Data-Constrained Language Model Pretraining: Improved Regularization and Scaling Laws

Zhiwei Xu[1], **Shihao Wu**[1], **Hanseul Cho**[2], **Wei Hu**[1*], **Yixin Wang**[1*]
[1]University of Michigan, [2]KAIST AI
{zhiweixu,wshihao,vvh,yixinw}@umich.edu,jhs4015@kaist.ac.kr

Abstract

Classical scaling laws for language model pretraining balance model size against training dataset size under a fixed compute budget, assuming abundant data and a single pass over the corpus. As training compute grows faster than the supply of natural language data, pretraining is likely to enter a data-constrained, compute-rich regime where models train for multiple epochs over a finite dataset. We study data-constrained pretraining along two axes, regularization and scaling. For regularization, we study masked-input regularization (MIR), an auxiliary next-token prediction loss on randomly masked inputs. MIR tests whether the random masking central to diffusion language models can benefit autoregressive pretraining without architectural changes or inference overhead. Across 72M to 1.4B parameter models, we find that MIR added on top of strong weight decay improves validation loss over autoregressive strong-weight-decay-only models, with downstream gains at 1.4B. For scaling, we propose SoftQ, a scaling law that couples model size and data size to capture their interaction under repeated data. Classical alternatives such as the Chinchilla law use an additive form that decouples these terms, making them misspecified in the data-constrained regime. We find that SoftQ fits data-constrained experiments substantially better than these alternatives, and estimates MIR's gains as equivalent to roughly 1.3 times as much unique training data. We release our code at https://github.com/yixinw-lab/dc_pretrain.

1 Introduction

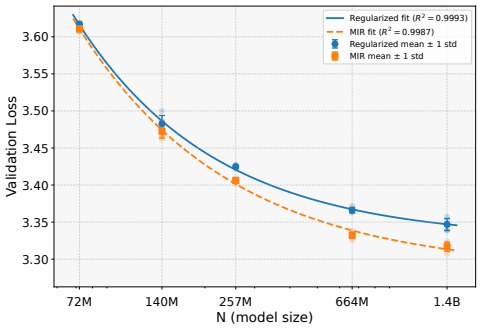
Scaling laws [Kaplan et al., 2020, Hoffmann et al., 2022] are widely used to choose model size and training-token budget for large language model pretraining. Classical scaling laws are largely compute-centric: they study how to allocate a fixed compute budget between parameters and tokens, assuming that unique training data can scale freely with compute. In this abundant-data setting, pretraining is typically performed with a single pass over a large corpus.

However, training compute is growing faster than the supply of natural language data [Villalobos et al., 2024, Sevilla and Roldán, 2024, Common Crawl, 2025], making data-constrained, compute-rich pretraining increasingly important. In this regime, the unique dataset is fixed, and additional compute is spent on larger models and multiple passes over the same corpus. Prior work has begun to study this setting: Muennighoff et al. [2023] tuned data repetition while fixing weight decay to 0.1 and proposed scaling laws based on effective resources that saturate with repetitions and excess parameters; Kim et al. [2026a] further showed that large weight decay is critical for preventing overfitting.

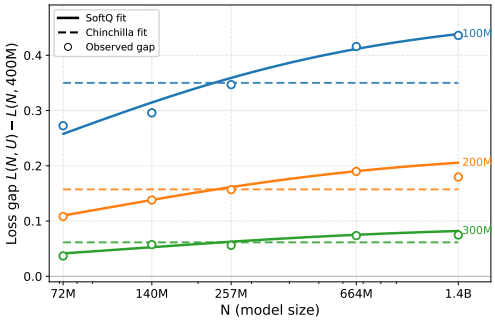
This shift raises two linked questions. The first concerns regularization: how can models avoid overfitting when compute increases but unique data does not? Prior work points to strong weight decay as one answer. A second possibility comes from masked diffusion language models (dLLMs),

*Equal advising.

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



(a) MIR improves a strong AR baseline.



(b) SoftQ captures data–model coupling.

Figure 1: **Overview of the main results.** Left: On DataComp-LM (DCLM) dataset [Li et al., 2024] with 100M unique training tokens, MIR improves validation loss over the strongly regularized autoregressive baseline across model sizes. Points show means over five random seeds, error bars show one standard deviation, and faint markers show individual runs. Right: On the strongly regularized baseline grid, we plot the loss gap $L(N, U) - L(N, 400\text{M})$ for unique data budget $U \in \{100\text{M}, 200\text{M}, 300\text{M}\}$. Chinchilla predicts a model-size-invariant gap for each U, while SoftQ tracks the empirical fan-out: the penalty from limited unique data grows with model size.

which typically use the same transformer architecture as autoregressive (AR) models but train by predicting randomly masked tokens. Under identical hyperparameters, dLLMs achieve lower validation loss than AR transformers in the data-constrained regime [Ni et al., 2025, Prabhudesai et al., 2025], suggesting that random masking may itself act as a form of regularization. However, these comparisons do not isolate masking from regularization strength: the dLLM advantage may be complementary to strong weight decay, or it may largely reflect insufficiently strong regularization in the AR baseline. This motivates our first question: how do random masking and weight decay interact, and how much does each contribute on top of the other?

The second question concerns scaling: what loss law describes the data-constrained, compute-rich regime? Chinchilla-style laws were fit to single-pass, abundant-data training and may not capture the validation-loss surface when unique data, rather than compute, is the binding resource. In particular, their additive form predicts that the loss gap between two unique-data budgets should be independent of model size. In this paper, we study both questions in the data-constrained, compute-rich regime.

Finding 1: Random masking provides regularization complementary to strong weight decay. We first ask how the two regularization mechanisms interact. We find that strong weight decay is not specific to AR pretraining: applying the AR-tuned weight decay to dLLMs substantially lowers their validation loss, and once both models are strongly regularized, their validation losses become comparable across the model sizes we study. Given that strong weight decay alone provides such substantial regularization, this makes it unclear whether random masking can still provide additional benefit once strong weight decay is already in use.

To isolate this effect, we study *masked-input regularization* (MIR), a minimal modification to standard AR pretraining. Let x denote a clean sequence and $\widetilde{x}$ a randomly masked version of the same sequence. Instead of optimizing only the standard next-token prediction loss $\mathcal{L}_{\text{NTP}}(x)$, MIR optimizes

$$\mathcal{L} = \mathcal{L}_{\text{NTP}}(x) + \lambda \mathcal{L}_{\text{NTP}}(\widetilde{x}).$$

Thus, the model trains on both clean and masked inputs, using the masked-input loss as an auxiliary regularizer. MIR requires no architectural changes and preserves standard autoregressive decoding at inference. Although it increases training compute, our setting is data-constrained and compute-rich, so we study MIR as a way to improve loss at a fixed unique-data budget, i.e., data efficiency rather than compute efficiency.

Across models from 72M to 1.4B parameters trained on DCLM [Li et al., 2024] and Stack-V2 [Lozhkov et al., 2024], MIR consistently improves validation loss on top of strong weight decay (Figure 1a). At 1.4B parameters, it also yields substantial downstream gains, including +10.2 points on BoolQ and +2.2 points on SciQ.

Finding 2: Chinchilla is misspecified in the data-constrained, compute-rich regime; a coupled scaling law fits better. To quantify how much unique data MIR is worth, we extend our experiments across five model sizes and four unique-data budgets and fit several scaling laws. The additive Chinchilla form [Hoffmann et al., 2022] fits poorly in this regime: it predicts that the validation-loss gap between two data budgets is independent of model size, whereas our experiments show that this gap grows with model size (Figure 1b).

We propose the *SoftQ scaling law*, a five-parameter form that couples model size and data size through a soft bottleneck motivated by the skill-learning view of scaling laws [Michaud, 2026]. SoftQ achieves better in-sample fit and out-of-sample prediction than Chinchilla, Quanta [Michaud, 2026], and Muennighoff-style [Muennighoff et al., 2023] laws on our dataset. The same ranking holds on an independent dataset from Kim et al. [2026a]. Using SoftQ as the baseline scaling law, we estimate MIR's gain over the strongly regularized baseline to be equivalent to roughly $1.3\times$ as much unique training data at the 200M–400M token budgets.

Contributions. We summarize our contributions as follows: (i) We show that large weight decay substantially improves dLLMs in the data-constrained regime, and that random masking further improves strongly regularized AR models. Building on this observation, we propose MIR, a minimal recipe that augments strongly regularized AR pretraining with an auxiliary masked-input next-token loss; we estimate MIR to be worth roughly $1.3\times$ as much unique training data at the 200M to 400M token budgets. (ii) We show that additive Chinchilla-style scaling laws do not fit the data-constrained, compute-rich regime, and propose SoftQ, a five-parameter scaling law that couples model and data size and substantially outperforms these alternatives.

2 Setup: Data-Constrained Autoregressive and Masked Pretraining

2.1 Data-Constrained and Compute-Rich Pretraining

Let N denote the number of model parameters, U the number of unique pretraining tokens, N_E the number of epochs over those tokens, and $D = UN_E$ the total number of training tokens. For a standard dense decoder-only transformer trained with next-token prediction, the training compute is approximately $C(N, D) \approx 6ND$.

Classical compute-optimal scaling laws [Kaplan et al., 2020, Hoffmann et al., 2022] model evaluation loss as a function of model size and training-token budget. In the abundant-data regime, the processed tokens can be treated as fresh samples, so the distinction between unique tokens and repeated tokens is not explicit. The standard compute-allocation problem is

$$(N^\star(C), D^\star(C)) = \arg\min_{N,D} L_{\text{eval}}(N, D) \quad \text{s.t.} \quad C(N, D) = C.$$

For example, Chinchilla-style parametric scaling writes $\widehat{L}(N, D) = E + AN^{-\alpha} + BD^{-\beta}$, and then chooses the point on this surface that minimizes loss under the training-compute constraint. Such laws are highly effective when new data is available, but they do not distinguish a token budget D consisting of fresh tokens from the same budget obtained by repeatedly training on a finite corpus.

In data-constrained, compute-rich pretraining, the unique-token budget U is fixed or bounded, and C is unbounded. Additional training compute can be spent by increasing the number of epochs, increasing model size, or changing regularization. Prior work studies several versions of this problem. Muennighoff et al. [2023] model repeated data under compute constraints by replacing raw token and parameter counts with effective resources that saturate as repetitions and excess parameters grow. Kim et al. [2026a] study a more compute-rich setting in which the unique data is fixed and the training recipe is tuned to estimate the best attainable loss at each model scale.

We follow the compute-rich perspective. For a fixed architecture family, optimizer class, data distribution, and evaluation protocol, define the optimized validation-loss envelope

$$L^\star(N, U) = \inf_{h \in \mathcal{H}} L_{\text{eval}}(N, U; h),$$

where h includes the tunable training hyperparameters, such as the number of epochs, learning-rate schedule, weight decay, and other regularization choices. In this formulation, $D = UN_E(h)$ determines the compute used by a particular training run, but compute is not the binding constraint

used to define $L^\star$. The goal is therefore to model the joint dependence of the best-achievable loss on model size N and unique data size U.

2.2 Autoregressive and Masked Diffusion Language Models

Let p_θ denote the transformer model and $\{x_i\}_{i=1}^n$ the training dataset, where each sample $x_i = [x_{i,0}, x_{i,1}, \ldots, x_{i,T-1}]$ is a sequence of length T. Autoregressive models predict tokens from left to right. The training objective $\mathcal{L}_{\text{NTP}}$ is $-\sum_{i=1}^{n}\sum_{t=0}^{T-1} \log p_\theta(x_{i,t} \mid x_{i,<t})/(nT)$. For each sequence x_i, dLLMs sample a mask ratio $r_i \sim \text{Unif}(0,1]$, and use a Bernoulli random variable $\text{Bern}(r_i)$ to decide whether to mask the token $x_{i,t}$ or not for each position $t \in [0, T)$. The model only predicts the true tokens at those masked positions. The training objective is

$$-\frac{1}{nT}\sum_{i=1}^{n}\left[\frac{1}{r_i}\sum_{t=0}^{T-1}\mathbb{I}(\widetilde{x}_{i,t} = \text{MASK})\log p_\theta(x_{i,t} \mid \widetilde{x}_i)\right],$$

where $\widetilde{x}_i$ represents the masked sample x_i.

3 Regularization in the Data-Constrained, Compute-Rich Regime

3.1 Weight Decay Transfers Across AR and dLLM Pretraining

Recent studies report that dLLMs outperform AR models in the data-constrained regime [Ni et al., 2025, Prabhudesai et al., 2025], using weight decay $\text{wd} = 0.1$ for both. Independently, Kim et al. [2026a] showed that large weight decay is critical for AR pretraining in this regime. We ask whether this benefit transfers to dLLMs and re-examine the AR–dLLM comparison under matched large-weight-decay treatment. On DCLM with 100M unique tokens, we compare four recipes at three model sizes (140M, 257M, and 664M): (i) Multi-epoch AR ($\text{wd} = 0.1$, tuned epochs); (ii) Multi-epoch dLLM ($\text{wd} = 0.1$, per Prabhudesai et al. [2025]); (iii) Strongly Regularized AR with epochs, learning rate, and weight decay jointly tuned following Kim et al. [2026a]; (iv) Strongly Regularized dLLM, which inherits the AR-tuned weight decay but keeps other hyperparameters at Prabhudesai et al. [2025] defaults. We report final-step validation loss for AR and the best across-epoch loss for dLLM.

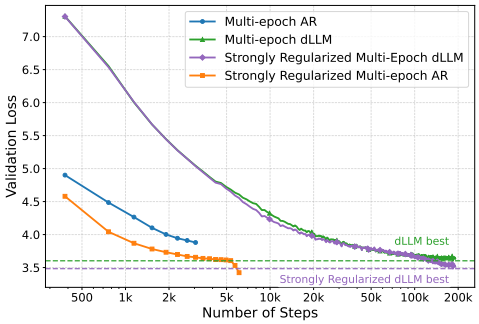


Figure 2: Validation Loss dynamics on DCLM 100M for the 257M model. Large weight decay substantially improves both multi-epoch AR and dLLM training; with both well regularized, their validation losses become comparable.

Figure 2 shows all four recipes at 257M. With $\text{wd} = 0.1$, we reproduce the finding that dLLM (3.60) outperforms multi-epoch AR (3.88). Large weight decay dramatically improves both: it reduces AR loss to 3.42 and, when ported to dLLM, reduces dLLM loss to 3.48. Since dLLM validation loss is the negative evidence lower bound (an upper bound on the negative log-likelihood) while AR loss is exact negative log-likelihood, the slightly higher dLLM validation loss does not imply worse performance. The two strongly regularized recipes have losses comparable at 140M, 257M, and 664M (see Table 4 in Appendix A), implying that the previously reported AR–dLLM gap is largely explained by insufficient AR regularization. Still, the fact that dLLMs avoid the repeated-epoch collapse seen in weakly regularized AR suggests that random input masking acts as an implicit regularizer in its own right. We next ask whether this masking signal can contribute additional gains on top of strong weight decay when added to AR training.

3.2 Masked Input Regularization

To capture this hypothesized benefit without abandoning the efficiency of standard AR decoding, we study masked-input regularization (MIR). The method samples a mask ratio r from a uniform distribution $\text{Unif}(r_{\min}, r_{\max})$ for each input sequence x. At each position $t \in [0, T-1]$, a Bernoulli random variable with success probability r determines whether to replace the token x_t with a

specialized [MASK] token. Let $\widetilde{x}$ denote this corrupted sequence. Without altering the model architecture, MIR adds an auxiliary next-token prediction loss on the masked sequence:

$$\mathcal{L} = \mathcal{L}_{\mathrm{NTP}}(x) + \lambda \mathcal{L}_{\mathrm{NTP}}(\widetilde{x}).$$

It requires two forward passes to calculate the training loss for each batch. MIR therefore increases per-step training compute. Because our focus is the data-constrained, compute-rich regime, we use MIR to study whether additional compute can improve loss at a fixed unique-data budget, rather than as a compute-efficiency method. See tuning details and regularization coefficient in Appendix A.7.

Remark 3.1. We study the data-constrained, compute-rich regime defined in Section 2.1, where unique data rather than compute is the binding resource, so we focus on improving data efficiency rather than compute efficiency. We further quantify MIR's gain in data efficiency in Section 4.2.

3.3 Theoretical Intuition: Reducing Memorization via Masking

We provide intuition for how masking improves validation loss by analyzing a toy context-specific noise model in Appendix C. This model decomposes each sequence into three parts: a context-specific component that enables memorization and acts as noise for generalization, a generalizable component that contains predictive features, and an output token to be predicted from the first two components. Under the data-constrained, compute-rich regime, we establish the following dynamic.

