OPT2025: 17th Annual Workshop on Optimization for Machine Learning

Domain-Aware Scaling Laws Uncover Data Synergy

Kimia Hamidieh HAMIDIEH@MIT.EDU
Massachusetts Institute of Technology
Lester Mackey LMACKEY@MICROSOFT.COM
Microsoft Resesarch
David Alvarez-melis DAALVARE@MICROSOFT.COM
Microsoft Resesarch & Harvard University

Abstract

Machine learning progress is often attributed to scaling model size and dataset volume, yet the composition of data can be just as consequential. Empirical findings repeatedly show that combining datasets from different domains yields non-trivial interactions: adding code improves mathematical reasoning, while certain mixtures introduce interference that suppresses performance. We refer to these effects collectively as data *synergy*—interaction effects whereby the joint contribution of multiple domains exceeds (positive synergy) or falls short of (interference) the sum of their isolated contributions. In this work, we formalize and quantify dataset interactions in large language models. Leveraging observational variation across open-weight LLMs with diverse pretraining mixtures, we estimate both direct domain-to-benchmark synergy (how one domain contributes to performance on another) and pretraining data synergy (capabilities that require co-occurrence of multiple domains). Our framework improves predictive accuracy over domain-agnostic scaling laws, recovers stable synergy patterns such as math–code complementarity, and yields interpretable maps of cross-domain transfer. These results demonstrate that understanding and exploiting data synergy is essential for designing data mixtures and curating corpora in the next generation of foundation models.

1. Introduction

Recent improvements in Large Language Models (LLMs) are strongly shaped by their pretraining data [15, 25], yet most formulations abstract away composition and interactions, reducing data into an undifferentiated token count [9, 13]. Practitioners frequently observe that adding data from one domain improves performance on seemingly unrelated tasks–for example, code data enhancing mathematical reasoning [1, 18], while other combinations lead to interference and degraded performance [16, 34]. We refer to these cross-domain interactions as **data synergy**. Such findings suggest that tokens are not interchangeable: what matters is not only how much data we train on, but also *what kinds of data are combined*.

There is growing evidence that interactions matter. Prior works find that continued pretraining on code improves reasoning-heavy benchmarks [32]; and targeted ablations reveal both positive transfer and occasional negative transfer across conceptually related sources [8, 31, 34]. These observations are not isolated anecdotes but point to regularities in how domain composition shapes learned representations; regularities that merit explicit modeling.

Most existing approaches overlook this dimension, and instead treat pretraining corpora as homogeneous. Classical scaling laws, for instance, relate loss to parameter count and total data but are domain-agnostic, since they assume all tokens contribute equally [9, 13]. and mixture-optimization methods often search over weights while implicitly assuming independent returns [30]. What is missing is an explicit, identifiably interaction-aware formulation: a way to separate the aggregate benefit of more data from domain-specific deviations, and to quantify when two domains together yield more (or less) than the sum of their parts.

In this paper, we present a framework for quantifying and modeling data synergy in LLMs. Rather than treating data as homogeneous, we exploit natural variation across open-weight models trained on diverse data mixtures. Our approach formalizes two complementary notions: (i) **Domain→benchmark synergy**, measuring how pre-training data from one domain affects the performance on another, and (ii) **Pretraining data synergy**, capturing domain-domain interaction effects that depend on co-occurrence of multiple domains in the training data. Our estimation procedure jointly fits these effects across many (model, benchmark) observations, provides sparse, interpretable maps of cross-domain interactions that generalize beyond observed mixtures.

Our contributions are as follows:

- We provide an operational definition of dataset synergy that links empirical observations to formal modeling.
- We introduce domain-aware estimators that improve predictive model performance over standard scaling laws by incorporating synergy terms.
- We recover stable, interpretable synergy patterns, such as the recurring complementarity between code and math, that provide actionable insights for data curation and acquisition.

2. Domain and Synergy-Aware Scaling Laws

2.1. Problem Setup

Let $M_1, \dots, M_m$ denote a set of language models (e.g., open-weight LLMs on Huggingface), and let $D_1, \dots, D_n$ be evaluation domains. For every model-domain pair we observe the loss value $L = \big[l_{i,j}\big] \in \mathbb{R}^{m \times n}$. Alongside L we collect model-level covariates: parameter count N_i, total pretraining tokens d_i, and composition (mixture shares) of the pretraining data $u_{i,k} \in [0,1]$, the fraction of tokens from training domain D_k that model M_i is pretrained on. Our goal is to quantify *cross-domain data synergy*: how training on pretraining domain(s) affects loss on a benchmark, after accounting for model scale and total tokens.

