarithms. The central size fits use ordinary least squares on log corrected spread versus log total parameter count. The Pythia fit uses the five small suites; DataDecide fits exclude sizes below 60M total parameters, which do not follow the same slope. The large Pythia pair is a transfer check across both size and training recipe, not a fitting point for that line. Regression standard errors measure residual variation under the fit, not all uncertainty from seed sampling, text sampling, filtering, or choice of fit interval.

Corrected pair estimates from retained small-suite members have about 1–3% relative variation above 31M, with roughly 6% at 14M. Transferring these errors to new protocols is not justified. Comprehensive document-level intervals are unavailable for the main pretrained comparison.

Reference lines in the broad plot. Figure 2 shows fitted lines for Pythia, DataDecide, dense Qwen pairs, and our character-level family; Appendix B.7.2 gives the character fit. The blue line is the five-size Pythia seed fit, with exponent -0.142. The green line independently fits both slope and amplitude from 13 DataDecide comparisons above 60M parameters: nine seed sets and four data-recipe ensembles. Its decay exponent is 0.1648 ± 0.0122 and its amplitude at 10^9 parameters is 0.02087 ± 0.00031 nats/byte ($\pm$ regression SE). This green line is not the independent C4 or Dolma seed-only fit reported in Table 4; it explicitly includes recipe ensembles.

Solid segments cover the sizes used for each fit, and dotted segments show extrapolation in either direction. The 7M and 29M DataDecide-C4 points are displayed but excluded from the fits. Neither extrapolation supplies evidence of a universal seed law across all displayed protocols.

The Qwen fit uses all five dense Qwen2.5/Qwen3 base-model pairs, including the 14B pair: $\widehat{\rho}^{\mathrm{G}}_{\infty} = 0.02891(N/10^9)^{-0.11784}$ nats/byte, with regression SE 0.01532, $R^2 = 0.9517$, and log-RMSE 0.03087. Its slope is somewhat shallower than the Pythia reference; these descriptive fits do not establish a common exponent.

Upward triangles mark the dense Qwen pairs. The smaller spread reduction at 14B remains visible. These comparisons change model generation, so they are not controlled seed interventions.

Figure 15 shows the corresponding arithmetic-spread and BALD comparisons. Table 8 reports the scaling coefficients for all three measures and every displayed family, with evaluation coverage and estimator details in Section C.2.

Table 4: Independent true-seed size fits. Figure 2 uses the Pythia row for its blue line; its green line fits an independent exponent to the combined DataDecide seed and recipe ensembles. Size ranges count total parameters. The fitted form is $\widehat{\rho}_{\infty}^{\mathrm{G}} = c(N/10^9)^{-\gamma}$. Standard errors are from the log regression, not complete sampling intervals.

Family	Sizes	N (M)	γ	SE	Log-RMSE
Pythia	5	14.1–405	0.142	0.019	0.039
DataDecide C4	5	76.4–1280	0.162	0.016	0.027
DataDecide Dolma	4	190.5–1280	0.152	0.018	0.017

Table 5: Recent-text seed measurements, including the separately labelled public 6.9B deduplicated-v0 pair. That pair is excluded from primary fits. Loss and corrected geometric spread are in nats/byte; fractions use the V4-Pro-anchored assumed floor $E = 0.33372841$. At 14M and 31M, S counts retained members but evaluated pools have $S_{\mathrm{pool}} = 2$; corrected spread averages those pairs. Other rows use the measured full selected ensemble with the $S/(S-1)$ correction.

Family	N (M)	S	$\bar{L}_S$	$\widehat{\rho}_{\infty}^{\mathrm{G}}$	$\widehat{f}_{\infty}^{\mathrm{G}}$
Pythia	14.1	6	0.91467	0.04758	8.19%
Pythia	30.5	9	0.82338	0.04412	9.01%
Pythia	70.4	10	0.75479	0.04224	10.03%
Pythia	162.3	9	0.65751	0.03440	10.62%
Pythia	405.3	8	0.58780	0.02974	11.70%
Pythia dedup v0	6857.3	2	0.49224	0.02391	15.09%
DataDecide C4	7.0	3	1.06419	0.03836	5.25%
DataDecide C4	28.8	3	0.78230	0.03868	8.62%
DataDecide C4	76.4	3	0.68822	0.03364	9.49%
DataDecide C4	190.5	3	0.61666	0.02732	9.66%
DataDecide C4	371.5	3	0.57274	0.02474	10.35%
DataDecide C4	597.7	3	0.54724	0.02428	11.37%
DataDecide C4	1279.9	3	0.50981	0.02082	11.82%
DataDecide Dolma	190.5	3	0.61719	0.02681	9.46%
DataDecide Dolma	371.5	3	0.56958	0.02352	9.97%
DataDecide Dolma	597.7	3	0.54308	0.02295	10.96%
DataDecide Dolma	1279.9	3	0.50481	0.01984	11.60%

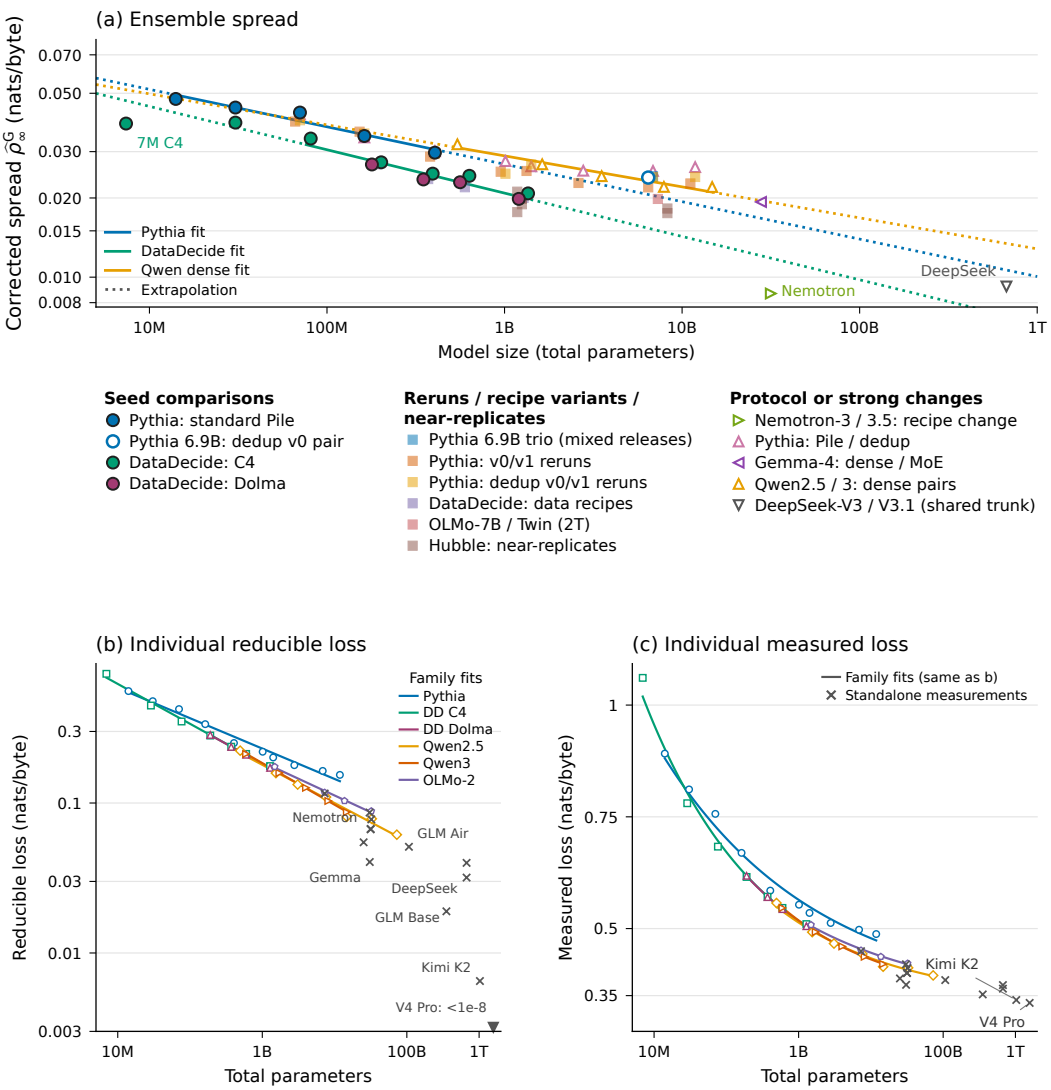

Figure 12: **Spread decreases slowly with model size.** (a): corrected geometric spread for 60 public-model comparisons; solid lines cover fitted ranges and dotted segments extrapolate. The DataDecide reference has its own fitted exponent. (b): individual-model reducible loss, using the default seed where available. (c): the same checkpoints' measured losses before subtracting the floor, with Kimi-K2-Base and DeepSeek-V4-Pro-Base labelled. In (b,c), curves fit six model families; gray markers show 13 additional checkpoints without fitted or connecting lines. The V4 point below the range in (b) is marked at the boundary. Model size counts total parameters. Power-law fits are straight on the log–log axes in (b). Appendices B and E.2 give selections and irreducible-loss assumptions.
In (b), $L(q) - E \propto N^{-\eta}$ with $E = 0.33372841$ nats/byte. Individual-model η estimates ($\pm$ regression SE) include Pythia 0.203 ± 0.010, DataDecide C4 0.266 ± 0.009, Dolma 0.264 ± 0.003, Qwen2.5 0.255 ± 0.014, and Qwen3 0.277 ± 0.002. For the separate seed cohorts in (a), spread exponents γ are Pythia 0.142 ± 0.019, C4 0.162 ± 0.016, and Dolma 0.152 ± 0.018.

On C4, the small-suite fit predicts 0.0204 nats/byte at the 6.9B deduplicated-v0 pair and 0.0189 at the 12B rerun; both measurements are about 0.0200. The corresponding recent-text large points are higher: the 6.9B deduplicated-v0 pair has 0.02391, and the 12B deduplicated rerun 0.02403. These are different fits and protocols, not interchangeable estimates of a universal amplitude.

The shallow Pythia trend appears on both evaluation distributions, while its amplitude and the descriptive cross-generation trends can change. For example, the dense Qwen-pair exponents are 0.094 on C4 and 0.118 on recent text. These comparisons therefore support robustness of the qualitative spread trend without establishing a corpus-independent exponent. Appendix B.5 gives additional member- and size-selection controls using corrected pair spreads at every Pythia size.

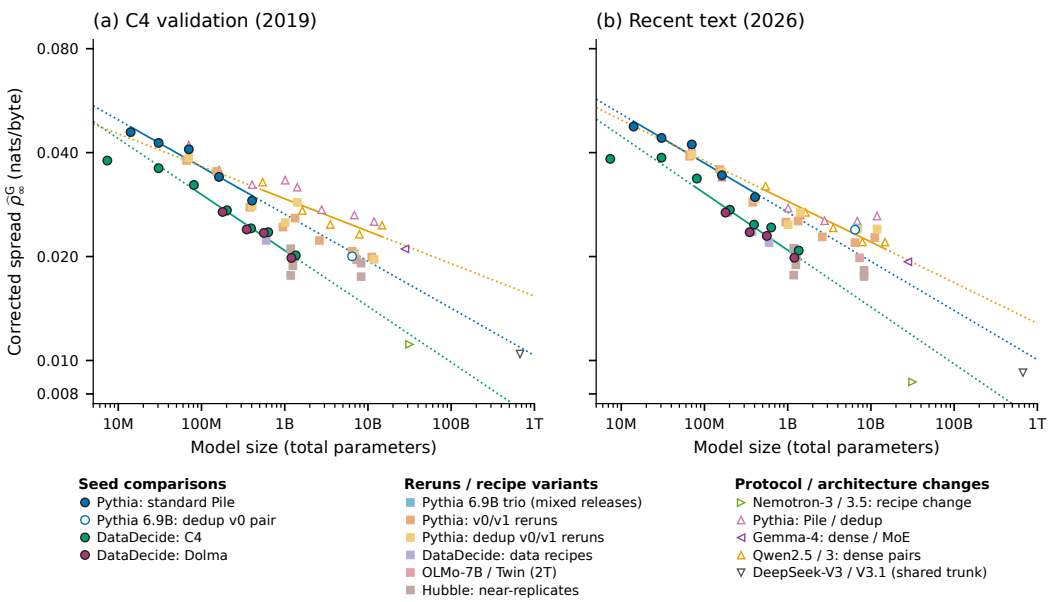

Figure 13: **Spread scaling on C4 validation and recent text.** The 60 public-model comparison definitions from Figure 2, evaluated separately on (a) C4 validation and (b) the recent-text sample. Colors, marker classes, display offsets, and fitting cohorts follow the main comparison. The three reference curves are refitted on each corpus; solid segments cover fitted sizes and dotted segments extrapolate. The green DataDecide reference fits its own amplitude and exponent: $\gamma = 0.162$ on C4 and 0.165 on recent text. Pythia's loss filter is applied separately to each corpus, retaining eight 14M members on C4 and six on recent text; other retained suite sizes agree. At 14M and 31M, points average corrected pair spreads. Larger seed suites use their measured full pools. The Pythia decay exponents are 0.136 on C4 and 0.142 on recent text. No irreducible-loss estimate enters these spread measurements.

B.5 Sensitivity to member and size selection

Table 6 tests the geometric-spread fit on C4 and recent text using the same loss threshold as the primary analysis. This audit averages corrected pair spreads at every Pythia size; the primary broad-plot fits instead mix pair averages and full-pool estimates. Its coefficients therefore need not equal those in Table 4. The available suites already omit the original 160M member and two 410M members, so the audit does not reconstruct every original training run. Filtering is applied separately on each evaluation corpus and need not retain identical memberships. Overlapping pairs are not independent replicates; leave-one-size-out ranges are sensitivity checks, not confidence intervals.

Filtering changes the Pythia decay exponent from 0.154 to 0.132 on C4 and from 0.169 to 0.138 on recent text. The shallow decline survives, but the 0.02–0.03 shift is material relative to the recovered-share growth rate. The C4 fit predicts 0.02098 nats/byte at the separate 6.9B pair, versus 0.02003 measured; the recent-text fit predicts 0.02104, versus 0.02391. Excluding the smallest DataDecide-C4 sizes also changes its exponent, as shown in the table.

B.6 Individual-model loss scaling

Figure 12b,c uses one checkpoint's loss $L(q)$ and its own total parameter count at each size; (b) subtracts the assumed floor, while (c) shows the measured loss directly. Each point describes one model. When multiple seeds are available we select the default seed consistently, without selecting on loss. The Pythia ladder uses ten current public checkpoints from 14M to 12B; this includes the default 160M checkpoint, which is excluded from the filtered spread cohort. DataDecide-C4 uses seven default-seed checkpoints, and DataDecide-Dolma uses the four default-seed checkpoints at the same training steps as the seed comparison. The recipe comparisons instead use final checkpoints. Dedicated individual-checkpoint evaluations supply the seven Qwen2.5, five dense Qwen3, and four OLMo-2 observations. We keep the two Qwen generations separate.

The standalone Qwen2.5-32B evaluation uses BF16 and gives 0.412227 nats/byte; this evaluation is used throughout its loss curve.

Table 6: Sensitivity of geometric-spread fits to member and size selection, using byte-normalized spreads. DD denotes DataDecide; Recent denotes the August 2026 text sample. Filtered Pythia fits exclude runs whose loss exceeds the suite median by more than 0.01 nats/byte. The reported decay exponent γ is positive. Deletion ranges omit one size at a time; they are sensitivity checks, not confidence intervals.

Corpus	Selection	Sizes	γ	Log-RMSE	Deletion range
C4	Pythia, available	5	0.154	0.035	$[0.138, 0.180]$
C4	Pythia, filtered	5	0.132	0.041	$[0.109, 0.152]$
C4	DD C4, all sizes	7	0.127	0.048	$[0.118, 0.154]$
C4	DD C4, $N \geq 60$M	5	0.162	0.018	$[0.152, 0.167]$
C4	DD Dolma	4	0.155	0.020	$[0.126, 0.160]$
Recent	Pythia, available	5	0.169	0.027	$[0.161, 0.185]$
Recent	Pythia, filtered	5	0.138	0.042	$[0.118, 0.158]$
Recent	DD C4, all sizes	7	0.128	0.062	$[0.118, 0.163]$
Recent	DD C4, $N \geq 60$M	5	0.162	0.027	$[0.137, 0.168]$
Recent	DD Dolma	4	0.152	0.017	$[0.140, 0.156]$

Each family is fitted independently by ordinary least squares of $\log(L(q) - E)$ against $\log N$, with $E = 0.33372841$ nats/byte, using all its displayed sizes. The reducible-loss exponents η ($\pm$ regression SE) are Pythia 0.20262 ± 0.00971 ($n = 10$), DataDecide C4 0.26581 ± 0.00923 ($n = 7$), Dolma 0.26433 ± 0.00271 ($n = 4$), Qwen2.5 0.25510 ± 0.01434 ($n = 7$), Qwen3 0.27748 ± 0.00237 ($n = 5$), and OLMo-2 0.22325 ± 0.01202 ($n = 4$). Fit segments stop at each family's measured endpoints. Subtracting the irreducible loss makes the fitted power law straight on logarithmic axes; measured deviations remain visible.

The SEs condition on this irreducible loss and checkpoint selection, and do not include seed or text resampling. These loss ladders differ from the spread cohorts in panel (a); Figure 5 retains its own matched loss/spread cohorts for recovered-share comparisons.

Ten additional individual checkpoints are shown as gray crosses: sparse Qwen3, two OLMo-3, two Nemotron, two Gemma-4, two DeepSeek, and GLM-4.5-Air. They do not enter the six fitted laws. Own total parameter counts are used even for sparse models: the DeepSeek checkpoints each have 671.0B parameters, the Nemotron checkpoints each 31.58B, and GLM-Air 106.85B.

The evaluation also includes GLM-4.5-Base (352.80B parameters, loss 0.3526895 nats/byte), Kimi-K2-Base ($1{,}026.41$B, 0.3402290), and DeepSeek-V4-Pro-Base ($1{,}573.00$B, 0.3337284). DeepSeek-V4-Pro-Base supplies the downward-rounded empirical cap; its positive residual of 6.94×10^{-9} nats/byte lies below panel (b)'s displayed range and is marked at the lower boundary. The remaining 49 checkpoints, including Kimi at residual 0.0065006 nats/byte, are shown at their measured reducible-loss ordinates in (b). Panel (c) shows all 50 checkpoints at their actual achieved losses. The 13 checkpoints outside the six fitted ladders, including Kimi and DeepSeek-V4-Pro-Base, appear as standalone points in both loss panels and do not enter the family fits. Architectural counts exclude quantization metadata and unused multi-token-prediction weights. Cross-tokenizer loss comparisons remain descriptive because the models can score different prefixes of the same text segments.