Theorem (Informal). *Under the context-specific noise model in the data-constrained, compute-rich regime, standard autoregressive pretraining can minimize training loss by relying almost entirely on the context-specific component, thereby memorizing patterns that do not generalize to unseen examples. In contrast, MIR regularizes the model's dependence on the context-specific components and encourages it to learn predictive patterns on the generalizable components, thereby strictly improving validation loss. Moreover, for a fixed data size, this improvement increases as model capacity grows.*

This informal theorem illustrates that MIR improves validation loss by reducing the model's dependence on context-specific noise and encouraging it to learn generalizable predictive features. We provide the formal model definition, assumptions, theorem statement, and proofs in Appendix C.

3.4 Empirical Results

We evaluate MIR against the strongly regularized AR model baseline along four axes: scaling behavior on natural language, whether the gain transfers to coding data, where the gain comes from, and whether it translates to downstream tasks. Throughout, the only difference between MIR and the baseline is the auxiliary masked-input loss; architecture, optimizer, and the per-cell-tuned (epochs, weight decay, learning rate) configuration are held fixed across the two recipes.

We find the following. (1) On DCLM with 100M unique tokens, MIR reduces validation loss at every model scale from 72M to 1.4B and on every matched random seed, with the average gain growing from roughly 0.006 at 72M to about 0.03 at 1.4B. This trend is consistent with the theoretical prediction in Section 3.3 that overparameterized models benefit more from masking-based regularization. (2) The benefit is not specific to natural language: with hyperparameters tuned only on DCLM, MIR also reduces validation loss at all five model sizes on the code-heavy Stack-V2 dataset. (3) A token-level analysis on the 1.4B model shows that MIR's gain comes from a broad set of validation positions rather than a few outliers. At the positions where MIR most outperforms the baseline, the true next token is itself usually a common one such as a function word or punctuation, and what makes prediction difficult is the preceding context: rare names, mixed scripts, broken word pieces, or noisy web text (Section 3.4). (4) The loss improvement directionally transfers to downstream tasks: the 1.4B MIR model outperforms the strongly regularized baseline on six of eight zero-shot metrics, including $+10.2$ points on BoolQ and $+2.2$ points on SciQ.

Experimental setup. To evaluate masked-input regularization, we train models on two distinct data distributions: standard natural language from DCLM [Li et al., 2024] and code-heavy text from Stack-V2 [Lozhkov et al., 2024]. For both datasets, the pretraining budget is fixed to 100M unique seed tokens and 10M tokens are reserved for validation. We tune hyperparameters only on DCLM data and test whether the improvement from MIR still exists on the Stack-V2 dataset.

We build a scaling ladder with five model sizes:

$$\text{ScalingLadder}(k) = (kW_1, kL_1, S_1, B_1),$$

where $W_1 = 1024$ is the embedding dimension when $k = 1$, $L_1 = 12$ is the number of layers when $k = 1$, $S_1 = 2048$ is the sequence length, $B_1 = 128$ is the total batch size, and $k \in \{0.5, 0.75, 1, 1.5, 2\}$. Across the scaling ladder, the attention head dimension is fixed at 64, while the depth, embedding dimension, MLP dimension, and number of attention heads increase with scale. The model size ranges from 72M to 1.4B. We use a Llama-style decoder-only transformer and use the same model architecture for all experiments. AdamW optimizer is used for all experiments. We use grid search to select the number of training steps, learning rate, and weight decay for each model. See Appendix A for details on the optimizer, model architecture, and hyperparameter search.

Validation loss improvements. Figure 1a visualizes validation loss across the scaling ladder for the DCLM 100M dataset, averaged over five random seeds. MIR improves validation loss over the strongly regularized baseline for every matched seed at every model scale. On the 1.4B parameter model, for example, MIR reduces the mean validation loss from 3.347 to 3.317. The average gain grows from roughly 0.006 loss at 72M parameters to about 0.03 loss for the two largest models, suggesting that MIR is especially useful when model capacity is high relative to the amount of unique training data. This trend is qualitatively consistent with our theoretical analysis, which predicts that larger overparameterized models are more prone to overfitting, and therefore benefit more from masking-based regularization.

Crucially, this regularization benefit generalizes beyond standard natural language. We repeat the same 100M token experiments to evaluate performance on code-heavy data: on Stack-V2, MIR reduces validation loss at all five model sizes, with absolute gains from 0.008 to 0.020 loss; see full numbers in Table 8 in the Appendix.

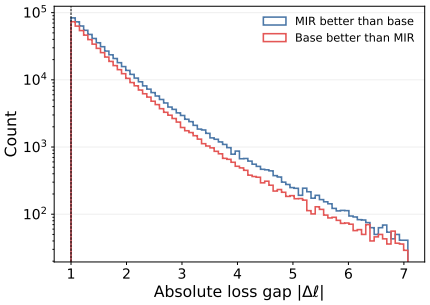
Where MIR helps: Token-level analysis. To localize where the validation-loss gain comes from, we compare the 1.4B regularized baseline and the 1.4B MIR model on the 10M DCLM eval dataset. For each position t, we compute the negative log-likelihood on the true target $y_t = x_{t+1}$ and define the token-level loss gap as $\Delta\ell_t = \ell_{\text{base}}(t) - \ell_{\text{MIR}}(t)$, so that positive values favor MIR. Figure 3 Left shows that the MIR-better tail is both larger and slightly heavier than the baseline-better tail after removing the center region $|\Delta\ell_t| < 1$: 6.61% of tokens satisfy $\Delta\ell_t \geq 1$, while 5.41% satisfy $\Delta\ell_t \leq -1$. Therefore, the overall loss gain is not driven by a few isolated outliers, but appears on a broad set of hard validation tokens.

The top positive-gap tokens reveal a clear qualitative pattern. We rank all validation positions by $\Delta\ell_t$, decode the top 0.1% MIR-better positions, and inspect the true token together with the preceding and following tokens. These high-gap examples are dominated by continuation problems rather than standalone rare targets: 62.6% are word or subword continuations, 16.3% occur in non-English or transliterated text, and 11.6% are punctuation tokens. Importantly, the true token is often a common token such as "and", "to", "of", "is", a comma, or a closing parenthesis. What makes these positions hard to predict is the local prefix context: non-English languages, rare names, mixed scripts, broken word pieces, or noisy web and markup text.

Representative examples illustrate how these gains arise in practice. Figure 3(right) shows cases where the baseline falls back to a generic continuation or keeps following the wrong local pattern, whereas MIR recovers the intended continuation. One example is a mixed-script entity name followed by a Japanese parenthetical gloss, where the correct next token is the closing parenthesis ")"; MIR predicts it correctly, while the baseline keeps extending the Japanese string. Taken together, these results suggest that MIR helps most when the next-token decision depends on robustness to unusual or noisy local prefix context rather than on simple frequency-based continuation. This pattern is consistent with our theoretical intuition that masking regularizes the model's dependence on irrelevant details in the prefix context and encourages it to learn predictive features that generalize across contexts.

Downstream evaluations. To understand whether the improved validation loss translates to capability gains on downstream tasks, we evaluate the two 1.4B models trained on DCLM dataset with $U =$ 100M across a suite of downstream tasks using `lm-evaluation-harness` [Gao et al., 2024].

Table 1 shows that the MIR-trained model achieves superior performance on six of the eight evaluated metrics. The improvements are particularly pronounced on reasoning and reading comprehension tasks, pushing accuracy on BoolQ up by 10.18 percentage points and SciQ up by 2.20 percentage



Ex. 1. Rather, since the verses of the Qur'an are a source of guidance, clear proofs, and the distinguisher between right and wrong, just as we ...
MIR $\arg\max p_{\text{MIR}} =$ `between`, p_{MIR}('`between`') = 0.7188
Baseline $\arg\max p_{\text{base}} =$ `of`, p_{base}('`between`') = 0.0001037

Ex. 2. Conversely, another trader watching for patterns in, say, Google (NASDAQ:GOOG) may notice a reverse head and shoulders and buy shares of Google, expecting ...
MIR $\arg\max p_{\text{MIR}} =$ `shoulders`, p_{MIR}('`shoulders`') = 0.4019
Baseline $\arg\max p_{\text{base}} =$ `a`, p_{base}('`shoulders`') = 0.0002055

Ex. 3. ... the Ventriloquist of the Dead is replaced by an obese shape-shifter called The Death Imitator (死の物真似師 (しのものまねし) Shino Mono Maneshi?, known as Ghost Kaiba in the English anime). The second Player Killer that the group encounters is ...
MIR $\arg\max p_{\text{MIR}} =$), p_{MIR}(')') = 0.3328
Baseline $\arg\max p_{\text{base}} =$ 売, p_{base}(':') = 1.6×10^{-6}

Ex. 4. 7 人之中有 4 人（57%）認為這篇評論有參考價值
MIR $\arg\max p_{\text{MIR}} =$ %, p_{MIR}('%') = 0.157177
Baseline $\arg\max p_{\text{base}} =$ <U+FFFD>, p_{base}('%') = 0.002207

Figure 3: Left: Absolute token-level loss-gap tails on all validation tokens for the 1.4B models after removing the center region $|\Delta\ell| < 1$. The positive tail, where MIR assigns higher probability to the true next token than the strongly regularized baseline, is both larger and slightly heavier. Right: Representative MIR-better tokens from the top 0.1% positive-gap set. In each example, the target next token is highlighted in red, and the probabilities assigned to that token by MIR and by the strongly regularized baseline are shown below. Many large-gap cases involve names, subword completions, mixed scripts, or noisy web and technical text, even when the true token itself is common.

points compared to the strongly regularized baseline. Because these models are trained at academic scale, with 1.4B parameters and 100M unique training tokens, we view the downstream evaluations as a coarse capability check: MIR shows a large gain on BoolQ and smaller mixed changes elsewhere, with the overall pattern directionally agreeing with its validation-loss improvement.

Table 1: Downstream zero-shot evaluation for 1.4B models trained on the DCLM data with U =100M.

Task	Random Guess	Regularized Baseline	+ MIR
ARC-Easy (acc_norm)	0.2500	0.3805 ± 0.0100	$\mathbf{0.3893 \pm 0.0100}$
BoolQ (acc)	0.5000	0.4511 ± 0.0087	$\mathbf{0.5529 \pm 0.0087}$
HellaSwag (acc_norm)	0.2500	0.2833 ± 0.0045	$\mathbf{0.2855 \pm 0.0045}$
PiQA (acc_norm)	0.5000	$\mathbf{0.5996 \pm 0.0114}$	0.5985 ± 0.0114
RACE (acc)	0.2500	0.2689 ± 0.0137	$\mathbf{0.2766 \pm 0.0138}$
SciQ (acc_norm)	0.2500	0.5780 ± 0.0156	$\mathbf{0.6000 \pm 0.0155}$
Lambada (acc)	$\sim$0.0000	$\mathbf{0.2271 \pm 0.0058}$	0.2261 ± 0.0058
Lambada (perplexity)	N/A	112.8966 ± 4.9091	$\mathbf{106.7115 \pm 4.5752}$

4 Scaling in the Data-Constrained, Compute-Rich Regime

In this section, we extend the experiments to a five-by-four grid of model sizes by unique-data budgets and use it to (a) show that the classical Chinchilla scaling law is misspecified in the data-constrained, compute-rich regime, (b) propose the SoftQ scaling law as a better-fitting alternative, and (c) quantify MIR's data efficiency gain over the strongly regularized baseline.

Constructing the baseline grid. We choose three additional unique-data budgets beyond the 100M used in Section 3.4: 200M, 300M, and 400M. The grid is thus five model sizes $\times$ four data sizes: $\{72\text{M}, 140\text{M}, 257\text{M}, 664\text{M}, 1.4\text{B}\} \times \{100\text{M}, 200\text{M}, 300\text{M}, 400\text{M}\}$. For each cell, we tune the number of epochs, weight decay, and learning rate; the optimal weight decay is consistently much larger than the standard value of 0.1, so following Kim et al. [2026a] we call this the strongly regularized recipe. See Appendix A for the hyperparameter search and best configurations. The result is a baseline dataset $\{(N, U, L)\}$ of 20 points, where L is the validation loss of the AR model of size N in the scaling ladder trained on U unique tokens with the best hyperparameters $(N_E, \text{weight decay}, \text{learning rate})$ for that cell.

4.1 The SoftQ scaling law

Why Chinchilla is Misspecified. The Chinchilla scaling law decomposes loss into irreducible entropy, finite-parameter error, and finite-data error:

$$L_{\text{Ch}}(N, U) = E + \frac{A}{N^{\alpha}} + \frac{B}{U^{\beta}}. \tag{1}$$

Its additive structure implies that the parameter and data terms are separable. Consequently, given a model with size N, the loss gap between two unique data budgets U_1, U_2 does not depend on N:

$$L_{\mathrm{Ch}}(N, U_1) - L_{\mathrm{Ch}}(N, U_2) = \frac{B}{U_1^{\beta}} - \frac{B}{U_2^{\beta}}.$$

This prediction is at odds with the expected behavior in data-constrained, compute-rich pretraining. The marginal value of additional unique data should depend on model size. For sufficiently small models, both U_1 and U_2 provide more unique information than the model can effectively exploit, so the losses obtained from the two data budgets should be similar. In this regime, the loss gap should be close to zero. For sufficiently large models, capacity is no longer the binding constraint, and the difference in available unique information between U_1 and U_2 should become visible in validation loss. The gap should therefore increase with N, reflecting a coupling between model size and unique data budget that Chinchilla's additive form cannot represent.

We verify this behavior empirically. Figure 1b shows the diagnostic directly: the loss gap between each smaller data budget and the 400M budget increases with model size, but Chinchilla predicts a constant gap for each budget. This motivates a coupled law rather than an additive one.

Existing coupled laws. Two prior laws have moved in this direction. Muennighoff et al. [2023] generalize the Chinchilla law by replacing raw data and parameter counts with effective model size N' and effective data size D' that saturate under repeated data and excess parameters. It includes the number of epochs N_E as an additional input to predict the validation loss:

$$L_{\mathrm{M}}(N, U, N_E) = E + \frac{A}{(N')^{\alpha}} + \frac{B}{(D')^{\beta}}, \quad D' = f(U, N_E),\ N' = g(N, U, N_E), \tag{2}$$

which has seven parameters to fit. Michaud [2026] derive a scaling law from the quanta-skill learning model. They assume that the use frequencies of skills follow a power law and obtain $L(N, U) - E \propto n(N, U)^{-\alpha}$, where $n(N, U)$ is the number of skills the model can learn given N parameters and U unique tokens. Under further assumptions, they show $n(N, U) \propto N$ when $U \to \infty$ and $n(N, U) \propto U^{1/(1+\alpha)}$ when $N \to \infty$. Concurrently, Merrill et al. [2026] proposed Expressivity-Aware Scaling Laws, which derived the same scaling properties. Setting $n(N, U) = \left(A/N + B/U^{1/(1+\alpha)}\right)^{-1}$ yields the Quanta scaling law:

$$L_{\mathrm{Q}}(N, U) = E + \left(\frac{A}{N} + \frac{B}{U^{1/(1+\alpha)}}\right)^{\alpha}, \tag{3}$$

where the marginal value of increasing model size depends on the available data through the outer exponent. We give the full expression of Muennighoff law in Appendix B and the detailed Quanta derivation in Appendix D.

SoftQ. Motivated by the skill-learning view of scaling, we propose the *SoftQ scaling law*, a soft-quanta law that combines the parameter-limited and data-limited regimes through a smooth bottleneck:

$$L_{\mathrm{SoftQ}}(N, U) = E + \left(\frac{A}{N^{\rho}} + \frac{B}{U^{\rho/(1+\alpha)}}\right)^{\alpha/\rho}. \tag{4}$$

The parameter ρ controls the sharpness of the transition between the parameter-limited and data-limited regimes. As $U \to \infty$, the law recovers a parameter-scaling limit $L - E \propto N^{-\alpha}$; as $N \to \infty$, it recovers a data-scaling limit $L - E \propto U^{-\alpha/(1+\alpha)}$. It has five fitted parameters, $\{A, B, E, \alpha, \rho\}$, matching the Chinchilla parameter count while explicitly coupling model size and data size. When $\rho = 1$, SoftQ reduces to the Quanta law, so SoftQ strictly nests Quanta as a special case while adding one parameter that controls the bottleneck sharpness.

