Problems with Chinchilla Approach 2: Systematic Biases in IsoFLOP Parabola Fits

Eric Czech
Open Athena AI Foundation

Abstract

Chinchilla Approach 2 is arguably the most widely adopted method for fitting neural scaling laws in practice. The method fits parabolas to IsoFLOP loss curves and extracts scaling exponents through a sequence of simple polynomial and linear regressions. We show that this parabolic approximation introduces systematic biases in compute-optimal allocation estimates, even on noise-free synthetic data under ideal experimental conditions. Two independent bias sources are identified: surface asymmetry ($\alpha \neq \beta$), which shifts intercept estimates, and off-center sampling, which distorts intercepts or exponents depending on whether the offset is constant or varies with compute budget. These biases compound in practice and grow with sampling grid width. We show that exploiting the partially linear structure of the Chinchilla loss surface, by separating linear coefficients from nonlinear exponents, eliminates these biases. Our realization of this approach, Variable Projection with Non-negative Least Squares (VPNLS), recovers all five surface parameters with machine precision across all conditions tested, while offering comparable data efficiency to Approach 2, high stability with full parametric inference, and no dependence on specialized IsoFLOP experiment designs.

1 Introduction

The Chinchilla paper [Hoffmann et al., 2022] introduced three approaches to scaling law estimation; of these, Approach 2 appears to have seen by far the broadest adoption. It has been used by leading AI labs including DeepMind [Hoffmann et al., 2022, Alabdulmohsin et al., 2023] (its creators), Meta [Grattafiori et al., 2024, Tay et al., 2025], DeepSeek [Bi et al., 2024], Microsoft [Jiang et al., 2025], Amazon [Haldar and Pinto, 2023], Waymo [Chen et al., 2025], and Arc Institute [Nguyen et al., 2024], among others. It is also a workhorse method for academic scaling law studies [Li et al., 2024, Nie et al., 2025, Wagh et al., 2024] and high-profile practitioner tutorials[1] from researchers like Andrej Karpathy.

The method's appeal lies in its stability and data efficiency relative to nonlinear optimization over all loss surface parameters. Rather than fitting all five parameters of the loss surface simultaneously, Approach 2 targets only the two scaling exponents, relying on second-order Taylor approximations that reduce each IsoFLOP curve to a simple parabola. This sacrifices recovery of the full loss surface but makes estimation far more stable and data-efficient, letting practitioners extract the most actionable quantities for compute allocation planning through a sequence of straightforward polynomial and linear fits, without ever touching a nonlinear optimizer.

Despite this broad adoption, the sensitivity of the method's core approximations and its behavior on loss surfaces that are less symmetric than the original Chinchilla form (where parameter and token scaling exponents are roughly equal) have not, to our knowledge, been studied in detail.

[1] https://github.com/karpathy/nanochat/discussions/420

Here we revisit the basics of how to apply a simple model like Chinchilla with high precision and stability, to validation loss alone, before considering more advanced extensions. We investigate through noise-free synthetic simulations that isolate systematic biases inherent to the method itself by eliminating all sources of statistical noise.

We show how these biases affect downstream decisions like dataset size selection for final training runs at large compute budgets. We show how extrapolation errors trace back to suboptimal IsoFLOP experiment design, and that pathologies in these designs can be observed in real, high-profile scaling law studies even if they are difficult to quantify precisely. Finally, we propose an alternative fitting method that is simple, stable, and free of these biases while building on the same intuitive computational shortcut: optimizing exponential terms separately from linear terms. We call this approach Variable Projection with Non-negative Least Squares (VPNLS).

This investigation is also motivated by a broader landscape of *analytical* extensions to the Chinchilla loss surface. A growing body of work adds or modifies terms in the original functional form to account for additional training configuration choices such as data repetition [Muennighoff et al., 2023, Goyal et al., 2024], overfitting [Yang et al., 2024], precision [Kumar et al., 2024], MoE sparsity [others, 2025c], data quality [others, 2025e], data mixtures [others, 2025d,f, Goyal et al., 2024], non-embedding parameters [Sardana and Frankle, 2024], and downstream task performance [others, 2024], to name a few. These extensions prescribe explicit functional forms rather than inferring scaling law structure automatically, and they build directly on the Chinchilla model as a foundation. A fitting method that recovers the base surface with higher precision may therefore offer a stronger starting point for these richer settings as well.

2 Preliminaries

Neural scaling laws describe how model performance improves with compute. The Chinchilla loss surface models this relationship as:

$$L(N, D) = E + \frac{A}{N^\alpha} + \frac{B}{D^\beta} \tag{1}$$

where N is the number of parameters, D is the number of training tokens, E is the irreducible loss, and A, B, α, β capture how quickly performance improves with scale.

Given a compute budget $C \approx 6ND$, the optimal allocation satisfies:

$$N^* \propto C^a \quad \text{where} \quad a = \frac{\beta}{\alpha + \beta} \tag{2}$$

$$D^* \propto C^b \quad \text{where} \quad b = \frac{\alpha}{\alpha + \beta} \tag{3}$$

Recovering the exponents a and b from empirical training runs is crucial for planning efficient large-scale training. Two canonical approaches exist:

2.1 Approach 2: IsoFLOP Parabolic Fitting

This method is presented in the Chinchilla paper. The key insight is that along a fixed-compute contour (IsoFLOP curve), loss as a function of $\log N$ is approximately parabolic near the optimum. The procedure has three steps:

1. **Sample IsoFLOP contours:** For each compute budget C, train models at various (N, D) pairs satisfying $C = 6ND$.

2. **Fit parabolas:** For each budget, fit $L = p(\log N)^2 + q(\log N) + r$ and extract the minimum N^*.
3. **Fit power laws:** Regress $\log N^*$ against $\log C$ to recover the exponent a (and similarly for D^*, b).

The appeal is simplicity: only polynomial fits, no nonlinear optimization. The parabolic approximation comes from a Taylor expansion of the loss surface around the optimum.

2.2 Approach 3: Direct Surface Fitting

The alternative is to fit all five parameters (E, A, B, α, β) simultaneously via nonlinear least squares. This avoids the parabolic approximation entirely but is notoriously unstable: highly sensitive to initialization and prone to converging to spurious local minima.

3 The Happy Path: Symmetric Surfaces

Before examining failure modes, we establish that Approach 2 works perfectly under ideal conditions. Consider a **symmetric** loss surface where $\alpha = \beta$:

$$L(N, D) = 1.69 + \frac{400}{N^{0.31}} + \frac{400}{D^{0.31}} \tag{4}$$

With equal exponents, the optimal allocation splits compute evenly between parameters and data. The true scaling exponents are:

$$a = b = \frac{0.31}{0.31 + 0.31} = 0.5 \tag{5}$$

We sample five IsoFLOP contours spanning 10^{17} to 10^{21} FLOPs, with 15 model sizes per curve, fit parabolas to each, and extract the optimal token count D^*. All simulations throughout this paper use these same five compute budgets and 15 points per IsoFLOP curve.

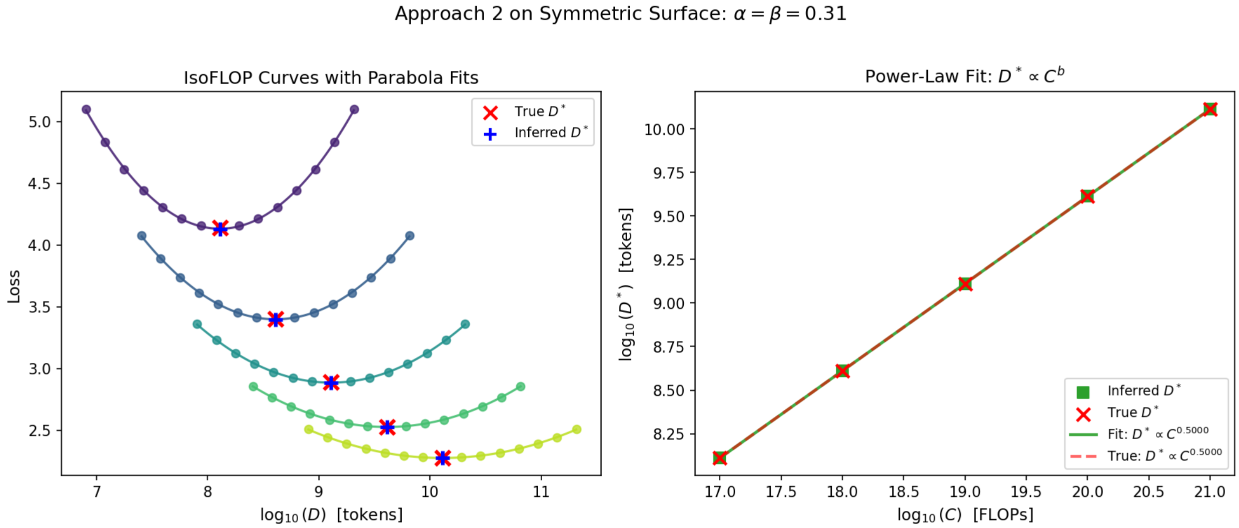

Figure 1: Approach 2 applied to a symmetric loss surface. Left: IsoFLOP curves with fitted parabolas. True (×) and inferred (+) optima are indistinguishable. Right: Power-law fit recovers the exact scaling exponent.

The results confirm perfect recovery of the token scaling exponent and intercept:

Parameter	True Value	Inferred Value	Relative Error
b (D^* exponent)	0.500000	0.500000	$+6.2 \times 10^{-12}\%$
b_0 (D^* intercept)	-0.389076	-0.389076	$-1.4 \times 10^{-10}\%$

Table 1: Approach 2 parameter recovery on the symmetric surface.

Key Result. On a symmetric loss surface with perfectly crafted IsoFLOP grid sampling, Approach 2 recovers both exponents and intercepts with machine-precision accuracy. When $\alpha = \beta$, the parabola vertex shift is zero, so the inferred optima coincide with the true optima.

This establishes our baseline. Approach 2 is precisely correct under ideal conditions that are unrealistic in practice. The problems arise when we deviate from these ideal conditions, as we show in the following sections where these conditions are perturbed in controlled ways.

4 Asymmetric Surfaces: Intercept and Extrapolation Errors

We repeat the exact same procedure as before: perfect sampling centers, no noise, identical methodology. The only change is that the loss surface is now **asymmetric** ($\alpha \neq \beta$).

4.1 What Happens

Simulation results show that when the loss surface is asymmetric, Approach 2 produces systematically wrong intercepts while exponents remain accurate. This is not statistical noise; it is a deterministic bias from fitting parabolas to a non-parabolic surface.

We test two configurations to see how the effect scales:

- **Chinchilla:** $\alpha = 0.34$, $\beta = 0.28$ (ratio ≈ 1.2)
- **Asymmetric:** $\alpha = 0.46$, $\beta = 0.15$ (ratio $= 3.0$)

The Asymmetric surface is not a contrived stress test. An exponent ratio of 3.0 is comparable to what has been observed in practice. DeepSeek [Bi et al., 2024] reports compute-optimal allocation exponents of $a = 0.73$, $b = 0.27$ for an OpenWebText2 variant, implying a loss surface exponent ratio of $\beta/\alpha \approx 2.7$. The asymmetry runs in the opposite direction from our Asymmetric surface ($\beta > \alpha$ rather than $\alpha > \beta$), but the degree of imbalance is similar, and it is the magnitude of the imbalance, not its direction, that drives the biases studied here.

Parameter	True Value	Inferred Value	Relative Error
b (D^* exponent)	0.548387	0.548387	$\approx 0\%$
b_0 (D^* intercept)	-0.555357	-0.578092	-4.1%

Table 2: Approach 2 parameter recovery on the Chinchilla surface.