B.7 Our character-level models

B.7.1 Training design and parameter counts

Our character-level models are trained from scratch on FineWeb-Edu with one byte per token, a vocabulary of 256 byte values, and 512-token contexts. The decoder-only transformer uses RMSNorm, SwiGLU feed-forward blocks, rotary position embeddings, query/key normalization, 64-dimensional attention heads, and no biases. Its input embeddings and output head are untied. Increasing the budget processes additional corpus positions; these experiments do not deliberately cycle over a fixed subset.

Let d be hidden width, $h = d/32$ the number of transformer layers, and D the number of training bytes. For the character-model scaling and branching studies, we use the approximate parameter count $N = 12hd^2$; D/N is the training budget in bytes per parameter. This convention approximates the block matrices and excludes the normalization weights and vocabulary matrices. The latter contain only $2(256)d = 512d$ parameters in total: 65,536 at width 128 and 393,216 at width 768. Across the six widths $128, 192, 256, 384, 512, 768$, analytic counts span 0.786M–169.869M, while

exact total counts span 0.858M–170.303M. Small differences beyond the vocabulary matrices arise from normalization weights and rounding the feed-forward width to a multiple of eight. A grid cell denotes one model size and data budget, with multiple seeds where available.

Training uses an 18-step linear warmup followed by a constant-learning-rate trajectory. For every measured budget D, a copy of the model and optimizer state at approximately $0.8D$ bytes completes a linear decay to zero over the remaining $0.2D$ bytes, with rounding to whole update steps. The constant-rate trajectory then resumes from its saved state, so each endpoint receives its own cooldown without changing the shared prefix. This matches optimization stage across budgets; an intermediate checkpoint from a longer run has not necessarily completed an equivalent decay. Nominal endpoint budgets are spaced by $\sqrt{2}$ from 6.25 to 400 bytes per parameter.

Muon updates the hidden matrices with momentum 0.95; AdamW updates the embeddings, output head, and normalization gains with $(\beta_1, \beta_2) = (0.9, 0.95)$ and $\epsilon = 10^{-10}$. Initial-recipe peak learning rates are 0.006087 for Muon, 0.066236 for embeddings, and 0.001775 for the head and normalization gains. Weight decay 0.001423 applies to hidden matrices and the output head. In increasing width order, initial batch sizes are $16, 32, 64, 128, 256, 512$ sequences. Retuning uses batches $1, 2, 8, 32$ at the first four widths and multiplies their peak rates by $1/4, 1/8, 1/4, 1/2$, respectively, rounded to six decimal places; widths 512 and 768 retain the initial settings. Shared-training and adapter interventions use the separately specified matched protocols in Appendix H.

The complete loss grid contains 78 cells at six widths, each with 13 nominal budgets from 6.25 to 400 bytes per parameter. It includes all three width-768 seeds at approximately 283 and 400 bytes per parameter. We distinguish an initial training recipe from a retuned recipe with adjusted batch sizes and learning rates at the smaller widths. The complete loss grid uses the retuned recipe; width 192 retains seeds 0 and 1. The primary spread grid combines the initial recipe at widths 128, 256, and 384 with the larger-width runs at 512 and 768. It omits width 192 because its seed runs behave inconsistently. At width 128, seeds 0 and 2 are retained because seed 1 has an anomalous late-training trajectory; other retained widths use three members. This spread grid has 65 cells. Its common fitted window contains 35 cells at 50–400 bytes per parameter and five sizes from 0.79M to 170M. The earlier 50–200 window is retained below as a sensitivity check.

B.7.2 Spread and reducible-loss fits

For the character-model overlay in Figure 2, we use the same 35 cells at 50–400 bytes per analytic parameter, convert corrected spreads to nats/byte by multiplying by $\ln 2$, and place them at exact total counts of 857,728, 6,460,672, 21,441,408, 50,678,272, and 170,303,232 parameters. Refitting on this total-count axis gives $\widehat{\rho}_\infty^{\mathrm{G}} = 0.217806 N_{\mathrm{total}}^{-0.118435}$ nats/byte, with exponent SE 0.001559 and log-RMSE 0.016212. Open diamonds show individual budgets and filled diamonds their geometric mean at each size. These held-out FineWeb-Edu measurements use 512-byte contexts; the public-model points use recent web text and up to 1,024 subword tokens, so absolute spread levels are not evaluated under a common protocol. The remaining character scaling analyses retain the analytic count $N = 12hd^2$ defined above, explaining their slightly different exponent of approximately 0.117.

The size-only fit over 50–400 bytes per parameter is $\widehat{\rho}_\infty^{\mathrm{G}} = 0.303554 N^{-0.116538}$ bits/byte, with log-RMSE 0.01720. Here log-RMSE is the root mean squared residual after taking natural logarithms. Allowing a data exponent gives $\widehat{\rho}_\infty^{\mathrm{G}} = cN^{-\gamma_N} D^{-\gamma_D}$, with $\gamma_N = 0.11784$, $\gamma_D = -0.00130$, and log-RMSE 0.01717. The negligible change in error supports using the size-only law; it does not determine the infinite-data limit. A milli-bit is 10^{-3} bits. Across widths, mean corrected spread decreases from about 62 to 33 milli-bits/byte. The smallest size has a nonmonotonic within-width range of 4.35 milli-bits/byte over the enlarged window; the corresponding range at width 768 is 0.36.

The orange series in Figure 3b is the invisible portion $L(q_S^{\mathrm{G}}) - E$ for the same 35 cells and retained seeds as the spread fit, evaluated on the same 524,288 held-out FineWeb-Edu bytes. We recover the actual finite-pool loss as $L(q_S^{\mathrm{G}}) = \bar{L}_S - [(S-1)/S]\widehat{\rho}_\infty^{\mathrm{G}}$, undoing the spread correction before subtraction. With $E = 0.593341$ bits/byte, log–log least squares gives $L(q_S^{\mathrm{G}}) - E = 16.6593 N^{-0.191399}$, with exponent SE 0.005118 and log-RMSE 0.054085. At the Kimi cap $E = 0.63764990$, its exponent is 0.203958 ± 0.005468. Orange shading shows this irreducible-loss sensitivity, separately from regression error. Both vertical axes are logarithmic. For comparison, total member reducible loss on these cells follows $\bar{L}_S - E = 16.1439 N^{-0.187090}$ at the reference E (exponent SE 0.004885, log-RMSE 0.051617), and has exponent 0.198868 ± 0.005215 at the higher E. The spread exponent has SE 0.001627.

These size-only fits describe a window in bytes per parameter, so loss decline combines increasing model size and data. Checkpoints at different budgets share the earlier portion of training; regression SEs are not independent-run confidence intervals. Restricting both quantities to the earlier 25 cells at 50–200 gives spread exponent 0.114469 ± 0.001931 and reducible-loss exponent 0.186542 ± 0.004371 at the reference irreducible loss. Thus extending the common window changes neither the shallow spread law nor the ordering of the exponents.

For the full 78-cell retuned grid, the direct member-loss fit and the separately fitted pooled-log-loss component at $E = 0.593341$ are

$$\bar{L}_S \simeq E + 14.36N^{-0.2296} + 9.19D^{-0.1450},$$
$$\widehat{L}_\infty^{\mathrm{G}} \simeq E + 15.90N^{-0.2451} + 8.90D^{-0.1436}. \tag{11}$$

Their RMSEs against their respective targets are 0.0193 and 0.0194 bits/byte. The second target subtracts the *measured* population-corrected spread of the same retuned members from their mean loss, avoiding transfer of the primary spread law between recipes. No infinite ensemble is directly evaluated. Adding the independently fitted retuned spread law gives member-loss RMSE 0.0198; Appendix E.3 compares the two approaches on held-out widths and identical irreducible-loss values.

B.7.3 Empirical constraints on the irreducible loss

The working irreducible loss is the joint extrapolation of losses from publicly released models with accessible weights, $E = 0.593341$ bits/byte on the 512-byte protocol. Its earlier fit-profile range, 0.531–0.640, is a sensitivity to the fitted model, not a confidence interval for conditional entropy. Kimi-K2-Base (Moonshot AI, 2025), a roughly one-trillion-parameter mixture-of-experts model, achieves 0.63764990 bits/byte on the matched 512-byte protocol, improving on GLM-4.5-Base's 0.677338 and Qwen2.5-72B's approximately 0.717. Gemma-4-31B achieves 0.848147 with its required beginning-of-sequence token. DeepSeek-V4-Pro-Base gives 0.68195265 bits/byte on this protocol, so Kimi remains the best measured character reference.

We therefore use 0.593341–0.63764990 as the working sensitivity range, rounding the achieved upper endpoint downward to eight decimal places for refitting. The lower endpoint is a fitted assumption, not an established lower bound. An achieved model loss supplies an empirical cap for this sensitivity analysis, not a statistical bound on population entropy; finite text sampling, tokenization, and possible training-data overlap limit that interpretation. GLM's long-context score on these texts is 0.453323 bits/byte, illustrating why irreducible-loss values from a different context protocol cannot replace the 512-byte value.

The spread law requires no estimate of irreducible loss. Both endpoints preserve its slower decay than reducible loss, while changing recovered fractions. All character-model loss comparisons use the same irreducible loss choices for both hypotheses (Appendix E.3); the recent-text and C4 pretrained studies retain their own protocol-specific values.

B.7.4 Sensitivity of the data-plateau fit

Figure 3a uses free-slope linear fits on all available budgets from 25 through 400. The horizontal axis is logarithmic, while these fits remain linear in bytes per parameter. Across the five sizes, their fitted changes are -1.93% to -0.09% of mean spread; for width-768 the change is -1.60%. Moving the cutoff to 50 gives changes from -1.32% to $+3.82\%$. These weak fitted trends do not remove the smallest model's visible decline and rebound. Checkpoints at different budgets share the earlier portion of training, so residual-based SEs do not include all dependence. The fitted cutoffs describe the measured data range; they do not identify a compute-optimal training budget.

At width 768, doubling the budget from 200 to 400 bytes per parameter reduces mean member loss by 1.83% and the actual three-member geometric-pool loss from 1.02170 to 1.00280 bits/byte, while corrected geometric spread falls by only 0.68%. These measurements use seeds 0–2 and the same 524,288 evaluation bytes.

Table 7: Fitted data trends for Figure 3a, including all three d768 seeds through 400 bytes per parameter. Slopes are changes in milli-bits/byte per additional 100 bytes/parameter, with residual-based regression SE. Each slope uses all available budgets above its cutoff; every size now reaches 400. Neither cutoff denotes a common compute optimum.

N (M)	Plotted range	25+: slope $\pm$ SE	50+: slope $\pm$ SE
0.79	25–400	-0.322 ± 0.779	$+0.672 \pm 0.392$
6.29	25–400	-0.022 ± 0.178	$+0.042 \pm 0.239$
21.23	25–400	-0.011 ± 0.062	$+0.026 \pm 0.056$
50.33	25–400	-0.113 ± 0.053	-0.144 ± 0.068
169.87	25–400	-0.141 ± 0.044	-0.086 ± 0.036

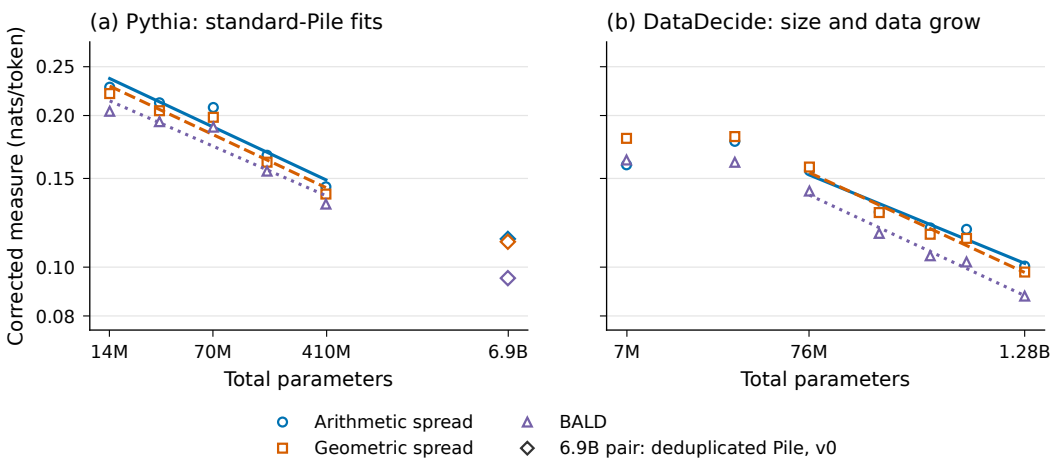

Figure 14: **The shallow size trend persists under both loss pools and BALD.** Corrected measures on matching members and recent-text positions. Points are measurements; lines are log–log least-squares fits to the five standard-Pile Pythia suites and DataDecide-C4 sizes above 60M parameters. DataDecide-C4 grows data with size. Open diamonds show the separate 6.9B deduplicated-v0 pair, excluded from the fits. Lines stop at their fitting ranges. Ensemble sizes vary along the ladders, and the approximate correction does not eliminate higher-order differences between measures.

C Pooling, entropy, and fixed difficulty groups

C.1 Pooling measures on matching members and positions

The recent-text comparison uses August 2026 Common Crawl text and evaluates 56 ensembles on 96 windows, or 89,485–92,160 matched positions per ensemble. For each context x and observed next token y, it computes arithmetic spread $\lambda_S(x,y) = \log q^{\mathrm{A}}_{S,x}(y) - S^{-1}\sum_s \log q_{s,x}(y)$, geometric spread $\rho^{\mathrm{G}}_S(x) = -\log Z_x$, and BALD $J_S(x) = H(q^{\mathrm{A}}_{S,x}) - S^{-1}\sum_s H(q_{s,x})$. The corpus quantities ρ^{G}_S and J_S are their context averages. All three use the same members and scoring positions. The reported comparison statistics apply the approximate $S/(S-1)$ correction to each measure. For heterogeneous pairs this factor is only a descriptive rescaling; it does not estimate an exchangeable seed population (Appendix B). Figure 14 shows the standard-Pile Pythia and DataDecide-C4 seed ladders, with the public 6.9B deduplicated-v0 pair marked separately.

Across these ensembles, arithmetic/geometric ensemble-average spread ratios are approximately 0.89–1.05, with an overall ratio near 1.00. The pooled mean losses differ by less than 0.014 nats/token. BALD and geometric spread rank positions almost identically, with Spearman correlations 0.997–1.000. Arithmetic spread has weaker per-position association with geometric spread, about 0.50–0.81, because it depends on which token occurred. Close corpus means therefore do not justify replacing one tokenwise statistic by another. The pooling definitions identify their difference exactly as the observed-label log ratio between the arithmetic and geometric pools. Its expectation can change with context length independently of geometric spread, so these recent-text averages do not impose equal context-scaling slopes.

BALD/geometric ratios are around 0.955 for larger seed suites, 0.90 for triplets, and 0.85–0.88 for pairs. Suite size changes with model size in parts of the Pythia ladder, so aggregate exponent differences mix functional and finite-ensemble effects. In the combined Pythia seed/rerun ladder, the decay exponents are 0.121 for geometric spread, 0.141 for BALD, and 0.130 for arithmetic spread. The fixed-pair deduplicated rerun ladder gives about 0.098, 0.099, and 0.095; The five dense Qwen cross-generation pairs give about 0.118, 0.117, and 0.120. These broader protocols remain distinct from the primary seed fits.

The comparison of our character-level models likewise supports similar mean spreads. Arithmetic/geometric spread has median 0.932 in both grids: a range 0.866–1.035 over 69 settings under the baseline training recipe and 0.873–1.050 over 76 under the retuned recipe (Appendix B.7). These pooling comparisons use a broader set of sizes and budgets than the spread-scaling fit.

C.2 PUBLIC-MODEL SCALING UNDER ALL THREE ENSEMBLE MEASURES

Figure 15 gives the arithmetic-spread and BALD counterparts for the public-model comparisons in Figure 2, retaining their parameter counts, colors, and comparison classes. All plotted measures are normalized by scored bytes and multiplied by $S/(S-1)$. For heterogeneous comparisons this factor remains a descriptive rescaling, not a seed-population correction. Arithmetic spread uses the 96-window recent-text evaluation for all 60 comparisons. BALD uses 96 windows for 53 comparisons. The other seven use two windows: the standard-Pile Pythia v0/v1 reruns at 70M, 160M, 410M, and 6.9B, plus Nemotron, Gemma, and DeepSeek. These shorter evaluations are marked separately and excluded from every fit. The matched-family table therefore uses four standard-Pile rerun sizes; the three large-model families with only two-window BALD have no eligible matched fit.

There is one additional estimator difference from Figure 2: at 14M and 31M, its geometric points average corrected retained-seed pairs, whereas the arithmetic and BALD measurements use the full pools of six and nine retained seeds. The other Pythia sizes use the same retained full pools across measures. Table 8 separates the plotted reference laws from family fits that use matching full-pool estimators and the intersection of 96-window coverage across all three measures.

For each measure we fit $d(N) = c(N/10^9)^{-\gamma}$ by unweighted least squares in log space, where c is the corrected measure at one billion parameters in nats/byte. The three plotted reference curves follow Figure 2: Pythia, DataDecide, and dense Qwen pairs each fit their own exponent and amplitude for each measure. Family fits separate DataDecide-C4, DataDecide-Dolma, and recipe ensembles, excluding C4 sizes below 60M. Hubble training budgets and depth variants are kept separate. A single size cannot identify both coefficients; two-size fits have no residual degrees of freedom for a regression standard error. Standard errors describe regression residuals conditional on the chosen checkpoints and text, not complete seed or text-sampling uncertainty. Amplitude errors use the delta method.