4.2 Scaling Laws Comparison and MIR data efficiency

We compare Chinchilla, Quanta, Muennighoff, and SoftQ on three diagnostics: (1) full fit on the strongly regularized baseline results; (2) held-out fit, training on the 100M/200M/300M points and predicting the five 400M points; and (3) full fit on an independent baseline dataset provided by Kim et al. [2026a]. For the fitting protocol, Chinchilla, Quanta, and SoftQ use the Approach-3-style objective of Hoffmann et al. [2022]: Huber loss with threshold $\delta = 10^{-3}$ on log-loss residuals. For the Muennighoff-style law, our main comparison uses a dataset-adapted two-stage protocol:

Table 2: Scaling laws comparison results. Lower is better.

		Full fit			Held-out 400M		[Kim et al.] Full fit	
Law	k	RMSE	MAE	AIC	RMSE	MAE	RMSE	AIC
Chinchilla	5	0.02653	0.01802	-135.18	0.03106	0.02540	0.04041	-92.68
Quanta	4	0.01252	0.00889	-167.23	0.01497	0.01207	0.02375	-111.69
Muennighoff	7	0.02335	0.01713	-136.29	0.03252	0.02711	0.03299	-95.17
SoftQ	**5**	**0.00801**	**0.00520**	**-183.06**	**0.00595**	**0.00471**	**0.00785**	**-145.10**

fit the base Chinchilla coefficients on the same split, then hold them fixed while fitting only the decay constants $R_N^\star$ and $R_D^\star$. We report RMSE and MAE on the raw validation-loss scale, and an SSE-based Gaussian AIC: $n\log(\mathrm{RSS}/n) + 2k$, where k is the number of fitted parameters.

Table 2 shows that SoftQ is the strongest baseline law across all three diagnostics. It gives the best in-sample fit on the full baseline dataset, the best data-axis extrapolation to the held-out 400M budget, and the best fit on the external scaling law datasets from Kim et al. [2026a]. The held-out result is especially important for data-efficiency estimation, because we later ask how much additional unique data the regularized baseline would need to match an MIR asymptote. Figure 1b visualizes this fit: SoftQ reproduces the empirical fan-out across data budgets that Chinchilla cannot. Eq. (14) in Appendix B gives its full expression.

We train the model with MIR on the same grid. For each unique-token budget, we fit the parameter-scaling asymptote $L_{\mathrm{MIR},U}(N) = E_U + A_U/N^{\alpha_U}$ using the five model sizes at that budget. See Figure 8 for the fitted curves. Taking $N \to \infty$ in Eq. (14) gives the regularized-baseline infinite-model curve $L_{\mathrm{Reg},\infty}(U) = 0.306 + 2.249\,U^{-0.125}$. For each MIR asymptote E_U, we solve $L_{\mathrm{Reg},\infty}(U_{\mathrm{eq}}) = E_U$ and report $U_{\mathrm{eq}}/U_{\mathrm{MIR}}$. Under this baseline law, MIR consistently improves unique-data efficiency: at 200M–400M unique tokens, the regularized baseline would need about 1.28–1.34× as much unique data to match the MIR infinite-model asymptote. For completeness, we also use the other three scaling laws to calculate the data efficiency ratios. SoftQ gives the most conservative data efficiency ratio at $U = 400$M among all scaling laws. See Appendix B.6 for details.

5 Related Work

Classical Scaling Laws Empirical scaling laws have provided a central tool for predicting language-model loss as a function of model size, data, and compute. Hestness et al. [2017], Rosenfeld et al. [2020] found that deep-learning generalization curves often follow power laws across model and dataset scales. For language modeling, Kaplan et al. [2020] showed that cross-entropy loss scales predictably with parameter count, dataset size, and training compute. Henighan et al. [2020] extended similar power-law behavior to other autoregressive generative domains. Hoffmann et al. [2022] revised the compute-optimal allocation problem and argued that model size and training tokens should be increased at comparable rates, leading to the Chinchilla recipe. These laws are highly effective in the abundant-data setting, but they typically treat processed tokens as fresh samples and therefore do not explicitly distinguish unique data from repeated epochs. This distinction becomes important once the available corpus size, rather than compute, becomes the binding resource.

Data-constrained Pretraining Muennighoff et al. [2023] studied repeated-data training and proposed effective-resource scaling laws that account for the diminishing value of repeated tokens and excess parameters; they found that modest repetition can be close to fresh data, but that the marginal value of repetition eventually decays. Kim et al. [2026a] sharpened this into an infinite-compute, fixed-data viewpoint, showing that simply increasing epochs and parameters can overfit, and that much stronger regularization, especially substantially larger weight decay than standard practice, can improve the best attainable loss. Recent work has also explored data-side and benchmark-driven approaches to this regime. Kim et al. [2026b] generate document-level synthetic rephrases and show that scaling these generations improves validation loss on web text. The NanoGPT Slowrun benchmark [Q Labs, 2026] similarly operationalizes fixed-data, high-compute language modeling by fixing 100M FineWeb [Penedo et al., 2024] tokens and ranking methods by validation loss with no compute limit.

Masked Diffusion Language Model Discrete and masked diffusion language models provide an alternative to left-to-right factorization by corrupting tokens and learning to reverse the corruption process. Sahoo et al. [2024] proposed masked diffusion language models with effective training recipes. Nie et al. [2025], Bie et al. [2025] scaled up the model and data size to train large-scale diffusion language models. In the data-constrained setting, Prabhudesai et al. [2025] and Ni et al. [2025] report that masked diffusion models can outperform autoregressive models under repeated-data training, attributing the gains to factors such as any-order prediction, dense denoising supervision, and implicit Monte Carlo augmentation.

Masking, noising, and denoising objectives Training on corrupted inputs has a long history as a regularization and representation-learning principle. In NLP, BERT popularized masked language modeling for bidirectional representation learning [Devlin et al., 2019], while BART and T5 extended masking and denoising ideas to sequence-to-sequence pretraining through masked-span reconstruction, arbitrary text corruption, and span corruption [Lewis et al., 2020, Raffel et al., 2020]. These objectives use masking as the main pretraining task and often change the architecture or inference interface relative to decoder-only autoregressive language modeling. MIR instead keeps the standard causal next-token objective and autoregressive decoding, using masking only as an auxiliary input perturbation during training. Several works are closer to MIR in spirit because they inject masking or dropout into autoregressive or limited-data training. Zhang et al. [2020] used word and token dropout as data augmentation and regularization in sequence modeling. Zhuang et al. [2025] proposed Mask-Enhanced Autoregressive Prediction, which masks a small fraction of input tokens and then performs standard next-token prediction to improve retrieval and long-context behavior. Wang et al. [2025] masked low-entropy tokens to regularize multi-epoch training on limited domain data.

Closely related is Gao et al. [2025], which observed that random input masking drives data efficiency in dLLMs. Their study focused on how weight decay and input masking separately impact data efficiency. Tuning a 0.6B model, they showed that each technique independently reduces validation loss and narrows the dLLM-AR gap, finding that stronger regularization (0.5 weight decay, 0.1 MLP dropout) or dLLM-style inputs separately make AR models competitive with dLLMs. In contrast, our evaluation of MIR focuses on the interaction between input masking and weight decay. Instead of treating them as independent alternatives, we demonstrate that dLLM-style masking provides a complementary benefit to weight decay and dropout when deployed as an auxiliary loss, further lowering AR validation loss. Moreover, this gain widens as models scale. To quantify this gain, we further developed a new coupled scaling law (SoftQ) specifically for this data-constrained regime, confirming that MIR yields superior scaling behavior and quantifiable data-efficiency gains.

6 Discussion

We study data-constrained, compute-rich pretraining along two axes, regularization and scaling. First, large weight decay substantially reduces dLLM validation loss; MIR, an auxiliary next-token loss on randomly masked inputs, further improves AR model validation loss on top of large weight decay across 72M to 1.4B parameters. Second, the additive Chinchilla law is misspecified in this regime because it decouples model and data size; we propose the SoftQ scaling law, which couples them and fits both our experiments and an independent grid from prior work better than existing alternatives. Our study has several limitations. Experiments span up to 1.4B parameters and 400M unique tokens, small relative to frontier-scale pretraining. We held model architecture and optimizer fixed; varying these could yield further gains. Our protocol also relies on heavy per-cell hyperparameter search; a hyperparameter-transfer recipe for this regime is a natural next step.

7 Acknowledgments

We thank Eric Czech, Hrayr Harutyunyan, and Samip Dahal for helpful discussions and their invaluable feedback. This work was supported in part by funding from the DARPA AIQ program, the Office of Naval Research under grant N00014-23-1-2590, the National Science Foundation under grant No. 2310831, No. 2428059, No. 2435696, No. 2440954, a Michigan Institute for Data Science Propelling Original Data Science (PODS) grant, Two Sigma Investments LP, and LG Management Development Institute AI Research.

References

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.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, DDL Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 10, 2022.

Pablo Villalobos, Anson Ho, Jaime Sevilla, Tamay Besiroglu, Lennart Heim, and Marius Hobbhahn. Will we run out of data? limits of llm scaling based on human-generated data. *arXiv preprint arXiv:2211.04325*, 2024.

Jaime Sevilla and Edu Roldán. Training compute of frontier AI models grows by 4-5x per year, 2024. URL https://epoch.ai/blog/training-compute-of-frontier-ai-models-grows-by-4-5x-per-year. Accessed: 2026-04-29.

Common Crawl. Statistics of Common Crawl Monthly Archives: Crawl Size, 2025. URL https://commoncrawl.github.io/cc-crawl-statistics/plots/crawlsize. Accessed: 2026-04-28.

Niklas Muennighoff, Alexander Rush, Boaz Barak, Teven Le Scao, Nouamane Tazi, Aleksandra Piktus, Sampo Pyysalo, Thomas Wolf, and Colin A Raffel. Scaling data-constrained language models. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 50358–50376. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/9d89448b63ce1e2e8dc7af72c984c196-Paper-Conference.pdf.

Konwoo Kim, Suhas Kotha, Percy Liang, and Tatsunori Hashimoto. Pre-training under infinite compute. In *The Fourteenth International Conference on Learning Representations*, 2026a. URL https://openreview.net/forum?id=ck0aZTAnwK.

Jinjie Ni, Qian Liu, Longxu Dou, Chao Du, Zili Wang, Hang Yan, Tianyu Pang, and Michael Qizhe Shieh. Diffusion language models are super data learners. *arXiv preprint arXiv:2511.03276*, 2025.

Mihir Prabhudesai, Mengning Wu, Amir Zadeh, Katerina Fragkiadaki, and Deepak Pathak. Diffusion beats autoregressive in data-constrained settings. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=W5Ht05jF4c.

Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Gadre, Hritik Bansal, Etash Guha, Sedrick Keh, Kushal Arora, Saurabh Garg, Rui Xin, Niklas Muennighoff, Reinhard Heckel, Jean Mercat, Mayee Chen, Suchin Gururangan, Mitchell Wortsman, Alon Albalak, Yonatan Bitton, Marianna Nezhurina, Amro Abbas, Cheng-Yu Hsieh, Dhruba Ghosh, Josh Gardner, Maciej Kilian, Hanlin Zhang, Rulin Shao, Sarah Pratt, Sunny Sanyal, Gabriel Ilharco, Giannis Daras, Kalyani Marathe, Aaron Gokaslan, Jieyu Zhang, Khyathi Chandu, Thao Nguyen, Igor Vasiljevic, Sham Kakade, Shuran Song, Sujay Sanghavi, Fartash Faghri, Sewoong Oh, Luke Zettlemoyer, Kyle Lo, Alaaeldin El-Nouby, Hadi Pouransari, Alexander Toshev, Stephanie Wang, Dirk Groeneveld, Luca Soldaini, Pang Wei Koh, Jenia Jitsev, Thomas Kollar, Alexandros G. Dimakis, Yair Carmon, Achal Dave, Ludwig Schmidt, and Vaishaal Shankar. Datacomp-lm: In search of the next generation of training sets for language models. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 14200–14282. Curran Associates, Inc., 2024. doi: 10.52202/079017-0455. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/19e4ea30dded58259665db375885e412-Paper-Datasets_and_Benchmarks_Track.pdf.

Anton Lozhkov, Raymond Li, Loubna Ben Allal, Federico Cassano, Joel Lamy-Poirier, Nouamane Tazi, Ao Tang, Dmytro Pykhtar, Jiawei Liu, Yuxiang Wei, Tianyang Liu, Max Tian, Denis Kocetkov, Arthur Zucker, Younes Belkada, Zijian Wang, Qian Liu, Dmitry Abulkhanov, Indraneil Paul, Zhuang Li, Wen-Ding Li, Megan Risdal, Jia Li, Jian Zhu, Terry Yue Zhuo, Evgenii Zheltonozhskii,

Nii Osae Osae Dade, Wenhao Yu, Lucas Krauß, Naman Jain, Yixuan Su, Xuanli He, Manan Dey, Edoardo Abati, Yekun Chai, Niklas Muennighoff, Xiangru Tang, Muhtasham Oblokulov, Christopher Akiki, Marc Marone, Chenghao Mou, Mayank Mishra, Alex Gu, Binyuan Hui, Tri Dao, Armel Zebaze, Olivier Dehaene, Nicolas Patry, Canwen Xu, Julian McAuley, Han Hu, Torsten Scholak, Sebastien Paquet, Jennifer Robinson, Carolyn Jane Anderson, Nicolas Chapados, Mostofa Patwary, Nima Tajbakhsh, Yacine Jernite, Carlos Muñoz Ferrandis, Lingming Zhang, Sean Hughes, Thomas Wolf, Arjun Guha, Leandro von Werra, and Harm de Vries. Starcoder 2 and the stack v2: The next generation, 2024.

Eric J. Michaud. On neural scaling and the quanta hypothesis. *Learning Mechanics*, 2026. URL https://learningmechanics.pub/quanta.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. The language model evaluation harness, 07 2024. URL https://zenodo.org/records/12608602.

William Merrill, Yanhong Li, Tyler Romero, Anej Svete, Caia Costello, Pradeep Dasigi, Dirk Groeneveld, David Heineman, Bailey Kuehl, Nathan Lambert, et al. Olmo hybrid: From theory to practice and back. *arXiv preprint arXiv:2604.03444*, 2026.

Joel Hestness, Sharan Narang, Newsha Ardalani, Gregory Diamos, Heewoo Jun, Hassan Kianinejad, Md Mostofa Ali Patwary, Yang Yang, and Yanqi Zhou. Deep learning scaling is predictable, empirically. *arXiv preprint arXiv:1712.00409*, 2017.

Jonathan S. Rosenfeld, Amir Rosenfeld, Yonatan Belinkov, and Nir Shavit. A constructive prediction of the generalization error across scales. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=ryenvpEKDr.

Tom Henighan, Jared Kaplan, Mor Katz, Mark Chen, Christopher Hesse, Jacob Jackson, Heewoo Jun, Tom B Brown, Prafulla Dhariwal, Scott Gray, et al. Scaling laws for autoregressive generative modeling. *arXiv preprint arXiv:2010.14701*, 2020.

Konwoo Kim, Suhas Kotha, Yejin Choi, Tatsunori Hashimoto, Nick Haber, and Percy Liang. Data-efficient pre-training by scaling synthetic megadocs. *arXiv preprint arXiv:2603.18534*, 2026b.

Q Labs. NanoGPT Slowrun. https://github.com/qlabs-eng/slowrun, 2026. GitHub repository. Accessed: 2026-04-28.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. The fineweb datasets: Decanting the web for the finest text data at scale. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL https://openreview.net/forum?id=n6SCkn2QaG.

Subham S Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. *Advances in Neural Information Processing Systems*, 37:130136–130184, 2024.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, JUN ZHOU, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=KnqiC0znVF.

Tiwei Bie, Maosong Cao, Kun Chen, Lun Du, Mingliang Gong, Zhuochen Gong, Yanmei Gu, Jiaqi Hu, Zenan Huang, Zhenzhong Lan, et al. Llada2.0: Scaling up diffusion language models to 100b. *arXiv preprint arXiv:2512.15745*, 2025.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.703. URL https://aclanthology.org/2020.acl-main.703/.

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.

Huaao Zhang, Shigui Qiu, Xiangyu Duan, and Min Zhang. Token drop mechanism for neural machine translation. In Donia Scott, Nuria Bel, and Chengqing Zong, editors, *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4298–4303, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.379. URL https://aclanthology.org/2020.coling-main.379/.

Xialie Zhuang, Zhikai Jia, Jianjin Li, Zhenyu Zhang, Li Shen, Zheng Cao, and Shiwei Liu. Mask-enhanced autoregressive prediction: Pay less attention to learn more. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu, editors, *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 80516–80532. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/zhuang25b.html.