Chinchilla Surface.

Asymmetric Surface.

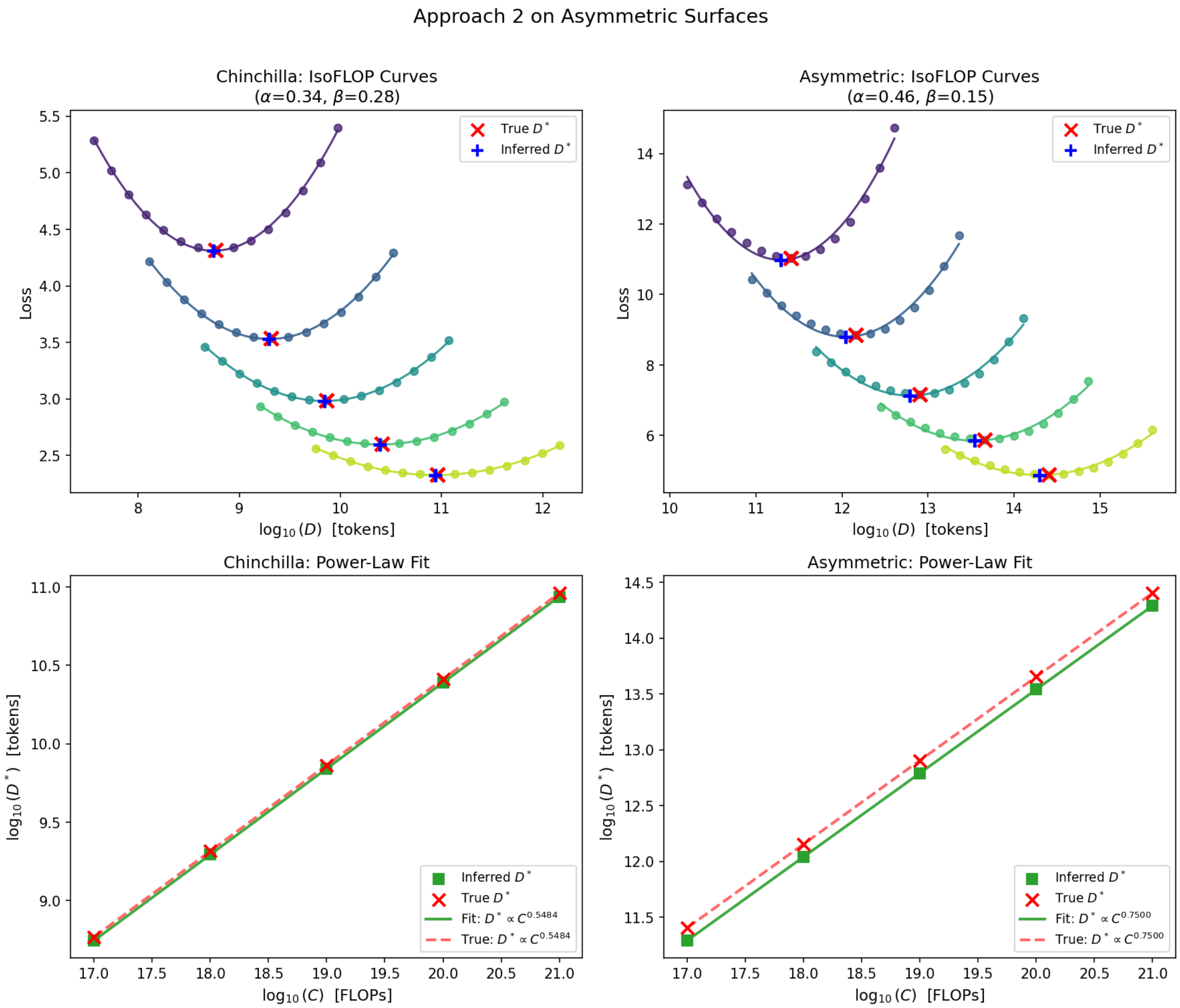

Figure 2: Approach 2 on asymmetric loss surfaces. Note the visible gap between true (dashed) and inferred (solid) power-law lines in the Asymmetric case. The exponents match perfectly, but the intercepts differ.

4.2 Why This Is Surprising

A few percent error in the intercept might seem minor, but consider that this simulation gave Approach 2 every advantage. The data is perfect: no measurement noise, with every point lying exactly on the true loss surface. The sampling is perfect too, with IsoFLOP grids centered precisely at the true optimum (something practitioners would not know how to do in practice). And the parameters are standard, taken directly from the Chinchilla paper rather than contrived to expose a potentially unrealistic weakness.

Key Result. Even under these ideal conditions, Approach 2 produces biased intercepts for asymmetric surfaces. The error is systematic, a property of the parabolic approximation, not statistical noise.

Parameter	True Value	Inferred Value	Relative Error
b (D^* exponent)	0.750000	0.750000	$\approx 0\%$
b_0 (D^* intercept)	-1.345791	-1.459957	-8.5%

Table 3: Approach 2 parameter recovery on the Asymmetric surface.

4.3 Why It Happens

The IsoFLOP loss curve is not a true parabola; it contains exponential terms. When a parabola is fit to this curve, the parabola's minimum (vertex) does not land exactly at the true optimum. It shifts slightly, and the key insight is that this shift depends only on the loss surface shape (α, β) and the sampling grid. It does not depend on compute budget. The sampling grid size becomes important here: wider grids amplify the mismatch between the true curve and its parabolic approximation, increasing the vertex shift.

Because the IsoFLOP parabola is fit in $\log N$ space (as described in the Approach 2 procedure), the vertex shift directly biases N^*. Since $C = 6ND$, analyzing the bias in either N^* or D^* is sufficient; we focus on N^* here since that is where the parabolic fit typically operates.

Since the vertex shift is constant across all compute budgets, it biases every inferred N^* by the same multiplicative factor. When fitting $\log N^*$ vs $\log C$ to extract scaling exponents:

- The **slope (exponent)** is unchanged: multiplying all N^* values by a constant factor adds a constant to $\log N^*$, which does not affect the slope.
- The **intercept** absorbs the entire error, biased by exactly that multiplicative factor.

The intercept error can be derived analytically in closed form. The parabola vertex shifts by δw (in log-space), giving an intercept error of:

$$\text{Intercept error} = 10^{\delta w} - 1 \tag{6}$$

where $\delta w = f(\alpha, \beta, W, n)$ depends only on the surface exponents and the sampling grid (width W in log-space, number of points n per IsoFLOP curve), not on C, E, A, or B. Here W spans $10^{-W/2}$ to $10^{W/2}$ times the optimal N^*, so $W = 2.41$ (the XL grid) means sampling from $\frac{1}{16}\times$ to $16\times$ the optimum. Key properties:

- $\delta w = 0$ when $\alpha = \beta$ (symmetric surfaces have no error)
- δw grows with $|\alpha - \beta|$ (more asymmetry, more error)
- δw grows with W (wider sampling range, more error)

For example, with the Chinchilla parameters ($\alpha = 0.34$, $\beta = 0.28$): the XS grid ($W = 0.60$) yields 0.3% intercept error, while the XL grid ($W = 2.41$) yields 4.1% error.

The full derivation[2] provides the closed-form expression for vertex shift δw as a function of α, β, W, and n. It also shows how this shift translates directly into intercept error, independent of compute budget.

Intuition via Taylor expansion. A parabola is a 2nd-order polynomial, equivalent to a 2nd-order Taylor expansion around the optimum. The approximation $L(w) \approx L(0) + \frac{1}{2}L''(0)w^2$ is only

[2] https://github.com/Open-Athena/scaling-law-analysis/blob/main/results/article/static/scaling_parameter_errors.pdf

valid when higher-order terms are negligible, i.e., when samples are close to the true minimum. As sampling range increases, 3rd and 4th order terms grow. For symmetric surfaces ($\alpha = \beta$), odd-order terms cancel by symmetry, preserving the vertex location. For asymmetric surfaces, they do not cancel, shifting the fitted vertex away from the true optimum.

4.4 Why It Matters

Extrapolation to higher compute budgets requires both exponents and intercepts to be correct. The previous subsection (Section 4.3) established that asymmetric loss surfaces produce provably biased intercepts even under ideal experimental conditions. Here we quantify what those errors mean in practical terms by examining compute-optimal token prediction: given a compute budget, how many tokens does the inferred scaling law predict?

Up to this point, all analysis has assumed a single fixed sampling grid width. We now examine how token prediction error varies with both compute budget and sampling grid width. For surfaces with asymmetric exponents, wider sampling grids amplify the parabola-fitting mismatch, increasing the constant vertex shift and thus the intercept bias. To make this comparison concrete, we first define what "wider" and "narrower" mean in quantitative terms.

A sampling grid of "$\pm k\times$" means the sampled values (whether model sizes or token counts) range from $\frac{1}{k}$ to k times the true optimum at each compute budget. The total range covered is k^2 (the ratio of largest to smallest), and the $\log_{10}$ of that ratio gives the number of decades the grid spans end-to-end. Table 4 shows the four grid widths used in this analysis.

Grid Name	$\pm k\times$	Sampling Range	Total Ratio	Decade Span
Extra Small (XS)	±2×	$\frac{1}{2}\times$ to $2\times$	4×	0.60
Small (S)	±4×	$\frac{1}{4}\times$ to $4\times$	16×	1.20
Large (L)	±8×	$\frac{1}{8}\times$ to $8\times$	64×	1.81
Extra Large (XL)	±16×	$\frac{1}{16}\times$ to $16\times$	256×	2.41

Table 4: Sampling grid widths used throughout this paper. Real experiments typically span 1–2 decades, placing the Small and Large grids within the realistic range. The Extra Large grid (±16×, ∼2.4 decades) is the default used in all single-grid analyses in the preceding sections.

The key observations from Figure 3 are:

- **Symmetric surfaces are unaffected:** When $\alpha = \beta$, all grid widths produce zero error.
- **Asymmetric surfaces underestimate:** Negative errors mean the inferred D^* is smaller than the true D^*. Following these predictions would undertrain the model.
- **Wider grids amplify error:** Moving from XS (±2×) to XL (±16×) grids increases error from 0.3% to 5.1% on Chinchilla, and from 1.7% to 23% on the Asymmetric surface.
- **Asymmetry magnifies everything:** The Asymmetric surface ($\alpha/\beta = 3$) shows roughly 4–5× larger errors than Chinchilla at each grid width.