2.2. Domain-agnostic scaling law

We begin with a domain-agnostic baseline that explains loss variation using only model size N in terms of number of parameters, and total pretraining tokens D. This serves as the baseline model against which composition- and synergy-aware refinements will be evaluated. Following Chinchilla scaling laws [9], we have the parametric form $L(N, D) = L_\infty + AN^{-\alpha} + BD^{-\beta}$, where L_∞ is the irreducible loss, A, B are scale coefficients for parameter and data terms, and α, β are the parameter and data scaling exponent. In practice we fit this baseline *per benchmark*, and allow these parameters to vary across evaluation domains, after mapping heterogeneous metrics to a common monotone-transformed pseudo-loss. Let $s = \log N$ and $d = \log D$, and for benchmark j define $e_j = \log L_{\infty,j}$, $a_j = \log A_j$, and $b_j = \log B_j$. Writing the parametric form in log-parameters gives the numerically-stable log-sum-exp (LSE) form:

$$\log L_j(s, d) = \mathrm{LSE}\big(e_j,\ a_j - \alpha_j s,\ b_j - \beta_j d\big), \tag{1}$$

where $\mathrm{LSE}(x_1, x_2, x_3) = \log\big(e^{x_1} + e^{x_2} + e^{x_3}\big)$. Equation (1) is our *domain-agnostic* baseline: the expected log-loss depends only on total parameters and total tokens. In the next subsections we enrich the *data* term $b - \beta \log D$ to account for training mixture composition and to estimate cross-domain synergy, while keeping the same overall form of the scaling law recoverable.

2.3. First-order Domain-Benchmark Synergy

We now modify the data term so that *different training domains can reduce loss at different rates* on each benchmark. Recall that $u_{i,k} \in [0, 1]$ is the fraction of tokens from training domain D_k that model M_i is pretrained on. It is easy to show that $\log d_i = \sum_k u_{i,k} \log(u_{i,k} d_i) + H(u_i)$, where $H(u_i) = -\sum_k u_{i,k} \log u_{i,k}$. To allow domain–specific data scaling exponent on benchmark j, $\beta_j + \gamma_{j,k}$ (faster if $\gamma_{j,k} > 0$, slower if $\gamma_{j,k} < 0$).

We introduce domain–specific data scaling exponent parameters $\{\gamma_{i,j}\}$, which modify the per-task data scaling exponents additively: $\beta_j + \gamma_{j,k}$. A positive $\gamma_{j,k}$ indicates that data from domain k yields greater-than-expected scaling – i.e, a synergistic effect with benchmark j. Conversely, a negative $\gamma_{j,k}$ implies diminishing returns, where data from domain k contributes less effectively than expected (interference). Substituting the $\log d_i$ decomposition equation into the third argument of the LSE in (1) and introducing these modifiers gives

$$\mathbb{E}[\, l_{i,j}\,] = \mathrm{LSE}\Big(e_j, a_j - \alpha_j s_i, b_j - \big[\beta_j \mathbf{1} + \gamma_{j,\cdot}\big]^\top z_i - \beta_j H(u_i)\Big), \tag{2}$$

with $z_{i,k} = u_{i,k} \log(u_{i,k} d_i)$ and synergy coefficients $\gamma_{j,k} \in \mathbb{R}$.

2.4. Second-order Pretraining Data Synergy

We hypothesize that certain training domains interact synergistically and intrinsically, such that their joint presence yields learning benefits irrespective of the specific downstream benchmark. These synergies reflect fundamental complementarities between domains, though their effect size is still mediated by the benchmark-specific data scaling coefficient. The gain from such co-occurrence is bottlenecked by the scarcer source and can be understood as producing "additional bonus tokens" only when both domains are present. To capture this, we model synergy with a pairwise term that vanishes if either domain is absent and scales with the smaller per-domain log-token budget.