Jiapeng Wang, Yiwen Hu, Yanzipeng Gao, Haoyu Wang, Shuo Wang, Hongyu Lu, Jiaxin Mao, Wayne Xin Zhao, Junyi Li, and Xiao Zhang. Entropy-guided token dropout: Training autoregressive language models with limited domain data. *arXiv preprint arXiv:2512.23422*, 2025.

Zitian Gao, Haoming Luo, Lynx Chen, Jason Klein Liu, Ran Tao, Joey Zhou, and Bryan Dai. What makes diffusion language models super data learners? *arXiv preprint arXiv:2510.04071*, 2025.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, et al. 2 olmo 2 furious. *arXiv preprint arXiv:2501.00656*, 2024.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, Gaël Liu, Francesco Visin, Kathleen Kenealy, Lucas Beyer, Xiaohai Zhai, Anton Tsitsulin, Robert Busa-Fekete, Alex Feng, Noveen Sachdeva, Benjamin Coleman, Yi Gao, Basil Mustafa, Iain Barr, Emilio Parisotto, David Tian, Matan Eyal, Colin Cherry, Jan-Thorsten Peter, Danila Sinopalnikov, Surya Bhupatiraju, Rishabh Agarwal, Mehran Kazemi, Dan Malkin, Ravin Kumar, David Vilar, Idan Brusilovsky, Jiaming Luo, Andreas Steiner, Abe Friesen, Abhanshu Sharma, Abheesht Sharma, Adi Mayrav Gilady, Adrian Goedeckemeyer, Alaa Saade, Alex Feng, Alexander Kolesnikov, Alexei Bendebury, Alvin Abdagic, Amit Vadi, András György, André Susano Pinto, Anil Das, Ankur Bapna, Antoine Miech, Antoine Yang, Antonia Paterson, Ashish Shenoy, Ayan Chakrabarti, Bilal Piot, Bo Wu, Bobak Shahriari, Bryce Petrini, Charlie Chen, Charline Le Lan, Christopher A. Choquette-Choo, CJ Carey, Cormac Brick, Daniel Deutsch, Danielle Eisenbud, Dee Cattle, Derek Cheng, Dimitris Paparas, Divyashree Shivakumar Sreepathihalli, Doug Reid, Dustin Tran, Dustin Zelle, Eric Noland, Erwin Huizenga, Eugene Kharitonov, Frederick Liu, Gagik Amirkhanyan, Glenn Cameron, Hadi Hashemi, Hanna Klimczak-Plucińska, Harman Singh, Harsh Mehta, Harshal Tushar Lehri, Hussein Hazimeh, Ian Ballantyne, Idan Szpektor, Ivan Nardini, Jean Pouget-Abadie, Jetha Chan, Joe Stanton, John Wieting, Jonathan Lai, Jordi Orbay, Joseph Fernandez, Josh Newlan, Ju yeong Ji, Jyotinder Singh, Kat Black, Kathy Yu, Kevin Hui, Kiran Vodrahalli, Klaus Greff, Linhai Qiu, Marcella Valentine, Marina Coelho, Marvin Ritter, Matt Hoffman, Matthew Watson, Mayank Chaturvedi, Michael Moynihan, Min Ma, Nabila Babar, Natasha Noy, Nathan Byrd, Nick Roy, Nikola Momchev, Nilay Chauhan, Noveen

Sachdeva, Oskar Bunyan, Pankil Botarda, Paul Caron, Paul Kishan Rubenstein, Phil Culliton, Philipp Schmid, Pier Giuseppe Sessa, Pingmei Xu, Piotr Stanczyk, Pouya Tafti, Rakesh Shivanna, Renjie Wu, Renke Pan, Reza Rokni, Rob Willoughby, Rohith Vallu, Ryan Mullins, Sammy Jerome, Sara Smoot, Sertan Girgin, Shariq Iqbal, Shashir Reddy, Shruti Sheth, Siim Põder, Sijal Bhatnagar, Sindhu Raghuram Panyam, Sivan Eiger, Susan Zhang, Tianqi Liu, Trevor Yacovone, Tyler Liechty, Uday Kalra, Utku Evci, Vedant Misra, Vincent Roseberry, Vlad Feinberg, Vlad Kolesnikov, Woohyun Han, Woosuk Kwon, Xi Chen, Yinlam Chow, Yuvein Zhu, Zichuan Wei, Zoltan Egyed, Victor Cotruta, Minh Giang, Phoebe Kirk, Anand Rao, Kat Black, Nabila Babar, Jessica Lo, Erica Moreira, Luiz Gustavo Martins, Omar Sanseviero, Lucas Gonzalez, Zach Gleicher, Tris Warkentin, Vahab Mirrokni, Evan Senter, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, Yossi Matias, D. Sculley, Slav Petrov, Noah Fiedel, Noam Shazeer, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Jean-Baptiste Alayrac, Rohan Anil, Dmitry, Lepikhin, Sebastian Borgeaud, Olivier Bachem, Armand Joulin, Alek Andreev, Cassidy Hardin, Robert Dadashi, and Léonard Hussenot. Gemma 3 technical report, 2025. URL https://arxiv.org/abs/2503.19786.

Team Olmo, Allyson Ettinger, Amanda Bertsch, Bailey Kuehl, David Graham, David Heineman, Dirk Groeneveld, Faeze Brahman, Finbarr Timbers, Hamish Ivison, et al. Olmo 3. *arXiv preprint arXiv:2512.13961*, 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Ailin Huang, Ang Li, Aobo Kong, Bin Wang, Binxing Jiao, Bo Dong, Bojun Wang, Boyu Chen, Brian Li, Buyun Ma, et al. Step 3.5 flash: Open frontier-level intelligence with 11b active parameters. *arXiv preprint arXiv:2602.10604*, 2026.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Shengding Hu, Yuge Tu, Xu Han, Ganqu Cui, Chaoqun He, Weilin Zhao, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Xinrong Zhang, Zhen Leng Thai, Chongyi Wang, Yuan Yao, Chenyang Zhao, Jie Zhou, Jie Cai, Zhongwu Zhai, Ning Ding, Chao Jia, Guoyang Zeng, dahai li, Zhiyuan Liu, and Maosong Sun. MiniCPM: Unveiling the potential of small language models with scalable training strategies. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=3X2L2TFr0f.

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

Appendix

A Experiment Details **15**

A.1 Compute, Architecture, and Scaling Ladder 15

A.2 Data and Evaluation Splits 16

A.3 AR Training Recipe 16

A.4 dLLM Baseline Protocol 16

A.5 Multi-epoch AR Epoch Search 17

A.6 Strongly Regularized Baseline Search 17

A.7 MIR Hyperparameter Tuning 18

A.8 Auxiliary Experimental Results 19

A.9 Dataset Licenses 19

B Details of the Scaling-Law Analysis **19**

B.1 Setup, Notation, and Fitting Objective 20

B.2 Candidate Scaling Laws 20

B.3 Fit Quality and Model Selection 21

B.4 Fitted Constants and Selected SoftQ Law 21

B.5 MIR Data-Efficiency Calculation 22

B.6 Sensitivity Analyses 23

B.7 Additional Visualizations 24

C Why Masking Reduces Memorization: A Toy Model **26**

C.1 Proof of Theorem C.4 29

C.2 Proof of Theorem C.6 32

C.3 Proof of Corollary C.7 33

C.4 Proof of Theorem C.8 34

D Derivation of Quanta Scaling Law **36**

A Experiment Details

This appendix describes the compute setup, architecture ladder, data splits, training recipes, hyperparameter searches, and auxiliary experimental results. See the full data generation and training code in Github. See the Wandb logs at WandB.

A.1 Compute, Architecture, and Scaling Ladder

All experiments can run on eight 80GB SXM H100 GPUs. The longest AR model run completes in under 24 hours. The longest dLLM model run completes in under 48 hours.

We use a Llama-style decoder-only transformer [Grattafiori et al., 2024] with QK norm and interleaved global-local self-attention as the model architecture. Compared to the architecture used in Kim et al. [2026a], we additionally use QK norm and interleaved local and global attention. QK norm is

Table 3: Scaling Ladder Details.

k	Layers	Embed dim	MLP dim	Heads	Head dim	Model size
0.5	6	512	1536	8	64	71,965,952
0.75	9	768	2048	12	64	140,983,680
1.0	12	1024	2816	16	64	257,190,400
1.5	18	1536	4096	24	64	664,200,960
2.0	24	2048	5632	32	64	1,439,273,984

widely used [OLMo et al., 2024, Team et al., 2025, Olmo et al., 2025, Yang et al., 2025] in recent open-source large language models to stabilize pretraining, and interleaved local and global attention is also widely used [Team et al., 2025, Olmo et al., 2025, Huang et al., 2026] to reduce compute and reduce KV cache size. We use the GPT-2 [Radford et al., 2019] tokenizer with one extra [MASK] token for random masking. The vocabulary size is 50258.

We follow the scaling ladder

$$\text{ScalingLadder}(k) = (kW_1, kL_1, S_1, B_1),$$

where $W_1 = 1024$ is the embedding dimension when $k = 1$, $L_1 = 12$ is the number of layers when $k = 1$, $S_1 = 2048$ is the sequence length, $B_1 = 128$ is the total batch size, and $k \in \{0.5, 0.75, 1, 1.5, 2\}$. Across the scaling ladder, the attention head dimension is fixed at 64, while the depth, embedding dimension, MLP dimension, and number of attention heads increase with scale. The resulting models span from 71,965,952 parameters to 1,439,273,984 parameters. Table 3 summarizes the full architecture ladder.

A.2 Data and Evaluation Splits

Following Kim et al. [2026a], we use DCLM-POOL [Li et al., 2024], an open-source pretraining dataset containing 240T tokens. We use the DCLM subset generated by Kim et al. [2026a] to construct datasets with 100M, 200M, 300M, and 400M unique training tokens. Each smaller-budget dataset is a subset of the corresponding larger-budget dataset. We always use the same evaluation dataset, which contains 10M tokens from DCLM.

We also use Stack-V2 [Lozhkov et al., 2024] to evaluate whether masked-input regularization is beneficial for pretraining on code data. The corresponding validation losses are reported in Table 8.

A.3 AR Training Recipe

Unless stated otherwise, AR experiments use the AdamW optimizer [Loshchilov and Hutter, 2019] with $\beta_1 = 0.9$, $\beta_2 = 0.95$, and $\epsilon = 10^{-8}$. This config is adopted from Kim et al. [2026a]. For all AR model pretraining, we use the Warmup-stable-decay (WSD) [Hu et al., 2024] learning-rate schedule with 1% of the total steps for linear warmup and 10% of the total steps for warmdown. We set dropout rate to be 0.1.

A.4 dLLM Baseline Protocol

For dLLM pretraining, we adopt the config used in Prabhudesai et al. [2025]: batch size 256, sequence length 2048, learning rate schedule with peak 2×10^{-4}, minimum 2×10^{-5}, 1% warmup, cosine decay, weight decay 0.1, and gradient clipping of 1.0. For the number of epochs, we adopt the optimal values reported in Prabhudesai et al. [2025]: 500 epochs for the 257M and 664M models and 800 epochs for the 140M model. We calculate validation loss after each epoch and report the lowest value. The 140M model achieves its lowest validation loss 3.646694 at epoch 789, the 257M model achieves 3.602763 at epoch 483, and the 664M model achieves 3.680272 at epoch 141. We set dropout rate to be 0.1.

Table 4 reports the DCLM 100M validation losses for the AR and dLLM recipes at the three model sizes where we run dLLM pretraining. The strongly regularized dLLM uses the AR-tuned weight decay while keeping the other dLLM hyperparameters fixed to the protocol above.

Table 4: DCLM 100M validation loss for AR and dLLM recipes at different model sizes. For AR recipes, we report final validation loss; for dLLM recipes, we report the best across-epoch validation loss.

	Model size		
Recipe	140M	257M	664M
Multi-Epoch dLLM	3.646694	3.602763	3.680272
Multi-Epoch AR	3.945268	3.879782	3.821800
Strongly Regularized dLLM	3.579445	3.483598	3.387994
Strongly Regularized AR	3.471395	3.422107	3.367138
MIR	**3.468458**	**3.404833**	**3.332668**

A.5 Multi-epoch AR Epoch Search

We search for the best number of epochs for each model size for multi-epoch AR. As shown in Figure 4, 16 epochs is the best for 140M, 8 epochs is the best for 257M, and 32 epochs is the best for 664M.

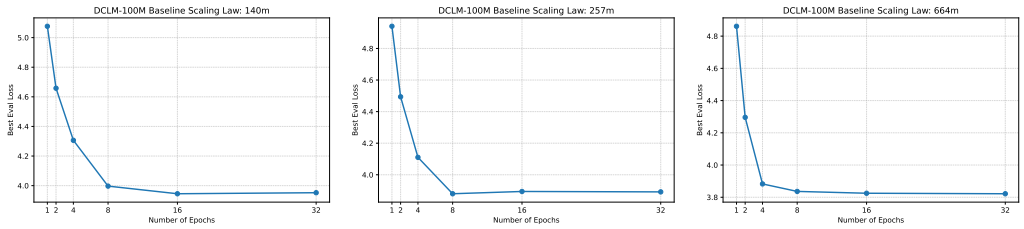


Figure 4: Validation loss vs. number of epochs. Weight decay is fixed to 0.1, peak learning rate is fixed to 2e-4. Left: model size 140M; Middle: model size 257M; Right: model size 664M.

A.6 Strongly Regularized Baseline Search

The strongly regularized baseline sweeps are conducted in the data-constrained DCLM setting described in the main text. We run separate searches at unique-data budgets of 100M, 200M, 300M, and 400M tokens. Within each budget, we use the same training and evaluation datasets across all model scales so that differences in performance can be attributed to model size and training objective rather than differences in data exposure.

We tune the optimization settings separately for each model scale and data budget. The search space consists of the number of training epochs, weight decay, and learning rate. In general, larger models prefer fewer epochs and stronger weight decay, while the selected learning rates remain in the range of 10^{-3} to 10^{-2}. We describe the full 100M sweeps first, then append the larger-budget searches used in the scaling-law analysis.

72M model. We search over epochs $\{16, 32, 64\}$, weight decay $\{0.4, 0.8, 1.6\}$, and learning rate $\{10^{-3}, 3 \times 10^{-3}, 10^{-2}\}$. We additionally run a refined sweep over epochs $\{16, 32, 64\}$, weight decay $\{0.1, 0.2, 0.4\}$, and learning rate $\{3 \times 10^{-3}, 10^{-2}, 3 \times 10^{-2}\}$. The best configuration is $(32, 0.4, 10^{-2})$.

140M model. We first search over epochs $\{8, 16, 32\}$, weight decay $\{0.8, 1.6, 3.2\}$, and learning rate $\{3 \times 10^{-4}, 10^{-3}, 3 \times 10^{-3}\}$. We then run an additional sweep over epochs $\{16, 32, 64\}$, weight decay $\{0.2, 0.4, 0.8, 1.6\}$, and learning rate $\{10^{-3}, 3 \times 10^{-3}, 10^{-2}, 3 \times 10^{-2}\}$. The best configuration is $(32, 0.8, 3 \times 10^{-3})$.

257M model. We search over epochs $\{8, 16, 32\}$, weight decay $\{0.8, 1.6, 3.2\}$, and learning rate $\{3 \times 10^{-4}, 10^{-3}, 3 \times 10^{-3}\}$. The best configuration is $(16, 1.6, 10^{-3})$.

664M model. We search over epochs $\{8, 16, 32\}$, weight decay $\{0.8, 1.6, 3.2\}$, and learning rate $\{3 \times 10^{-4}, 10^{-3}, 3 \times 10^{-3}\}$. The best configuration is $(16, 1.6, 10^{-3})$.

1.4B model. We search over epochs $\{4, 8, 16\}$, weight decay $\{1.6, 3.2, 6.4\}$, and learning rate $\{3 \times 10^{-4}, 10^{-3}, 3 \times 10^{-3}\}$. The best configuration is $(16, 3.2, 10^{-3})$.

Table 5 summarizes the selected hyperparameters at each scale.

Table 5: Best strongly regularized hyperparameter configuration in the 100M unique-token setting.