Key Result. Consider the Chinchilla surface with the Large grid (±8×), a practical sampling range for real experiments. When extrapolating to 10^{24} FLOPs, the true optimal token count is 4.04 trillion, but Approach 2 predicts only 3.92 trillion: a 2.9% underestimate, or roughly 117 billion fewer tokens than optimal. While 2.9% may seem modest, recall that this simulation uses

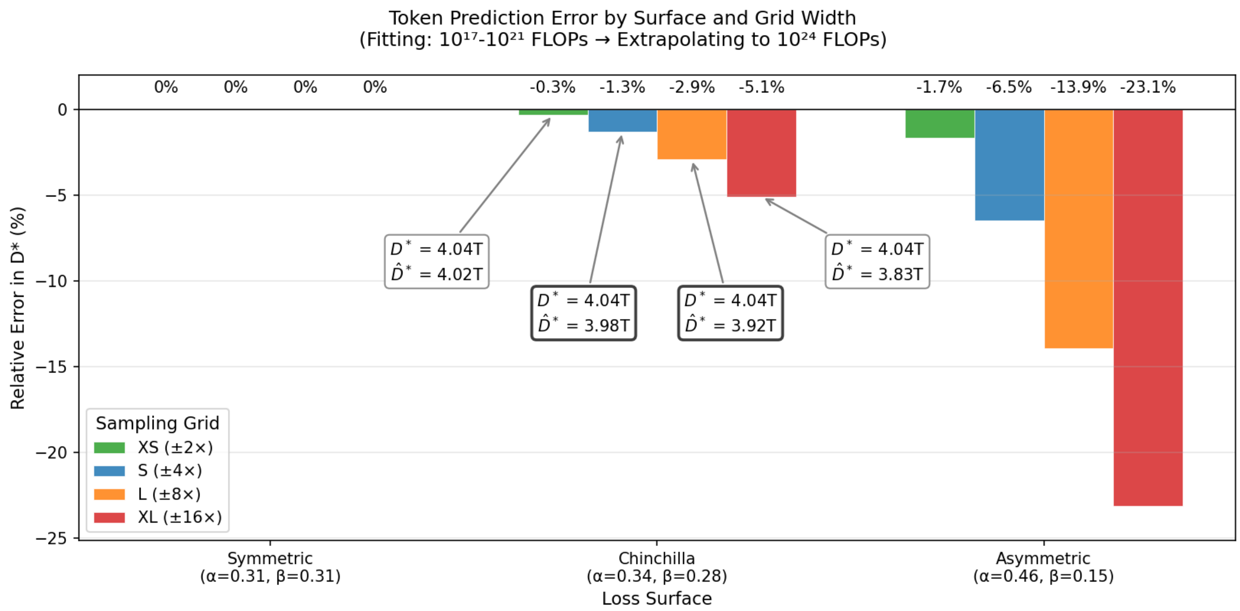

Figure 3: Relative error in compute-optimal token prediction when extrapolating from the training range (10^{17}–10^{21} FLOPs) to 10^{24} FLOPs. Negative values indicate underestimation: the inferred scaling law predicts fewer tokens than optimal. Bars are grouped by sampling grid width. Annotations for the Chinchilla surface show D^* (true compute-optimal token count) versus $\hat{D}^*$ (the Approach 2 estimate); the Small and Large grid annotations are emphasized as they fall within the realistic 1–2 decade range typical of scaling law experiments.

unrealistically ideal conditions: perfectly centered sampling grids at every compute budget and zero measurement noise. Real experiments, where the true optimum is unknown, data is noisy, and the scaling exponent imbalance may be larger than Chinchilla's modest $\alpha/\beta \approx 1.2$, can only do worse.

The full raw data underlying Figure 3 is provided in Appendix A.

5 Off-Center Sampling: Exponent and Extrapolation Errors

The previous sections assumed perfectly centered sampling. At every compute budget, the IsoFLOP grid was placed exactly at the true optimum. In practice, N^* is not known before running the experiment. Sampling centers are guesses, informed by prior estimates or heuristics, and they will likely be wrong by some amount.

This is a distinct source of error from the asymmetry bias examined earlier. Asymmetry errors arise from the shape of the loss surface ($\alpha \neq \beta$); off-center errors arise from where the sampling grid is placed. To isolate this new effect, we return to the symmetric surface ($\alpha = \beta = 0.31$) where asymmetry bias is zero by construction.

5.1 Constant Multiplicative Bias

The simplest form of off-center sampling is a constant multiplicative offset: every compute budget's sampling center is shifted by the same factor from the true optimum. A "3× offset" means each IsoFLOP grid is centered at $3 \times D^*$ instead of D^*, so the grid midpoint consistently sits at three times the true optimal token count.

Because this offset is the same at every compute budget, it has a familiar geometric effect where each parabola vertex shifts by a constant amount in log-space. This is the same mechanism as asymmetry bias. The slope of $\log D^*$ vs $\log C$ is unaffected (a constant additive shift in log-space does not change the slope), so the scaling exponent is preserved perfectly. The intercept, however, absorbs the entire error.

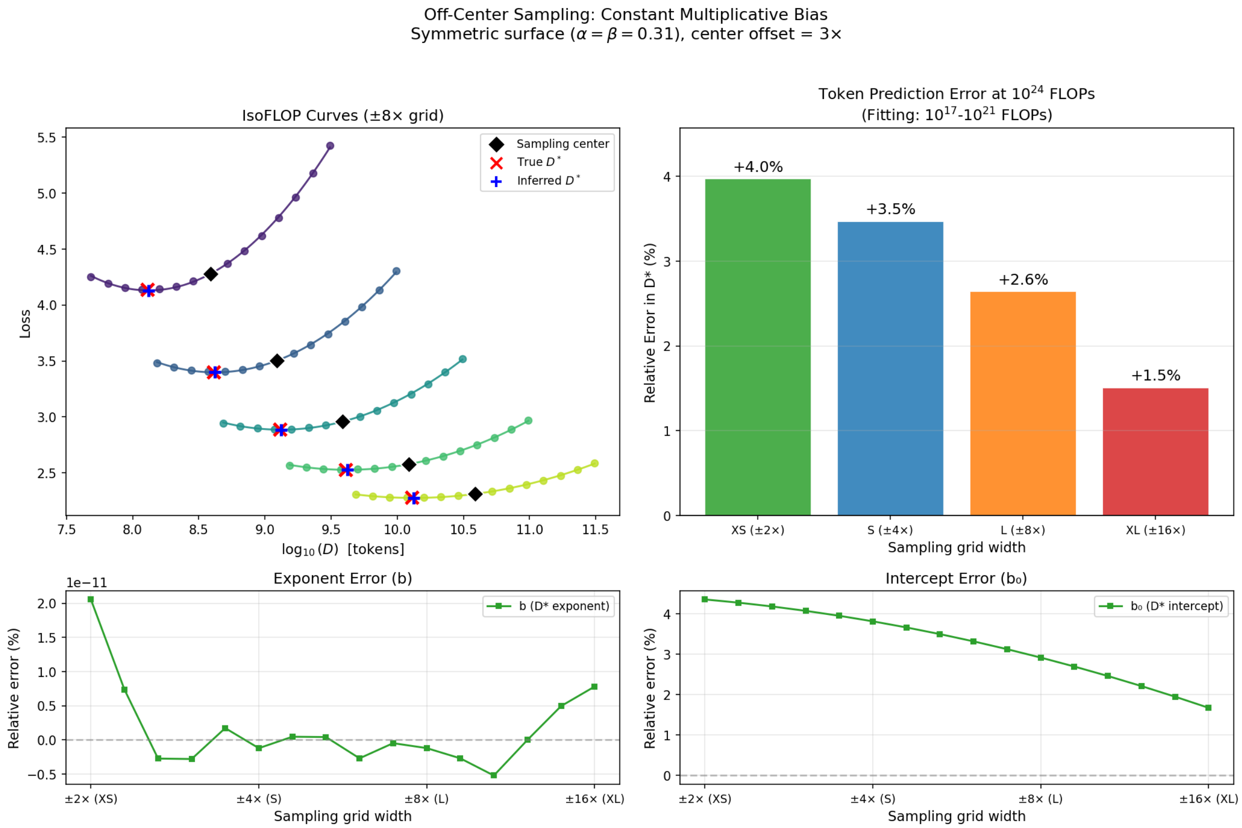

Figure 4: Effect of a constant 3× offset in sampling centers on the symmetric surface. Top left: IsoFLOP curves at the Large grid (±8×), with black diamonds marking the (off-center) sampling center, red × the true D^*, and blue + the inferred D^*. Top right: extrapolation error in compute-optimal token prediction at 10^{24} FLOPs for each grid width. Bottom row: exponent and intercept errors across grid widths from XS (±2×) to XL (±16×), plotted on the same y-axis scale. The exponent is recovered perfectly (flat at zero) while the intercept shows systematic bias that varies with grid width.

The extrapolation bar chart (top right of Figure 4) shows what this means for token prediction. All four grid widths overestimate D^*, with the narrowest grid (XS) producing the largest error. This is the reverse of the asymmetry bias pattern, where wider grids amplified error. Here, narrower grids are more sensitive to off-center placement because fewer samples lie near the true optimum.

The intercept error panel (bottom right) confirms the pattern across the full continuum of grid widths. The error is always positive (the inferred D^* overshoots) and decreases monotonically as the grid widens, reflecting how a wider sampling range brings more of the true loss curve's shape into the fit, partially compensating for the misplaced center.

Key Result. Consider the symmetric surface with the Large grid (±8×) and a 3× offset, where every IsoFLOP grid is centered at three times the true optimal token count. When extrapolating to 10^{24} FLOPs, the true optimal token count is 408.2 billion, but Approach 2 predicts 419.0 billion: a

2.6% overestimate, roughly 10.8 billion more tokens than optimal. Compare this with the Chinchilla asymmetry result at the same grid width: a 2.9% underestimate. The magnitudes are comparable, but the sources are entirely different. Asymmetry bias comes from the shape of the loss surface; off-center bias comes from where the grid is placed. In a real experiment, both act simultaneously.

5.2 Drifting Bias

When the offset varies with compute budget, a qualitatively different failure mode emerges. To illustrate this, we apply a linear drift. The sampling center starts at the true optimum for the lowest budget and drifts to 3× the true optimum at the highest budget, interpolating linearly in log-compute space.

Because the offset now differs across compute budgets, it no longer cancels in the slope of $\log D^*$ vs $\log C$. Both the exponent and the intercept are affected.

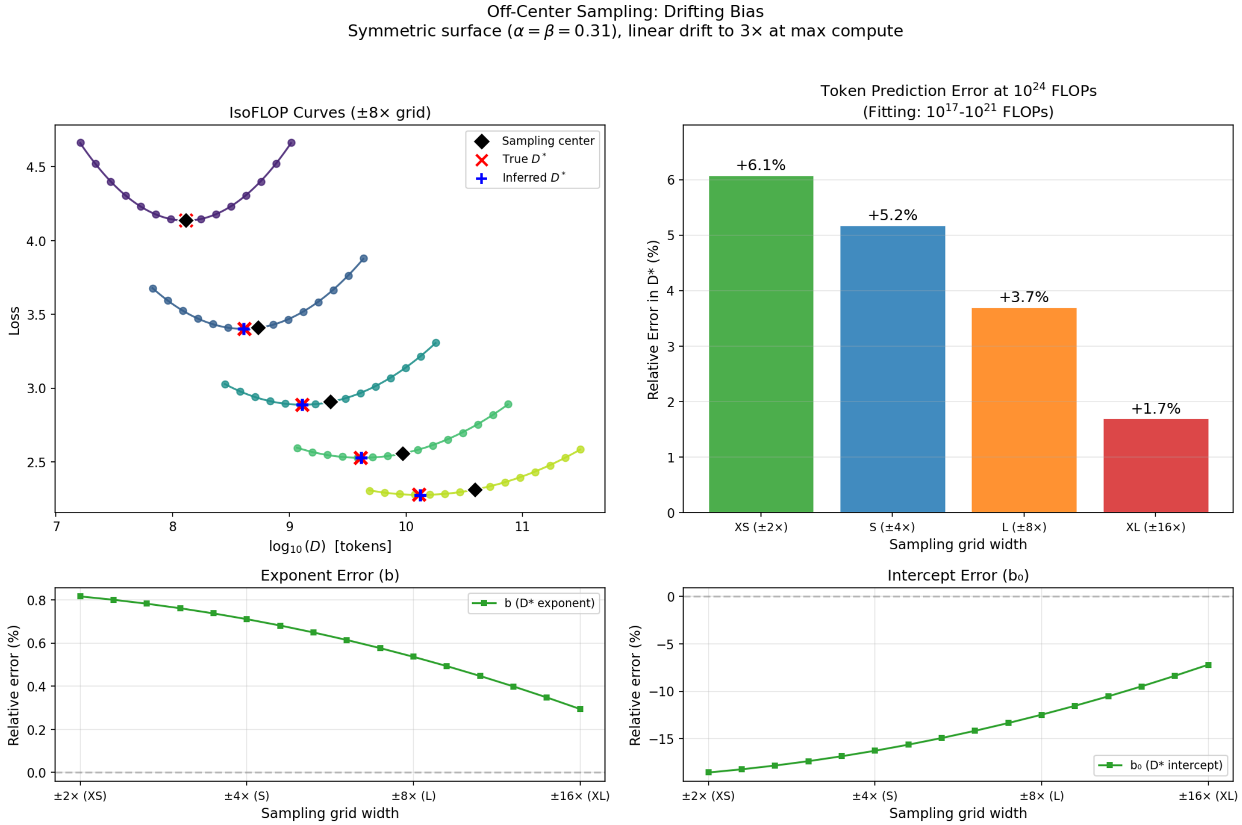

Figure 5: Effect of a linear drift in sampling centers (centered at true optimum for lowest budget, drifting to 3× at highest budget) on the symmetric surface. Unlike the constant bias case, the exponent error (bottom left) is now non-zero: the slope of $\log D^*$ vs $\log C$ is distorted because the offset varies across compute budgets.