Let $u_{i,k} \in [0, 1]$ be pretraining domain mixture weights and d_i the total token count and define

$$z_{i,k} = u_{i,k} \log(u_{i,k} d_i), \qquad \bar{z}_{i,k} = \log\big(1 + u_{i,k} d_i\big), \qquad \mathrm{softmin}_\tau(a, b) := -\tau \log\big(e^{-a/\tau} + e^{-b/\tau}\big).$$

Start from the first–order data term split into baseline and per–domain parts, $-\beta_j \sum_k z_{i,k} - \sum_k \gamma_{j,k}\, z_{i,k} - \beta_j H(u_i)$, and augment only the base effect with a co-occurrence correction

$$-\beta_j \sum_k \Big[z_{i,k} + \sum_{k' \neq k} \gamma_{j,k} \sigma_{kk'}\ \mathrm{softmin}_\tau\big(\bar{z}_{i,k}, \bar{z}_{i,k'}\big)\Big] - \sum_k \gamma_{j,k} z_{i,k} - \beta_j H(u_i),$$

that is with a cross-domain synergy correction modulated by domain-benchmark synergy strength. Using $\bar{z}$ (nonnegative and $= 0$ when $u_{i,k} = 0$) ensures the interaction truly vanishes if either domain is absent and preserves the desired $O(\log d_i)$ scaling. We then scale the last term by β_j to ensure that it moves at the same rate as the baseline. Symmetrizing (using $\sigma_{kk'} = \sigma_{k'k}$), gives,

$$\Phi_{i,j} = b_j - \big[\beta_j \mathbf{1}_K + \gamma_{j,\cdot}\big]^\top z_i - \beta_j\, H(u_i) - \beta_j \sum_{k < k'} (\gamma_{j,k} + \gamma_{j,k'})\, \sigma_{kk'}\ \mathrm{softmin}_\tau\big(\bar{z}_{i,k}, \bar{z}_{i,k'}\big) \tag{3}$$

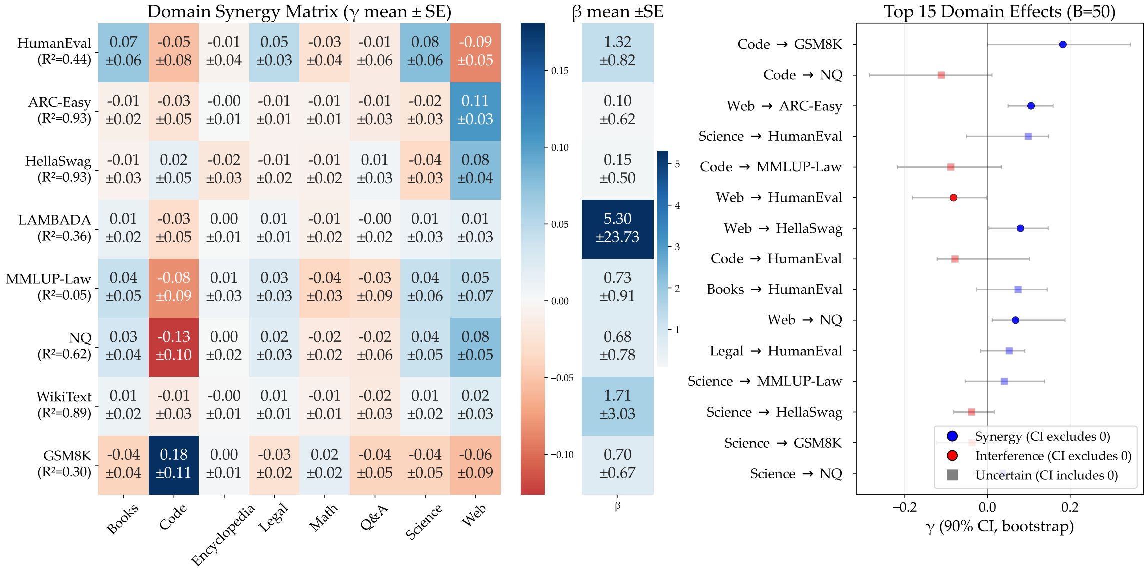


Figure 1: We visualize first-order domain→benchmark synergy heatmap $\gamma_{j,k}$ (median ± SE, B=50) between pretraining domains (columns) and benchmarks (rows), and right panel shows β_j with 90% CIs. A few synergies (e.g., code → GSM8K) are reliably positive, while most synergies are small or uncertain.

where $\Sigma = [\sigma_{kk'}]$ is symmetric with $\sigma_{kk} = 0$. The complete log-space law has the form $\log L_{i,j} = \mathrm{LSE}\big(e_j,\ a_j - \alpha_j s_i,\ \Phi_{i,j}\big)$. In Appendix C.2, we interpret the new data term as "effective" number of pretraining data.

3. Results

We estimate composition effects from *observational variation* across open-weight models and their publicly documented pre-training mixtures, rather than training models. Details on models and data are included in Appendix E.