Model size	Best (epochs, weight decay, lr)
72M	$(32, 0.4, 10^{-2})$
140M	$(32, 0.8, 3 \times 10^{-3})$
257M	$(16, 1.6, 10^{-3})$
664M	$(16, 1.6, 10^{-3})$
1.4B	$(16, 3.2, 10^{-3})$

Table 6 summarizes the selected hyperparameters for the larger unique-data budgets used in the scaling-law analysis. For 200M and 400M unique tokens, we run budget-specific sweeps. For the intermediate 300M budget, we evaluate candidate configurations inherited from the selected 200M and 400M settings at each model scale. The longest runs take around 2 hours at 200M, 6 hours at 300M, and 8 hours at 400M on 8 H100s.

Table 6: Selected strongly regularized hyperparameter configurations for the larger unique-data budgets. Each entry is (epochs, weight decay, lr).

Unique data	Model size	Best (epochs, weight decay, lr)
200M	72M	$(64, 0.2, 10^{-2})$
200M	140M	$(32, 0.4, 3 \times 10^{-3})$
200M	257M	$(16, 0.8, 10^{-3})$
200M	664M	$(16, 1.6, 10^{-3})$
200M	1.4B	$(16, 1.6, 10^{-3})$
300M	72M	$(64, 0.1, 10^{-2})$
300M	140M	$(64, 0.2, 3 \times 10^{-3})$
300M	257M	$(32, 0.8, 10^{-3})$
300M	664M	$(32, 0.8, 10^{-3})$
300M	1.4B	$(32, 1.6, 10^{-3})$
400M	72M	$(64, 0.1, 10^{-2})$
400M	140M	$(64, 0.2, 3 \times 10^{-3})$
400M	257M	$(32, 0.4, 10^{-3})$
400M	664M	$(32, 0.8, 10^{-3})$
400M	1.4B	$(32, 0.8, 10^{-3})$

A.7 MIR Hyperparameter Tuning

In MIR, for each sequence x, a mask ratio r is sampled from $\mathrm{Unif}(r_{\min}, r_{\max})$, then for each position $t \in [0, T-1]$, we use a Bernoulli random variable with success probability r to decide whether to mask x_t. Denote the masked version as $\widetilde{x}$. We optimize

$$\mathcal{L} = \mathcal{L}_{\mathrm{NTP}}(x) + \lambda \mathcal{L}_{\mathrm{NTP}}(\widetilde{x}).$$

We tune the values of $r_{\min}, r_{\max}, \lambda$ using the 1.4B model and DCLM 100M. See the results in Figure 5. The selected values are $r_{\min} = 0, r_{\max} = 0.5, \lambda = 0.4$. We also tried

$$\mathcal{L} = (1-\lambda)\mathcal{L}_{\mathrm{NTP}}(x) + \lambda \mathcal{L}_{\mathrm{NTP}}(\widetilde{x}),$$

but its performance was slightly worse than $\mathcal{L}_{\mathrm{NTP}}(x) + \lambda \mathcal{L}_{\mathrm{NTP}}(\widetilde{x})$.

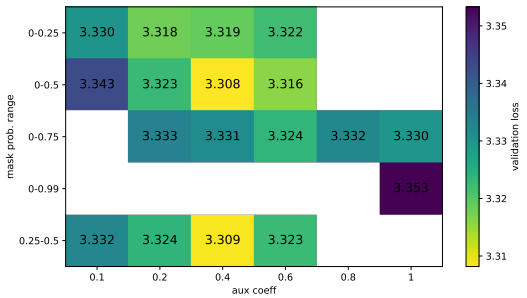


Figure 5: Tuning the mask ratio bounds ($r_{\min}, r_{\max}$) and regularization coefficient λ.

A.8 Auxiliary Experimental Results

Table 7 reports the best evaluation loss across model scales in the 100M unique-token setting. Table 8 reports the Stack-V2 validation losses.

Table 7: Best evaluation loss across model scales in the 100M unique-token setting (seed 42).

Recipe	72M	140M	257M	664M	1.4B
Single-epoch	4.866105	4.960820	5.025738	5.019738	5.302995
Strongly Regularized Recipe (Baseline)	3.615903	3.471395	3.422107	3.367138	3.339578
MIR	3.613621	3.468458	3.404833	3.332668	3.308170

Table 8: Validation loss on the Stack-V2 100M unique token dataset. MIR consistently outperforms the strongly regularized baseline across all model scales.

Recipe	72M	140M	257M	664M	1.4B
Regularized Baseline	1.064	1.020	1.005	0.996	0.983
MIR	**1.054**	**1.012**	**0.985**	**0.988**	**0.967**

A.9 Dataset Licenses

Dataset	Use in this paper	Version / URL	License and terms
DCLM-Pool	Natural-language pretraining and validation data.	Link	CC BY 4.0. DCLM-Pool is derived from Common Crawl and is also subject to the Common Crawl Terms of Use. We cite the original DCLM paper and do not redistribute the raw dataset.
The Stack v2	Code-heavy pretraining data for the Stack-v2 experiments.	Link. Version: 2.1.0	No single dataset-wide content license; Hugging Face lists the license as "other". The dataset contains source code from repositories with various original licenses. User must comply with upstream licenses, including attribution clauses where relevant, the Stack-v2 access terms, Software Heritage principles for language-model training, and validated removal-request updates. We do not redistribute raw Stack-v2 files.

Table 9: Existing datasets used in this paper and their licenses or terms of use.

B Details of the Scaling-Law Analysis

This appendix gives the full scaling-law definitions, fitting protocol, fitted constants, residual diagnostics, and the plots moved out of the main text.

B.1 Setup, Notation, and Fitting Objective

We use N for model size and U for unique training tokens, both measured in billions. The baseline grid contains 5 model sizes $\{72\text{M}, 140\text{M}, 257\text{M}, 664\text{M}, 1.4\text{B}\}$ and 4 unique-token budgets $\{100\text{M}, 200\text{M}, 300\text{M}, 400\text{M}\}$, for 20 strongly regularized baseline points. The external grid is provided in Kim et al. [2026a]. For our dataset and the external one, the repetition variable in the Muennighoff-style law is $R_D = \text{epochs} - 1$, where the epoch count is taken from the best configuration or run identifier.

For Chinchilla, Quanta, and SoftQ, we minimize the Approach-3-style objective of Hoffmann et al. [2022]:

$$\min_{\theta} \sum_{i} \text{Huber}_{10^{-3}} \left(\log \widehat{L}_\theta(N_i, U_i) - \log L_i \right). \tag{5}$$

All reported RMSE, MAE, and RSS values are computed afterward on the raw validation-loss scale. AIC is the SSE-based Gaussian criterion

$$\text{AIC} = n \log(\text{RSS}/n) + 2k, \tag{6}$$

where constants independent of the model are omitted. Since the fitted objective is Huber loss on log residuals, this AIC should be read as a common raw-loss summary criterion rather than the exact likelihood optimized during fitting.

B.2 Candidate Scaling Laws

The Chinchilla law is

$$L_{\text{Ch}}(N, U) = E + \frac{A}{N^\alpha} + \frac{B}{U^\beta}. \tag{7}$$

Its additive form implies a data-independent model-size gap: $L_{\text{Ch}}(N_1, U) - L_{\text{Ch}}(N_2, U)$ does not depend on U. Figure 7 shows that the observed DCLM baseline gaps vary with the unique-token budget.

The Quanta-motivated joint law is

$$L_{\text{Q}}(N, U) = E + \left(\frac{A}{N} + \frac{B}{U^{1/(1+\alpha)}} \right)^\alpha. \tag{8}$$

It couples the parameter and data axes before applying the outer power, so the marginal value of additional parameters depends on the available data.

The Muennighoff-style law replaces raw resources by effective resources:

$$L_{\text{M}}(N, U, R_D) = E + \frac{A}{(N')^\alpha} + \frac{B}{(D')^\beta}, \tag{9}$$

$$D' = U + U R_D^\star \left(1 - \exp[-R_D / R_D^\star]\right), \tag{10}$$

$$N' = U_N + U_N R_N^\star \left(1 - \exp[-R_N / R_N^\star]\right). \tag{11}$$

Given the base Chinchilla coefficients, we compute the one-epoch optimal parameter count

$$N_{\text{opt}}(U) = \left(\frac{\alpha A}{\beta B} \right)^{1/\alpha} U^{\beta/\alpha}, \tag{12}$$

then set $U_N = \min\{N, N_{\text{opt}}(U)\}$ and $R_N = N/U_N - 1$. It has seven parameters to fit: $\{A, B, E, \alpha, \beta, R_N^\star, R_D^\star\}$. Our main comparison uses a dataset-adapted two-stage protocol: fit the base Chinchilla coefficients on the relevant split, then fit only $R_N^\star$ and $R_D^\star$. This is the appropriate comparison if the goal is to evaluate the effective-resource functional form on our loss scale. The literal fixed-C4 coefficients from Muennighoff et al. [2023] are included as an ablation in Table 19.

SoftQ is

$$L_{\text{SoftQ}}(N, U) = E + \left(A N^{-\rho} + B U^{-\rho/(1+\alpha)} \right)^{\alpha/\rho}. \tag{13}$$

When $\rho = 1$, this reduces to the Quanta law. The parameter ρ controls the softness of the transition between the parameter-limited and data-limited regimes.

B.3 Fit Quality and Model Selection

Table 10: Full fit on all 20 strongly regularized baseline points. Lower is better.

Law	# params	RMSE	MAE	AIC
Chinchilla	5	0.026528	0.018016	-135.18
Quanta	4	0.012517	0.008889	-167.23
Muennighoff	7	0.023345	0.017130	-136.29
SoftQ	5	**0.008015**	**0.005204**	**-183.06**

Table 11: Fit on the DCLM 100M/200M/300M baseline points and evaluation on the held-out 400M points.

Law	Train RMSE	Train MAE	Held-out RMSE	Held-out MAE
Chinchilla	0.024636	0.016223	0.031063	0.025396
Quanta	0.012430	0.008853	0.014975	0.012073
Muennighoff	0.023208	0.016216	0.032519	0.027111
SoftQ	**0.008850**	**0.005502**	**0.005955**	**0.004708**

Table 12: Held-out residuals on DCLM 400M, predicted minus observed.

Law	72M	140M	257M	664M	1.4B
Chinchilla	-0.060050	-0.024451	-0.011210	+0.017412	+0.013857
Quanta	+0.028961	+0.011817	-0.009568	-0.004273	-0.005744
Muennighoff	-0.061890	-0.025911	-0.011861	+0.018631	+0.017262
SoftQ	-0.002392	+0.001378	-0.008994	-0.001468	-0.009308

SoftQ has the best aggregate held-out RMSE and MAE. It is closest to zero on four of the five held-out model sizes; Quanta is slightly closer at the 1.4B point.

Table 13: Full fit on the regularized-baseline points provided by Kim et al. [2026a].

Law	# params	RMSE	MAE	AIC
Chinchilla	5	0.040412	0.025554	-92.68
Quanta	4	0.023750	0.014726	-111.69
Muennighoff	7	0.032989	0.022119	-95.17
SoftQ	5	**0.007854**	**0.005955**	**-145.10**

B.4 Fitted Constants and Selected SoftQ Law

See Table 14, 15, and 16 for the fitted constants of each scaling law in each scenario. Specifically, on the full DCLM grid, the fitted SoftQ law is

$$\boxed{L_{\mathrm{SoftQ}}(N,U) = 0.30565 + \left(39.2962\,N^{-0.79608} + 92.4362\,U^{-0.69676}\right)^{0.17906}} \tag{14}$$

with N and U in billions. We therefore use Eq. (14) as the regularized-baseline law for the MIR data-efficiency calculation.

Table 14: Fitted constants on all 20 DCLM baseline points. For Muennighoff, A, α, B, β, E are the first-stage Chinchilla coefficients.

Law	A	α	B	β	ρ	E	Extra
Chinchilla	0.1294	0.5167	0.5357	0.2924	–	2.1116	–
Quanta	242.5882	0.1354	564.4767	–	–	0.2283	–
Muennighoff	0.1294	0.5167	0.5357	0.2924	–	2.1116	$R_N^\star = 31.39$, $R_D^\star = 0.024$
SoftQ	39.2962	0.1425	92.4362	–	0.7961	0.3056	–

Table 15: Fitted constants for the held-out extrapolation experiment, trained only on the DCLM 100M/200M/300M points.

Law	A	α	B	β	ρ	E	Extra
Chinchilla	0.1363	0.4788	0.9823	0.1926	–	1.6310	–
Quanta	799.5772	0.1263	1769.1342	–	–	5.4×10^{-8}	–
Muennighoff	0.1363	0.4788	0.9823	0.1926	–	1.6310	$R_N^\star = 90.51$, $R_D^\star = 0.008$
SoftQ	128.7280	0.1287	295.7854	–	0.7853	1.9×10^{-6}	–

Table 16: Fitted constants on the external grid provided by Kim et al. [2026a].

Law	A	α	B	β	ρ	E	Extra
Chinchilla	0.0543	1.1551	0.3659	0.4594	–	2.6590	–
Quanta	0.1342	0.4959	0.4205	–	–	2.3197	–
Muennighoff	0.0543	1.1551	0.3659	0.4594	–	2.6590	$R_N^\star = 2.13$, $R_D^\star = 0.096$
SoftQ	0.0613	0.5905	0.2565	–	1.4468	2.4360	–

B.5 MIR Data-Efficiency Calculation

Table 17: MIR parameter-scaling fits used for data-efficiency estimation.

MIR unique data	A_U	α_U	MIR asymptote E_U
100M	0.03829	0.82186	3.27997
200M	0.13293	0.49592	2.95596
300M	0.13939	0.51307	2.83953
400M	0.15617	0.51006	2.74826

Using the full-DCLM SoftQ fit, the baseline infinite-model curve is

$$L_{\mathrm{Reg},\infty}(U) = 0.30565 + 2.24905\, U^{-0.12476},$$

where U is in billions. Solving $L_{\mathrm{Reg},\infty}(U_{\mathrm{eq}}) = E_U$ gives the data-efficiency ratios reported in Table 18.

Table 18: MIR unique-data efficiency relative to the strongly regularized baseline, using SoftQ to model the baseline infinite-model curve.

MIR unique data	MIR asymptote E_U	Baseline-equivalent U_{eq}	Data efficiency
100M	3.27997	106.4M	1.06×
200M	2.95596	268.2M	1.34×
300M	2.83953	384.5M	1.28×
400M	2.74826	515.9M	1.29×

B.6 Sensitivity Analyses

The original Muennighoff paper fixes the base Chinchilla coefficients to a C4-calibrated law and fits only $R_N^\star, R_D^\star$. Because those base coefficients are on a different corpus and loss scale, they are not the main comparison in this paper. Table 19 reports the literal fixed-C4 variant for completeness.

Table 19: Literal fixed-C4 Muennighoff variant. This fixes the base Chinchilla law to the coefficients from Muennighoff et al. [2023] and fits only $R_N^\star, R_D^\star$.

Dataset / split	RMSE	MAE	AIC	Notes
DCLM full fit	0.060257	0.047360	-108.37	$R_N^\star = 119.82$, $R_D^\star = 9.995$
DCLM held-out 400M	0.071734	0.064728	–	train RMSE $= 0.055268$
Kim et al full fit	0.120231	0.098342	-63.79	$R_N^\star = 10^8$, $R_D^\star = 0.927$

Table 20: MIR data efficiency under each fitted regularized-baseline law. We fit each baseline law on the same 20 DCLM regularized points. U_{eq} is the amount of unique data the corresponding regularized-baseline infinite-model curve needs to match the MIR asymptote E_U.

	Chinchilla		Quanta		Muennighoff		SoftQ	
MIR U	U_{eq}	Eff.	U_{eq}	Eff.	U_{eq}	Eff.	U_{eq}	Eff.
100M	69.5M	0.70×	115.0M	1.15×	92.4M	0.92×	106.4M	1.06×
200M	211.1M	1.06×	294.7M	1.47×	280.6M	1.40×	268.2M	1.34×
300M	350.6M	1.17×	424.9M	1.42×	466.1M	1.55×	384.5M	1.28×
400M	554.3M	1.39×	572.7M	1.43×	737.0M	1.84×	515.9M	1.29×

The main paper reports MIR data efficiency using SoftQ because it is the selected baseline law by full-fit AIC, held-out prediction, and the external check using data from Kim et al. [2026a]. For completeness, we also compute the same quantity under the Chinchilla, Quanta, and Muennighoff-style laws fitted on the full DCLM strongly regularized baseline grid. The Chinchilla, Quanta, and SoftQ fits use the Approach-3 log-Huber objective in Eq. (5). The Muennighoff-style fit uses the two-stage protocol described above: fit the base Chinchilla coefficients on the same DCLM grid, then hold those coefficients fixed and fit only $R_N^\star$ and $R_D^\star$.