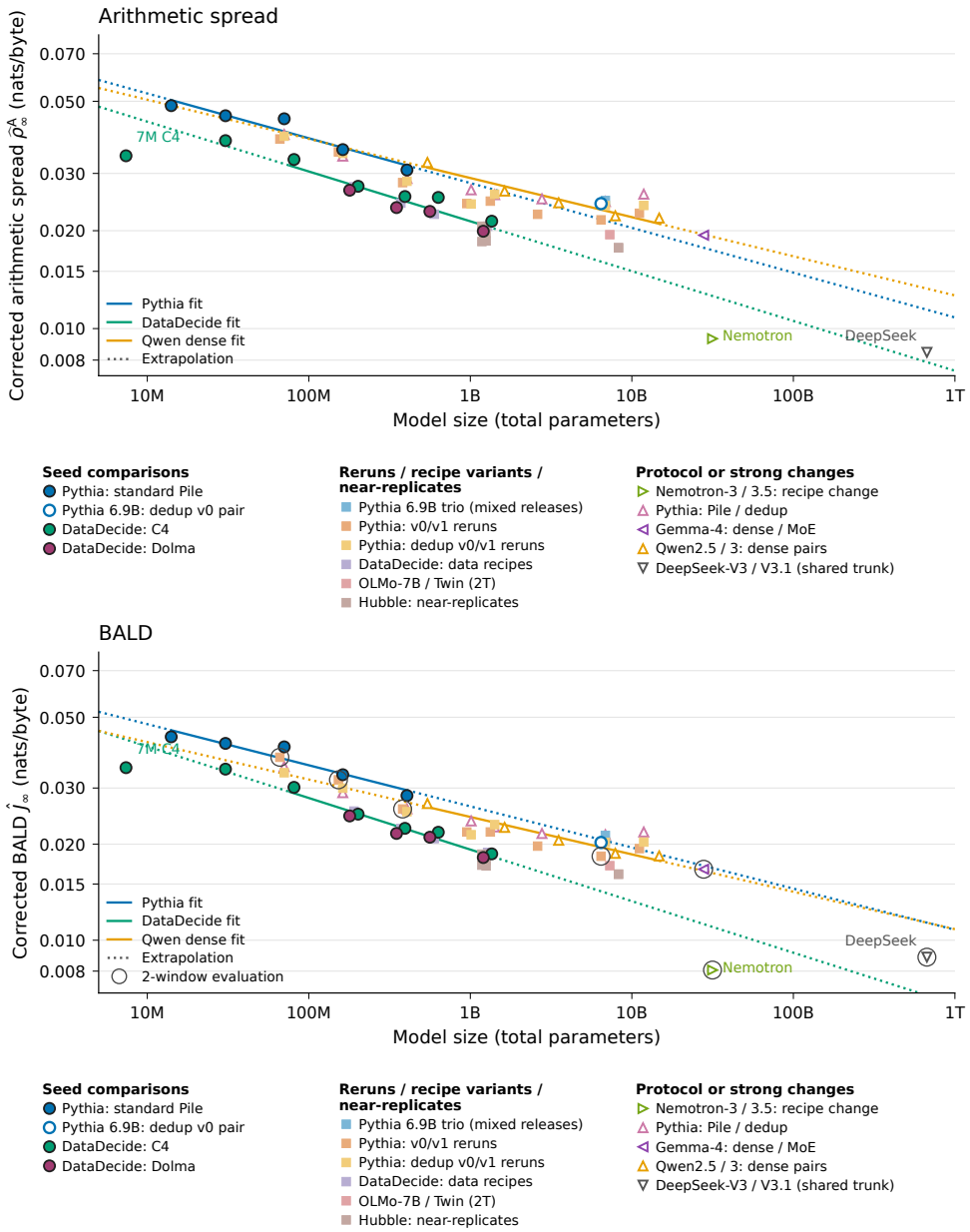

Figure 15: **Arithmetic spread (top) and BALD (bottom) for the public-model comparisons.** Both use the colors, size counts, and comparison classes of Figure 2 and the $S/(S-1)$ rescaling. Arithmetic spread uses 96 windows throughout; rings mark seven two-window BALD evaluations excluded from fits. Pythia's 14M and 31M points use full retained pools. Solid lines cover fitted sizes; dotted lines extrapolate. DataDecide fits its own exponent for each measure. Table 8 distinguishes these plotted laws from matched-family fits.

Table 8: **Scaling coefficients for all three ensemble measures and eligible multi-size families in Figure 2.** The law is $d(N) = c(N/10^9)^{-\gamma}$. Each cell gives c (nats/byte) on the first line and γ on the second, with regression standard errors where estimable; n counts eligible comparisons. The first block reproduces the plotted reference laws. The remaining blocks use matching 96-window cohorts and full-pool estimators across the three measures. Families with fewer than two matched sizes are omitted. Details and exclusions are given in Section C.2.

Model family / reference	n	Geometric	Arithmetic	BALD
Plotted reference laws				
Pythia reference	5	0.02687 ± 0.00148 0.142 ± 0.019	0.02802 ± 0.00183 0.138 ± 0.023	0.02628 ± 0.00176 0.129 ± 0.023
DataDecide reference	13	0.02087 ± 0.00031 0.165 ± 0.012	0.02138 ± 0.00036 0.153 ± 0.014	0.01924 ± 0.00019 0.162 ± 0.008
Dense Qwen reference	5	0.02891 ± 0.00073 0.118 ± 0.015	0.02904 ± 0.00076 0.120 ± 0.016	0.02435 ± 0.00049 0.117 ± 0.012
Matched family laws: seeds				
Pythia, standard-Pile seeds	5	0.02703 ± 0.00157 0.138 ± 0.020	0.02802 ± 0.00183 0.138 ± 0.023	0.02628 ± 0.00176 0.129 ± 0.023
DataDecide-C4 seeds	5	0.02163 ± 0.00050 0.162 ± 0.016	0.02244 ± 0.00059 0.143 ± 0.018	0.01946 ± 0.00036 0.163 ± 0.013
DataDecide-Dolma seeds	4	0.02072 ± 0.00037 0.152 ± 0.018	0.02079 ± 0.00033 0.147 ± 0.016	0.01899 ± 0.00032 0.152 ± 0.017
Matched family laws: reruns, recipes, and near-replicates				
Pythia, standard-Pile reruns	4	0.02506 ± 0.00072 0.046 ± 0.021	0.02424 ± 0.00064 0.033 ± 0.020	0.02173 ± 0.00057 0.052 ± 0.019
Pythia, deduplicated reruns	7	0.02835 ± 0.00089 0.097 ± 0.018	0.02782 ± 0.00098 0.095 ± 0.020	0.02419 ± 0.00067 0.099 ± 0.016
DataDecide, recipe ensembles	4	0.02054 ± 0.00026 0.154 ± 0.013	0.02116 ± 0.00025 0.147 ± 0.012	0.01940 ± 0.00020 0.156 ± 0.010
Hubble, 100B tokens	2	0.01769 0.005	0.01859 0.022	0.01729 0.034
Hubble, 500B tokens	2	0.02140 0.076	0.02091 0.079	0.01884 0.074
Matched family laws: protocol or architecture changes				
Pythia, Pile / deduplicated	8	0.02928 ± 0.00084 0.088 ± 0.017	0.02855 ± 0.00081 0.085 ± 0.017	0.02489 ± 0.00063 0.091 ± 0.015
Qwen2.5 / 3, dense	5	0.02891 ± 0.00073 0.118 ± 0.015	0.02904 ± 0.00076 0.120 ± 0.016	0.02435 ± 0.00049 0.117 ± 0.012

C.3 Difficulty is fixed before comparing model sizes

The difficulty analysis uses OLMo-7B token entropy to assign the same recent-text positions to six fixed percentile groups: 0–20, 20–40, 40–60, 60–80, 80–95, and 95–100. The five standard-Pile Pythia suites show faster spread decay in the easy groups under all three measures (Table 10). Reference entropy is a difficulty proxy, not measured irreducible randomness. Fixing groups avoids changing their membership with evaluated model size. **Controls using the same reference and positions.** We recompute the five-suite comparison from cached predictions using OLMo-7B for both token and word scores. All three groupings use the same 90,100 positions whose complete whitespace-delimited words lie within the scored interval. Word-sum scores add the OLMo token entropies within each word; word-mean scores divide that sum by the number of its OLMo tokens. Each score defines six fixed percentile groups, with membership held fixed across sizes. Table 9 shows that the weaker contrast persists with a common reference, but length normalization substantially

Table 9: Geometric decay exponents using one reference model, the same five standard-Pile Pythia suites, and identical scored positions. Entries are exponent ± regression standard error. The errors condition on the evaluated text and selected ensembles; they do not include full sampling uncertainty.

Percentile	OLMo token	OLMo word sum	OLMo word mean
0–20%	0.331 ± 0.029	0.252 ± 0.023	0.263 ± 0.026
20–40%	0.204 ± 0.027	0.157 ± 0.024	0.190 ± 0.025
40–60%	0.120 ± 0.021	0.111 ± 0.020	0.140 ± 0.021
60–80%	0.072 ± 0.018	0.096 ± 0.019	0.094 ± 0.018
80–95%	0.050 ± 0.019	0.113 ± 0.020	0.057 ± 0.019
95–100%	0.059 ± 0.022	0.120 ± 0.019	0.056 ± 0.022

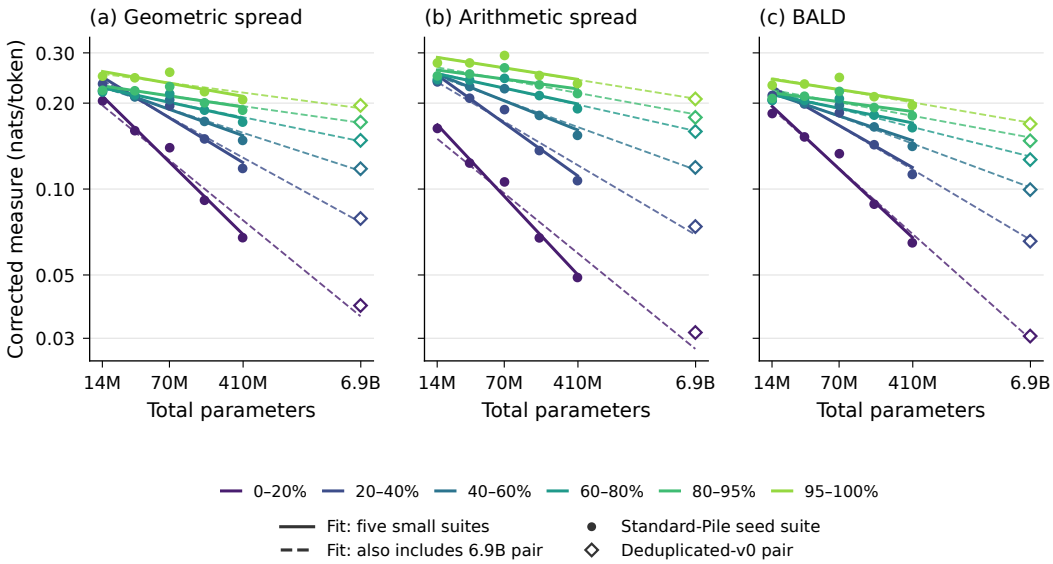

Figure 16: **Pooling and fit selection can be compared directly.** The same fixed token groups under geometric spread, arithmetic spread, and BALD. Filled points are the five standard-Pile suites; open diamonds are the separate 6.9B deduplicated-v0 pair. Solid lines fit only the five suites, as in Figure 4; dashed lines refit after including the public pair, giving a six-size comparison across recipes. All points remain visible. Each line stops at its fitting range; the dashed fit spans two training recipes.

restores the easy–hard contrast: the geometric easiest/hardest exponents are $0.331/0.059$ for token entropy, $0.252/0.120$ for word sums, and $0.263/0.056$ for word means. BALD and arithmetic spread show the same qualitative pattern. Separating initial and continuation positions while retaining these fixed thresholds also preserves differences between token populations. These controls establish sensitivity to score aggregation and token composition; they do not identify an irreducible word-level entropy.

Fitted decay rates. For each difficulty group b, let $d_b(N)$ be the mean corrected measure at size N. We fit $\log d_b(N) = \log A_b - \gamma_b \log(N/10^9)$, where $A_b > 0$ is its amplitude at one billion parameters and γ_b is its decay exponent, with equal weight per ensemble. Connecting measured points highlights local departures from a power law; fitted straight lines summarize the overall slope. Figure 16 compares pooling choices and fitting ranges while keeping the measurements visible. For the five-suite geometric fit, log-space R^2 is 0.978 in the easiest bin and about 0.69–0.70 in the two hardest bins; visible bends limit the claim to approximate, finite-range trends.

Sensitivity to the large point and reruns. Adding the public 6.9B pair changes the easiest-bin exponents from $0.330/0.359/0.314$ to $0.275/0.274/0.300$ for geometric/arithmetic/BALD, respectively. Thus the easy–hard contrast persists, but its numerical rate depends on the fitting range and recipe selection. The broader 19-comparison Pythia fit, which also includes version reruns and the mixed trio, gives $0.265/0.271/0.289$ in the easiest bin. It is a broader protocol comparison, not 19 independent seed suites.

Table 10: Fixed-token-bin fits and sensitivity to the separate large pair. The five-size columns use only standard-Pile Pythia suites at 14M–410M; six-size columns also include the public deduplicated-v0 6.9B pair. R^2 and SE describe unweighted log–log regressions. These are descriptive fit diagnostics, not independent-seed confidence intervals. The matching five-size fits appear in Figure 4.

Measure	Percentile	γ_5	SE_5	R^2_5	γ_6	R^2_6
Geometric	0–20%	0.330	0.029	0.978	0.275	0.974
Geometric	20–40%	0.205	0.027	0.952	0.185	0.978
Geometric	40–60%	0.120	0.021	0.914	0.109	0.964
Geometric	60–80%	0.073	0.018	0.845	0.069	0.942
Geometric	80–95%	0.050	0.019	0.694	0.047	0.872
Geometric	95–100%	0.059	0.022	0.700	0.045	0.793
Arithmetic	0–20%	0.359	0.029	0.981	0.274	0.953
Arithmetic	20–40%	0.241	0.028	0.960	0.199	0.965
Arithmetic	40–60%	0.134	0.026	0.897	0.122	0.958
Arithmetic	60–80%	0.072	0.022	0.790	0.075	0.933
Arithmetic	80–95%	0.045	0.024	0.545	0.061	0.862
Arithmetic	95–100%	0.053	0.025	0.590	0.054	0.840
BALD	0–20%	0.314	0.032	0.970	0.300	0.989
BALD	20–40%	0.192	0.030	0.933	0.200	0.981
BALD	40–60%	0.113	0.024	0.877	0.125	0.964
BALD	60–80%	0.068	0.021	0.771	0.084	0.933
BALD	80–95%	0.045	0.022	0.577	0.062	0.875
BALD	95–100%	0.052	0.025	0.583	0.058	0.855

Word-continuation positions also have faster geometric decay than word-initial positions: approximately 0.19 versus 0.10 for Pythia and 0.24 versus 0.09 for DataDecide-C4. Thus token composition affects the numerical contrast. Binning by each evaluated model's own loss would change membership with size and, for arithmetic spread, reuse the observed-label probability in both the grouping and score.

D CHANGING THE AVAILABLE CONTEXT

We hold model weights and target passages fixed while changing the preceding text. Pythia seed ensembles cover up to 2,048 context tokens; Qwen generation pairs extend to 32,768. The latter compare training recipes as well as seeds.

D.1 MODELS AND EVALUATION

Pythia uses three members at each of five sizes, 14M–410M: the default release and seeds 1–2, except at 160M, which uses seeds 1–3. Each was trained on 300B tokens. GPT-NeoX-20B is the reference (Black et al., 2022); students and reference receive the same truncated context and are normalized over the same 50,277 assigned token IDs. Table 11 gives checkpoints and revision prefixes.

The Qwen pairs are Qwen2.5-0.5B/Qwen3-0.6B-Base, 1.5B/1.7B, and 3B/4B. We check vocabulary and tokenizer alignment over 151,665 shared assigned IDs. Models use their native context limits, without positional extension or chat templates. We do not compute a Qwen reference KL.

We sample 128 passages from 77 length-eligible books in the PG-19 test split (Rae et al., 2019), with one or two passages per book and sampling seed 2026091421. Each contributes 128 consecutive targets, giving 16,384 targets per context setting. Target blocks do not overlap, although their contexts can. Both tokenizer families use the same books and text anchors. Within a family, every length scores exactly the same targets. Test-split membership does not establish exclusion from pretraining.

Context budgets C are powers of two, from 1 to each family's maximum. Each fresh forward pass scores

$$B(C) = \min\{128, \max[1, \lfloor C/16 \rfloor]\} \tag{12}$$

consecutive targets, with $C - B(C) + 1$ to C preceding tokens. This range is less than 6.25% of the budget; plots use midpoint $c = \sqrt{(C - B + 1)C}$. For $C \leq 16$, each target has exactly C preceding tokens. We recompute truncated prefixes rather than removing entries from a longer cache.

D.2 MEASURES AND FITTING

We measure mean member entropy, BALD, geometric spread, and realized arithmetic spread using Section 2.2's definitions. BALD and both spreads are multiplied by $S/(S-1)$, with $S = 3$ for Pythia and $S = 2$ for Qwen. For Qwen this is only a descriptive rescaling. Values are in nats/token, on separate axes for the two tokenizers. Pythia also uses the context-matched reference $r_{C,X}$ to compute

$$e(C) = \mathbb{E}_X \left[\frac{1}{S} \sum_s \mathrm{KL}(r_{C,X} \| q_{s,C,X}) \right]. \tag{13}$$

This is a reference-based proxy for reducible loss, not the loss above true conditional entropy. Both reference and student distributions change with context.

Points average the 16,384 targets. Error bars and shading show one standard deviation across the 128 passage averages, describing text heterogeneity rather than uncertainty in the mean. For BALD and both spreads, unweighted least squares fits

$$\bar{d}(c) = a + b \log_2(c/32) \tag{14}$$

over budgets $C > 16$: seven Pythia and eleven Qwen settings. Intercepts and slopes are free. The cutoff is fixed, and entropy and reference KL are not fitted. Table 12 reports all coefficients, including poorly linear and increasing series.

Paired intervals resample the 77 books 2,000 times (seed 2026091422), retaining all passages from each sampled book and the same weights across models, contexts, and measures. They condition on the selected corpus and checkpoints. As a split-book check, all 26 of the 37 endpoint changes whose 95% intervals exclude zero have the same sign in both fixed book halves.

D.3 RESULTS

Geometric spread initially contracts: from one to 128 context tokens it falls 40.3% for Pythia-14M and 67.2% for the smallest Qwen pair. Beyond 128 tokens, Pythia entropy falls 3.9–6.6%, while geometric spread changes by -4.5% to $+3.3\%$. Arithmetic spread rises for 31M, 70M, and 160M, with paired intervals excluding zero. The fits from 32 tokens include more of the initial decline and

give geometric endpoint changes of $-15.7, -11.1, -7.4, -1.2, -3.9\%$. Weak dependence at longer contexts therefore does not mean constancy across the entire fitted range.