Compare the bottom-left panels of Figures 4 and 5: constant bias produces a flat line at zero (exponent preserved), while drifting bias produces a non-zero exponent error that varies with grid width.

Key Message. Constant bias preserves exponents; any compute-dependent bias pattern distorts them. The distinction matters because exponent errors compound during extrapolation, while intercept errors remain fixed.

6 IsoFLOP Curves in the Wild

The previous sections used synthetic, noise-free simulations to isolate Approach 2's biases under controlled conditions. A natural question is whether the conditions that trigger these biases, asymmetric loss surfaces and imperfectly centered sampling, actually arise in practice. To get a sense of this, we can look at IsoFLOP curves published in three of the most prominent scaling law studies [Hoffmann et al., 2022, Grattafiori et al., 2024, Bi et al., 2024].

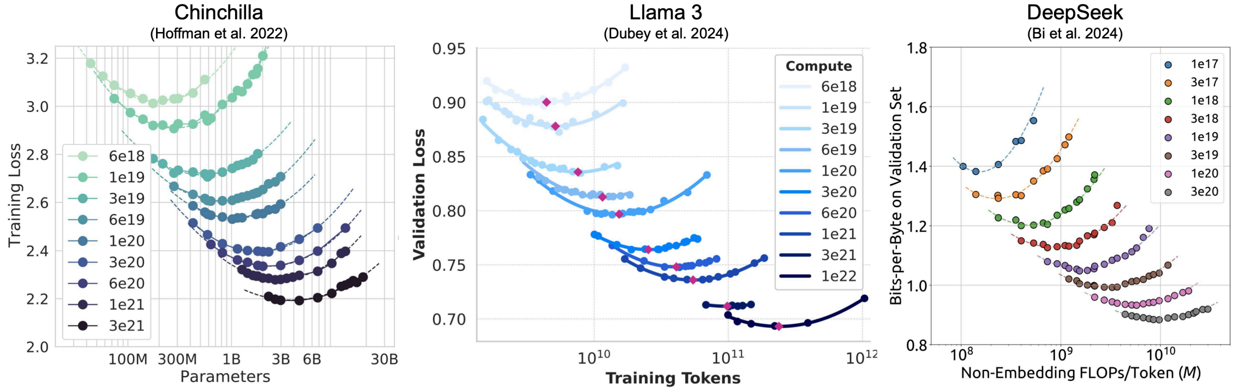

Figure 6: IsoFLOP curves from three published scaling law studies. Left: Chinchilla (training loss vs parameters). Center: Llama 3 (validation loss vs training tokens). Right: DeepSeek (bits-per-byte vs FLOPs/token). Each panel shows curves at multiple compute budgets, fit using Approach 2.

Published scaling law studies frequently report asymmetric loss surfaces. Some degree of asymmetry is nearly universal across modalities, and several reported exponent estimates reach the same levels of imbalance as the most extreme configuration in our simulations. The related biases documented in this paper therefore apply broadly.

Beyond surface asymmetry, the IsoFLOP curves in Figure 6 also show visible signs of off-center sampling and drift:

- **Off-center sampling:** At some compute budgets, the sampling grid does not appear centered at the curve minimum, placing more points on one side of the optimum than the other.
- **Drifting centers:** The degree of off-centering appears to vary across compute budgets rather than remaining constant, which is the drifting-bias pattern that distorts both exponents and intercepts.

To be clear, this is not a criticism of these studies. These are among the most careful and influential scaling law analyses published. The point is a more general one: the conditions under which Approach 2's biases activate, asymmetric surfaces and imperfect sampling centers, appear to be the norm rather than the exception. The idealized conditions of the Happy Path (Section 3; symmetric surface, perfectly centered grids) are the special case.

6.1 Compounding Errors

Given evidence that both surface asymmetry and off-center sampling are present in real studies, we can simulate what happens when these biases act simultaneously. Using the same three loss surfaces from earlier sections, we combine them with the 3× drift and 3× constant offset from the off-center analysis. We fit Approach 2 on compute budgets from 10^{17} to 10^{21} FLOPs and extrapolate D^* predictions to 10^{24} FLOPs across all four grid widths.

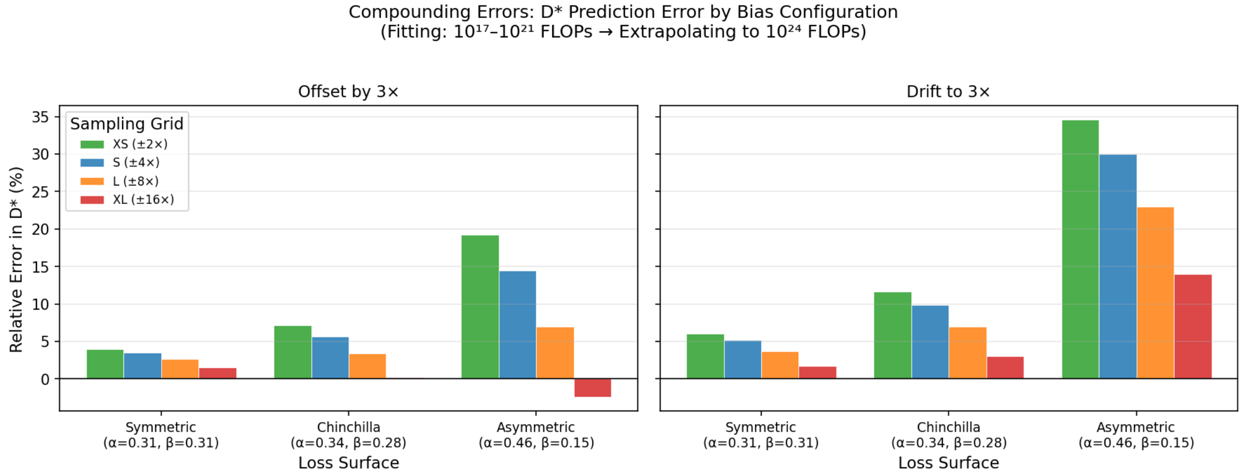

Figure 7: Relative error in D^* at 10^{24} FLOPs with off-center sampling on all three loss surfaces. Left: constant 3× center offset at every budget. Right: linear drift to 3× at the highest compute budget. Bars are grouped by sampling grid width (XS through XL). Negative values indicate underestimation; positive values indicate overestimation. On the symmetric surface, the constant offset results correspond to Figure 4 and the drift results correspond to Figure 5; the asymmetric surfaces reveal how these sampling biases interact with the inherent asymmetry bias.

Comparing with the baseline in Figure 3, where asymmetry bias alone produces errors up to −5% on Chinchilla and −23% on the Asymmetric surface, the two bias sources interact in opposite directions. Off-center sampling pushes errors positive (overestimating D^*), while asymmetry bias pushes errors negative (underestimating). The net error depends on which source dominates. With narrow grids, asymmetry bias is negligible and the sampling bias determines the error: drift to 3× produces +6% on the symmetric surface and +12% on Chinchilla. With wider grids, asymmetry bias grows and begins to offset the sampling bias. On Chinchilla with a constant 3× offset, this cancellation is nearly perfect with the XL grid (+0.24%), but this is only coincidental.

On the Asymmetric surface, the drift configuration produces the largest errors in the figure: +35% with the XS grid and still +14% with XL. Even the constant offset configuration reaches +19% with XS before the asymmetry bias partially offsets it with wider grids.

These 3× perturbations are representative of realistic conditions. The IsoFLOP curves they produce on the symmetric surface (top-left panels of Figures 4 and 5) show sampling centers that are visibly displaced from the curve minima, with the displacement either uniform across budgets (constant offset) or growing toward higher budgets (drift). Both patterns are qualitatively similar to what is observed in the published studies shown in Figure 6, where sampling grids are not perfectly centered and the degree of off-centering varies across compute budgets. A 3× factor means the sampling center sits at three times the true optimal token count, which is likely within the range of uncertainty practitioners face when choosing sampling centers before the optimum is known.

Figure 11 in the appendix provides a more detailed view: it shows how D^* extrapolation errors evolve across compute budgets from 10^{22} to 10^{25} FLOPs, revealing which bias sources produce errors that grow with extrapolation distance (drift) versus those that remain roughly constant (surface asymmetry and constant offsets), and how these patterns vary across multiple drift rates and center offset magnitudes.

Key Result. Multiple bias sources act simultaneously in any real experiment. Surface asymmetry and off-center sampling each produce meaningful errors on their own. When they happen to act in the same direction, the combined error exceeds either one alone: on the Asymmetric surface with drift to 3×, errors reach 35% even when using the narrowest grid, where the parabolic approximation is most accurate. When they oppose, partial cancellation can occur, but this depends on the specific combination of surface geometry, offset magnitude, and grid width, making it unreliable in practice.

The full raw data underlying Figure 7 is provided in Appendix B.

7 Robust Fits: Unbiased Estimation with Linear Separation

The previous sections showed that Approach 2's parabolic approximation introduces systematic biases in intercepts (from asymmetry) and potentially exponents (from off-center sampling), and that the conditions driving these biases are visible in published scaling law studies. The natural alternative is Approach 3, which fits all five surface parameters (E, A, B, α, β) simultaneously via nonlinear least squares. This avoids the parabolic approximation entirely but brings its own set of problems.

7.1 Problems with Direct Surface Fitting

A recent survey of over 50 scaling law papers [Moeini et al., 2025] documents the landscape of fitting practices and their failure modes. The problems described below apply to scaling law fitting in general, not just Chinchilla forms, but they are directly relevant because Approach 3 involves the same kind of nonlinear optimization. Over half of the papers surveyed do not fully specify their fitting procedure (optimizer, loss function, or initialization), which compounds reproducibility challenges.

The most common optimizers for scaling law fits are BFGS and L-BFGS. Some studies use SGD-family optimizers like Adam and Adagrad, though these are noted as sometimes poorly suited for curve fitting due to limited data efficiency. At least one study [Goyal et al., 2024] forgoes optimization entirely in favor of pure grid search because fitted solutions are too unstable.

In practice, this instability takes several forms. Results are sensitive to initialization: different starting points for the optimizer can lead to substantially different fitted parameters. Results are also sensitive to optimizer hyperparameters such as convergence tolerance and gradient estimation method. And the optimizer frequently converges to local minima rather than the global optimum.