For each law, we define the regularized-baseline infinite-model curve $L_{\text{Reg},\infty}(U)$ and solve $L_{\text{Reg},\infty}(U_{\text{eq}}) = E_U$, where E_U is the MIR parameter-scaling asymptote in Table 17. The data efficiency ratio is $U_{\text{eq}}/U_{\text{MIR}}$. For Chinchilla, Quanta, and SoftQ, the infinite-model curves are obtained by taking $N \to \infty$. For the Muennighoff-style law, we take both $N \to \infty$ and the saturated repeated-data limit $R_D \to \infty$, giving

$$L_{\text{M},\infty}(U) = E + \frac{A}{\{(1+R_N^\star)N_{\text{opt}}(U)\}^\alpha} + \frac{B}{\{(1+R_D^\star)U\}^\beta},$$

where

$$N_{\text{opt}}(U) = \left(\frac{\alpha A}{\beta B}\right)^{1/\alpha} U^{\beta/\alpha}.$$

With the fitted constants in Table 14, the resulting one-dimensional curves are

$$\begin{aligned}
L_{\text{Ch},\infty}(U) &= 2.11164 + 0.53575\,U^{-0.29241},\\
L_{\text{Q},\infty}(U) &= 0.22834 + 2.35787\,U^{-0.11924},\\
L_{\text{M},\infty}(U) &= 2.11164 + 0.58227\,U^{-0.29241},\\
L_{\text{SoftQ},\infty}(U) &= 0.30565 + 2.24905\,U^{-0.12476},
\end{aligned}$$

where U is measured in billions of unique tokens.

The alternative-law estimates vary substantially. In particular, the additive Chinchilla fit gives a sub-unity ratio at 100M because its infinite-model curve already predicts a loss below the MIR asymptote at 100M, which is another symptom of the decoupled law being misspecified in this

regime. Quanta and Muennighoff generally produce larger ratios than SoftQ at 200M–400M, while SoftQ gives the most conservative estimates among the coupled laws that also passed the held-out and external-data checks. For this reason, we keep the SoftQ-based ratios as the main-text estimate and report the other laws only as sensitivity analyses. We also observe that the data efficiency ratio difference under different scaling laws generally shrinks as the unique data size increases.

B.7 Additional Visualizations

Figure 6 gives the absolute-loss view of the Chinchilla and SoftQ fits. Figures 7 and 8 provide additional views of the baseline and MIR scaling results.

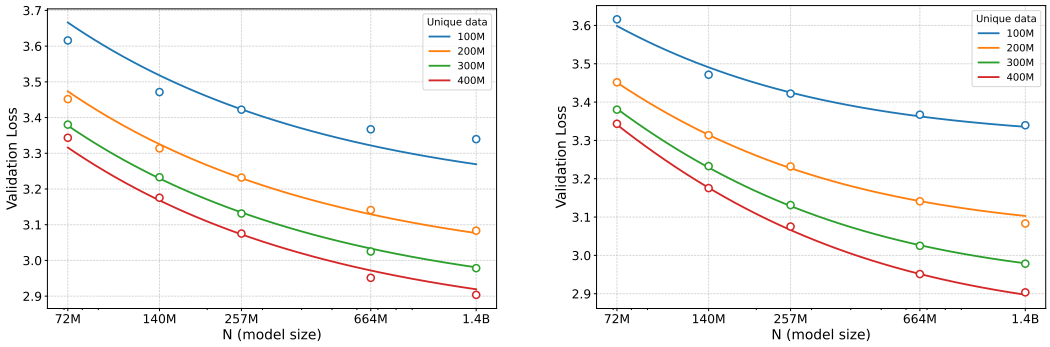


Figure 6: Absolute-loss view of the fitted Chinchilla and SoftQ laws on the 20 strongly regularized baseline points. Left: Chinchilla fit. Right: SoftQ fit. Points are observed validation losses and curves are model predictions.

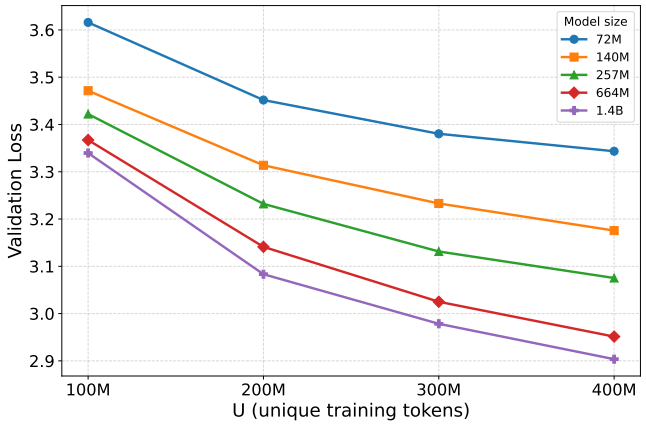


Figure 7: Regularized baseline validation loss as a function of unique training data size U across five model sizes. The changing separation between curves contradicts the data-independent model-size gap implied by the additive Chinchilla form.

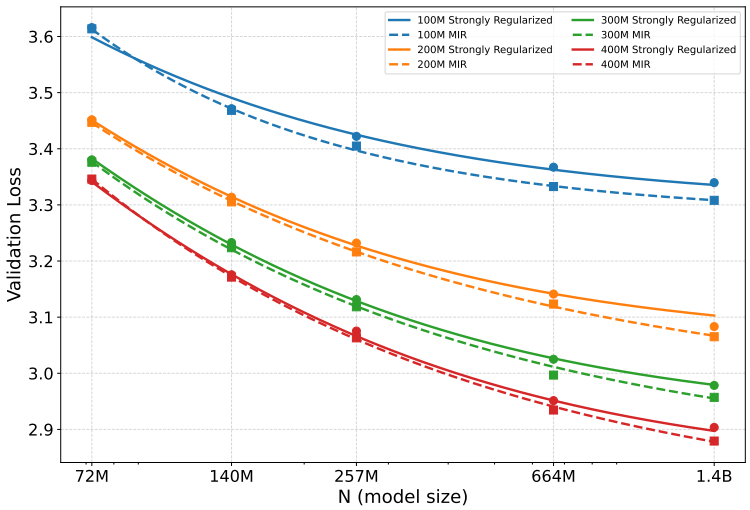


Figure 8: Scaling curves across four unique-data budgets for the strongly regularized baseline and MIR. MIR improves validation loss at most model-size and data-budget pairs, and the asymptotic fits in Table 17 quantify the infinite-model limit.

C Why Masking Reduces Memorization: A Toy Model

This section gives a toy model for the intuition stated in Section 3.3 of the main text: masked-input regularization reduces validation loss by reducing dependence on context-specific components (noise) and preserving a signal through generalizable components. The intention in this section is not to model transformer pretraining in full, but to isolate one mechanism that becomes important in the data-constrained, compute-rich regime.

We decompose each training sequence into three parts: a context-specific component, a generalizable component, and an output token. The context-specific component can identify individual training examples and therefore enables memorization. The generalizable component contains predictive information that also appears in validation examples. A sufficiently large model can fit the finite training set through the context-specific component alone; however, this fit does not transfer to validation examples with unseen context-specific components. Masking changes this because it can sometimes hide the context-specific component while leaving the generalizable component visible. On such masked inputs, prediction through memorization is unavailable, so the model is encouraged to learn predictive patterns from the generalizable component. Specifically, we introduce the following context-specific noise model.

Definition C.1 (Context-Specific Noise Model)**.** The training set consists of examples

$$(C_i, S_i, Y_i), \;\; i = 1, \ldots, n,$$

where C_i is a context-specific component, $S_i \in \mathbb{R}^d$ is a generalizable component, and $Y_i \in \{-1, +1\}$ is the output token to be predicted. In this model, we consider binary prediction for simplicity and clarity. We assume

$$\|S_i\|_2 \le B.$$

The population validation distribution has the same joint distribution of (S, Y), but its context-specific components are unseen during training.

Let

$$\mu := \mathbb{E}[YS] \in \mathbb{R}^d, \;\; \Sigma := \mathbb{E}[SS^\top] \in \mathbb{R}^{d\times d},$$

and assume $\mu \neq 0$. On the finite training set, define

$$\widehat{\mu} := \frac{1}{n}\sum_{i=1}^{n} Y_i S_i, \;\; \widehat{\Sigma} := \frac{1}{n}\sum_{i=1}^{n} S_i S_i^\top.$$

Let $\mathbf{S} \in \mathbb{R}^{n\times d}$ be the matrix with i-th row $S_i^\top$, and let $Y = (Y_1, \ldots, Y_n)^\top$.

Definition C.2 (Clean and MIR Objectives)**.** For model size m, let

$$\phi_i = \phi_m(C_i) \in \mathbb{R}^m, \;\; \Phi_m = (\phi_1, \ldots, \phi_n)^\top \in \mathbb{R}^{n\times m}, \;\; G_m = \Phi_m \Phi_m^\top.$$

The model prediction score on example i is

$$\theta_i = \phi_i^\top w + b^\top S_i,$$

where $w \in \mathbb{R}^m$ models the context-specific memorization component and $b \in \mathbb{R}^d$ models the generalizable component. We consider squared and logistic losses,

$$\ell_{\mathrm{sq}}(y, \theta) = \frac{1}{2}(y - \theta)^2, \;\; \ell_{\log}(y, \theta) = \log(1 + \exp(-y\theta)).$$

To model masking, let $r \in [0, 1]$ be a sampled mask ratio. Conditional on r, let $V_{C,i}, V_{S,i} \in \{0, 1\}$ be independent visibility indicators with

$$\mathbb{P}(V_{C,i} = 1 \,|\, r) = \mathbb{P}(V_{S,i} = 1 \,|\, r) = 1 - r.$$

The masked prediction score on example i is then

$$V_{C,i}\phi_i^\top w + V_{S,i} b^\top S_i.$$

Thus masking may remove the context-specific component, the generalizable component, both, or neither. For loss $\ell \in \{\ell_{\mathrm{sq}}, \ell_{\log}\}$, the clean objective is

$$\widehat{J}^{(m)}_{\ell,\mathrm{clean}}(w,b) = \frac{1}{n}\sum_{i=1}^{n} \ell\big(Y_i, \phi_i^\top w + b^\top S_i\big) + \frac{\rho_w}{2n}\|w\|_2^2 + \frac{\rho_b}{2}\|b\|_2^2,$$

and the MIR objective is

$$\widehat{J}^{(m)}_{\ell,\mathrm{MIR}}(w,b) = \widehat{J}^{(m)}_{\ell,\mathrm{clean}}(w,b) + \frac{\lambda}{n}\sum_{i=1}^{n} \mathbb{E}_M\left[\ell\big(Y_i, V_{C,i}\phi_i^\top w + V_{S,i} b^\top S_i\big)\right],$$

where the expectation is over the masking randomness.

Assumption C.3 (Growing Context-Specific Capacity)**.** For every m, $G_m \succ 0$. Let $a_m := \lambda_{\min}(G_m)$, then $a_m \to \infty$ as $m \to \infty$.

Assumption C.3 captures the data-constrained, compute-rich regime: the number of training examples is fixed, while the capacity of the context-specific memorization component grows. In this regime, for any fixed vector of prediction scores on the finite training set, the context-specific component can represent that vector with vanishing regularization cost as $m \to \infty$.

Theorem C.4 (Behavior of the generalizable component in Clean and MIR training)**.** Let

$$h := \mathbb{E}[(1-r)^2], \quad q := \mathbb{E}[r(1-r)], \quad \beta := \lambda q,$$

and assume $\beta > 0$. Define

$$\alpha := 1 + \lambda h, \quad \delta := \alpha + \beta, \quad \eta := \delta - \frac{\alpha^2}{\delta} = \frac{\beta(\delta+\alpha)}{\delta}.$$

Under Assumption C.3, let $b^{(m)}_{\mathrm{clean,sq}}$, $b^{(m)}_{\mathrm{MIR,sq}}$, $b^{(m)}_{\mathrm{clean,log}}$, and $b^{(m)}_{\mathrm{MIR,log}}$ denote the b-coordinates of minimizers of the corresponding objectives. Then, as $m \to \infty$,

$$b^{(m)}_{\mathrm{clean,sq}} \to 0, \quad b^{(m)}_{\mathrm{MIR,sq}} \to \bar{b}_{\mathrm{sq}} := \beta\left(\rho_b I_d + \eta\widehat{\Sigma}\right)^{-1}\widehat{\mu}.$$

For logistic loss,

$$b^{(m)}_{\mathrm{clean,log}} \to 0, \quad b^{(m)}_{\mathrm{MIR,log}} \to \bar{b}_{\log},$$

where $\bar{b}_{\log}$ is the unique minimizer of

$$b \mapsto \beta\frac{1}{n}\sum_{i=1}^{n}\log\big(1+\exp(-Y_i b^\top S_i)\big) + \frac{\rho_b}{2}\|b\|_2^2.$$

Moreover, if $\widehat{\mu} \neq 0$, then $\widehat{\mu}^\top \bar{b}_{\mathrm{sq}} > 0$ and $\bar{b}_{\mathrm{sq}} \neq 0$. For logistic loss, if $\widehat{\mu} \neq 0$, then

$$\bar{b}_{\log} \neq 0, \quad \|\bar{b}_{\log}\|_2 \leq \frac{\beta B}{\rho_b}.$$

The theorem formalizes the memorization effect. Clean training can fit the finite training set through the context-specific component alone, so the coefficient on the generalizable component vanishes as context-specific capacity grows. MIR does not have this degeneracy: because masking sometimes hides the context-specific component, the limiting objective retains a nonzero training signal for the generalizable component.

Assumption C.5 (Validation Contexts Are Unseen)**.** For validation examples, the context-specific memorization features learned on the training set are unavailable. We model this as

$$\phi_m(C_{\mathrm{val}}) = 0.$$

Therefore validation predictions depend only on the generalizable logit $b^\top S$.

This assumption does not say that validation text contains no patterns related to the training text. It says only that the example-specific context features used to memorize the finite training corpus do not transfer to unseen validation examples.

Theorem C.6 (MIR Improves Validation Risk)**.** Under the assumptions of Theorem C.4 and Assumption C.5, define

$$R_{\mathrm{sq}}(b) := \mathbb{E}\big[(Y - b^\top S)^2\big] = 1 - 2\mu^\top b + b^\top \Sigma b,$$

and

$$R_{\log}(b) := \mathbb{E}\big[\log\big(1 + \exp(-Y b^\top S)\big)\big].$$

For squared loss, if $2\mu^\top \bar{b}_{\mathrm{sq}} - \bar{b}_{\mathrm{sq}}^\top \Sigma \bar{b}_{\mathrm{sq}} > 0$, then, for all sufficiently large m,

$$R_{\mathrm{sq}}\Big(b^{(m)}_{\mathrm{MIR,sq}}\Big) < R_{\mathrm{sq}}\Big(b^{(m)}_{\mathrm{clean,sq}}\Big).$$

This condition holds automatically when $\widehat{\mu} = \mu$ and $\widehat{\Sigma} = \Sigma$. For logistic loss, if $\mu^\top \bar{b}_{\log} > \frac{B^2}{4}\|\bar{b}_{\log}\|_2^2$, then, for all sufficiently large m,

$$R_{\log}\Big(b^{(m)}_{\mathrm{MIR,log}}\Big) < R_{\log}\Big(b^{(m)}_{\mathrm{clean,log}}\Big).$$

In particular, this logistic condition holds for sufficiently small β/ρ_b whenever $\mu^\top \widehat{\mu} > 0$. Moreover, defining

$$\Delta_{\mathrm{sq},m} := R_{\mathrm{sq}}\Big(b^{(m)}_{\mathrm{clean,sq}}\Big) - R_{\mathrm{sq}}\Big(b^{(m)}_{\mathrm{MIR,sq}}\Big),$$

and

$$\Delta_{\log,m} := R_{\log}\Big(b^{(m)}_{\mathrm{clean,log}}\Big) - R_{\log}\Big(b^{(m)}_{\mathrm{MIR,log}}\Big),$$

we have

$$\Delta_{\mathrm{sq},m} \to R_{\mathrm{sq}}(0) - R_{\mathrm{sq}}(\bar{b}_{\mathrm{sq}}) > 0$$

under the squared-loss condition, and

$$\Delta_{\log,m} \to R_{\log}(0) - R_{\log}(\bar{b}_{\log}) > 0$$

under the logistic-loss condition.