For Qwen, entropy falls 11.4–13.1% from 128 to 32,768 tokens. BALD and geometric spread decline modestly for the middle and largest pairs and remain approximately flat for the smallest. Its arithmetic spread rises 5.0% (95% interval 1.1–9.0%); the other arithmetic changes have intervals containing zero. Figure 17 compares the measures and Table 13 gives all endpoint changes. Qwen geometric spread appears in Figure 3c.

All five Pythia ensembles move farther from the context-matched reference as context grows (Figure 18, right). Most of the rise occurs before 32–64 tokens. From 128 to 2,048, reference KL rises 1.6–4.0%, with every paired interval excluding zero. Over the full range, reference next-token loss improves by 3.775 nats/token, compared with 2.404–3.371 for students. This supports the interpretation that the reference benefits more from context, although realized losses do not decompose forward KL. The reference's probabilities also change the weighting in that KL.

The corrected reference-relative share falls from 37.7% to 9.3% for 14M and 51.3% to 19.1% for 410M; finite three-member shares are two thirds of these values. Thus context scaling does not support a growing recovered share. A fixed full-context reference would define a different target. Arithmetic and geometric trends can also differ because their pointwise difference is $\log[q^{\mathrm{A}}_{S,x}(y)/q^{\mathrm{G}}_{S,x}(y)]$, which depends on the observed token.

Numerical checks. Pythia and NeoX use FP32 with TF32 disabled. Qwen uses FP32 through 64 tokens and a 16-bit transformer with FP32 output projection thereafter. Checks against FP32 at 128, 512, and 2,048 tokens differ by at most 0.081% for the largest pair and 0.238% for the middle pair. At 32,768, the largest pair differs by at most 0.263% on one 128-target passage. This last check does not cover the whole corpus. Exact memory-efficient attention agrees with native attention within 1.2×10^{-5} total variation at 2,048 tokens. We also check target indexing, vocabulary alignment, probability normalization, the pooling and reference-KL identities, and $0 \le J_S \le \log S$ up to roundoff. Lower-precision reference calculations that failed the accuracy checks are excluded.

Table 11: Exact context-study members and total parameter counts. Revision prefixes identify the evaluated Hugging Face releases. The GPT-NeoX revision was recovered from local download metadata and checked against all 46 previously recorded safetensor shard hashes. All entries are final pretrained checkpoints.

Checkpoint	Parameters	Revision
EleutherAI/pythia-14m	14067712	`cf967c0a9a`
EleutherAI/pythia-14m-seed1	14067712	`0c929f682e`
EleutherAI/pythia-14m-seed2	14067712	`d34e2feab8`
EleutherAI/pythia-31m	30494720	`e556ace21b`
EleutherAI/pythia-31m-seed1	30494720	`9d2c5119aa`
EleutherAI/pythia-31m-seed2	30494720	`7a2c0cfe77`
EleutherAI/pythia-70m	70426624	`a39f36b100`
EleutherAI/pythia-70m-seed1	70426624	`b922141806`
EleutherAI/pythia-70m-seed2	70426624	`9df96128bf`
EleutherAI/pythia-160m-seed1	162322944	`5ed55222b3`
EleutherAI/pythia-160m-seed2	162322944	`6e9b9f6270`
EleutherAI/pythia-160m-seed3	162322944	`a74886a25b`
EleutherAI/pythia-410m	405334016	`9879c9b5f8`
EleutherAI/pythia-410m-seed1	405334016	`33803c4f5a`
EleutherAI/pythia-410m-seed2	405334016	`da2549c8d3`
Qwen/Qwen2.5-0.5B	494032768	`060db6499f`
Qwen/Qwen3-0.6B-Base	596049920	`da87bfb608`
Qwen/Qwen2.5-1.5B	1543714304	`8faed761d4`
Qwen/Qwen3-1.7B-Base	1720574976	`ea980cb0a6`
Qwen/Qwen2.5-3B	3085938688	`3aab1f1954`
Qwen/Qwen3-4B-Base	4022468096	`906bfd4b4d`
EleutherAI/gpt-neox-20b	20554567680	`c292233c83`

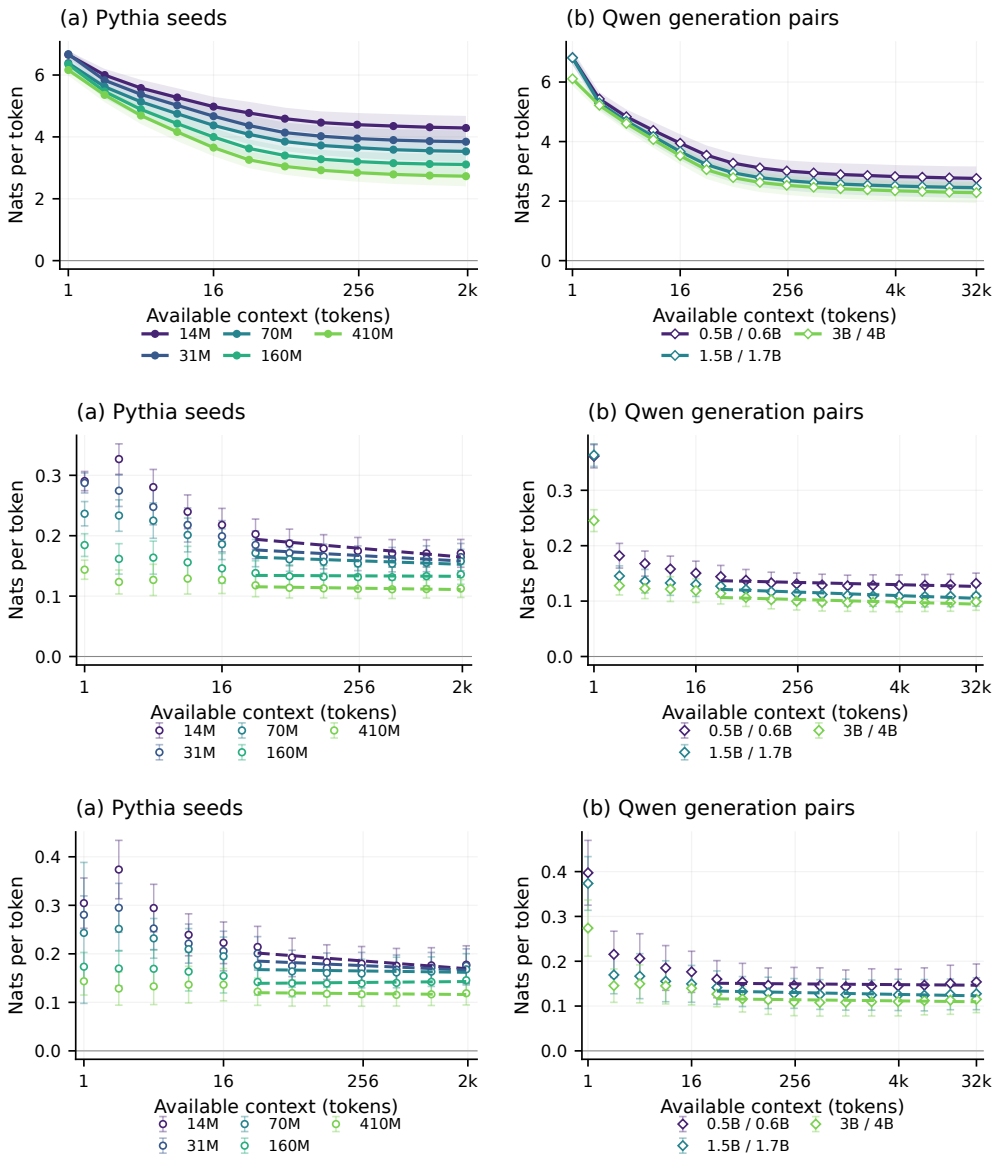

Figure 17: **Context dependence of entropy and ensemble disagreement.** Rows show mean member entropy, BALD, and arithmetic spread; columns show Pythia seed ensembles and Qwen generation pairs. Means use fixed PG-19 targets. Bands and bars are one passage standard deviation. BALD and spread use the $S/(S-1)$ rescaling; dashed lines fit budgets above 16 tokens. Entropy falls at longer contexts, while disagreement changes more weakly and arithmetic spread can rise.

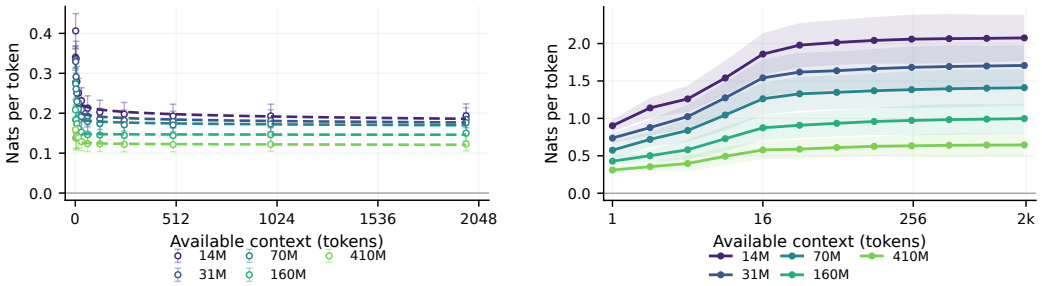

Figure 18: **Pythia spread and reference-based reducible loss.** Left: corrected geometric spread, with free fits above 16 tokens and a linear context axis. Right: mean forward KL from context-matched GPT-NeoX-20B to the three members, with a logarithmic context axis. Both use the same 16,384 targets; bars and shading give one passage standard deviation. Reference KL increases even where spread is nearly flat.

Table 12: Context fits after the first 16 tokens. The fitted mean is $a + b\log_2(c/32)$, where c is the actual context midpoint. Means and coefficients are multiplied by $S/(S-1)$. Intercepts are nats/token; slopes and their paired 95% book-bootstrap intervals are milli-nats/token per doubling. R^2 describes the fit to measured means; change is the fitted endpoint change (%). Pythia uses seven settings from 32 to 2,048; Qwen uses eleven from 32 to 32,768. No slope is forced to zero.

Models	Measure	a	$10^3 b$	95% interval	R^2	Change
Pythia 14M	BALD	0.1939	-4.888	$[-5.278, -4.480]$	0.790	-15.1
Pythia 14M	Geometric	0.2206	-5.792	$[-6.305, -5.252]$	0.777	-15.7
Pythia 14M	Arithmetic	0.2014	-5.370	$[-6.345, -4.434]$	0.689	-15.9
Pythia 31M	BALD	0.1764	-3.053	$[-3.422, -2.665]$	0.595	-10.3
Pythia 31M	Geometric	0.1988	-3.695	$[-4.181, -3.181]$	0.581	-11.1
Pythia 31M	Arithmetic	0.1849	-3.087	$[-3.950, -2.222]$	0.339	-10.0
Pythia 70M	BALD	0.1643	-1.959	$[-2.333, -1.566]$	0.470	-7.1
Pythia 70M	Geometric	0.1834	-2.257	$[-2.754, -1.746]$	0.437	-7.4
Pythia 70M	Arithmetic	0.1675	-0.923	$[-1.854, +0.078]$	0.111	-3.3
Pythia 160M	BALD	0.1343	-0.250	$[-0.660, +0.153]$	0.044	-1.1
Pythia 160M	Geometric	0.1480	-0.304	$[-0.822, +0.210]$	0.044	-1.2
Pythia 160M	Arithmetic	0.1392	+0.625	$[-0.133, +1.411]$	0.296	+2.7
Pythia 410M	BALD	0.1155	-0.778	$[-1.214, -0.356]$	0.557	-4.0
Pythia 410M	Geometric	0.1258	-0.815	$[-1.329, -0.332]$	0.508	-3.9
Pythia 410M	Arithmetic	0.1199	-0.617	$[-1.312, +0.044]$	0.366	-3.1
Qwen 0.5B/0.6B	BALD	0.1368	-1.027	$[-1.330, -0.737]$	0.454	-7.5
Qwen 0.5B/0.6B	Geometric	0.1585	-1.116	$[-1.504, -0.733]$	0.380	-7.1
Qwen 0.5B/0.6B	Arithmetic	0.1508	-0.430	$[-1.046, +0.209]$	0.075	-2.9
Qwen 1.5B/1.7B	BALD	0.1211	-1.609	$[-1.921, -1.314]$	0.774	-13.3
Qwen 1.5B/1.7B	Geometric	0.1393	-1.888	$[-2.299, -1.499]$	0.775	-13.6
Qwen 1.5B/1.7B	Arithmetic	0.1334	-1.059	$[-1.656, -0.454]$	0.500	-8.0
Qwen 3B/4B	BALD	0.1065	-1.190	$[-1.471, -0.926]$	0.575	-11.2
Qwen 3B/4B	Geometric	0.1208	-1.333	$[-1.688, -0.997]$	0.529	-11.1
Qwen 3B/4B	Arithmetic	0.1161	-0.651	$[-1.175, -0.099]$	0.165	-5.6

Table 13: Measured changes (%) from context budget 128 to 2,048 for Pythia and 32,768 for Qwen. Intervals resample the same 77 books 2,000 times at both endpoints. These are endpoint comparisons, not fitted changes or tests of monotonicity. The finite-member correction cancels from every percentage.

Models	Measure	Change (%)	95% interval
Pythia 14M	Entropy	-3.91	$[-4.63, -3.25]$
Pythia 14M	BALD	-4.55	$[-5.43, -3.68]$
Pythia 14M	Geometric	-4.51	$[-5.54, -3.49]$
Pythia 14M	Arithmetic	-2.90	$[-5.26, -0.65]$
Pythia 14M	Reference KL	+1.64	$[+0.65, +2.63]$
Pythia 31M	Entropy	-4.46	$[-5.25, -3.74]$
Pythia 31M	BALD	-0.60	$[-1.57, +0.32]$
Pythia 31M	Geometric	-0.46	$[-1.57, +0.64]$
Pythia 31M	Arithmetic	+2.98	$[+0.88, +5.09]$
Pythia 31M	Reference KL	+2.61	$[+1.54, +3.76]$

Table 13 continued.

Models	Measure	Change (%)	95% interval
Pythia 70M	Entropy	-5.18	$[-6.06, -4.41]$
Pythia 70M	BALD	+0.67	$[-0.26, +1.63]$
Pythia 70M	Geometric	+1.08	$[-0.02, +2.22]$
Pythia 70M	Arithmetic	+4.71	$[+2.56, +6.89]$
Pythia 70M	Reference KL	+2.98	$[+1.83, +4.24]$
Pythia 160M	Entropy	-5.21	$[-6.16, -4.34]$
Pythia 160M	BALD	+3.07	$[+1.96, +4.23]$
Pythia 160M	Geometric	+3.33	$[+2.01, +4.64]$
Pythia 160M	Arithmetic	+4.93	$[+2.81, +7.01]$
Pythia 160M	Reference KL	+4.01	$[+2.45, +5.83]$
Pythia 410M	Entropy	-6.62	$[-7.63, -5.66]$
Pythia 410M	BALD	-0.24	$[-1.43, +1.01]$
Pythia 410M	Geometric	-0.03	$[-1.33, +1.33]$
Pythia 410M	Arithmetic	+1.01	$[-1.23, +3.29]$
Pythia 410M	Reference KL	+2.95	$[+0.72, +5.42]$
Qwen 0.5B/0.6B	Entropy	-11.38	$[-12.79, -10.08]$
Qwen 0.5B/0.6B	BALD	-0.76	$[-2.42, +0.89]$
Qwen 0.5B/0.6B	Geometric	+0.30	$[-1.77, +2.47]$
Qwen 0.5B/0.6B	Arithmetic	+5.01	$[+1.06, +8.97]$
Qwen 1.5B/1.7B	Entropy	-12.04	$[-13.50, -10.68]$
Qwen 1.5B/1.7B	BALD	-6.60	$[-8.34, -4.96]$
Qwen 1.5B/1.7B	Geometric	-6.61	$[-8.63, -4.73]$
Qwen 1.5B/1.7B	Arithmetic	-1.43	$[-5.22, +2.16]$
Qwen 3B/4B	Entropy	-13.08	$[-14.60, -11.67]$
Qwen 3B/4B	BALD	-3.50	$[-5.52, -1.73]$
Qwen 3B/4B	Geometric	-3.15	$[-5.49, -0.95]$
Qwen 3B/4B	Arithmetic	+0.88	$[-3.51, +5.18]$

Table 14: **Separate component fits for public and character-level models.** Public-model exponents use 17 recent-text cells at their shared fitted floor; the character fit uses 35 cells at its separate reference floor. Projected shares use Equation (7) at 10^{12} total parameters. Brackets give the public floor-profile sensitivity or the character reference-to-cap sensitivity described below. These are extrapolations, not observations.

Family	Invisible exponent α	Visible exponent γ	Share at 1T (%)
Pythia	0.29189	0.19988	20.6 [12.1, 22.6]
DataDecide-C4	0.24764	0.16184	18.3 [10.3, 20.2]
DataDecide-Dolma	0.26524	0.15189	21.0 [11.2, 23.4]
Dense Qwen2.5/3	0.29133	0.11784	36.9 [17.6, 42.3]
Character	0.19559	0.11844	13.1 [13.1, 15.7]

E VISIBLE-SHARE SCALING AND LOSS-LAW COMPARISONS

This appendix supports Section 3.3. We first fit separate visible and invisible laws, then vary the assumed irreducible loss and compare the summed law with a direct law for total loss.

E.1 SEPARATE VISIBLE AND INVISIBLE POWER LAWS

The public-model fits in Figure 5 use 17 recent-text cells with matched mean-member losses and corrected geometric spreads: three standard-Pile Pythia suites above 60M, five DataDecide-C4 sets, four DataDecide-Dolma sets, and five dense Qwen2.5/3 pairs. The separate 6.9B Pythia pair is displayed but excluded from these fits. Parameter count includes embeddings; for unequal members it is the geometric mean of member counts. These cohorts are narrower than those used for the main spread comparison in Figure 2.