Initialization is the most studied source of variability. Common mitigations include grid search over thousands of starting points (running the optimizer from each and keeping the best fit), random sampling of starting points, evaluating a coarse grid without optimization and seeding the optimizer from the single best candidate, or initializing from previously published parameter values. Yet the survey's own experiments show that full-grid optimization over 4500 starting points can yield results that diverge significantly from reported figures, evidence of "the difficulty of optimizing over this space, and the presence of many local minima."

A simpler alternative is to log-linearize the power law and fit with linear regression. However, the log transformation changes the error distribution and exaggerates errors at small loss values, biasing parameter estimates. This bias is easily observed in simulations like ours. The survey also finds that the choice of loss function (whether Log-Huber, Huber, MSE, or MAE) affects fitted parameters unpredictably across datasets, and non-MSE objectives can introduce systematic bias in parameter estimates. Our goal is to identify a fitting method that is simple, stable, and efficient rather than to address outliers or other statistical concerns, so we use MSE for all fits.

The survey's experimental analysis varies optimizer, loss function, and initialization strategy across three datasets. The overarching finding is that none of these choices reliably eliminates instability, and results shift unpredictably between datasets. A key contributor is the high dimensionality of the joint five-parameter optimization, which creates a complex loss landscape with many local minima and interacting sensitivities. Reducing the dimensionality of the nonlinear search is one way to make the problem more tractable.

As an example of what "complex loss landscape" means concretely, consider the Hessian of the residual sum of squares (RSS) objective for a five-parameter fit on noise-free data from the Asymmetric surface ($\alpha = 0.465$, $\beta = 0.155$), using five IsoFLOP contours from 10^{17} to 10^{21} FLOPs with 15 points per curve. Its eigenvalues reveal how sensitive the objective is to perturbations along each parameter direction [Ghorbani et al., 2019], and the condition number κ (the ratio of the largest to the smallest eigenvalue) measures how difficult the landscape is for gradient-based methods to navigate. For this surface, the five eigenvalues span from approximately 8×10^{-6} to 3×10^{6}, giving $\kappa \approx 3.5 \times 10^{11}$. The two flattest directions (smallest eigenvalues) point almost entirely along the linear coefficients A and B. Near the optimum, perturbing either coefficient barely changes the RSS, making them effectively underdetermined by the data even when the data are perfect. The steepest directions are dominated by the scaling exponents α and β.

Quasi-Newton methods like L-BFGS, which are among the most common optimizers for scaling law fits, build an approximate inverse Hessian to scale gradient steps across parameter directions. When eigenvalues span 12 orders of magnitude, the gradient signal along the flat A/B directions is negligible compared to the steep α/β directions, and convergence criteria are often satisfied by progress in the steep directions before the flat directions are resolved. Separating the linear parameters from the nonlinear search eliminates the ill-conditioned directions entirely. The resulting two-dimensional landscape over (α, β) has a Hessian condition number of $\kappa \approx 11$ in our example, a reduction by a factor of roughly 3×10^{10}. This motivates an algorithm that exploits the partially linear structure of the Chinchilla loss surface to search only the well-conditioned two-dimensional subspace.

7.2 Variable Projection (VPNLS)

The Chinchilla loss surface has a partially linear structure that can be exploited. For any fixed values of α and β, the remaining parameters (E, A, B) enter the model linearly and can be solved exactly via least squares. This is the same computational shortcut that motivates Approach 2 (optimizing exponential terms separately from linear terms), but applied here without the parabolic approximation.

The algorithm searches over (α, β) and, at each candidate pair, solves for (E, A, B) via non-negative least squares (NNLS). A coarse 32×32 grid search identifies a good starting region, and a Nelder-Mead simplex optimizer refines it. The linear separation is maintained throughout. The optimizer only ever navigates the two-dimensional (α, β) surface, never the full five-parameter space. We term this method Variable Projection with Non-negative Least Squares (VPNLS).

The choice of Nelder-Mead over L-BFGS-B is deliberate. VPNLS uses NNLS for the inner solve to guarantee that E, A, and B remain non-negative, preventing physically meaningless fits. However, NNLS has no closed-form gradient with respect to the outer parameters (α, β). Switching to ordinary least squares would restore differentiability but cannot enforce non-negativity. With NNLS, L-BFGS-B must rely on finite-difference gradients, which creates a set of interacting tuning parameters (`eps`, `jac`, `ftol`, `gtol`, `maxcor`, `maxls`) where tight tolerances demand gradient accuracy that finite differences cannot reliably provide.

Nelder-Mead avoids this entirely. Its few settings (`xatol`, `fatol`) are independent and work

```
function VPNLS(data):
  function objective($\alpha$, $\beta$):
    $\mathbf{X} \leftarrow [\mathbf{1},\ N^{-\alpha},\ D^{-\beta}]$          // design matrix
    $(E, A, B) \leftarrow$ NNLS($\mathbf{X}, L$)                  // linear solve, $E, A, B \geq 0$
    return $\|L - \mathbf{X} \cdot [E, A, B]\|^2$
  $(\alpha_0, \beta_0) \leftarrow \arg\min$ objective($\alpha$, $\beta$)          // coarse $32 \times 32$ grid
  $(\alpha^*, \beta^*) \leftarrow$ NelderMead(objective, start=$(\alpha_0, \beta_0)$)   // refine in 2D
  $(E^*, A^*, B^*) \leftarrow$ NNLS($\mathbf{X}(\alpha^*, \beta^*), L$)        // recover linear params
  return $(E^*, A^*, B^*, \alpha^*, \beta^*)$
```

well out of the box. Nelder-Mead scales poorly to high dimensions, but variable projection reduces the search to just two dimensions, which is exactly the regime where simplex methods excel.

7.3 Method Comparison (Parameter Recovery)

To validate this choice, we compare nine method configurations on noise-free synthetic data across three loss surfaces (symmetric, Chinchilla, and high imbalance) and 20 sampling ranges. This is the best case for gradient-based methods since the data contains no noise that could obscure gradient information.

The configurations fall into two groups. The first uses 5D direct optimization (Approach 3), fitting all five parameters jointly with L-BFGS-B using either analytical gradients, forward finite differences, or central finite differences. The second uses 2D variable projection over (α, β) only, comparing VPNLS (Nelder-Mead), L-BFGS-B with four finite-difference configurations (default ε, central differences, $\varepsilon = 10^{-6}$, and $\varepsilon = 10^{-10}$), and a fine 256^2 grid search with no local refinement. Both groups use the same total initialization budget: the 5D methods search a $4^5 = 1{,}024$-point grid over all five parameters, while the 2D methods search a $32^2 = 1{,}024$-point grid over (α, β) only, so that accuracy differences are more likely to reflect the optimizer and loss-landscape geometry than an initialization advantage.

In the left panel, each dot shows the typical (geometric mean) parameter recovery error for one method, and the horizontal bar shows the range from best to worst case across 60 scenarios. The right panel breaks this down by parameter, showing the worst-case error for each.

Consider the best Approach 3 configuration (5D L-BFGS-B with analytical gradients). Even with exact gradients on noise-free data, the worst-case errors reach about 1.4% for B and about 0.75% for A, compared with VPNLS errors on the order of 10^{-8}%—a gap of roughly seven orders of magnitude. Figure 10 in the appendix breaks this down by surface and sampling range, also revealing that Approach 3's errors can vary systematically with sampling range on certain surfaces.

Looking at the full set of methods, a clear hierarchy emerges. High-resolution grid search (256^2) is stable across all conditions but provides the poorest overall precision among 2D methods, limited by grid resolution.

5D direct optimization (Approach 3) is more accurate on average than grid search but highly variable across conditions. The 5D configurations that rely on finite-difference gradients rather than analytical gradients perform particularly poorly and serve as a useful negative control. They demonstrate what high variability and instability look like, and Approach 3 with analytical gradients exhibits a similar pattern centered around more accurate parameter estimates. The full per-parameter breakdown (Figure 10) shows these instability patterns in detail.

L-BFGS-B with 2D variable projection can match VPNLS precision, but the optimizer fails to converge in a non-trivial fraction of scenarios even in this relatively small test suite. The choice of

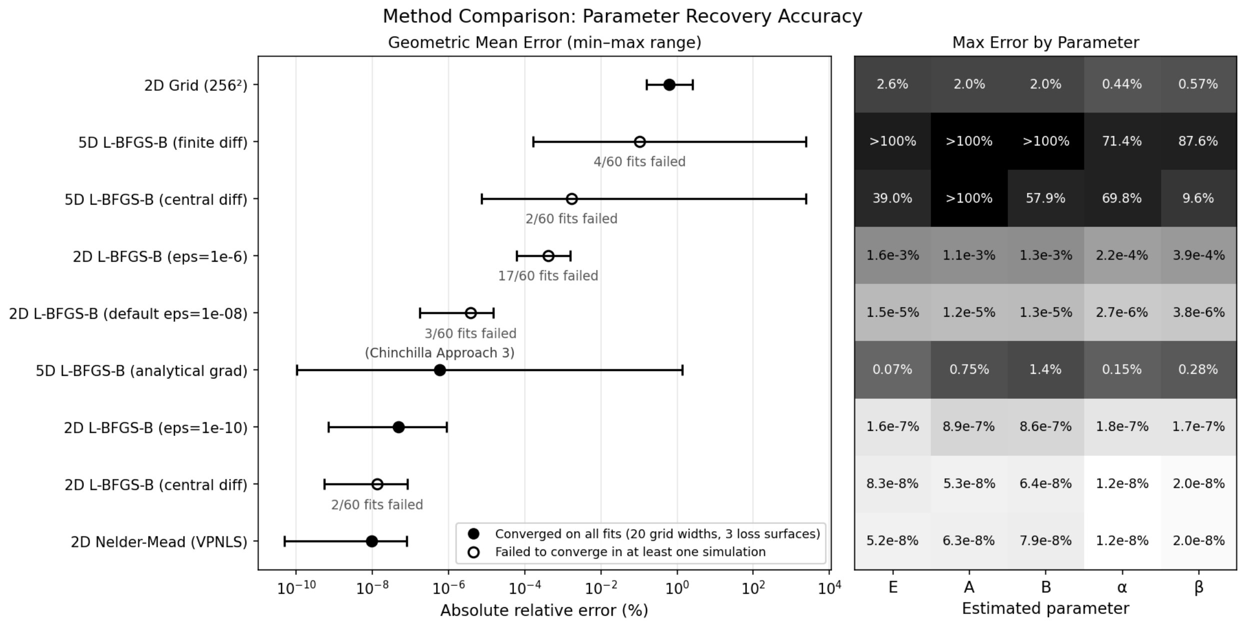

Figure 8: Comparison of nine fitting methods on noise-free synthetic data across three loss surfaces and 20 sampling ranges (60 fits total per method). Left: geometric mean of |relative error| (%) pooled across all surfaces, grid widths, and parameters, with horizontal bars spanning the min-to-max range. Filled dots indicate convergence on all 60 fits; open dots indicate at least one failure (count annotated). Right: maximum |relative error| (%) per parameter over successful fits, on a log-scale colormap. Methods are sorted by geometric mean error, with the worst at top.

finite-difference scheme matters considerably. By default, scipy's L-BFGS-B approximates gradients with forward differences: each partial derivative is estimated as $(f(x+h) - f(x))\,/\,h$. Passing `jac='3-point'` to `scipy.optimize.minimize` switches to 3-point central differences, where each partial is estimated as $(f(x+h) - f(x-h))\,/\,2h$. The central formula is generally more accurate for smooth objectives because it samples symmetrically around the point of interest. In our tests, this closes the precision gap with Nelder-Mead (from roughly 10^{-5}% to 10^{-8}% error), but introduces sporadic line search failures. Notably, these failures can be false positives. The optimizer has already reached the true minimum, with residual sum of squares near machine zero, but the line search cannot verify further progress because function values are too small to distinguish. In scipy, this surfaces as `result.success = False` with an `ABNORMAL` status from `scipy.optimize.minimize`, even though the returned parameters are correct.