Corollary C.7 (Empirical Signal)**.** Suppose (S_i, Y_i) are independent, $\|S_i\|_2 \le B$, and $\mu = \mathbb{E}[YS] \neq 0$. Then

$$\mathbb{P}\big(\mu^\top \widehat{\mu} > 0\big) \ge 1 - \exp\left(-\frac{n\|\mu\|_2^2}{2B^2}\right).$$

In particular, with high probability, the empirical generalizable component is aligned with the population predictive direction.

The previous results compare clean and MIR training in the limit. The next result makes the dependence on model size explicit for a simplified squared-loss objective. This simplified objective replaces the full expected masked loss by the term that appears when the context-specific component is hidden while the generalizable component remains visible. It is not the full MIR objective, but it isolates the part of masking that forces prediction from the generalizable component.

Theorem C.8 (Increasing Benefit with Growing Model Size)**.** Consider the squared-loss objective

$$\widehat{J}^{(m)}_{\mathrm{key,sq}}(w, b) := \widehat{J}^{(m)}_{\ell_{\mathrm{sq}},\mathrm{clean}}(w, b) + \frac{\beta}{2n}\|Y - \mathbf{S}b\|_2^2, \qquad \beta > 0.$$

Let $b^{(m)}_{\mathrm{key,sq}}$ be its b-coordinate minimizer, and define

$$\Delta_{\mathrm{key},m} := R_{\mathrm{sq}}\Big(b^{(m)}_{\mathrm{clean,sq}}\Big) - R_{\mathrm{sq}}\Big(b^{(m)}_{\mathrm{key,sq}}\Big).$$

Assume $G_m = mI_n, \widehat{\mu} = \mu, \widehat{\Sigma} = \Sigma, \mu \neq 0$. Then $\Delta_{\mathrm{key},m}$ is strictly increasing in m. Moreover, let

$$\Sigma = U\mathrm{diag}(\lambda_1, \ldots, \lambda_d)U^\top, \;\; U^\top \mu = (\mu_1, \ldots, \mu_d)^\top.$$

For each $\lambda_j > 0$, define

$$\kappa_j := \frac{\beta\lambda_j}{\rho_b}.$$

Then

$$\lim_{m\to\infty} \Delta_{\mathrm{key},m} = \sum_{\lambda_j>0} \frac{\mu_j^2}{\lambda_j} \frac{\kappa_j(\kappa_j+2)}{(1+\kappa_j)^2} > 0.$$

Theorem C.8 illustrates the increasing benefit of masking the context-specific component as model size increases. The condition $G_m = mI_n$ is an idealized isotropic-capacity assumption and is stronger than needed for the main intuition; it is used here to obtain a simple closed-form expression and a monotonicity statement. More general Gram matrices with growing eigenvalues would lead to a similar conclusion, although the closed-form expression would be less transparent. The assumptions $\widehat{\mu} = \mu$ and $\widehat{\Sigma} = \Sigma$ remove finite-sample realization error from the training sequences from the statement. They ensure that the empirical generalizable signal in the training set is aligned with the population signal that determines validation risk. Under these conditions, any difference between clean training and the masked objective comes from the use of context-specific memorization. In finite samples, these assumptions can be interpreted as a population-aligned simplification: when n is large, $\widehat{\mu}$ and $\widehat{\Sigma}$ concentrate around μ and Σ, so the same conclusion is stable up to small perturbation terms. In what follows, we prove the theoretical results in this section.

Throughout the proofs, we write

$$u := \Phi_m w \in \mathbb{R}^n.$$

Whenever $G_m = \Phi_m \Phi_m^\top \succ 0$, every $u \in \mathbb{R}^n$ is representable as $\Phi_m w$, and the minimum-norm representative satisfies

$$\min_{w:\Phi_m w = u} \|w\|_2^2 = u^\top G_m^{-1} u.$$

Indeed, $w_\star = \Phi_m^\top G_m^{-1} u$ satisfies $\Phi_m w_\star = u$. For any other feasible $w = w_\star + v$, we have $\Phi_m v = 0$, and hence

$$\langle w_\star, v\rangle = u^\top G_m^{-1} \Phi_m v = 0.$$

Thus

$$\|w\|_2^2 = \|w_\star\|_2^2 + \|v\|_2^2 \geq \|w_\star\|_2^2 = u^\top G_m^{-1} u.$$

Therefore the optimization over (w, b) is equivalent to optimization over (u, b), with regularization term

$$\frac{\rho_w}{2n} u^\top G_m^{-1} u.$$

C.1 Proof of Theorem C.4

Proof. We first prove the squared-loss claims. In (u, b)-coordinates, the clean squared-loss objective is

$$\frac{1}{2n}\|Y - u - \mathbf{S}b\|_2^2 + \frac{\rho_w}{2n} u^\top G_m^{-1} u + \frac{\rho_b}{2}\|b\|_2^2.$$

For fixed b, the first-order condition in u is

$$u - (Y - \mathbf{S}b) + \rho_w G_m^{-1} u = 0.$$

Therefore

$$u^\star(b) = (G_m + \rho_w I_n)^{-1} G_m (Y - \mathbf{S}b).$$

Profiling out u, the clean squared-loss objective becomes

$$\frac{1}{2n}(Y - \mathbf{S}b)^\top T_m (Y - \mathbf{S}b) + \frac{\rho_b}{2}\|b\|_2^2, \;\; T_m := \rho_w (G_m + \rho_w I_n)^{-1}.$$

Differentiating with respect to b gives

$$b_{\mathrm{clean,sq}}^{(m)} = \left(\rho_b I_d + \frac{1}{n}\mathbf{S}^\top T_m \mathbf{S}\right)^{-1} \frac{1}{n}\mathbf{S}^\top T_m Y.$$

The eigenvalues of T_m are

$$\frac{\rho_w}{\lambda_j(G_m) + \rho_w},$$

and hence

$$\|T_m\|_{\mathrm{op}} \le \frac{\rho_w}{a_m + \rho_w} \to 0.$$

It follows that

$$b^{(m)}_{\mathrm{clean,sq}} \to 0.$$

We next consider MIR with squared loss. Let

$$s_0 := \mathbb{E}[r^2].$$

Up to the additive constant $\lambda s_0 (2n)^{-1} \|Y\|_2^2$, the expected four-case squared-loss objective is

$$\alpha \frac{1}{2n}\|Y - u - \mathbf{S}b\|_2^2 + \beta \frac{1}{2n}\|Y - \mathbf{S}b\|_2^2 + \beta \frac{1}{2n}\|Y - u\|_2^2 + \frac{\rho_w}{2n} u^\top G_m^{-1} u + \frac{\rho_b}{2}\|b\|_2^2.$$

The first-order condition in u is

$$\alpha(u + \mathbf{S}b - Y) + \beta(u - Y) + \rho_w G_m^{-1} u = 0.$$

Since $\delta = \alpha + \beta$, this gives

$$u^\star(b) = M_m(\delta Y - \alpha \mathbf{S}b), \;\; M_m := (\delta I_n + \rho_w G_m^{-1})^{-1}.$$

The first-order condition in b is

$$\frac{\alpha}{n}\mathbf{S}^\top(u + \mathbf{S}b - Y) + \frac{\beta}{n}\mathbf{S}^\top(\mathbf{S}b - Y) + \rho_b b = 0.$$

Equivalently,

$$\frac{\alpha}{n}\mathbf{S}^\top u + (\delta \widehat{\Sigma} + \rho_b I_d) b - \delta \widehat{\mu} = 0.$$

Substituting $u^\star(b) = M_m(\delta Y - \alpha \mathbf{S}b)$ yields

$$b^{(m)}_{\mathrm{MIR,sq}} = \left(\rho_b I_d + \delta \widehat{\Sigma} - \frac{\alpha^2}{n}\mathbf{S}^\top M_m \mathbf{S}\right)^{-1} \left(\delta \widehat{\mu} - \frac{\alpha\delta}{n}\mathbf{S}^\top M_m Y\right).$$

Because $\|G_m^{-1}\|_{\mathrm{op}} \to 0$, we have

$$M_m \to \delta^{-1} I_n$$

in operator norm. Therefore

$$\frac{1}{n}\mathbf{S}^\top M_m Y \to \frac{1}{\delta}\widehat{\mu}, \;\; \frac{1}{n}\mathbf{S}^\top M_m \mathbf{S} \to \frac{1}{\delta}\widehat{\Sigma}.$$

It follows that

$$b^{(m)}_{\mathrm{MIR,sq}} \to \beta(\rho_b I_d + \eta \widehat{\Sigma})^{-1}\widehat{\mu} = \bar{b}_{\mathrm{sq}}.$$

If $\widehat{\mu} \neq 0$, then

$$\widehat{\mu}^\top \bar{b}_{\mathrm{sq}} = \beta \widehat{\mu}^\top (\rho_b I_d + \eta \widehat{\Sigma})^{-1} \widehat{\mu} > 0,$$

because $\rho_b I_d + \eta \widehat{\Sigma} \succ 0$. Hence $\bar{b}_{\mathrm{sq}} \neq 0$.

We now prove the logistic-loss claims. Let

$$g(z) := \log(1 + e^{-z}).$$

Choose $\tau_m \to \infty$ such that $\tau_m^2 / a_m \to 0$. For clean logistic training, evaluate the objective at $u = \tau_m Y$ and $b = 0$. Then

$$g(Y_i u_i) = g(\tau_m) \to 0,$$

and

$$\frac{\rho_w}{2n} u^\top G_m^{-1} u \le \frac{\rho_w}{2} \frac{\tau_m^2}{a_m} \to 0.$$

Hence the minimum clean logistic objective converges to zero. Since the objective is nonnegative and contains the term $\rho_b\|b\|_2^2/2$, every clean logistic minimizer satisfies

$$b^{(m)}_{\mathrm{clean,log}} \to 0.$$

For MIR logistic training, define

$$F_{\log}(b) := \beta\frac{1}{n}\sum_{i=1}^{n} g(Y_i b^\top S_i) + \frac{\rho_b}{2}\|b\|_2^2.$$

This function is strongly convex and therefore has a unique minimizer $\bar{b}_{\log}$. The expected four-case MIR logistic objective in (u, b)-coordinates is

$$\begin{aligned}\widehat{J}^{(m)}_{\ell_{\log},\mathrm{MIR}}(u,b) := \alpha\frac{1}{n}\sum_{i=1}^{n} g(Y_i(u_i + b^\top S_i)) + \beta\frac{1}{n}\sum_{i=1}^{n} g(Y_i b^\top S_i) + \beta\frac{1}{n}\sum_{i=1}^{n} g(Y_i u_i)\\ + \lambda s_0 \log 2 + \frac{\rho_w}{2n}u^\top G_m^{-1} u + \frac{\rho_b}{2}\|b\|_2^2.\end{aligned}$$

All terms except $F_{\log}(b)$ and the constant $\lambda s_0 \log 2$ are nonnegative. Hence, for every (u, b),

$$\widehat{J}^{(m)}_{\ell_{\log},\mathrm{MIR}}(u,b) \geq F_{\log}(b) + \lambda s_0 \log 2.$$

Now evaluate the MIR logistic objective at $b = \bar{b}_{\log}$ and $u = \tau_m Y$. Then the context-specific-only margin is $Y_i u_i = \tau_m$, while the margin satisfies

$$Y_i(u_i + \bar{b}_{\log}^\top S_i) = \tau_m + Y_i \bar{b}_{\log}^\top S_i \geq \tau_m - B\|\bar{b}_{\log}\|_2 \to \infty.$$

Thus the corresponding logistic losses vanish, and the context-specific regularization again tends to zero as we choose τ_m such that $\tau_m^2/a_m \to 0$. Therefore

$$\inf_{u,b} \widehat{J}^{(m)}_{\ell_{\log},\mathrm{MIR}}(u,b) \leq F_{\log}(\bar{b}_{\log}) + \lambda s_0 \log 2 + o(1).$$

Combining the lower and upper bounds gives

$$F_{\log}\Big(b^{(m)}_{\mathrm{MIR,log}}\Big) \leq F_{\log}(\bar{b}_{\log}) + o(1).$$

By strong convexity of $F_{\log}$,

$$b^{(m)}_{\mathrm{MIR,log}} \to \bar{b}_{\log}.$$

Finally,

$$\nabla F_{\log}(0) = -\frac{\beta}{2}\widehat{\mu}.$$

Thus, if $\widehat{\mu} \neq 0$, zero is not the minimizer and $\bar{b}_{\log} \neq 0$. At the minimizer $\bar{b}_{\log}$, the first-order condition for $F_{\log}$ gives

$$0 = \beta\frac{1}{n}\sum_{i=1}^{n} g'(Y_i \bar{b}_{\log}^\top S_i) Y_i S_i + \rho_b \bar{b}_{\log}.$$

Equivalently,

$$\rho_b \bar{b}_{\log} = -\beta\frac{1}{n}\sum_{i=1}^{n} g'(Y_i \bar{b}_{\log}^\top S_i) Y_i S_i.$$

Taking Euclidean norms and using the triangle inequality,

$$\begin{aligned}\rho_b\|\bar{b}_{\log}\|_2 &\leq \beta\frac{1}{n}\sum_{i=1}^{n} \big|g'(Y_i \bar{b}_{\log}^\top S_i)\big|\,|Y_i|\|S_i\|_2\\ &\leq \beta B,\end{aligned}$$

because $|Y_i| = 1$, $\|S_i\|_2 \le B$, and $|g'(z)| \le 1$. Therefore

$$\|\bar{b}_{\log}\|_2 \le \frac{\beta B}{\rho_b}.$$

□

C.2 Proof of Theorem C.6

Proof. By Assumption C.5, validation prediction scores are $b^\top S$. Therefore validation risks depend only on the coefficient b of the generalizable component.

For squared loss,

$$R_{\mathrm{sq}}(b) - R_{\mathrm{sq}}(0) = -2\mu^\top b + b^\top \Sigma b.$$

Thus $R_{\mathrm{sq}}(b) < R_{\mathrm{sq}}(0)$ whenever

$$2\mu^\top b - b^\top \Sigma b > 0.$$

By Theorem C.4,

$$b^{(m)}_{\mathrm{clean,sq}} \to 0, \quad b^{(m)}_{\mathrm{MIR,sq}} \to \bar{b}_{\mathrm{sq}}.$$

If

$$2\mu^\top \bar{b}_{\mathrm{sq}} - \bar{b}_{\mathrm{sq}}^\top \Sigma \bar{b}_{\mathrm{sq}} > 0,$$

then continuity gives

$$R_{\mathrm{sq}}\left(b^{(m)}_{\mathrm{MIR,sq}}\right) < R_{\mathrm{sq}}\left(b^{(m)}_{\mathrm{clean,sq}}\right)$$

for all sufficiently large m.

We next show that the squared-loss condition holds automatically when $\widehat{\mu} = \mu$ and $\widehat{\Sigma} = \Sigma$. In this case,

$$\bar{b}_{\mathrm{sq}} = \beta(\rho_b I_d + \eta\Sigma)^{-1}\mu.$$

Let

$$A := (\rho_b I_d + \eta\Sigma)^{-1}.$$

Since A and Σ commute,

$$\bar{b}_{\mathrm{sq}}^\top \Sigma \bar{b}_{\mathrm{sq}} = \beta^2 \mu^\top A\Sigma A\mu \le \frac{\beta}{\eta}\,\beta\mu^\top A\mu = \frac{\beta}{\eta}\,\mu^\top \bar{b}_{\mathrm{sq}}.$$

Because

$$\eta = \frac{\beta(\delta+\alpha)}{\delta} > \beta,$$

we have $\beta/\eta < 1$. Also,

$$\mu^\top \bar{b}_{\mathrm{sq}} = \beta\mu^\top A\mu > 0,$$

since $\mu \neq 0$ and $A \succ 0$. Therefore

$$2\mu^\top \bar{b}_{\mathrm{sq}} - \bar{b}_{\mathrm{sq}}^\top \Sigma \bar{b}_{\mathrm{sq}} > \mu^\top \bar{b}_{\mathrm{sq}} > 0.$$

For logistic loss,

$$\nabla R_{\log}(0) = -\frac{1}{2}\mu.$$

Moreover, with $\sigma(t) = (1+e^{-t})^{-1}$, the Hessian satisfies

$$\nabla^2 R_{\log}(b) = \mathbb{E}\left[\sigma(Yb^\top S)\sigma(-Yb^\top S)SS^\top\right] \preceq \frac{1}{4}\mathbb{E}[SS^\top] \preceq \frac{B^2}{4} I_d.$$

Hence Taylor's expansion gives

$$R_{\log}(b) \le R_{\log}(0) - \frac{1}{2}\mu^\top b + \frac{B^2}{8}\|b\|_2^2.$$

Thus $R_{\log}(b) < R_{\log}(0)$ whenever

$$\mu^\top b > \frac{B^2}{4}\|b\|_2^2.$$

Applying this condition at $b = \bar{b}_{\log}$, and using Theorem C.4, gives

$$R_{\log}\left(b^{(m)}_{\mathrm{MIR},\log}\right) < R_{\log}\left(b^{(m)}_{\mathrm{clean},\log}\right)$$

for all sufficiently large m.