For each public-model family, ordinary least squares of $\log \widehat{\rho}_\infty^{\mathrm{G}}$ on $\log N$ determines the visible law $V_\infty(N) = cN^{-\gamma}$. Holding these spread fits fixed, we estimate a common E and separate positive amplitudes and decay exponents for $I_\infty(N) = AN^{-\alpha}$ by minimizing

$$R(E, \theta) = \sum_{i=1}^{17} \left[\log(E + I_\infty(N_i) + V_\infty(N_i)) - \log \bar{L}_i\right]^2 . \tag{15}$$

Here θ collects the four public-model families' invisible-law amplitudes and exponents, and each summand uses the component laws for cell i's family. Each cell receives equal weight. The fitted total reducible loss is $I_\infty + V_\infty$, with $E = 0.32435935$ nats/byte. The invisible component refers to the residual after population-corrected spread, not the loss of the measured finite ensemble; the correction has only a descriptive interpretation for cross-generation Qwen pairs. Panel (a) uses observed corrected spread and the residual $\bar{L}_i - E - \widehat{\rho}_{\infty,i}^{\mathrm{G}}$ alongside their fitted component laws. Each component is divided by its own measured value at the smallest fitted size in that family. Table 14 gives the fitted exponents and projections. Pythia's visible exponent differs from Figure 2 because the present fit includes only its three suites above 60M.

For the public families, we propagate floor sensitivity by profiling E while refitting all eight invisible-law parameters and holding the spread laws fixed. With 17 loss observations and nine loss-fit parameters, the nominal 95% residual F-profile rule retains values satisfying

$$R(E) \le R_{\min}\left[1 + \frac{F_{1,8}(0.95)}{8}\right], \qquad R(E) = \min_\theta R(E, \theta). \tag{16}$$

The uncapped profile interval is $[0.261474, 0.358120]$ nats/byte. After imposing the empirical cap $E \le 0.33372841$ discussed in Appendix E.2, the retained interval is $[0.261474, 0.33372841]$ nats/byte. Its lower endpoint comes from the fit profile; its upper endpoint is the cap, rather than a fitted confidence limit. The same floor value applies to every public-model family at each profile point. Shaded bands in panel (b) take the minimum and maximum of $V_\infty/(I_\infty + V_\infty)$ over that profile; marker intervals evaluate $\widehat{\rho}_\infty^{\mathrm{G}}/(\bar{L}_S - E)$ over the same floor interval. Solid curves span measured fitting ranges and dashed curves extend to 10^{12} parameters without adding observations.

This is approximate conditional uncertainty in the floor under the specified loss model and residual criterion. At each floor, the envelope uses the refitted best component laws without propagating their parameter uncertainty at that fixed floor. It also excludes uncertainty in the fitted spread laws,

seed and text sampling, and systematic model or evaluation errors; it is not an entropy confidence interval or a full prediction interval. Positive fitted component laws keep their ratio in $(0, 1)$ without clipping. Corrected empirical ratios need not obey the exact finite-ensemble fraction bound because the population correction is approximate. The direct-loss-law controls in Appendix E.2 use the same upper cap with a separately specified lower sensitivity endpoint. The matched recent-text shared-floor fit here is also distinct from the conditional loss-law comparisons with their own cohorts and protocol-specific floors in Appendix E.3.

Character-level component fit. The black series in Figure 5 uses the same 35 primary character-model cells as Figure 2, spanning five sizes and seven budgets at 50–400 bytes per analytic parameter (Appendix B.7.2). The plotted axis uses exact total counts. Character losses and spreads are converted from bits to nats per byte. Their separate irreducible-loss range is $E = 0.593341$–0.63764990 bits/byte, with the lower reference value used for central points and curves; these cells do not enter the public-model shared-floor fit above. We define measured components as $V_i = \widehat{\rho}^{\mathrm{G}}_{\infty,i}$ and $I_i = \bar{L}_i - E - V_i$, and measured share as $V_i/(\bar{L}_i - E)$. This invisible component differs from the finite-pool quantity $L(q^{\mathrm{G}}_S) - E$ used in Figure 3b, which equals $I_i + V_i/S$.

We fit the visible law by log–log ordinary least squares over all 35 cells, then hold it fixed and minimize equal-cell squared log residuals of the predicted full member loss $E + I_\infty + V_\infty$, separately at each character floor. At the reference floor, $V_\infty = 0.0187134(N/10^9)^{-0.118435}$ and $I_\infty = 0.211161(N/10^9)^{-0.195587}$ nats/byte. At the cap, the invisible exponent is 0.208190. The projected 1T share varies from 13.1% to 15.7% across this assumed range. Character component markers in panel (a) are geometric means across the seven budgets at each size, normalized by the corresponding component's smallest-size mean. In panel (b), open black diamonds show the 35 individual cells and filled diamonds their geometric-mean shares at each size. Shading refits the invisible law across the character floor range; marker intervals vary that floor in the measured ratios. This is a separate sensitivity assumption, not the public-model floor profile or a sampling interval. Size and training data increase together over this fixed budget window, and the character evaluation uses different text and context lengths from the public-model evaluation.

E.2 Dependence on the unknown irreducible loss

For the direct-loss-law controls below and the ensemble-versus-larger-model comparisons, the assumed recent-text irreducible loss is 0.33372841 nats/byte, rounded downward from the lowest achieved single-model loss among the evaluated models (DeepSeek-V4-Pro-Base, 0.3337284169). This improves the Kimi-K2-Base anchor of 0.34022899 and the earlier GLM-4.5-Base anchor of 0.35268947 nats/byte and keeps the best measured model's reducible loss positive. It does not identify the unknown population conditional entropy. The achieved loss supplies an empirical cap for sensitivity analysis, not a statistical bound on population entropy. Cross-tokenizer normalization and finite text sampling add further uncertainty.

We evaluate the pinned Kimi-K2-Base release with its native FP8 weights, preserving the FP32 routing-correction buffers and using BF16 for the remaining weights. No chat template is applied. The character protocol retains all 1,024 source chunks and scores 521,264 decoded target bytes (112,496 tokens); the recent-text protocol retains all 96 documents and scores 431,140 bytes (87,517 tokens), with the same 1,024-token limit and 64-token warm-up as the earlier references. Tokenizer-default special tokens, decoded-byte normalization, and source-segment boundaries are retained. Scoring the full output vocabulary and the earlier tokenizer-length slice gives identical losses for this checkpoint. Kimi is an additional single-model reference, not a new ensemble-spread measurement; it is not evaluated on C4 here.

We also evaluate the pinned DeepSeek-V4-Pro-Base release (DeepSeek-AI, 2026), using the official native FP8 checkpoint with SGLang 0.5.19. Its native compressed-attention cache stores non-RoPE keys and values in FP8 and RoPE components in BF16. The checkpoint receipts, live server configuration, and GPU checks are recorded; the scoring API does not expose resident tensor dtypes. We preserve tokenizer-default behavior without adding a beginning-of-sequence token or a chat template, verify exact input-token identities, and retain all per-token log probabilities. The character protocol scores all 1,024 chunks, comprising 521,257 target bytes and 111,513 tokens; the recent-text protocol scores all 96 documents, comprising 431,669 bytes and 88,104 tokens, under the same context limits and warm-up. The tokenizer and output vocabularies both contain 129,280 tokens, so the full-vocabulary and legacy-slice losses coincide. The character loss is 0.68195265 bits/byte,

Table 15: Achieved individual-model losses on the two evaluation corpora, in nats per scored byte. Each corpus uses the same 96 source segments across these models, subject to tokenizer-dependent truncation. These are measured cross-entropies, not conditional-entropy estimates; the corpora have different text distributions. Kimi-K2-Base and DeepSeek-V4-Pro-Base were evaluated only on recent text; dashes denote unmeasured C4 losses.

Model	C4 validation	Recent text
Qwen2.5-72B	0.479487	0.395215
DeepSeek-V3-Base	0.430436	0.365561
DeepSeek-V3.1-Base	0.442147	0.373619
GLM-4.5-Air-Base	0.450985	0.384772
GLM-4.5-Base	0.417401	0.352689
Kimi-K2-Base	—	0.340229
DeepSeek-V4-Pro-Base	—	0.333728

Table 16: Decay-rate differences, fitted as $\widehat{f}_\infty \propto N^{\eta-\gamma}$, on the same 12 primary seed sets above 60M parameters, excluding the public 6.9B deduplicated-v0 pair. Positive values indicate a growing recovered share. The pooled regression uses a common intercept across families and is descriptive; the family fits preserve the same floor dependence.

Assumed E	Pooled	Pythia	DD C4	DD Dolma
0	-0.088	-0.057	-0.055	-0.047
0.3	0.037	0.061	0.056	0.077
0.32	0.055	0.077	0.073	0.097
0.333728	0.069	0.088	0.086	0.112

leaving Kimi as the better character reference; DeepSeek improves the recent-text cap. Neither model adds a C4 or ensemble-spread measurement.

Table 15 reports achieved losses rather than fitted asymptotes on both text sources. GLM-4.5-Base reaches 0.417401 nats/byte on C4 validation and 0.352689 on recent text; Kimi-K2-Base reaches 0.340229 on recent text and DeepSeek-V4-Pro-Base lowers this reference to 0.333728, without adding a C4 measurement. The recent-text construction in Appendix B.2 addresses the memorization concern, but none of these achieved losses is a measured conditional entropy; different corpora require separate irreducible-loss assumptions. In particular, the earlier fitted C4 value $E = 0.436514$ retained for the conditional loss-law comparisons in Appendix E.3 exceeds this achieved C4 loss. It should not be interpreted as an empirical entropy cap. Those comparisons remain conditional on their stated value, while Figure 13 requires no irreducible-loss assumption. Lower recent-text losses alone cannot establish less memorization or a more accurate entropy estimate, because the underlying text distribution also changes.

The primary fraction fits use 12 seed sets with total $N \geq 60$M and a consistent recipe within each ladder: three standard-Pile Pythia suites, five DataDecide-C4 sets, and four DataDecide-Dolma sets. The separate 6.9B pair is displayed but excluded from these fits. Each uses the same corrected spread and retained mean-member loss in numerator and denominator. Figure 19 and Table 16 vary only the irreducible loss. The positive trend holds at $E = 0.300$, 0.320, and 0.33372841 within every family; it need not hold for all mathematically possible irreducible-loss values. The zero-irreducible-loss result is included to make that limitation visible. No lower bound near 0.300 has been established by these experiments. More precisely, let $x_i = \log N_i$ and d_i be each measured corrected spread. The finite-range share slope changes sign at the root

$$\sum_i (x_i - \bar{x}) \log\left[\frac{d_i}{L_i - E_{\text{crit}}}\right] = 0. \tag{17}$$

Using the same observations gives $E_{\text{crit}} = 0.24914$ for the pooled 12-set regression, 0.18996 for Pythia, 0.20088 for DataDecide-C4, and 0.17154 for DataDecide-Dolma, all in nats/byte. The separate dense-Qwen root is 0.14397. Over the admissible interval checked here, fitted growth occurs above these roots. They depend on the chosen regression and models and are neither entropy estimates nor statistical bounds.

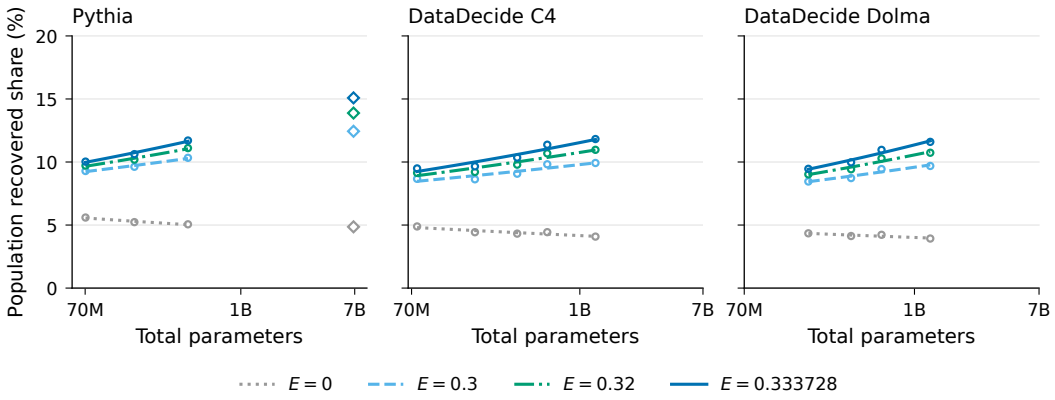

Figure 19: **Recovered fractions depend on the assumed irreducible loss.** The primary seed measurements from Figure 20, evaluated under four values of E in nats/byte. Curves are power-law fits to the same seed sets; open diamonds show the excluded 6.9B deduplicated-v0 pair. The zero-irreducible-loss control reverses the general trend, so growth is not implied by the spread measurements alone. The three nonzero choices are sensitivity assumptions, not an entropy confidence interval.

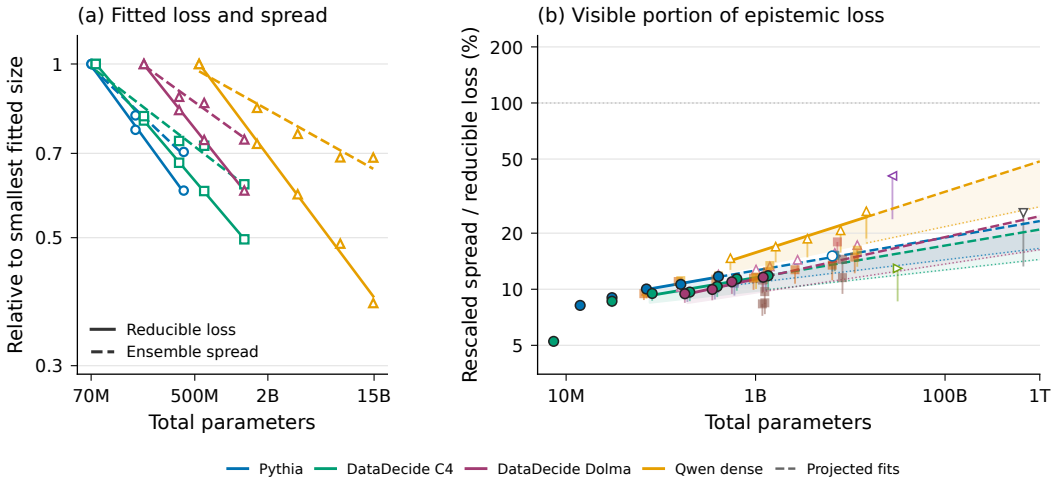

Figure 20: **Direct-power-law alternative for total epistemic loss.** Left: reducible loss and spread, normalized within each family; power laws fit the 12 primary seed sets above 60M and, separately, five dense Qwen2.5/3 pairs. Right: all 60 ensemble comparisons, with the colors and symbols of Figure 2. Ranges vary the assumed irreducible loss from 0.300 to 0.33372841 nats/byte. These ranges are sensitivity assumptions, not a fitted-floor interval. Qwen is a cross-generation ensemble. Figure 5 instead fits separate visible and invisible laws and estimates the floor from their sum.

As a selection-sensitivity check, including the public 6.9B pair gives pooled share exponents 0.064, 0.079, and 0.091 at irreducible-loss values 0.300, 0.320, and 0.33372841; the zero-irreducible-loss exponent is -0.037. The primary 12-set fits preserve this direction at the nonzero irreducible-loss values while keeping a consistent Pythia training recipe.

Hubble's 8B near-replicates have corrected shares of about 11.5–14.5% at the assumed irreducible loss, but do not extend the pure-seed inference to that family.

Figure 20 includes all 60 model comparisons, with one vertical interval per observation for $E \in [0.300, 0.33372841]$. The three seed-family curves are unweighted log–log regressions on the same 12 primary seed sets; their bands refit the ratios at both endpoints. The intervals describe the effect of an assumed irreducible loss, rather than sampling uncertainty. The approximate population correction means the estimated share need not obey the exact finite-ensemble fraction bound. Both the spread normalization and assumed irreducible loss must therefore be identified when interpreting broad heterogeneous comparisons.

The figure extends each Pythia, DataDecide, and dense Qwen irreducible-loss sensitivity band from that family's largest fitted size to 10^{12} total parameters. The projected boundaries evaluate the same

fitted power laws at $E = 0.300$ and 0.33372841; no additional observations enter the fit. Solid curves mark the measured fitting ranges and dashed curves the extrapolation. The shading represents sensitivity to the assumed irreducible loss, not a confidence or prediction interval.

The separate Qwen fit uses all five dense Qwen2.5/3 pairs, from 0.543B to 14.769B in geometric-mean total member parameters. At $E = 0.33372841$, the spread exponent is 0.117842 ± 0.015322, the reducible-loss exponent is 0.281060 ± 0.010035, and the recovered-share growth exponent is 0.163218 ± 0.023419. At $E = 0.300$ the share exponent is 0.106588 ± 0.018301; at zero irreducible loss it is -0.035561 ± 0.013071. These are residual regression standard errors conditional on the irreducible loss and selected pairs. The projected share at 10^{12} parameters ranges from 27.7% to 48.6% under the two displayed irreducible-loss values. This is a descriptive projection across model generations, not evidence from additional seed suites or an observed trillion-parameter Qwen ensemble.

E.3 One loss law or a sum of component laws?

Mean member log loss equals the loss of a specified pooled predictor plus the spread from forming that pool. This decomposition is exact, but it does not determine the functional form of either component. The pooled-log-loss power law used below is a diagnostic approximation over the measured range. If total epistemic loss and spread retained unequal exponents with the spread decaying more slowly at arbitrarily large sizes, their difference would eventually become negative. Consequently those simultaneous power laws cannot remain valid indefinitely, and this comparison does not justify asymptotic extrapolation. In particular, a power law for total epistemic loss and a different power law for ensemble spread do not imply a power law for their difference. We compare these two empirical descriptions on the same prediction target: measured mean member loss.

E.3.1 Matched targets, irreducible-loss values, and validation

Let N be model parameter count, D the training-token count, and E an assumed irreducible loss. A cell is one model-size and training-budget setting. Write L_i for its mean member log loss and $\rho^{\mathrm{G}}_{S,i} = L_i - L(q^{\mathrm{G}}_{S,i})$ for the spread from the normalized geometric pool of its S members. We use the approximate population correction $\widehat{\rho}^{\mathrm{G}}_{\infty,i} = S\rho^{\mathrm{G}}_{S,i}/(S-1)$. Approach A fits the conventional law directly:

$$\widehat{L}_{\mathrm{A}}(N, D) = E + AN^{-a} + BD^{-b}. \tag{18}$$

Approach B fits the same form to the pooled-log-loss target $R_i = L_i - \widehat{\rho}^{\mathrm{G}}_{\infty,i}$, independently fits a size-only law to the measured spread, and adds their predictions:

$$\begin{aligned} \widehat{R}(N, D) &= E + A'N^{-a'} + B'D^{-b'}, \\ \widehat{\rho}^{\mathrm{G}}_{\infty}(N) &= cN^{-\gamma}, \qquad \widehat{L}_{\mathrm{B}}(N, D) = \widehat{R}(N, D) + \widehat{\rho}^{\mathrm{G}}_{\infty}(N). \end{aligned} \tag{19}$$

The spread law has no data exponent. Final-checkpoint comparisons omit the D term from both loss laws. Since R_i uses the approximate $S/(S-1)$ correction, it estimates a population-pool loss (the invisible portion plus irreducible loss); it is not the loss of an evaluated infinite ensemble. It also includes shared optimization and data limitations, so it should not be identified with the best-in-family capacity loss A_N, defined as the lowest single-model loss at size N minus E.