L-BFGS-B remains a viable alternative to Nelder-Mead for practitioners willing to tune settings carefully and who understand that certain convergence errors from libraries like scipy may not necessarily be problematic. That said, VPNLS with Nelder-Mead is simpler, requires less tuning, and recovers parameter estimates with precision at least as high as any other method tested. It technically achieves the most precise estimates, though the margin over a well-configured L-BFGS-B with 3-point central differences is small.

The full method comparison data is provided in Appendix C.

Key Result. On noise-free synthetic data, VPNLS eliminates the biases inherent in the parabolic approximation and avoids the fragile gradient tuning that complicates L-BFGS-B when used with variable projection. All five loss surface parameters (E, A, B, α, β) are recovered with machine precision, and extrapolation to higher compute budgets is exact.

7.4 Method Comparison (Exponent Inference)

The parameter recovery results above are noise-free, which is obviously not representative of practice. We now extend the compounding errors scenario (Section 6.1; Asymmetric surface with a 3× drift at the ±8× grid) to a statistical setting. Gaussian noise is added to loss values at three levels ($\sigma = 0.05, 0.1, 0.2$), the number of compute budgets varies from 2 to 4, and the number of points per IsoFLOP curve ranges from 4 to 32. Each configuration is repeated with 256 independent noise realizations, yielding 9,216 fits per method and 36,864 fits in total. Figure 12 in the appendix shows what the noisy IsoFLOP samples look like at each noise level.

Because a scaling law study is typically run once rather than repeated hundreds of times, sporadic optimizer failures that produce large errors in a minority of fits are arguably the most consequential practical risk. The comparison below emphasizes maximum errors alongside typical accuracy for this reason.

The parameter recovery comparison (Section 7.3) evaluated all five surface parameters, but Approach 2 does not estimate them individually. Here we focus on the scaling exponents $a = \beta/(\alpha+\beta)$ and $b = \alpha/(\alpha+\beta)$, which determine compute-optimal allocation and are the quantities all four methods can be compared on directly. Figure 9 pools these exponent errors across all experimental conditions.

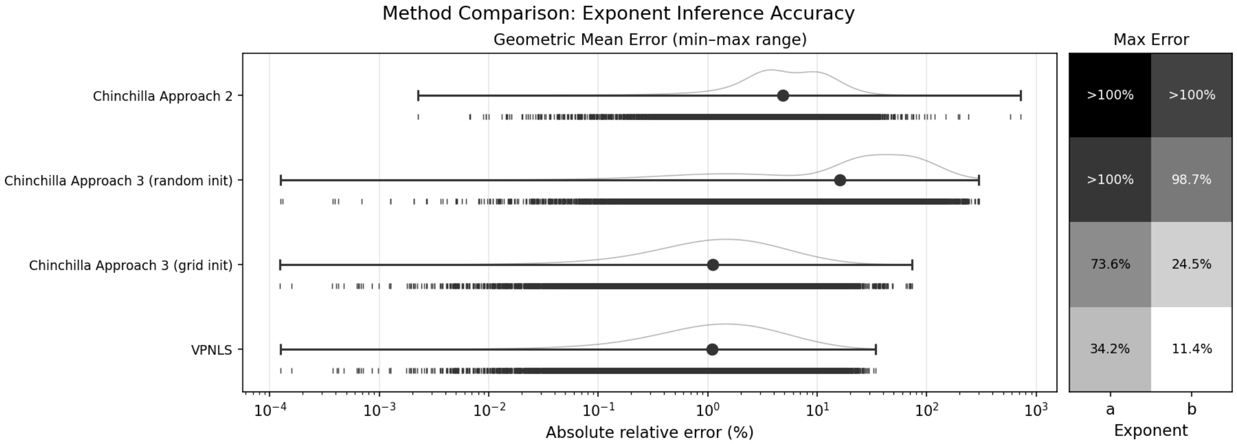

Figure 9: Comparison of four fitting methods on noisy synthetic data, pooled across three noise levels, three budget counts, four points-per-curve settings, and 256 noise realizations (9,216 fits per method). Left: geometric mean of |relative error| (%) with min-to-max range bars. Right: maximum |relative error| (%) per exponent. Methods sorted by worst-case error, worst at top. Surface: Asymmetric ($\alpha = 0.465$, $\beta = 0.155$) with 3× drift, ±8× grid, compute budgets 10^{17}–10^{21} FLOPs.

Method	GMean err%	Max a err%	Max b err%
VPNLS	1.09	34.2	11.4
Approach 3 (grid init)	1.11	73.6	24.5
Approach 2	4.84	715.5	238.5
Approach 3 (random init)	16.2	296.0	98.7

Table 5: Geometric mean and maximum |relative error| (%) on scaling exponents a and b, pooled across 3 noise levels × 3 budget counts × 4 points-per-curve settings × 256 seeds = 9,216 fits per method.

Randomly initialized Approach 3 performs the worst overall, with a geometric mean error of

16.2% and a maximum error on the a exponent exceeding 290%. This confirms the sensitivity to initialization documented in Section 7.1 and serves as a reference for what unconstrained 5D optimization looks like without careful seeding.

Approach 2 produces tighter distributions than randomly initialized Approach 3, but its geometric mean error of 4.8% is more than four times higher than VPNLS. Unlike the optimizer-related failures that affect Approach 3, this error reflects the structural bias from the parabolic approximation (Section 4). Even with 32 points per curve and low noise, the systematic inaccuracy persists. The maximum error on the a exponent exceeds 700%, driven by configurations with few points per curve where the parabolic fit to each IsoFLOP curve is less constrained (Figure 13).

Grid-initialized Approach 3 is substantially more accurate, with a geometric mean error of 1.1%. As in the parameter recovery comparison (Section 7.3), both methods use equal-sized initialization grids ($4^5 = 1{,}024$ for Approach 3 and $32^2 = 1{,}024$ for VPNLS), so any accuracy difference reflects the optimizer rather than an initialization advantage. Despite similar typical accuracy, Approach 3's maximum error on the a exponent reaches 73.6%, compared to 34.2% for VPNLS.

The gap between grid-initialized Approach 3 and VPNLS is most visible in the tails of the error distribution. Both methods achieve nearly identical typical accuracy (geometric mean of 1.1% vs 1.1%), but Approach 3's eight largest errors range from 66% to 74%, whereas VPNLS's eight largest all fall below 35%. The detailed breakdown in Figure 13 shows that these sporadic Approach 3 failures appear across different noise levels, budget counts, and dataset sizes without a clear pattern that would help anticipate them.

Key Result. On asymmetric loss surfaces with realistic noise, Approach 2's structural bias persists. Its geometric mean exponent error (4.8%) is more than four times that of VPNLS (1.1%), and its worst-case error on the a exponent exceeds 700%. Approach 3 without careful initialization fares even worse, with a geometric mean of 16.2% and worst-case errors approaching 300%. With grid initialization, Approach 3 matches VPNLS in typical accuracy but produces sporadic large errors (max 73.6% vs 34.2% on a) that appear unpredictably across IsoFLOP experiment design conditions. VPNLS offers the most reliable accuracy overall.

8 Conclusion

The Approach 2 biases documented in this paper are structural, not statistical. They exist on noise-free data with perfect experimental conditions and, as the noisy method comparison (Section 7.4) confirms, persist under realistic noise levels with varying amounts of data.

Two independent sources of error compound in practice. Surface asymmetry ($\alpha \neq \beta$) biases intercepts, and off-center sampling biases intercepts or exponents depending on whether the offset is constant or varies with compute budget. Both act simultaneously in any real experiment, and published scaling law studies frequently show clear signs of off-center sampling and report high levels of asymmetry, at least among those that publish IsoFLOP curves. At practical grid widths with Chinchilla-like asymmetry, token count errors of 5% or more are typical; on more asymmetric surfaces, the errors reach 20% or more.

A practical alternative exists. VPNLS (Variable Projection with Non-negative Least Squares) recovers all five surface parameters with machine precision on noise-free data. Under noise, VPNLS achieves typical accuracy on scaling exponents comparable to well-initialized Approach 3 while producing tighter worst-case errors. It uses the same intuitive linear separation that makes Approach 2 appealing and is straightforward to implement.

Because VPNLS recovers the full loss surface rather than just scaling exponents, it may also provide a more precise foundation for the analytical extensions to the Chinchilla model discussed in the introduction (Section 1). These extensions build on the same functional form and in most cases retain the partially linear structure that variable projection exploits, making them a natural direction for future work.

Practitioners using Approach 2 should be aware that intercept estimates carry a systematic bias that grows with exponent asymmetry and sampling grid width. This bias persists under noise: even with generous data and low noise levels, Approach 2's exponent errors remain several times larger than those of methods that fit the full surface. When precision matters for extrapolation to large compute budgets, VPNLS offers one robust alternative, though the underlying principle is more general. Any method that exploits the linear separability of the Chinchilla loss surface can avoid the parabolic approximation while retaining much of Approach 2's simplicity.

8.1 Limitations

Several limitations scope the conclusions of this study.

- **Irreducible loss dominance at large scale.** At sufficiently large compute budgets, scaling properties are dominated entirely by the irreducible loss E. When token counts and model sizes at fixed compute budgets are large enough, the Chinchilla surface reaches E asymptotically and all training configurations become equally effective, meaning that extrapolations are irrelevant and compute-optimal training is no longer informed by scaling laws. We assume this study is only relevant to practitioners working in a regime where downstream model quality can still effectively be informed by scaling law extrapolations per the Chinchilla model.

- **No quantification of downstream cost.** We do not connect token extrapolation error to under- or over-training, model performance, or the ultimate cost of Approach 2's errors in FLOPs or dollars. We avoid this because it is difficult to do well, alternatives to Approach 2 can be justified by theory and simulation alone, and those alternatives are easy to implement at effectively no extra computational cost.

- **Assumed correctness of the Chinchilla loss surface.** We assume the Chinchilla loss surface model $L(N, D) = E + A/N^\alpha + B/D^\beta$ is correct in practice. While there is substantial evidence legitimizing this model [others, 2025a], alternatives exist, including the Kaplan loss model [Kaplan et al., 2020], refined analytical surfaces like Farseer [others, 2025b] and MuPT [Yang et al., 2024], and agent-discovered functional forms [others, 2025g].

- **Qualitative characterization of published study errors.** Likely errors in published studies are characterized qualitatively rather than quantified. We believe the qualitative characterization is compelling enough on its own to justify that real IsoFLOP sampling pathologies occur in practice, but they are difficult to quantify precisely because they do not follow the convenient theoretical model we use for those pathologies in our simulations.

References

Ibrahim M. Alabdulmohsin, Behnam Neyshabur, and Xiaohua Zhai. Getting ViT in shape: Scaling laws for compute-optimal model design. In *NeurIPS*, 2023.

Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, et al. DeepSeek LLM: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*, 2024.

Yilun Chen et al. Scaling laws of motion forecasting and planning – technical report. *arXiv preprint arXiv:2506.08228*, 2025.

Behrooz Ghorbani, Shankar Krishnan, and Ying Xiao. An investigation into neural net optimization via hessian eigenvalue density. *arXiv preprint arXiv:1901.10159*, 2019.