It remains to justify the stated sufficient condition for logistic loss. Let

$$L_{\mathrm{emp}}(b) := \frac{1}{n}\sum_{i=1}^{n} g(Y_i b^\top S_i), \quad g(z) := \log(1 + e^{-z}).$$

The minimizer $\bar{b}_{\log}$ satisfies

$$\rho_b \bar{b}_{\log} = -\beta \nabla L_{\mathrm{emp}}(\bar{b}_{\log}).$$

Since $\|S_i\|_2 \le B$, the gradient ∇L_{emp} is Lipschitz on $\mathbb{R}^d$, and

$$\nabla L_{\mathrm{emp}}(0) = -\frac{1}{2}\widehat{\mu}.$$

Also, from the optimality equation and $\|\nabla L_{\mathrm{emp}}(b)\|_2 \le B$,

$$\|\bar{b}_{\log}\|_2 \le \frac{\beta B}{\rho_b}.$$

Therefore, as $\beta/\rho_b \to 0$,

$$\bar{b}_{\log} = \frac{\beta}{2\rho_b}\widehat{\mu} + o(\beta/\rho_b).$$

If $\mu^\top \widehat{\mu} > 0$, then

$$\mu^\top \bar{b}_{\log} = \frac{\beta}{2\rho_b}\mu^\top \widehat{\mu} + o(\beta/\rho_b),$$

whereas

$$\|\bar{b}_{\log}\|_2^2 = O((\beta/\rho_b)^2).$$

Hence, for sufficiently small β/ρ_b,

$$\mu^\top \bar{b}_{\log} > \frac{B^2}{4}\|\bar{b}_{\log}\|_2^2.$$

The asymptotic gain results follow from the same convergence and continuity:

$$\Delta_{\mathrm{sq},m} \to R_{\mathrm{sq}}(0) - R_{\mathrm{sq}}(\bar{b}_{\mathrm{sq}}) > 0,$$

and

$$\Delta_{\log,m} \to R_{\log}(0) - R_{\log}(\bar{b}_{\log}) > 0.$$

□

C.3 Proof of Corollary C.7

Proof. Let

$$a := \frac{\mu}{\|\mu\|_2}.$$

Then

$$a^\top \widehat{\mu} = \frac{1}{n}\sum_{i=1}^{n} Y_i a^\top S_i.$$

The summands satisfy

$$|Y_i a^\top S_i| \le B, \quad \mathbb{E}[Y_i a^\top S_i] = a^\top \mu = \|\mu\|_2.$$

Hoeffding's inequality [Vershynin, 2018] gives

$$\mathbb{P}(a^\top \widehat{\mu} \le 0) = \mathbb{P}(a^\top \widehat{\mu} - \|\mu\|_2 \le -\|\mu\|_2) \le \exp\left(-\frac{n\|\mu\|_2^2}{2B^2}\right).$$

Since $a^\top \widehat{\mu} > 0$ is equivalent to $\mu^\top \widehat{\mu} > 0$, the result follows. □

C.4 Proof of Theorem C.8

Proof. Under $G_m = mI_n$,

$$T_m = \rho_w (G_m + \rho_w I_n)^{-1} = \frac{\rho_w}{m + \rho_w} I_n.$$

Write

$$t_m := \frac{\rho_w}{m + \rho_w}.$$

Using the profiled squared-loss formula from the proof of Theorem C.4, and using $\widehat{\mu} = \mu$ and $\widehat{\Sigma} = \Sigma$, we obtain

$$b^{(m)}_{\text{clean,sq}} = t_m (\rho_b I_d + t_m \Sigma)^{-1} \mu.$$

For the key-ablating objective, profiling out u gives

$$\frac{1}{2n}(Y - \mathbf{S}b)^\top T_m (Y - \mathbf{S}b) + \frac{\beta}{2n}\|Y - \mathbf{S}b\|_2^2 + \frac{\rho_b}{2}\|b\|_2^2.$$

Differentiating with respect to b yields

$$b^{(m)}_{\text{key,sq}} = (t_m + \beta)(\rho_b I_d + (t_m + \beta)\Sigma)^{-1} \mu.$$

Let

$$\Sigma = U \operatorname{diag}(\lambda_1, \dots, \lambda_d) U^\top, \;\; U^\top \mu = (\mu_1, \dots, \mu_d)^\top.$$

If $\lambda_j = 0$, then $\mu_j = 0$ since $\mu = \mathbb{E}[YS]$ and $\Sigma = \mathbb{E}[SS^\top]$. Indeed, $\lambda_j = 0$ implies that the corresponding projection of S is zero almost surely, and hence its correlation with Y is also zero. Thus only terms with $\lambda_j > 0$ contribute to the risk.

For $\alpha_0 > 0$, define

$$b(\alpha_0) := \alpha_0 (\rho_b I_d + \alpha_0 \Sigma)^{-1} \mu.$$

In the eigenbasis of Σ, the j-th coordinate is, for $\lambda_j > 0$,

$$b_j(\alpha_0) = \frac{\mu_j}{\lambda_j} \frac{\alpha_0 \lambda_j / \rho_b}{1 + \alpha_0 \lambda_j / \rho_b}.$$

Let

$$s(x) := \frac{x}{1 + x}.$$

The reduction in squared risk from using $b(\alpha_0)$ instead of 0 is

$$R_{\text{sq}}(0) - R_{\text{sq}}(b(\alpha_0)) = 2\mu^\top b(\alpha_0) - b(\alpha_0)^\top \Sigma b(\alpha_0).$$

Write $U^\top b(\alpha_0) = (b_1(\alpha_0), \dots, b_d(\alpha_0))^\top$. In the eigenbasis of Σ,

$$\mu^\top b(\alpha_0) = \sum_{j=1}^d \mu_j b_j(\alpha_0), \qquad b(\alpha_0)^\top \Sigma b(\alpha_0) = \sum_{j=1}^d \lambda_j b_j(\alpha_0)^2.$$

Therefore

$$R_{\text{sq}}(0) - R_{\text{sq}}(b(\alpha_0)) = \sum_{j=1}^d \left\{2\mu_j b_j(\alpha_0) - \lambda_j b_j(\alpha_0)^2\right\}.$$

If $\lambda_j = 0$, then $\mu_j = 0$, so the corresponding term is zero. Thus only the terms with $\lambda_j > 0$ remain. For such j,

$$b_j(\alpha_0) = \frac{\mu_j}{\lambda_j} s\left(\frac{\alpha_0 \lambda_j}{\rho_b}\right), \qquad s(x) := \frac{x}{1+x}.$$

Substituting this expression gives

$$\begin{aligned} 2\mu_j b_j(\alpha_0) - \lambda_j b_j(\alpha_0)^2 &= 2\mu_j \frac{\mu_j}{\lambda_j} s\left(\frac{\alpha_0 \lambda_j}{\rho_b}\right) - \lambda_j \left[\frac{\mu_j}{\lambda_j} s\left(\frac{\alpha_0 \lambda_j}{\rho_b}\right)\right]^2 \\ &= \frac{2\mu_j^2}{\lambda_j} s\left(\frac{\alpha_0 \lambda_j}{\rho_b}\right) - \frac{\mu_j^2}{\lambda_j} s^2\left(\frac{\alpha_0 \lambda_j}{\rho_b}\right) \\ &= \frac{\mu_j^2}{\lambda_j} s\left(\frac{\alpha_0 \lambda_j}{\rho_b}\right)\left[2 - s\left(\frac{\alpha_0 \lambda_j}{\rho_b}\right)\right]. \end{aligned}$$

Hence

$$R_{\mathrm{sq}}(0) - R_{\mathrm{sq}}(b(\alpha_0)) = \sum_{\lambda_j > 0} \frac{\mu_j^2}{\lambda_j} s\left(\frac{\alpha_0 \lambda_j}{\rho_b}\right)\left[2 - s\left(\frac{\alpha_0 \lambda_j}{\rho_b}\right)\right].$$

Since

$$b^{(m)}_{\mathrm{clean,sq}} = b(t_m), \;\; b^{(m)}_{\mathrm{key,sq}} = b(t_m + \beta),$$

the gain $\Delta_{\mathrm{key},m}$ is a sum over $\lambda_j > 0$ of terms of the form

$$\frac{\mu_j^2}{\lambda_j}\left[s(x+\kappa_j)\{2 - s(x+\kappa_j)\} - s(x)\{2 - s(x)\}\right],$$

where

$$x = \frac{t_m \lambda_j}{\rho_b}, \;\; \kappa_j = \frac{\beta \lambda_j}{\rho_b} > 0.$$

Note that

$$s(x+\kappa) - s(x) = \frac{\kappa}{(1+x)(1+x+\kappa)}$$

and

$$2 - s(x+\kappa) - s(x) = \frac{\kappa + 2x + 2}{(1+x)(1+x+\kappa)}.$$

We have that each nonzero spectral contribution equals

$$\frac{\mu_j^2}{\lambda_j} \frac{\kappa_j(\kappa_j + 2x + 2)}{(1+x)^2(1+x+\kappa_j)^2}.$$

For fixed $\kappa > 0$, define

$$F_\kappa(x) := \frac{\kappa(\kappa + 2x + 2)}{(1+x)^2(1+x+\kappa)^2}.$$

Then

$$F'_\kappa(x) = -\frac{2\kappa\left(\kappa^2 + 3\kappa x + 3\kappa + 3x^2 + 6x + 3\right)}{(1+x)^3(1+x+\kappa)^3} < 0.$$

Since

$$t_m = \frac{\rho_w}{m + \rho_w}$$

is strictly decreasing in m, each nonzero spectral contribution to $\Delta_{\mathrm{key},m}$ is strictly increasing in m. Because $\mu \neq 0$, at least one such contribution is nonzero. Hence $\Delta_{\mathrm{key},m}$ is strictly increasing in m.

Finally, $t_m \to 0$, so $x \to 0$ for every j, and

$$\lim_{m \to \infty} \Delta_{\mathrm{key},m} = \sum_{\lambda_j > 0} \frac{\mu_j^2}{\lambda_j} \frac{\kappa_j(\kappa_j + 2)}{(1+\kappa_j)^2} > 0.$$

□

D Derivation of Quanta Scaling Law

To make the paper self-contained and easier for readers to follow, this section summarizes the Quanta argument from Michaud [2026] that we use in our paper. The background is skill learning: next-token prediction is assumed to require a large collection of discrete predictive skills, called *quanta*. A model either learns a quantum or it does not, and scaling improves performance by allowing the model to learn more quanta in descending order of usefulness.

Loss as a Function of Learned Quanta. Index quanta by decreasing use frequency. Let p_k be the probability that the k-th quantum is needed on a randomly drawn token. The Quanta model assumes a Zipf tail

$$p_k = \frac{1}{Z} k^{-(1+\alpha)}, \qquad Z = \sum_{k=1}^{\infty} k^{-(1+\alpha)}, \qquad \alpha > 0. \tag{15}$$

In the simplest monogenic version of the model, each token depends mainly on one quantum. Suppose learning a quantum lowers the loss on those tokens from b to a, with $b > a$. If the model has learned the first n quanta, its expected loss is

$$L(n) = \sum_{k=1}^{n} a p_k + \sum_{k=n+1}^{\infty} b p_k = a + (b-a) \sum_{k=n+1}^{\infty} p_k \approx a + \frac{b-a}{\alpha Z} n^{-\alpha}, \tag{16}$$

where the last line uses the standard tail approximation

$$\sum_{k=n+1}^{\infty} k^{-(1+\alpha)} \approx \int_{n}^{\infty} x^{-(1+\alpha)} dx = n^{-\alpha}/\alpha.$$

Therefore

$$L(n) \approx E + C n^{-\alpha}, \tag{17}$$

where $E = a$ is the irreducible loss floor and $C > 0$ absorbs the remaining constants. This is the key step: a Zipf distribution over skill frequencies induces a power law in the loss as a function of the number of learned skills.

Parameter Scaling. If data is abundant, then the bottleneck is model capacity. Assume each quantum requires approximately c_N parameters to represent. A model with N parameters can then learn

$$n_N \approx \frac{N}{c_N} \tag{18}$$

quanta. Substituting this into Eq. (17) gives

$$L(N, \infty) \approx E + A_N N^{-\alpha}, \tag{19}$$

so the parameter-scaling exponent is

$$\alpha_N = \alpha. \tag{20}$$

Data Scaling. In the data-constrained multi-epoch regime, repeated passes over the same corpus do not create new rare skills. The relevant resource is the number of unique tokens U. Assume that learning the k-th quantum requires at least τ tokens in the unique dataset that use that quantum. Then the last quantum that can be learned, denoted n_U, satisfies

$$U p_{n_U} \approx \tau. \tag{21}$$

Using $p_k \propto k^{-(1+\alpha)}$, we obtain

$$n_U \approx \left(\frac{U}{Z\tau} \right)^{1/(1+\alpha)}. \tag{22}$$

Substituting again into Eq. (17) yields

$$L(\infty, U) \approx E + A_U U^{-\alpha/(1+\alpha)}, \tag{23}$$

so the data-scaling exponent is

$$\alpha_U = \frac{\alpha}{1+\alpha}. \tag{24}$$

This is why the data exponent is smaller than the parameter exponent in the basic Quanta picture.

A Quanta-Motivated Joint Law. The single-axis derivations above do not uniquely determine a joint (N, U) law, but they do imply that the number of learned quanta is jointly limited by parameter capacity and unique-data coverage. A hard bottleneck view would write

$$n(N, U) \lesssim \min\left\{\gamma_N N,\ \gamma_U U^{1/(1+\alpha)}\right\}, \tag{25}$$

for some constants $\gamma_N, \gamma_U > 0$. Equivalently,

$$n(N, U)^{-1} \gtrsim \max\left\{\frac{1}{\gamma_N N},\ \frac{1}{\gamma_U U^{1/(1+\alpha)}}\right\}. \tag{26}$$

For fitting, it is convenient to replace this hard maximum by a smooth additive envelope in inverse-skill space,

$$n(N, U)^{-1} \approx \frac{A'}{N} + \frac{B'}{U^{1/(1+\alpha)}}. \tag{27}$$

Substituting this into $L - E \propto n^{-\alpha}$ gives the Quanta-motivated joint law

$$L_{\mathrm{Q}}(N, U) = E + \left(\frac{A}{N} + \frac{B}{U^{1/(1+\alpha)}}\right)^{\alpha}, \tag{28}$$

which is exactly the form used in Eq. (8). This coupling should be read as a smooth interpolation motivated by the Quanta asymptotes. It is attractive because it recovers both derived limits:

$$L_{\mathrm{Q}}(N, \infty) = E + A^{\alpha} N^{-\alpha}, \tag{29}$$

$$L_{\mathrm{Q}}(\infty, U) = E + B^{\alpha} U^{-\alpha/(1+\alpha)}. \tag{30}$$

Thus, the Quanta picture explains why the benefit of increasing model size should depend on the available unique data: both resources control the number of skills that can be learned, and the loss is governed by that shared latent quantity.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We believe the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of the paper in the Discussion section.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We provide full set of assumptions, theorem statements, and a complete and rigorous proof in the Appendix C.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We open source all training and data generation code at github repository for reproduction. We also explain the experiment details in the Appendix A.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We open source all training and data generation code at github for reproduction. We also explain the experiment details in the Appendix A.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: We specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results in the main paper and the Appendix A.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments in Figure 1a and Table 1.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We detail the computational resources in the Appendix and specify that each experiment conducted in the paper can be reproduced using 8 NVIDIA H100 (80GB) GPUs within 48 hours.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: the research conducted in the paper conform with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: This work is a fundamental study of data-constrained pretraining and does not introduce a deployed system, it may have indirect societal effects by making language-model pretraining more data-efficient. The efficiency gains could broaden access to model training,

but could also lower the barrier for misuse of language models, so downstream release and deployment decisions should still consider standard safety, privacy, and abuse-mitigation practices.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: Given the experiment scale and the opensource datasets we use, we believe this paper does not have such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We only use open-source datasets and public model architectures in this paper, and have cited relevant papers and technical reports. We report the license in Appendix A.9.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.

- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We open-source our code with proper comments and docstrings.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: the paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: the paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: We do not use LLMs to develop the methods.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.