Both loss fits minimize the Huber penalty on residuals in the logarithm of their respective full loss targets: the penalty is quadratic near zero and linear beyond an absolute residual of 0.01. The spread fit uses ordinary least squares of $\log \widehat{\rho}^{\mathrm{G}}_{\infty,i}$ against $\log N_i$. We score A and the *sum* in B against the same original L_i, using root mean squared error (RMSE) in the original loss units. Each cell receives equal weight. At a fixed irreducible loss, B fits six parameters rather than A's four for an N, D law, or four rather than two for a size-only law.

Every paired comparison uses the same cells, seed exclusions, held-out sizes, and irreducible loss. Character-model fits use $E = 0.593341$ and the Kimi cap 0.63764990 bits/byte. A separate control with freely fitted irreducible loss allows $0 \le E \le 1.5$ in both approaches; it is an identifiability check, not an entropy interval. Pythia and C4 Qwen use the study's working value $E = 0.436514$ nats/byte, and recent-text Qwen uses 0.33372841. These different texts and context protocols require separate irreducible-loss values. The C4 value is a fitted asymptote rather than a measured conditional entropy; its paired comparison is conditional on that assumption.

Cross-validation omits an entire width, model size, or Qwen pair. Both the pooled-log-loss law and the spread law are refitted on the retained cells; the omitted cell's measured spread is never added to

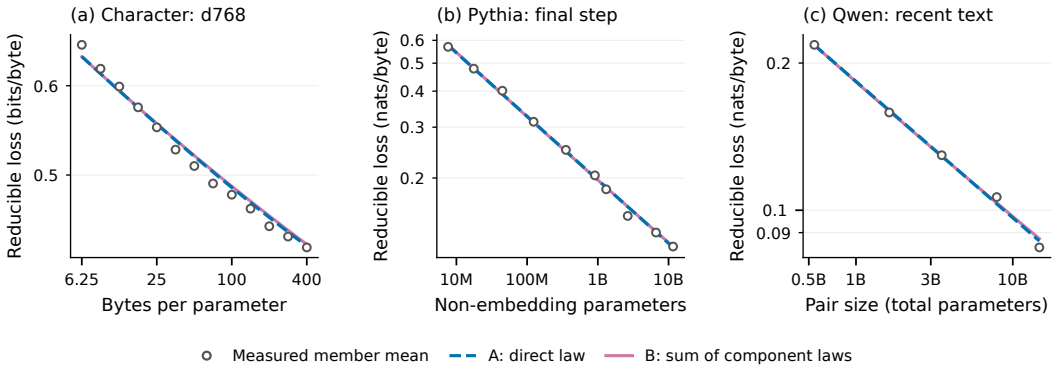

Figure 21: **Direct and component laws make similar member-loss predictions.** (a): hidden width $d = 768$ (labelled d768) across training budgets, including the 283 and 400 bytes-per-parameter cells; both laws are fitted to the complete 78-cell character grid. (b): Pythia's ten final checkpoints, including the larger rerun pairs. (c): the five dense Qwen pairs on recent text. Points are measured mean member losses minus the stated irreducible loss. Curves predict that same quantity: A directly, B by adding its independently fitted pooled log loss and spread. Logarithmic axes show deviations from simple power laws; Table 17 evaluates errors on untransformed member loss and adds held-out-size tests.

its prediction. Errors are pooled over the omitted cells. These are predictive comparisons, not formal tests of statistical equivalence.

E.3.2 Evaluation populations

Character models. The retuned training recipe adjusts batch sizes and learning rates at smaller widths (Appendix B.7). Its complete grid has 78 cells: six hidden widths $d \in \{128, 192, 256, 384, 512, 768\}$, each with 13 budgets from 6.25 to 400 bytes per parameter. It includes the three width-768 seeds at approximately 283 and 400 bytes per parameter. All cells use the same 524,288-byte evaluation, with seeds 0–2 except at width 192, where seeds 0 and 1 are retained. Geometric and arithmetic spreads and Bayesian active learning by disagreement (BALD), the arithmetic-pool entropy minus mean member entropy, are evaluated on these same members and positions. The held-out unit is a width.

This comparison uses the retuned recipe at all small widths. Its fitted geometric spread is $0.489N^{-0.142}$ bits/byte over all 78 cells. It differs from the primary $0.304N^{-0.117}$ law in Figure 3, which uses the specified original-recipe seed selection and 35 cells at 50–400 bytes per parameter. The two laws describe different training populations and windows; neither is substituted for the other.

Pythia. We score 90 size–checkpoint cells on the 96 C4 segments, with a 1,024-token context and a 64-token warm-up. Five small sizes (14M–410M release labels) use three true seeds each; five larger sizes (1B–12B) use main/v0 release-rerun pairs. The latter need not differ only in seed. Here N excludes embeddings and $D = 2{,}097{,}152$ times the training step. There are ten final-step cells at step 143,000, and 46 cells from step 32,000 onward. These cohorts differ from the larger retained seed suites and recent-text evaluation in the primary figures.

Qwen. We compare five dense Qwen2.5/Qwen3 pairs, from 0.5B/0.6B through 14B/14B, on C4 and recent text separately. Each pair uses the geometric mean of its members' total parameter counts. These are cross-generation comparisons, so interpreting $2\rho_2^{\mathrm{G}}$ as an independent-seed population spread is only a descriptive approximation. The target subtracted in B is the corrected spread, not the stored finite-pair spread.

E.3.3 Similar accuracy, different component exponents

On the complete character grid at $E = 0.593341$, A and B have in-sample RMSEs of 0.0193 and 0.0198 bits/byte, and leave-one-width-out RMSEs of 0.0290 and 0.0294. Raising the shared irreducible loss to the Kimi cap 0.63764990 gives held-out errors of 0.0303 and 0.0310. Allowing the irreducible loss to vary reverses their small ordering: 0.0304 versus 0.0301. The fitted irreducible-loss values in this control are 0.533 for A and 0.523 for B's pooled-log-loss component.

Pythia's final-step comparison gives 0.0056 versus 0.0061 nats/byte in sample and 0.0075 versus 0.0085 on omitted sizes. Qwen's five dense pairs slightly favor A on C4; on recent text A has the smaller in-sample error, while the omitted-pair errors are nearly tied (0.003210 for A and 0.003197

Table 17: Direct and component fits have similar predictive errors, without a consistent winner. RMSE is evaluated on the same original member-mean loss for A and B. B uses geometric spread with a size-only law. Held-out errors pool all omitted cells; both components are refitted without the held-out size. Character results use the updated 78-cell grid. The fits with freely estimated irreducible loss share bounds $0 \leq E \leq 1.5$. Units differ between character and pretrained studies.

Cohort	E	Cells	A: fit	B: fit	A: held out	B: held out
Character models: bits/byte, leave one of six widths out						
Character, N, D	0.593341	78	0.0193	0.0198	0.0290	0.0294
Character, N, D	0.637650	78	0.0200	0.0203	0.0303	0.0310
Character, N, D	free	78	0.0191	0.0197	0.0304	0.0301
Pythia: nats/byte, leave one of ten sizes out						
Final, size only	0.436514	10	0.0056	0.0061	0.0075	0.0085
Steps $\geq$ 32k	0.436514	46	0.0100	0.0104	0.0124	0.0135
All steps	0.436514	90	0.0365	0.0366	0.0434	0.0419
Qwen2.5/3: nats/byte, leave one pair out						
C4, dense	0.436514	5	0.0028	0.0035	0.0047	0.0052
Recent, dense	0.333728	5	0.0020	0.0022	0.0032	0.0032

for B). Thus the numerical advantage depends on the irreducible loss and model selection; the separate spread law gives no consistent improvement in predicting member loss.

The pooled-log-loss exponent is steeper in every main comparison. On characters, $a = 0.230$ becomes $a' = 0.245$ at the reference irreducible loss, while $b = 0.145$ and $b' = 0.144$ remain close. Pythia's final-step size exponents are 0.220 and 0.233; the dense Qwen exponents are 0.301 and 0.337 on C4, and 0.279 and 0.312 on recent text. Removing the more slowly declining spread therefore changes the fitted size exponent even when the summed prediction barely changes. This supports reporting the spread separately when it is the object of interest; it does not establish two exact asymptotic powers.

Other measures and joint fitting. On the 78-cell character grid, replacing geometric spread by arithmetic spread or BALD gives B held-out RMSEs of 0.0291 and 0.0297 at the reference irreducible loss, versus 0.0294 for geometric spread. BALD is an uncertainty diagnostic rather than the exact visible portion. A third approach fits $E + AN^{-a} + BD^{-b} + cN^{-\gamma}$ jointly to member loss alone, without observing a separate spread target. Its in-sample error at the reference irreducible loss is 0.0193, but its held-out error rises to 0.0364; with free E, it is 0.0445. At the higher fixed irreducible-loss values it nearly collapses to A. These controls give no predictive benefit from identifying two size powers through member loss alone.

E.3.4 Limits of the data axis and extrapolation

Pythia's intermediate checkpoints are taken during training, without a separate learning-rate decay phase for each checkpoint. On all 90 cells, both approaches fit poorly (RMSE about 0.0365 nats/byte); the data-term amplitude approaches its allowed ceiling. Restricting to the late checkpoints improves the fit, but the profile over a fixed data exponent is nearly flat. The size-only final-step result is therefore the clearest Pythia comparison. A size-only spread term cannot remove residual structure along the training trajectory.

Fitting only Pythia's five small true-seed sizes and predicting the five larger main runs gives final-step RMSEs of 0.0093 for A and 0.0115 for B; the late-checkpoint equivalents are 0.0156 and 0.0166. On C4, the five dense Qwen pairs predict Qwen2.5-72B's single-model loss as 0.4867 (A) or 0.4919 (B), against the measured 0.4795 nats/byte. Both approaches overpredict these larger-model losses, with a larger error for B. These finite extrapolations favor the direct law, while leaving the separately measured spread law useful for describing ensemble effects.

The experiments consequently support comparable finite-range fits, with no consistent predictive gain from splitting the loss. They do not establish that the invisible portion is exactly a power law, nor that subtracting a fitted spread from a conventional loss law must preserve a power law. Either description can approximate the measured range, subject to the stated irreducible loss, seed-population, and extrapolation limits.

F CONTEXT-DEPENDENT UNCERTAINTY

F.1 TARGET AND SCORES

For S students $q_{N,s,x}$ and a stronger reference r_x, the target in Figure 6 is

$$e_N(x) = \frac{1}{S}\sum_{s=1}^{S} \mathrm{KL}(r_x \| q_{N,s,x}). \tag{20}$$

It measures expected reducible loss if outcomes follow the reference, whose distribution is only a proxy for the unknown truth. We use $S = 3$ for character, Pythia, and DataDecide ensembles, and $S = 2$ for Qwen pairs. We select references by loss on separate text, without using uncertainty-prediction scores.

Predictors use one designated member's entropy $h_s(x) = H(q_{N,s,x})$, corrected geometric spread $g(x) = \frac{S}{S-1}[-\log Z_{N,x}]$, and raw BALD $B(x) = H(q^{\mathrm{A}}_{N,S,x}) - \frac{1}{S}\sum_{s=1}^{S} H(q_{N,s,x})$. All use full distributions over a shared vocabulary. Arithmetic spread is excluded because it also requires an observed token or a target distribution. Entropy and the hidden probe use character seed 0, the default Pythia seed (seed 1 at 160M), DataDecide's default seed, or the Qwen2.5 member. The displayed finite-member correction does not turn the cross-generation Qwen pairs into independent seed replicates.

F.2 MODELS, TEXT, AND REFERENCES

Character models. Four retuned student sizes have $0.858, 6.461, 21.441, 50.678$M total parameters, with seeds 0–2 at 50 bytes per analytic parameter. Model size and data grow together. From an 8 MiB FineWeb validation stream, we fix 256 nonoverlapping 512-byte windows (seed 2026091406), with 32 sampled positions after at least 64 context bytes in each. Eight 1 MiB source regions contribute 32 windows each. Even-indexed regions train predictors and odd-indexed regions test them, giving 4,096 contexts per half. These regions are stream blocks, not certified independent documents.

The reference is `facebook/blt-7b` (Pagnoni et al., 2024), selected over EvaByte (EvaByte, 2025) on 64 separate segments with 28,736 targets. A replacement must improve both raw and byte-conditioned loss, with a wholly negative paired 95% window-bootstrap interval for the conditioned difference. BLT improves on EvaByte by 0.0832 nats/byte (interval $[0.0735, 0.0930]$); region resampling gives $[0.0753, 0.0915]$. Its 10.55B stored parameters include 3.84B hash embeddings; a separate 99.5M entropy model sets patch boundaries. Forecasts condition on the 256 ordinary byte outputs. Excluded special-token mass averages 0.00015 but reaches 0.3322, so conditioning can matter at individual positions. BLT's aggregate NLL-minus-entropy of -0.0024 does not establish full calibration.

BLT uses release `b65201dc`, implementation `9774ed4f`, and patch threshold 1.335442066192627. FP32 causal-attention checks agree with the native implementation within 4.0×10^{-6} total variation; changing future bytes changes earlier forecasts by at most 4.4×10^{-6}.

Pythia. Five three-seed suites span 14M–410M at a common 300B-token budget, with memberships as in Appendix D.1. The reference is MPT-30B (The Databricks AI Research Team, 2023), selected against GPT-NeoX-20B and DCLM-7B (Li et al., 2024) on 16 separate recent-text windows. Their losses are 2.066, 2.341, and 2.191 nats/token, respectively; MPT minus NeoX has paired 95% interval $[-0.320, -0.234]$. On the study positions, MPT, NeoX, and DCLM give 2.132, 2.372, and 2.405, illustrating that reference quality can change with evaluation text.

The study uses 96 C4 windows of 1,024 tokens, with 32 positions per window after the 64-token warm-up: 3,072 contexts. Alternating four-window blocks define training and test halves of 1,536 contexts. Token IDs and merges match across students and references; distributions are normalized over the 50,277 assigned IDs. MPT's released BF16 weights are cast to FP32, with TF32 disabled and FP64 reductions. Native-forward log probabilities agree within 2.1×10^{-4} nats; excluded padded-output mass is at most 4.8×10^{-7}.

DataDecide. C4-trained students cover six sizes from 6.96M to 597.7M; Dolma-trained students cover three from 190.5M to 597.7M, each with three seeds. We omit 1.28B here because reference quality becomes less certain as students approach it. OLMo-7B remains the reference after a separate 64-document benchmark: its loss is 2.002 nats/token, versus 2.005 for the July 2024 OLMo release and 2.140 for OLMoE-1B-7B-0125 (Muennighoff et al., 2024). Neither candidate passes the same

replacement rule. Candidate-minus-original intervals are $[-0.033, 0.041]$ and $[0.089, 0.190]$, using 5,000 paired window resamples.

From the 96 recent-text windows we sample 32 positions after warm-up (seed 2026092301), giving 3,072 contexts shared across sizes. Correlations use all positions; C4 prediction uses alternating four-window blocks for training and test (1,536 contexts each). The 12 training blocks form four consecutive validation folds. Distributions use the same 50,280 assigned IDs, FP32 inference, and FP64 reductions, without adding special tokens. Removed padding mass is at most 1.20×10^{-4} and full-forward checks agree within 7.8×10^{-5} nats. The reference's study loss is 1.994 nats/token, below the largest C4 and Dolma students' 2.511 and 2.470.

Qwen. Five dense base-model pairs compare Qwen2.5 with Qwen3 at nominal sizes $0.5/0.6$, $1.5/1.7$, $3/4$, $7/8$, and $14/14$B (Qwen Team, 2024; 2025). These compare generations and training recipes, rather than random seeds. The size coordinate is the geometric mean of the two actual parameter counts. Qwen2.5 supplies the primary entropy and hidden states at every size, fixed before observing results. The fixed reference is Qwen2.5-72B. On a separate pilot of 64 documents whose URLs are disjoint from the study, with 32 positions per document, its loss is 1.9451 nats/token, versus 2.0394 for Qwen2.5-14B and 2.0796 for Qwen3-14B. Reference-minus-student paired 95% intervals are $[-0.1251, -0.0650]$ and $[-0.1760, -0.0943]$, respectively, from 5,000 whole-document bootstrap resamples. Both raw and assigned-vocabulary losses pass this quality check before reference study inference or uncertainty fitting.

The study uses 96 recent-web documents, each truncated to at most 1,024 tokens, with 32 sampled targets after at least 64 context tokens (seed 2026092508): 3,072 contexts. Alternating four-document blocks define training and test halves of 1,536 contexts, also evaluated in reverse. All eleven student and reference `tokenizer.json` files are byte-identical. Primary distributions retain all 151,665 assigned token IDs, including 22 shared special tokens, and exclude only unassigned padding outputs. Reconditioning on the 151,643 ordinary tokens is a sensitivity analysis. Inference uses FP32 weights and forward passes, with TF32 disabled; released BF16 weights are cast to FP32. Logits are normalized in FP64, log probabilities are stored in FP32, and these are renormalized in FP64 for reductions. Excluded padding mass is at most 2.6×10^{-5}.

F.3 Correlations and prediction

Table 18 gives Pearson and Spearman correlations at each fixed size. Character, Pythia, and Qwen use the primary test halves; DataDecide uses all sampled contexts. Qwen correlations on all 3,072 contexts are an additional sensitivity check. No coefficients, bins, or cross-size pooling enter these correlations. Entropy correlates more strongly with reference-based total loss than with spread at every character and Pythia size; this ordering is mixed for DataDecide. Entropy is also not uniformly better than spread at ranking total loss: at Pythia-410M their Spearman correlations with that target are 0.696 and 0.735, and at DataDecide-C4 597.7M they are 0.728 and 0.843.