Sachin Goyal et al. Scaling laws for data filtering – data curation cannot be compute agnostic. In *CVPR*, 2024.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Siddhant Haldar and Lerrel Pinto. Scaling laws for imitation learning in single-agent games. *Transactions on Machine Learning Research*, 2023.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

Ethan Jiang et al. Exploring scaling laws for EHR foundation models. *arXiv preprint arXiv:2505.22964*, 2025.

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

Tanishq Kumar et al. Scaling laws for precision. *arXiv preprint arXiv:2411.04330*, 2024.

Zhengyang Li et al. Scaling laws for diffusion transformers. *arXiv preprint arXiv:2410.08184*, 2024.

Armin Moeini et al. (mis)fitting: A survey of scaling laws. In *ICLR*, 2025.

Niklas Muennighoff, Alexander M. Rush, Boaz Barak, Teven Le Scao, Aleksandra Piktus, Nouamane Tazi, Sampo Pyysalo, and Thomas Wolf. Scaling data-constrained language models. *arXiv preprint arXiv:2305.16264*, 2023.

Eric Nguyen, Michael Poli, Matthew G. Durrant, et al. Sequence modeling and design from molecular to genome scale with Evo. *bioRxiv preprint 2024.02.27.582234*, 2024.

Shen Nie et al. Scaling behavior of discrete diffusion language models. *arXiv preprint arXiv:2512.10858*, 2025.

others. Establishing task scaling laws via compute-efficient model ladders. *arXiv preprint arXiv:2412.04403*, 2024.

others. Evaluating the robustness of chinchilla compute-optimal scaling. *arXiv preprint arXiv:2509.23963*, 2025a.

others. Predictable scale: Part II, Farseer: A refined scaling law in large language models. *arXiv preprint arXiv:2506.10972*, 2025b.

others. Towards greater leverage: Scaling laws for efficient mixture-of-experts language models. *arXiv preprint arXiv:2507.17702*, 2025c.

others. Scaling laws for optimal data mixtures. *arXiv preprint arXiv:2507.09404*, 2025d.

others. Scaling laws revisited: Modeling the role of data quality in language model pretraining. *arXiv preprint arXiv:2510.03313*, 2025e.

others. Scaling laws are redundancy laws. *arXiv preprint arXiv:2509.20721*, 2025f.

others. Can language models discover scaling laws? *arXiv preprint arXiv:2507.21184*, 2025g.

Nikhil Sardana and Jonathan Frankle. Reconciling Kaplan and Chinchilla scaling laws. *Transactions on Machine Learning Research*, 2024.

Yi Tay et al. Training compute-optimal transformer encoder models. In *EMNLP*, 2025.

Neeraj Wagh et al. Scaling laws for compute optimal biosignal transformers. 2024.

Xinghua Yang et al. MuPT: A generative symbolic music pretrained transformer. *arXiv preprint arXiv:2404.06393*, 2024.

A Extrapolation Error Raw Data

Surface	Grid	True D^*	Inferred D^*	Abs Error	Rel Error
Symmetric ($\alpha = \beta = 0.31$)					
	XS (±2×)	408.2B	408.2B	≈ 0	$\approx 0\%$
	S (±4×)	408.2B	408.2B	≈ 0	$\approx 0\%$
	L (±8×)	408.2B	408.2B	≈ 0	$\approx 0\%$
	XL (±16×)	408.2B	408.2B	≈ 0	$\approx 0\%$
Chinchilla ($\alpha = 0.34$, $\beta = 0.28$)					
	XS (±2×)	4.04T	4.02T	−13.2B	−0.33%
	S (±4×)	4.04T	3.98T	−52.5B	−1.30%
	L (±8×)	4.04T	3.92T	−117.2B	−2.90%
	XL (±16×)	4.04T	3.83T	−205.8B	−5.10%
Asymmetric ($\alpha = 0.465$, $\beta = 0.155$)					
	XS (±2×)	45.1Q	44.3Q	−755T	−1.67%
	S (±4×)	45.1Q	42.2Q	−2.9Q	−6.50%
	L (±8×)	45.1Q	38.8Q	−6.3Q	−13.91%
	XL (±16×)	45.1Q	34.7Q	−10.4Q	−23.12%

Table 6: Extrapolation error raw data underlying Figure 3. Training range: 10^{17}–10^{21} FLOPs; evaluation budget: 10^{24} FLOPs. B = billion, T = trillion, Q = quadrillion.

B Compounding Errors Raw Data

Config	Surface	Grid	True D^*	Inferred D^*	Rel Error
Offset $3\times$ *(sampling center at* $3\times$ *true optimum at every budget)*					
	Symmetric	XS	408.2B	424.5B	+3.97%
	Symmetric	S	408.2B	422.4B	+3.47%
	Symmetric	L	408.2B	419.0B	+2.65%
	Symmetric	XL	408.2B	414.4B	+1.51%
	Chinchilla	XS	4.04T	4.32T	+7.11%
	Chinchilla	S	4.04T	4.27T	+5.69%
	Chinchilla	L	4.04T	4.17T	+3.38%
	Chinchilla	XL	4.04T	4.05T	+0.24%
	Asymmetric	XS	45.1Q	53.8Q	+19.22%
	Asymmetric	S	45.1Q	51.6Q	+14.41%
	Asymmetric	L	45.1Q	48.2Q	+6.96%
	Asymmetric	XL	45.1Q	44.0Q	−2.42%
Drift to $3\times$ *(center drifts from true optimum to* $3\times$ *at highest budget)*					
	Symmetric	XS	408.2B	433.0B	+6.07%
	Symmetric	S	408.2B	429.4B	+5.17%
	Symmetric	L	408.2B	423.3B	+3.70%
	Symmetric	XL	408.2B	415.1B	+1.69%
	Chinchilla	XS	4.04T	4.50T	+11.61%
	Chinchilla	S	4.04T	4.43T	+9.83%
	Chinchilla	L	4.04T	4.32T	+6.94%
	Chinchilla	XL	4.04T	4.16T	+3.05%
	Asymmetric	XS	45.1Q	60.7Q	+34.57%
	Asymmetric	S	45.1Q	58.7Q	+30.04%
	Asymmetric	L	45.1Q	55.5Q	+22.97%
	Asymmetric	XL	45.1Q	51.4Q	+14.00%

Table 7: Compounding errors raw data underlying Figure 7. Training range: 10^{17}–10^{21} FLOPs; evaluation budget: 10^{24} FLOPs. B = billion, T = trillion, Q = quadrillion.

C Method Comparison Data

Method	Failures	Max E	Max A	Max B	Max α	Max β
2D Nelder-Mead (VPNLS)	0/60	5.2×10^{-8}	6.3×10^{-8}	7.9×10^{-8}	1.2×10^{-8}	2.0×10^{-8}
2D L-BFGS-B (central)	2/60	8.3×10^{-8}	5.3×10^{-8}	6.4×10^{-8}	1.2×10^{-8}	2.0×10^{-8}
2D L-BFGS-B (default ε)	3/60	1.5×10^{-5}	1.2×10^{-5}	1.3×10^{-5}	2.7×10^{-6}	3.8×10^{-6}
2D L-BFGS-B ($\varepsilon = 10^{-10}$)	0/60	1.6×10^{-7}	8.9×10^{-7}	8.6×10^{-7}	1.8×10^{-7}	1.7×10^{-7}
2D L-BFGS-B ($\varepsilon = 10^{-6}$)	17/60	1.6×10^{-3}	1.1×10^{-3}	1.3×10^{-3}	2.2×10^{-4}	3.9×10^{-4}
2D Grid (256^2)	0/60	2.58	2.03	2.03	0.44	0.57
5D L-BFGS-B (analytical)	0/60	0.07	0.75	1.4	0.15	0.28
5D L-BFGS-B (central)	2/60	39.0	2,361	57.9	69.8	9.6
5D L-BFGS-B (finite diff)	4/60	132	2,361	2,335	71.4	87.6

Table 8: Method comparison: maximum |relative error| (%) per parameter across 60 fits (3 surfaces × 20 sampling ranges), computed over successful (converged) fits only. Failure counts show convergence failures out of 60 total fits. Methods sorted by precision (best to worst).

D Detailed Method Comparison

Full per-parameter, per-surface, per-sampling-range error breakdown for all nine method configurations.

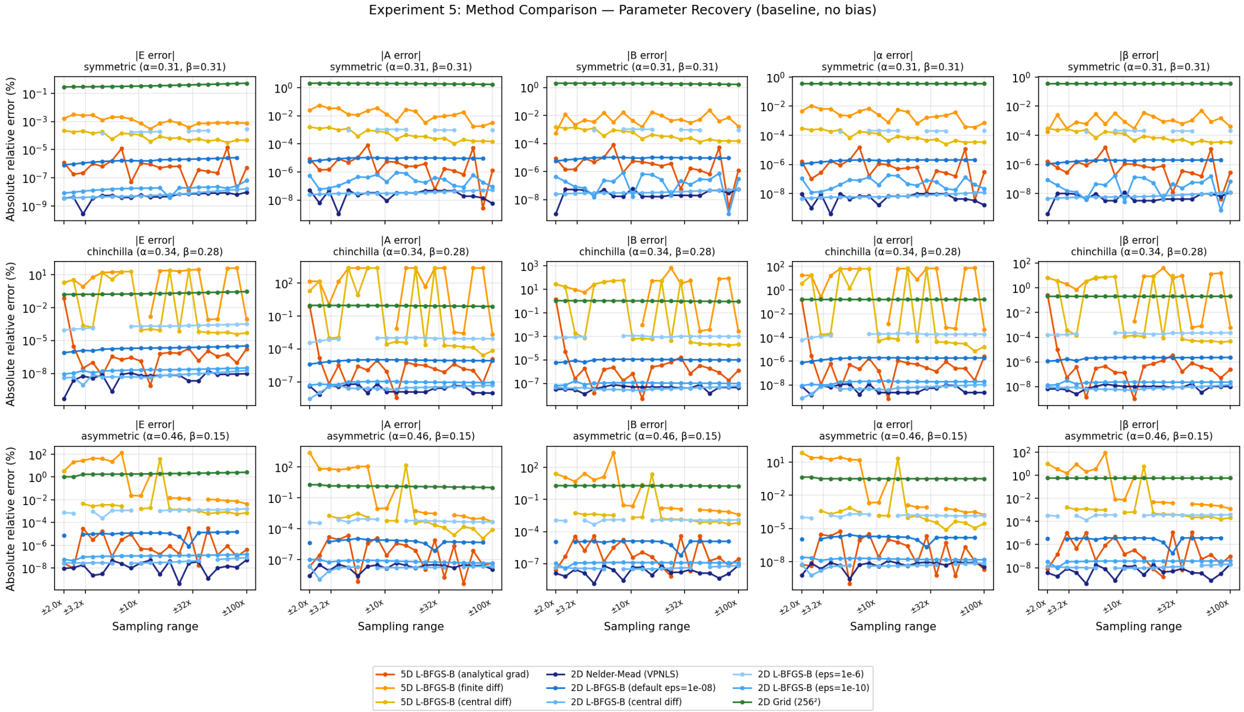

Figure 10: Detailed method comparison: absolute relative error vs sampling range for all nine configurations. Rows correspond to loss surfaces (symmetric, Chinchilla, Asymmetric); columns correspond to parameters (E, A, B, α, β).

E Combined Extrapolation Error by Compute Budget

Detailed view of D^* extrapolation error as a function of compute budget, showing how errors evolve from 10^{22} to 10^{25} FLOPs across sampling ranges, loss surfaces, and bias configurations.