3.1. First-order Domain→Benchmark Synergy

We estimate the first-order synergy of each pre-training domain on each benchmark, as explained in Appendix D. Figure 1 visualizes the estimated matrix $\Gamma = \{\gamma_{j,k}\}$ with the benchmark-specific data exponent β_j. To quantify uncertainty, we report standard errors from B=50 bootstrap resamples (80% subsampling) and show 90% confidence intervals for β_j. The resulting map is sparse and interpretable: a few domain→benchmark synergies are reliably positive, while most γ values are small or statistically indistinguishable from zero synergy. Notably, code pretraining exhibits positive first-order synergy to math reasoning benchmark GSM8K, which is consistent with findings of prior work [1, 18].

3.2. Second-order Pre-training Data Synergy

Beyond first-order effects, we estimate *pairwise* data synergy between pre-training domains to capture gains that materialize only when two domains co-occur, across all benchmarks. We fit the

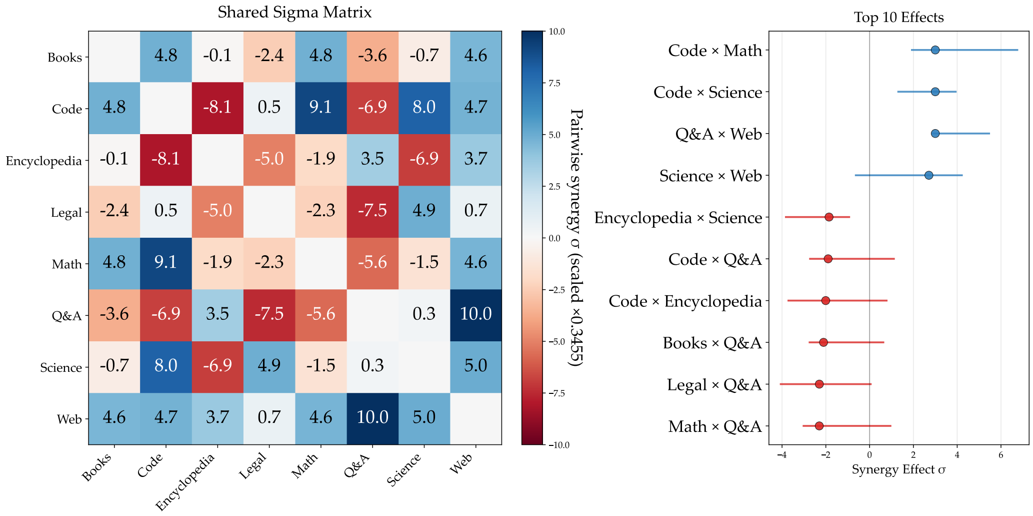


Figure 2: Second-order pretraining domain×domain synergy matrix $\Sigma = \{\sigma_{i,j}\}$ shows positive synergy between domains such as *Code×Math*, *Q&A×Web/Science*.

symmetric synergy matrix $\Sigma = \{\sigma_{k,k'}\}$ along with other parameters as in Equation 3 evaluate uncertainty via $B = 50$ bootstrap resamples (80% subsampling). Figure 2 summarizes the results the shared pairwise synergy across domains, along with the confidence intervals. Positive entries indicate domain-domain synergy (e.g. the expected *Code×Math* synergy), while negative entries show interference. We also observe negative synergy between *Code×Encyclopedia/Q&A*, which reflects that having both domains present in the pre-training dataset, may hurt average performance on selected domains.

We also assess how predictive the model is on held-out samples. Figure 3 compares our domain-aware estimator with a domain-agnostic ($\gamma_{i,j} = 0$) baseline. Across a number of benchmarks, the synergy-aware models show better loss on residuals of held-out examples, on these datasets. A comparison of all fits is shown in Figure 4 Appendix F.1. These gains show that modeling composition, via a small number of non-zero $\gamma_{j,k}$ (and universal $\sigma_{k,k'}$) adds explanatory power beyond total token count, which is especially effective for benchmarks that are different in distribution from typical pre-training datasets, such as in `HumanEval`, a coding benchmark.