For the three-member ensembles, the entropy-providing member also enters the spread. Removing it leaves positive rank correlations: 0.49–0.81 across character sizes, 0.41–0.62 for Pythia, and 0.44–0.73 for DataDecide, compared with 0.51–0.82, 0.44–0.63, and 0.47–0.75 using all three members. These ranges describe sizes, not confidence intervals. This check does not apply to Qwen pairs, where removing one member leaves no disagreement score.

Single-score predictors are affine least-squares fits; combinations use affine ridge. Tables 19, 20, and 21 report held-out results with primary and reversed text splits. At 50.7M character parameters, entropy plus BALD gives $R^2 = 0.253$ and $r_s = 0.905$, compared with 0.204 and 0.880 for entropy alone. At Pythia-410M the corresponding pairs are $0.228/0.788$ and $0.177/0.696$. Good rankings therefore coexist with modest numerical accuracy.

For Qwen, affine spread has higher held-out R^2 than affine entropy at every size on both splits. At the $14/14$B pair, primary-split R^2 is 0.540 for spread, 0.111 for entropy, and 0.524 for their combination. The Adam probe has negative R^2 beyond the smallest pair on both splits. These fits assess the specified linear head with limited supervision; they do not establish that the hidden states lack information about reference-based reducible loss. On the primary split, reconditioning on ordinary tokens changes R^2 by at most 0.0037 for the three reported scalar methods and 0.0091 for Adam across all sizes.

Reference choice materially affects the target. Holding Pythia-410M students, scores, and splits fixed, entropy plus spread has $R^2 = 0.221$ with NeoX, 0.226 with MPT, and 0.464 with DCLM. The higher

Family	Size	h versus g		h versus e		g versus e	
		r	r_s	r	r_s	r	r_s
Character	0.858M	0.187	0.509	0.478	0.760	0.263	0.582
	6.46M	0.296	0.734	0.487	0.842	0.320	0.788
	21.4M	0.354	0.791	0.467	0.869	0.353	0.835
	50.7M	0.356	0.824	0.453	0.880	0.353	0.863
Pythia	14M	0.242	0.442	0.456	0.592	0.330	0.504
	31M	0.326	0.515	0.476	0.626	0.392	0.586
	70M	0.405	0.582	0.462	0.641	0.451	0.625
	160M	0.381	0.625	0.471	0.687	0.375	0.679
	410M	0.412	0.631	0.431	0.696	0.389	0.735
DataDecide C4	6.96M	0.349	0.466	0.493	0.587	0.356	0.513
	28.8M	0.380	0.574	0.511	0.691	0.433	0.671
	76.4M	0.440	0.647	0.506	0.718	0.482	0.736
	190.5M	0.530	0.714	0.456	0.734	0.522	0.781
	371.5M	0.489	0.739	0.412	0.742	0.518	0.827
	597.7M	0.393	0.732	0.374	0.728	0.517	0.843
DataDecide Dolma	190.5M	0.544	0.715	0.470	0.732	0.525	0.795
	371.5M	0.537	0.738	0.443	0.747	0.505	0.816
	597.7M	0.528	0.747	0.424	0.750	0.549	0.845
Qwen	0.5/0.6B	0.355	0.712	0.430	0.738	0.494	0.795
	1.5/1.7B	0.226	0.684	0.404	0.738	0.491	0.832
	3/4B	0.281	0.701	0.358	0.750	0.563	0.835
	7/8B	0.320	0.728	0.380	0.770	0.543	0.863
	14/14B	0.306	0.729	0.335	0.745	0.739	0.875

Table 18: Correlations across individual contexts at each fixed size. Each pair of columns gives Pearson r and Spearman r_s. h is the entropy of one designated member, g is corrected geometric spread, and e is reference-based total reducible loss. Character, Pythia, and Qwen entries use their primary test sets (1,536 contexts for Pythia and Qwen); DataDecide uses the same 3,072 selected recent-text positions for both scores and OLMo-7B as reference. Qwen uses Qwen2.5 entropy, two-member generation-pair spread, and Qwen2.5-72B as reference.

			R^2			Spearman r_s		
Family	N (M)	Split	h_s	$h_s + g$	$h_s + B$	h_s	$h_s + g$	$h_s + B$
Character	0.86	P	0.227	0.259	0.261	0.760	0.774	0.774
Character	0.86	R	0.264	0.299	0.303	0.772	0.784	0.783
Character	6.46	P	0.236	0.270	0.275	0.842	0.861	0.861
Character	6.46	R	0.256	0.295	0.295	0.845	0.863	0.864
Character	21.44	P	0.216	0.256	0.261	0.869	0.890	0.891
Character	21.44	R	0.237	0.282	0.286	0.868	0.890	0.892
Character	50.68	P	0.204	0.246	0.253	0.880	0.904	0.905
Character	50.68	R	0.194	0.233	0.234	0.875	0.899	0.900
Pythia	14.07	P	0.203	0.255	0.258	0.592	0.645	0.647
Pythia	14.07	R	0.263	0.313	0.316	0.627	0.666	0.668
Pythia	30.49	P	0.223	0.284	0.288	0.626	0.684	0.688
Pythia	30.49	R	0.257	0.317	0.322	0.642	0.705	0.707
Pythia	70.43	P	0.208	0.286	0.285	0.641	0.706	0.707
Pythia	70.43	R	0.255	0.308	0.316	0.658	0.722	0.725
Pythia	162.32	P	0.214	0.256	0.263	0.687	0.752	0.754
Pythia	162.32	R	0.257	0.325	0.339	0.685	0.760	0.768
Pythia	405.33	P	0.177	0.226	0.228	0.696	0.786	0.788
Pythia	405.33	R	0.228	0.337	0.346	0.695	0.797	0.804

Table 19: Numerical prediction with the same single-member entropy and reference-KL targets as the correlation study. g denotes corrected three-member geometric spread and B raw BALD. P: primary text split; R: reversed split. Each single entropy score is fitted affinely; combinations use affine ridge. The primary entropy rank correlations exactly match Table 18. Character uses BLT and Pythia uses MPT-30B.

Size	Split	h_s	g	$h_s + g$	Adam
6.96M	P	0.240	0.106	0.273	0.074
	R	0.246	0.144	0.288	0.135
28.8M	P	0.262	0.154	0.319	-0.036
	R	0.255	0.207	0.325	-0.054
76.4M	P	0.274	0.225	0.348	-0.111
	R	0.227	0.235	0.317	-0.343
191M	P	0.208	0.271	0.313	-0.048
	R	0.198	0.250	0.304	-0.107
371M	P	0.166	0.290	0.317	-0.116
	R	0.162	0.226	0.267	-0.161
598M	P	0.131	0.278	0.328	-0.096
	R	0.136	0.226	0.248	-0.155

Table 20: Held-out R^2 for DataDecide C4 relative to OLMo-7B. P and R denote the primary and reversed text splits. h_s is default-seed entropy and g is corrected geometric spread; single scores use affine fits, and $h_s + g$ uses affine ridge. The Adam probe uses the same seed's hidden states. Negative predictions are retained when computing R^2.

DCLM score accompanies worse study loss; easier prediction of reference KL need not indicate a better reference. Primary NeoX and MPT targets have rank correlations of 0.916–0.940 across sizes.

F.3.1 Fitting and evaluation

Each scalar combination has an intercept and ridge regularization, with penalty strength selected from $10^{-6}, 10^{-5}, \ldots, 10^{2}$; a constant predictor is also a candidate. Four text folds within the training half select by validation mean squared error (MSE). Each character fold contains one source region; Pythia, DataDecide, and Qwen folds group complete four-window blocks. Features are centered, scalar scores are standardized, and feature vectors are rescaled to have mean squared length one. All transformations use training folds only. After normalization, the objective is summed squared error

Size	Split	h_s	g	$h_s + g$	Adam
0.5/0.6B	P	0.181	0.234	0.316	0.104
	R	0.281	0.290	0.386	0.150
1.5/1.7B	P	0.159	0.210	0.328	-0.225
	R	0.256	0.270	0.370	-0.290
3/4B	P	0.125	0.309	0.360	-0.420
	R	0.227	0.338	0.387	-0.752
7/8B	P	0.143	0.295	0.330	-0.655
	R	0.200	0.303	0.365	-0.940
14/14B	P	0.111	0.540	0.524	-1.553
	R	0.150	0.497	0.527	-1.938

Table 21: Held-out R^2 for Qwen2.5/Qwen3 base-model pairs relative to Qwen2.5-72B, using all 151,665 shared assigned tokens. P and R exchange the two text halves. h_s is Qwen2.5 entropy and g is corrected two-member geometric spread; single scores use affine fits and $h_s + g$ uses affine ridge. Adam uses Qwen2.5 hidden states. These are comparisons across generations and training recipes, not seed replicates. Negative predictions and negative R^2 are retained.

plus $n\lambda\|w\|^2$, where n is the number of training contexts, λ is the penalty strength, and w contains the non-intercept coefficients. The chosen fit is refitted to the training half. Ties within 10^{-12} prefer stronger regularization.

The *Adam probe* uses the designated member's middle or final hidden state, with the final state taken after output normalization. Adapting the regression-head protocol of Ahdritz et al. (2024), it uses an affine regression head, MSE loss, Adam with learning rate 10^{-5}, and validation-based early stopping. We use batches of 64, stop after 100 epochs without validation improvement, and impose a 100,000-epoch cap. Three training folds fit the head and the fourth selects stopping and layer. All reported runs stop before the cap. The middle state is taken after $\lceil h/2 \rceil$ of the model's h transformer blocks. Initialization uses random seed 42; the 64-bit Adam implementation was checked against PyTorch. This protocol retains its validation fold instead of refitting on the full training half. Thus the character head fits on 3,072 contexts and validates on 1,024; Pythia, DataDecide, and Qwen use 1,152 and 384, respectively. Held-out targets never select a coefficient, penalty, layer, or stopping epoch.

We report held-out $R^2 = 1 - \mathrm{MSE} \,/\, \mathrm{Var}(e_N)$, with MSE and target variance both evaluated on the test contexts, and Spearman correlation r_s for ranking. Negative predictions remain in all accuracy calculations. Negative R^2 means worse squared error than predicting the test-set mean. The primary and reversed splits exchange text halves and are sensitivity checks, not independent replications. Adjacent contexts share text, and these point estimates are conditional on the chosen members and reference.

Entropy and the hidden probe require one member's forward pass; ensemble scores require three for character, Pythia, and DataDecide, or two for Qwen. Hidden probes also estimate many more coefficients than scalar fits. For example, the largest Pythia probe fits 1,024 hidden coordinates from 1,152 training contexts with correlated neighboring positions. The results therefore describe prediction with the stated, limited supervision and do not establish a compute-matched advantage over representation methods.

G ENSEMBLING VERSUS LARGER MODELS

G.1 SELECTION AND COMPARISON CRITERION

The comparisons underlying Figure 7 include the complete seven-size standard-Pile Pythia v0/v1 rerun series at nominal 70M–6.9B and all five dense Qwen2.5/3 pairs at 0.5B–14B. All have exactly two members and a larger measured reference. Panels (a,b) show measured single-model losses from the Pythia current-release and both dense Qwen generation ladders, together with fitted single and pair curves. In the measured pair comparisons, the first six Pythia rerun pairs lose to the next larger model, while the final pair reverses that loss ordering. The 12B Pythia pair is outside the selection because no larger Pythia checkpoint was evaluated. These are release and generation comparisons, not controlled seed suites. Table 22 reports every pair in this comparison set. Table 23 adds the public 6.9B Pythia pair, together with selected finite-pool losses, best-member baselines, and total parameter budgets.

Writing q_{big} for the larger reference model, an S-member ensemble has lower loss exactly when $\rho_S^P > \bar{L}_S - L(q_{\mathrm{big}})$. This is the finite-ensemble criterion in Equation (8). Applying the same $S/(S-1)$ rescaling to both sides leaves the comparison unchanged. Comparing $\bar{L}_S - \widehat{\rho}_\infty^P$ directly with a larger model would instead describe an estimated population pool. It would incorrectly credit the measured pair with spreads that require additional members.

G.2 PREDICTIONS FROM THE FITTED SCALING LAWS

The finite-member factor $(S-1)/S$ makes the predicted geometric spread of a pair half the fitted population spread. We therefore evaluate

$$\widehat{L}(q_2^{\mathrm{G}}) = \frac{\widehat{L}(q_1) + \widehat{L}(q_2)}{2} - \frac{1}{2} c_{\mathrm{G}} (N/10^9)^{-\gamma_{\mathrm{G}}}, \qquad N = \sqrt{N_1 N_2}.$$

Here N_s is member s's total parameter count, and $\widehat{L}(q_s) = E + A_s(N_s/10^9)^{-\eta_s}$ is its fitted loss, with amplitude A_s and exponent η_s. The spread fit has amplitude c_{G} and exponent γ_{G}. The individual-model curves appear in Figure 12b, with $E = 0.33372841$ nats/byte. Each member uses its own exact total parameter count; the geometric mean is used only to evaluate spread, following its fitted size convention. The prediction budget remains $N_1 + N_2$. These empirical loss curves follow the recorded training histories; Qwen generations retain separate curves, while the common-data threshold in Equation (9) is an idealization.

For Pythia, the current-release fit supplies both member losses: $A = 0.230673$ and $\eta = 0.202624$. The independent-seed spread fit supplies $c_{\mathrm{G}} = 0.026871$ and $\gamma_{\mathrm{G}} = 0.142287$. For the 6.9B release pair, the fitted member loss is 0.489889 and the finite spread is 0.010216, giving 0.479673 nats/byte versus the measured 0.485628. This transfers the seed-spread law to a release-rerun pair and extrapolates beyond its 14M–410M fitting range.

For Qwen, we average the separate Qwen2.5 and Qwen3 loss fits, whose (A, η) values are $(0.179779, 0.255096)$ and $(0.184549, 0.277479)$. The dense generation-pair spread fit has $(c_{\mathrm{G}}, \gamma_{\mathrm{G}}) = (0.028912, 0.117842)$. At the 14B pair, the fitted mean member loss is 0.422670 and the finite spread is 0.010526, giving 0.412144 versus the measured 0.406655. Calculations use the unrounded fit coefficients. The individual-model loss fits include these model sizes, and the Qwen spread fit includes the 14B pair. Thus these comparisons describe the existing fits rather than an independent test of their prediction accuracy.

Table 22 includes all seven Pythia rerun pairs and five dense Qwen pairs. Its additional spread-fit column uses the measured mean member loss and predicts only the ensemble spread. This gives 0.486386 for the 6.9B Pythia pair and 0.407137 for the 14B Qwen pair. Their closer agreement shows that fitting the member losses contributes to the error in the full prediction. The full predictions differ from the measured pool losses by about 0.005–0.006 nats/byte in these two examples, comparable to or larger than their measured advantages over the larger references. The fitted laws therefore do not resolve such small advantages precisely.

For the smooth projections in Figure 7, we set $N_1 = N_2 = N$ and compare the fitted pair loss with a single model of size $2N$, with total parameter budget $N_{\mathrm{tot}} = 2N$. Circles in panels (a,b) show the measured single-model losses. Panel (c) shows conditional projections using fitted losses for both the single model and the pair at the same total parameter budget. Table 22 retains the measured pair comparisons. At $E = 0.33372841$ nats/byte, the Qwen pair crosses the fitted Qwen2.5

Table 22: Predicted and measured two-member geometric-ensemble losses for all pairs in Figure 7 (nats/byte). *Both fits* average the fitted member losses and subtracts half the fitted population spread. *Spread fit* instead uses the measured mean member loss. *Measured* is the actual pooled loss; *Reference* is the measured larger-model loss. Member labels are nominal sizes; all calculations use exact total parameter counts. Pythia references use the current release; Qwen references use Qwen2.5. These evaluate existing fits, including in-sample Qwen pairs, rather than a held-out prediction protocol.

Members	Larger model	Both fits	Spread fit	Measured	Reference
Pythia release reruns					
70M / 70M	160M	0.70902	0.72658	0.72666	0.66935
160M / 160M	410M	0.64975	0.64100	0.64054	0.58493
410M / 410M	1B	0.59544	0.57205	0.57297	0.55341
1B / 1B	1.4B	0.55044	0.54001	0.54084	0.53496
1.4B / 1.4B	2.8B	0.53596	0.52487	0.52500	0.51250
2.8B / 2.8B	6.9B	0.50968	0.50132	0.50154	0.49741
6.9B / 6.9B	12B	0.47967	0.48639	0.48563	0.48774
Qwen2.5 / Qwen3 dense pairs					
0.5B / 0.6B	1.5B	0.53232	0.53613	0.53567	0.49202
1.5B / 1.7B	3B	0.47992	0.47855	0.47878	0.46652
3B / 4B	7B	0.45141	0.45084	0.45120	0.44333
7B / 8B	14B	0.42743	0.42880	0.42911	0.41471
14B / 14B	32B	0.41214	0.40714	0.40665	0.41223

Table 23: Measured losses and total parameter budgets for selected Pythia and Qwen pairs. Losses are nats/byte; N_{ens} and N_{big} are billions of total parameters. Best is the lowest member loss; geometric and arithmetic columns give actual two-member pool losses. Pythia references use the current standard-Pile release; the public pair uses deduplicated Pile and v0. Q7, Q14, and Q32 denote Qwen2.5 references. Qwen pairs combine dense Qwen2.5/3 models. None of the best members beats its reference. Recipes differ and budgets are not exactly matched.

Pair → reference	N_{ens}	N_{big}	Best	Geom.	Arith.	Ref.
Pythia 2.8B reruns → 6.9B	5.550	6.857	0.51250	0.50154	0.50171	0.49741
Pythia 6.9B public pair → 12B	13.715	11.846	0.49148	0.48028	0.48012	0.48774
Pythia 6.9B reruns → 12B	13.715	11.846	0.49579	0.48563	0.48580	0.48774
Qwen 3B + 4B → Q7	7.108	7.616	0.46009	0.45120	0.45112	0.44333
Qwen 7B + 8B → Q14	15.806	14.770	0.43693	0.42911	0.42902	0.41471
Qwen 14B + 14B → Q32	29.538	32.764	0.41471	0.40665	0.40673	0.41223

baseline at 83.5B total parameters; neither Qwen3 nor Pythia has a crossover through 1T. Refitting the member-loss laws over irreducible-loss values $E \in [0.300, 0.33372841]$, with the spread law fixed, moves the Qwen2.5 crossover across 83.5–94.9B. No Qwen3 or Pythia crossover occurs through 1T over this sensitivity grid. These are conditional projections, and their small predicted advantages are not resolved by the fitting errors above.