F IsoFLOP Samples with Noise

G Exponent Inference Error Breakdown

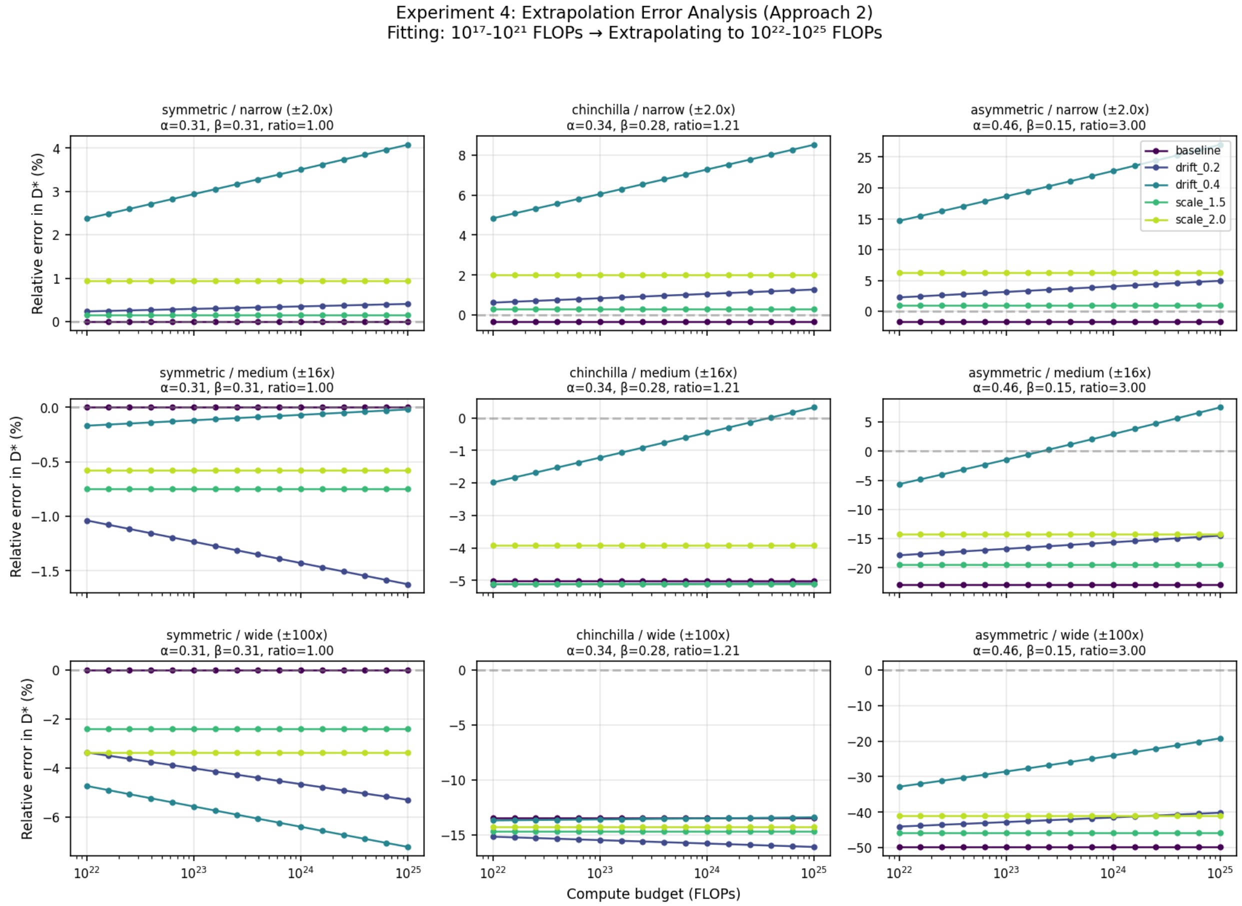

Figure 11: Combined extrapolation error by compute budget. Rows correspond to sampling ranges (narrow, medium, wide); columns correspond to loss surfaces (symmetric, Chinchilla, Asymmetric). Each panel shows relative D^* error vs extrapolation compute budget with one curve per bias configuration.

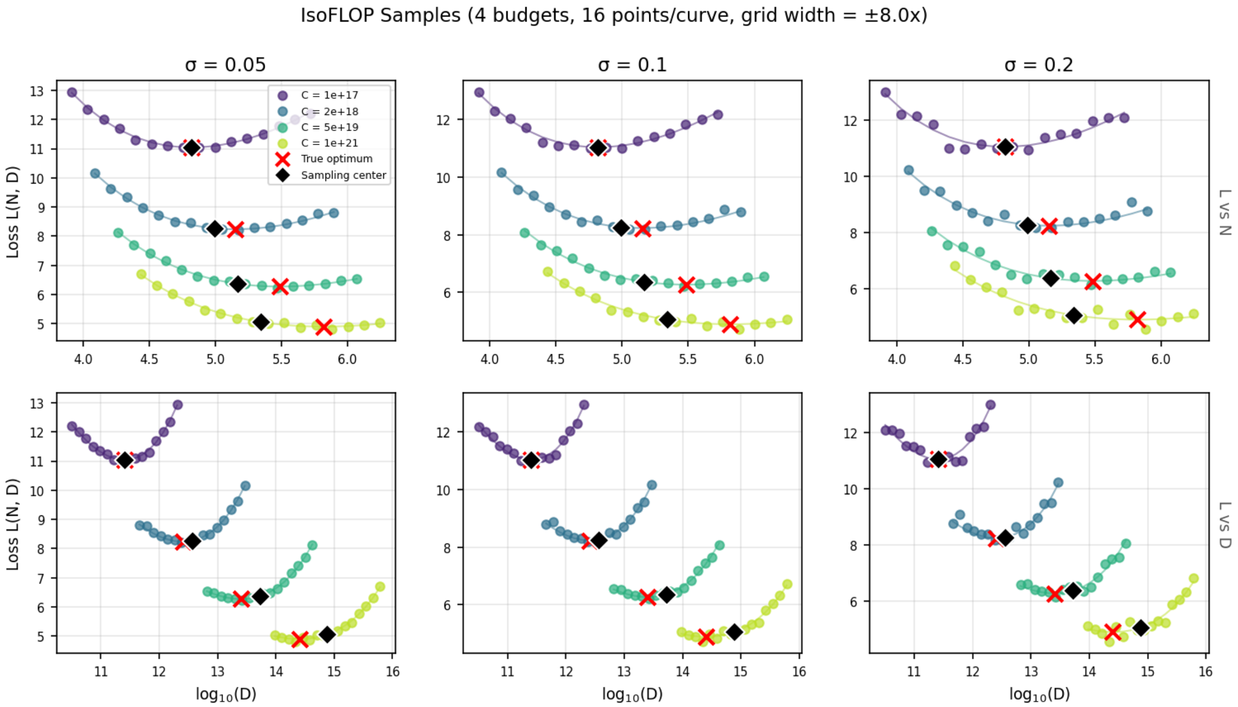

Figure 12: Noisy IsoFLOP samples used in the exponent inference comparison (Figure 9). Columns correspond to noise levels ($\sigma = 0.05, 0.1, 0.2$); rows show loss versus $\log_{10}(N)$ (top) and $\log_{10}(D)$ (bottom). Scatter points are noisy observations; solid curves show the noiseless reference surface. Red × marks the true compute-optimal point at each budget; black diamonds mark the sampling centers, which drift away from the true optima at higher compute budgets. Surface: Asymmetric ($\alpha = 0.465$, $\beta = 0.155$), 4 budgets (10^{17}–10^{21} FLOPs), 32 points per curve, ±8× grid.

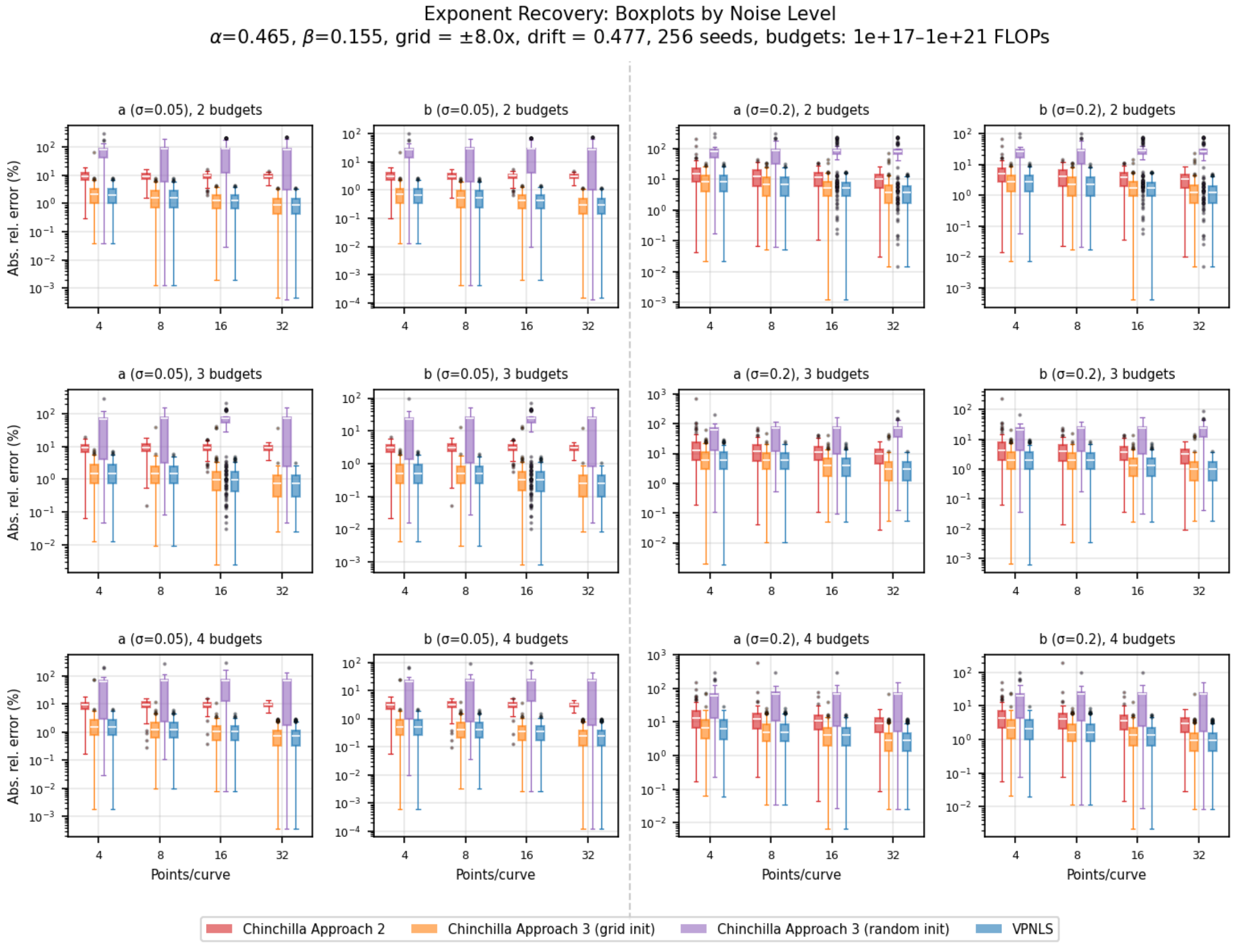

Figure 13: Per-condition breakdown of exponent inference errors from Figure 9. Rows correspond to number of compute budgets (2, 3, 4); left columns show the lowest noise level ($\sigma = 0.05$) and right columns show the highest ($\sigma = 0.2$), each split by exponent (a and b). Within each panel, boxplots show absolute relative error (%) on a log scale for each method at each points-per-curve setting (4, 8, 16, 32), over 256 noise realizations. Approach 3's sporadic large errors (outlier points above the upper whiskers) appear across conditions without concentrating in any single noise level, budget count, or dataset size.