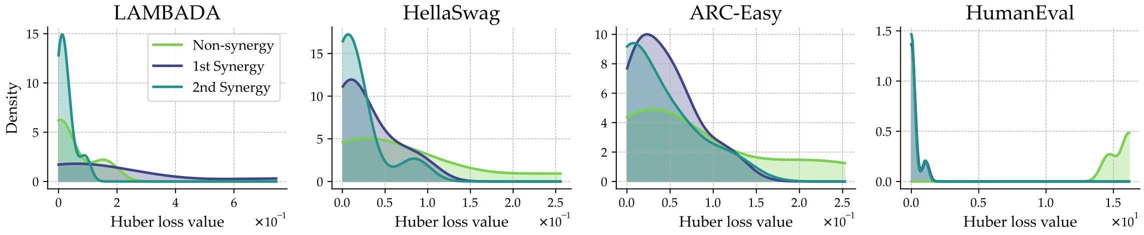


Figure 3: We compare first and second-order domain-aware scaling model vs. the domain-agnostic one (Chinchilla) on a number of benchmarks. The synergy-aware estimators achieves lower test loss on held-out models in these benchmarks.

4. Conclusion

We formalize and quantify data synergy in LLM pretraining, and show that explicitly modeling domain-benchmark and pretraining data interactions can lead to interpretable estimates of data synergy, and improve predictive fit across multiple benchmarks. Our approach is limited by its observational design, as estimates rely on noise or incomplete mixture metadata, and we only rely on eight benchmarks to estimate pretraining data synergy. Future work will validate these measured synergies by training on domains we find to be synergistic. Beyond methodology, our estimates support practical applications in data curation, mixing, and acquisition, and performing synergy-aware mixture optimization to target specific capabilities.

References

[1] Zhangir Azerbayev, Hailey Schoelkopf, Keiran Paster, Marco Dos Santos, Stephen McAleer, Albert Q Jiang, Jia Deng, Stella Biderman, and Sean Welleck. Llemma: An open language model for mathematics. *arXiv preprint arXiv:2310.10631*, 2023.

[2] Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR, 2023.

[3] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

[4] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

[5] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[6] Xinyang Geng and Hao Liu. Openllama: An open reproduction of llama, May 2023. URL https://github.com/openlm-research/open_llama.

[7] Dirk Groeneveld, Iz Beltagy, Pete Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Harsh Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, et al. Olmo: Accelerating the science of language models. *arXiv preprint arXiv:2402.00838*, 2024.

[8] Jiawei Gu, Zacc Yang, Chuanghao Ding, Rui Zhao, and Fei Tan. Cmr scaling law: Predicting critical mixture ratios for continual pre-training of language models. *arXiv preprint arXiv:2407.17467*, 2024.

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

[10] Peter J Huber. Robust estimation of a location parameter. In *Breakthroughs in statistics: Methodology and distribution*, pages 492–518. Springer, 1992.

[11] Jikai Jin, Vasilis Syrgkanis, Sham Kakade, and Hanlin Zhang. Discovering hierarchical latent capabilities of language models via causal representation learning. *arXiv preprint arXiv:2506.10378*, 2025.

[12] Feiyang Kang, Yifan Sun, Bingbing Wen, Si Chen, Dawn Song, Rafid Mahmood, and Ruoxi Jia. Autoscale: Scale-aware data mixing for pre-training llms. *arXiv preprint arXiv:2407.20177*, 2024.

[13] 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.

[14] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019.

[15] Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Yitzhak Gadre, Hritik Bansal, Etash Guha, Sedrick Scott Keh, Kushal Arora, et al. Datacomp-lm: In search of the next generation of training sets for language models. *Advances in Neural Information Processing Systems*, 37:14200–14282, 2024.

[16] Mingxin Li, Zhijie Nie, Yanzhao Zhang, Dingkun Long, Richong Zhang, and Pengjun Xie. Improving general text embedding model: Tackling task conflict and data imbalance through model merging. *arXiv preprint arXiv:2410.15035*, 2024.

[17] Qian Liu, Xiaosen Zheng, Niklas Muennighoff, Guangtao Zeng, Longxu Dou, Tianyu Pang, Jing Jiang, and Min Lin. Regmix: Data mixture as regression for language model pre-training. *arXiv preprint arXiv:2407.01492*, 2024.

[18] Zimu Lu, Aojun Zhou, Ke Wang, Houxing Ren, Weikang Shi, Junting Pan, Mingjie Zhan, and Hongsheng Li. Mathcoder2: Better math reasoning from continued pretraining on model-translated mathematical code. *arXiv preprint arXiv:2410.08196*, 2024.

[19] Yingwei Ma, Yue Liu, Yue Yu, Yuanliang Zhang, Yu Jiang, Changjian Wang, and Shanshan Li. At which training stage does code data help llms reasoning? *arXiv preprint arXiv:2309.16298*, 2023.