G.3 Measured outcomes and budget limitations

The two 6.9B Pythia pairs beat the current-release 12B reference under both pooling rules, although neither best member does. Their geometric losses are 0.480281 for the public deduplicated-v0 pair and 0.485628 for the version-rerun pair. Both use 13.715B total parameters versus 11.846B for the reference. The public pair also changes the training corpus and release; these comparisons do not isolate an ensemble benefit at a matched budget.

The Qwen2.5/3 14B pair uses 29.538B parameters and attains 0.406655 nats/byte versus 0.412227 for Qwen2.5-32B (32.764B parameters). Its best member remains 0.00248 above the reference, so pooling creates the advantage; arithmetic pooling gives the same conclusion. All Qwen comparisons share a tokenizer but mix generations.

Budgets sum actual member counts. Equal total parameters roughly match leading dense inference operations, but not latency or memory traffic; MoE total parameters do not measure active inference compute. Along $D \propto N$, training cost scales approximately as N^2, so S size-N runs match the

leading training cost of one size-$\sqrt{SN}$ run. Neither these examples nor the fitted powers establish a general crossover under matched latency or training cost.

Table 24: **Retained pair spread at 200 bytes per parameter.** Entries under branch points are the retained shares R defined in Section 3.5. The independent-pair column reports $2\rho^{\mathrm{G}}_{2,\mathrm{ind}}$ in milli-bits per byte.

Width	N (M)	Independent	$b = 1$	$b = 5$	$b = 25$	$b = 125$
128	0.79	76.72	1.043	0.971	0.722	0.281
192	2.65	60.65	0.899	0.837	0.659	0.247
256	6.29	51.21	0.949	0.885	0.700	0.284
384	21.23	43.90	0.944	0.866	0.687	0.307
512	50.33	40.04	0.926	0.846	0.678	0.318
768	169.87	34.12	0.905	0.830	0.688	0.383

H Shared-training and LoRA ensemble studies

H.1 Matched data and training states

The branching study uses the original deployed byte-model recipe, separately from the retuned loss-law fits in Appendix E.3. The vocabulary has 256 byte values and context length is 512. The six widths $128, 192, 256, 384, 512, 768$ use respectively $4, 6, 8, 12, 16, 24$ transformer layers, with parameter count $N = 12(d/32)d^2$ as defined in Appendix B.7.1. We write $t = D/N$ for the evaluation budget and b for the branch point, both in bytes per parameter. Muon and AdamW learning rates, batch sizes, and the cooldown grid are held fixed between each parent, its continuations, and its independent reference. This is single-pass training over additional corpus positions, not repeated training on a fixed subset.

Each parent warms up for 18 steps, then follows a constant-learning-rate trajectory through 160 bytes per parameter. A separate decay continuation at each evaluation budget lasts one quarter of the preceding trajectory's length, giving a final budget of 200 bytes per parameter. Eleven nominal evaluation budgets are spaced by $\sqrt{2}$ from 6.25 to 200. The experimental branch points at $1, 5, 25, 125$ bytes per parameter all lie after warmup and before the final decay. They therefore vary how much of the stable training trajectory is shared; they do not separately randomize the warmup and decay schedules.

The parent saves weights and optimizer moments at each branch point. One continuation resumes each saved state with the remaining data reordered. A blockwise permutation of training-stream positions preserves the corpus positions consumed by every cooldown endpoint; a branch composes a new permutation of the remaining positions with the parent's permutation. Parent initialization seed 0 and permutation seed 1 are compared with reference initialization seed 1 and permutation seed 2; branches use permutation seed 7. No new warmup or optimizer reset is introduced. Branches at 1 and 5 have 11 evaluated budgets each, the branch at 25 has six (35.4–200), and the branch at 125 has only the final budget. At the first budget for the branch at 5, the separate updates are entirely in the decay phase.

The earlier sequential-parent pilot systematically favored the permuted continuations in loss. Its pairs mix training-order effects with branching, so they are excluded from the reported laws. The corrected experiment permutes both the parent and independent reference from the start. The additional initialization-only comparison uses the three original seeds with the same sequential stream, averaging their three pairwise spreads. It is a different reference recipe and is not the denominator of R.

H.2 Evaluation and available measurements

All models are evaluated on the same 1,024 held-out sequences of length 512. For each pair, we compute $\rho^{\mathrm{G}}_2 = \mathbb{E}_X[-\log \sum_y \sqrt{q_{1,X}(y) q_{2,X}(y)}]$ and convert nats to bits per byte. The spread plot labels $2\rho^{\mathrm{G}}_2$ as ρ^{G}_∞, using the factor $S/(S-1) = 2$ for pairs. Shared-history pairs are correlated, so this rescaling does not justify extrapolation to an independent infinite ensemble. Their actual geometric spread remains exactly ρ^{G}_2.

The branching study contains 66 size–budget evaluations and 174 parent–branch comparisons, with 29 comparisons at each of the six widths. Table 24 includes every width at the final budget, including those excluded from individual fits. The branching laws use full-parameter pairs only; the LoRA comparison is reported separately in Appendix H.5.

There is one parent, one independent reference, and one continuation per branch point at each width. Evaluations at successive budgets share training history, and the four ratios at a width share their denominator. Consequently, the fitted cells are not independent experimental replicates. Regression errors quantify fit residuals under these selections, not uncertainty over alternative parents, permutations, or training seeds.

H.3 Full-budget laws and their fitting range

Figure 8 fits $2\rho_2^{\mathrm{G}} = cN^{-\gamma}$ by ordinary least squares in log space. Width 128 is omitted from both panels: its earliest branch exceeds the independent reference, giving a negative lost share for which the log fit is undefined. This exclusion also removes a visibly different smallest-width regime; Table 24 retains its observations. Every size-law fit uses the same five widths (192–768).

The independent-pair exponent is 0.13395 ± 0.00756; the initialization-only exponent is 0.12687 ± 0.00455. Branch exponents are 0.13548 ± 0.00249, 0.14038 ± 0.00144, 0.12890 ± 0.00251, and 0.03907 ± 0.00991 in increasing order of branch depth. All uncertainties are residual regression SEs across five sizes. The shallow late-branch exponent follows directly from corrected spreads of 13.48, 12.73, and 13.06 milli-bits/byte at widths 384, 512, and 768.

The lost-share fit uses the four fractions $b/T \in \{0.005, 0.025, 0.125, 0.625\}$ and all 20 positive observations at retained widths. It gives amplitude 0.84606 and exponent 0.46741 ± 0.02102, with natural-log RMSE 0.16046. The exponent near one half describes an empirical range, not a boundary law. In particular, extrapolating the fit to $b = T$ incorrectly leaves nonzero retained spread, whereas identical final states have zero spread. Extrapolating to zero shared bytes is also untested: all measured branches share their initialization and early updates. These limitations and the late-branch size dependence constrain attempts to reconstruct independent-seed spread from a cheaper continuation.

Intermediate budgets. The final-budget law does not transfer unchanged across training. Fitting separate powers of b/T and t/T gives retained-share RMSE 0.041, versus 0.083 for a function of b/t alone, on 145 comparisons excluding width 192's early-budget excursion. Including all 174 comparisons gives 0.055 and 0.094; leaving out each retained width gives 0.043 and 0.084. These checks use a different selection and loss function from the final-budget log fit. They support predictable attenuation, but not a universal function of shared fraction alone.

H.4 Training cost and correcting attenuated spread

For a pair evaluated at T, sharing a fraction $f = b/T$ reduces the nominal training work from $2NT$ to $N(2T - b)$ bytes, saving $f/2$ of the two independent runs' work at fixed architecture. This counts the final pair's training paths, excluding the extra cooldowns used to collect intermediate measurements. Sharing one eighth saves 6.25% while losing roughly a third of the spread; sharing five eighths saves 31.25% while losing roughly two thirds.

Conversely, a measured branched-pair spread could estimate independent-pair spread by dividing by $1 - 0.85f^{0.47}$ within the fitted range. This is not a separately validated estimator, and small retained shares amplify its error. The cost calculation concerns full-parameter training paths; an adapter's fraction of trainable parameters is not itself a training-time speedup.

H.5 LoRA protocol and rank sweep

The LoRA comparisons use the same parents at all five widths $d = 192, 256, 384, 512, 768$, with analytic counts $N = 2.65, 6.29, 21.23, 50.33, 169.87$M. At $b = 125$ bytes per parameter, the parent's constant-learning-rate state is frozen in its entirety, including embeddings, output head, and normalization parameters. Rank-r updates are added to the combined query/key/value projection, attention output projection, and the three feed-forward matrices in every block. Each adapted matrix is $W + BA$, with entries of A initialized independently as $\mathcal{N}(0, 1/n_{\mathrm{in}})$ and $B = 0$; the scale α/r is one. Thus both members start with identical predictions from the shared base.

Ranks are $8, d/8, d/2$, with two members per rank: 30 completed training runs. Adapter initialization seeds are $1, 2$ and remaining-stream permutation seeds are $7, 8$, composed with the parent's permutation. The first member uses the same remaining-data order as the full-parameter branch at 125. Both members consume the same remaining token set as that branch, continuing the stable phase to 160 bytes per parameter and then decaying to $T = 200$.

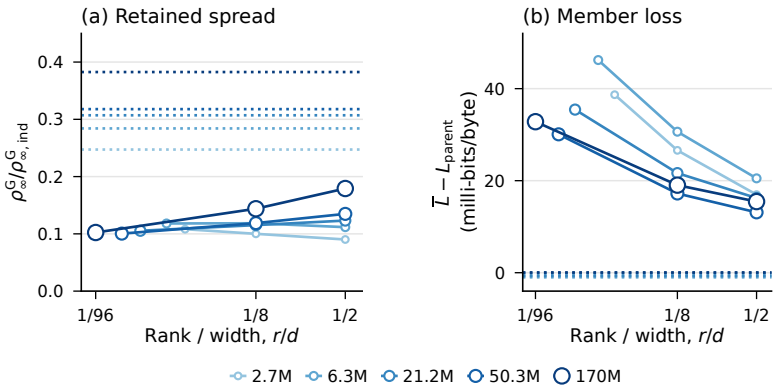

Figure 22: **LoRA further reduces spread.** At $b = 125$, $T = 200$: (a) retained spread; (b) mean member loss above the parent. Dotted lines show full-parameter branches.

Adapters use a fresh AdamW optimizer with learning rate 0.003, $(\beta_1, \beta_2) = (0.9, 0.95)$, $\epsilon = 10^{-10}$, zero weight decay, and a 100-step linear re-warm. The remaining stable/decay timing matches the full branch. In contrast, full-parameter branches retain the parent's Muon/AdamW states without re-warming. This comparison therefore changes the trainable subspace, optimizer, and restart treatment together; it does not isolate rank as a causal explanation for the spread difference.

A five-run learning-rate bracket at width 192 selected 0.003. For rank 24, rates $0.001, 0.003, 0.01$ gave losses $1.5702, 1.5674, 1.5707$ bits/byte; for rank 8, rates $0.003, 0.01$ gave $1.5780, 1.5785$. These tuning losses use 65,536 held-out bytes, while all pair comparisons use the common 524,288-byte evaluation set. The bracket was not repeated at larger widths or the highest rank.

If h is the feed-forward width and $\ell = d/32$ the number of blocks, the adapters contain $\ell r(9d + 3h)$ trainable parameters, versus $\ell(4d^2 + 3dh)$ base block-matrix parameters. The implemented h values are $512, 688, 1024, 1368, 2048$. The three rank choices therefore train about 1.5–6%, 18%, and 71% of the block-matrix count. These fractions exclude the frozen embeddings, head, and norms. Even the largest adapters retain only 36–47% of full-branch spread; increasing rank does not restore the full-parameter spread.

H.6 LoRA pair spread and prediction loss

Across all fifteen pairs, LoRA retains about 9–18% of independent-pair spread, compared with 25–38% for full-parameter branches at the same widths and branch point. This is a further reduction by a factor of 2.1–3.7. Rank effects depend on width: spread decreases with rank at $d = 192$, changes little at 256, and increases at $384, 512, 768$. At $d = 768$, the retained shares are $10.2\%, 14.4\%, 17.9\%$ for ranks $8, 96, 384$, versus 38.3% for the full-parameter branch. Figure 8b shows the fixed-relative-rank series $r = d/8$: its line joins measurements and its shaded band spans ranks $8, d/8, d/2$ at each width. Cooled adapters are merged into ordinary model weights before evaluation. Figure 22 compares only the two LoRA members at the same rank, the parent–full-branch pair, and the independent reference; mixed adapter–full-model pairs are not used. Let L_p be the full parent's loss, L_B the full continuation's loss, and $\bar{L}_{\mathrm{L}}$ the mean loss of the two LoRA members. Their geometric pool $q^{\mathrm{G}}_{2,\mathrm{L}}$ satisfies

$$L(q^{\mathrm{G}}_{2,\mathrm{L}}) = \bar{L}_{\mathrm{L}} - \rho^{\mathrm{G}}_{2,\mathrm{L}} = \bar{L}_{\mathrm{L}} - \tfrac{1}{2}\rho^{\mathrm{G}}_{\infty,\mathrm{L}}. \tag{21}$$

As above, $\rho^{\mathrm{G}}_{\infty}$ denotes the displayed rescaling, not an independently validated infinite-ensemble limit. Table 25 gives all fifteen comparisons. The independent reference spreads are $60.65, 51.21, 43.90, 40.04, 34.12$ milli-bits/byte; full-branch spreads are $14.99, 14.55, 13.48, 12.73, 13.06$ in increasing width order.

The mean member penalty relative to the parent falls with rank at every width, from 30–46 milli-bits/byte at rank 8 to 13–21 at rank $d/2$. The LoRA pool recovers only 1.75–3.28 milli-bits/byte and remains 10.48–43.76 above one full continuation. The full continuation itself differs from the parent by only 0.06–1.07 milli-bits/byte. Thus reduced spread here accompanies a higher prediction loss, rather than demonstrating reduced total epistemic loss.

Each width uses one frozen parent and one pair per rank, reusing adapter and order seeds across ranks. These fifteen pairs are not repeated independent estimates of a population mean. We therefore report the measured range without confidence intervals or a fitted LoRA size law. The evidence covers only one late branch point, five widths up to 170M parameters, and this optimizer recipe; it does not show

Table 25: **LoRA rank sweep after shared training.** All pairs branch at $b = 125$ and are evaluated at $T = 200$. Spreads and loss differences are in milli-bits/byte. $R_{\mathrm{L}} = \rho^{\mathrm{G}}_{2,\mathrm{L}}/\rho^{\mathrm{G}}_{2,\mathrm{ind}}$ and $Q_{\mathrm{L}} = \rho^{\mathrm{G}}_{2,\mathrm{L}}/\rho^{\mathrm{G}}_{2,\mathrm{branch}}$ compare LoRA spread with independent and full-parameter pairs. The last two columns distinguish the mean member's penalty relative to the parent from the geometric pool's penalty relative to one full continuation.

Width	Rank	$\rho^{\mathrm{G}}_{\infty,\mathrm{L}}$	R_{L}	Q_{L}	$\bar{L}_{\mathrm{L}} - L_p$	$L(q^{\mathrm{G}}_{2,\mathrm{L}}) - L_B$
192	8	6.56	0.108	0.438	38.69	36.47
192	24	6.07	0.100	0.405	26.59	24.62
192	96	5.47	0.090	0.365	16.96	15.30
256	8	6.05	0.118	0.416	46.22	43.76
256	32	6.07	0.119	0.417	30.64	28.18
256	128	5.72	0.112	0.393	20.51	18.22
384	8	4.63	0.105	0.343	35.45	33.92
384	48	5.06	0.115	0.375	21.64	19.91
384	192	5.41	0.123	0.401	16.22	14.31
512	8	4.02	0.100	0.316	30.07	28.13
512	64	4.75	0.119	0.374	17.22	14.90
512	256	5.40	0.135	0.424	13.12	10.48
768	8	3.49	0.102	0.268	32.82	31.02
768	96	4.91	0.144	0.376	19.03	16.52
768	384	6.11	0.179	0.468	15.46	12.34

that every LoRA ensemble, or every approximation built from a late trajectory, has the same retained fraction.

I Related work

Uncertainty as prediction error. The excess-risk view predates this work. Lahlou et al. (2023) include model misspecification in epistemic uncertainty, while proper-score decompositions separate calibration from information missing in a forecast. These connections are introduced in Section 2. Feature Gaps applies a conditional-entropy/forward-KL decomposition to contextual question answering using an ideally prompted model as reference (Bakman et al., 2026). Our reference models likewise provide proxies, rather than true conditional distributions. Decision-theoretic treatments also distinguish predictive beliefs, predictive performance, and data dispersion (Bickford Smith et al., 2025).

Pooling and scaling. Geometric spread is an established ensemble-diversity term: the identities in Section 2.2 build on loss-diversity and proper-score bias–variance decompositions (Wood et al., 2023; Gruber & Buettner, 2023). Autoregressive ensemble uncertainty is studied by Malinin & Gales (2021). Rosso et al. (2025) report shallow size dependence and weak data dependence in language-model fine-tuning. We examine these questions in pretraining and compare spread with the loss it recovers. The distinction matters because absolute disagreement and its share of reducible loss can move in opposite directions.

Ensemble budgets and shared training. The main text compares our results with ensemble scaling in image classification and language-model pretraining. An advantage from ensembling is not universal: optimally regularized random-feature regression favors one model at a fixed total feature count (Ruben et al., 2025). Snapshot ensembles collect predictors along one training trajectory (Huang et al., 2017); our branching study instead varies the shared prefix and compares continuations with independent pretraining. Bayesian adapters also define different populations: BLoB learns a variational low-rank distribution (Wang et al., 2024), while the methods discussed in Section 3.5 use local, probabilistic, or input-dependent adapter distributions. Their uncertainty estimates are conditional on a shared backbone; our experiment does not evaluate those posteriors.

Calibration and prediction targets. Entropy calibration over generated histories (Braverman et al., 2020; Cao et al., 2025) differs from recalibration on observed text. Neither top-label ECE nor a small achieved correction establishes full forecast calibration. Likewise, predicting a stronger model's confidence or entropy, as in the probe study used in the main text, differs from predicting reference-to-student KL. Our linear head adapts that regression protocol to the latter target; its limited supervision does not establish a general advantage over representation-based uncertainty methods. Semantic entropy groups sampled answers by meaning (Kuhn et al., 2023; Farquhar et al., 2024); it addresses answer reliability under generation randomness, rather than spread across training seeds.