[20] Ian Magnusson, Nguyen Tai, Ben Bogin, David Heineman, Jena D Hwang, Luca Soldaini, Akshita Bhagia, Jiacheng Liu, Dirk Groeneveld, Oyvind Tafjord, et al. Datadecide: How to predict best pretraining data with small experiments. *arXiv preprint arXiv:2504.11393*, 2025.

[21] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016.

[22] Jorge Nocedal. Updating quasi-newton matrices with limited storage. *Mathematics of computation*, 35(151):773–782, 1980.

[23] Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Quan Ngoc Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The lambada dataset: Word prediction requiring a broad discourse context. *arXiv preprint arXiv:1606.06031*, 2016.

[24] Jason Phang, Herbie Bradley, Leo Gao, Louis Castricato, and Stella Biderman. Eleutherai: Going beyond" open science" to" science in the open". *arXiv preprint arXiv:2210.06413*, 2022.

[25] Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, et al. Dolma: An open corpus of three trillion tokens for language model pretraining research. *arXiv preprint arXiv:2402.00159*, 2024.

[26] Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari Morcos. Beyond neural scaling laws: beating power law scaling via data pruning. *Advances in Neural Information Processing Systems*, 35:19523–19536, 2022.

[27] Saeid Asgari Taghanaki, Aliasgahr Khani, and Amir Khasahmadi. Mmlu-pro+: Evaluating higher-order reasoning and shortcut learning in llms. *arXiv preprint arXiv:2409.02257*, 2024.

[28] Tristan Thrush, Christopher Potts, and Tatsunori Hashimoto. Improving pretraining data using perplexity correlations. *arXiv preprint arXiv:2409.05816*, 2024.

[29] Maurice Weber, Dan Fu, Quentin Anthony, Yonatan Oren, Shane Adams, Anton Alexandrov, Xiaozhong Lyu, Huu Nguyen, Xiaozhe Yao, Virginia Adams, et al. Redpajama: an open dataset for training large language models. *Advances in neural information processing systems*, 37:116462–116492, 2024.

[30] Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy S Liang, Quoc V Le, Tengyu Ma, and Adams Wei Yu. Doremi: Optimizing data mixtures speeds up language model pretraining. *Advances in Neural Information Processing Systems*, 36:69798–69818, 2023.

[31] Jiasheng Ye, Peiju Liu, Tianxiang Sun, Jun Zhan, Yunhua Zhou, and Xipeng Qiu. Data mixing laws: Optimizing data mixtures by predicting language modeling performance. *arXiv preprint arXiv:2403.16952*, 2024.

[32] Huimu Yu, Xing Wu, Haotian Xu, Debing Zhang, and Songlin Hu. Codepmp: Scalable preference model pretraining for large language model reasoning. *arXiv preprint arXiv:2410.02229*, 2024.

[33] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*, 2019.

[34] Junhao Zheng, Qianli Ma, Zhen Liu, Binquan Wu, and Huawen Feng. Beyond anti-forgetting: Multimodal continual instruction tuning with positive forward transfer. *arXiv preprint arXiv:2401.09181*, 2024.

Appendix A. Related Work

Scaling laws. A line of work shows that LLM loss follows smooth power laws as model capacity and data grow. Early results characterize scaling trends across parameters, data, and compute [13], while subsequent analyses refine the compute–data trade-off and argue that many models were under-trained for their size and recommend that token counts scale roughly with parameters to remain compute-optimal [9]. We treat these as the domain-agnostic baseline and study differences induced by heterogeneous data mixtures.

Domain-aware scaling and mixture optimization. Beyond total token count, several works show that *which* tokens matter. Data pruning and curation can "beat" naive power-law scaling by shifting the effective constants and exponents in favorable ways [26]. Mixture-optimization approaches explicitly treat domain composition as a control variable: AutoScale automates scaling-law fitting and uses the fitted laws to recommend data-mixture and broader training-design choices with minimal additional training [12]. RegMix frames mixture selection as a regression problem with regularization to stabilize estimates under limited supervision [17]. More recently, "data mixing laws" relate performance to mixture weights and document nonlinear returns, including phase-transition-like effects as specific domains are increased [31]. Our formulation contributes to this line by modeling domain effects via the empirical domain frequencies $u_{i,k}$ (fraction of tokens from domain k in the dataset used for model i), which allows the data term's exponent to vary for different domains and compositions, rather than only with D.

Observational inference of skills and benchmark structure. Orthogonal to controlled pretraining, a complementary literature uses observational variation across many (model, task) points to infer latent capability structure and transfer patterns. Perplexity-correlation methods identify promising upstream corpora for a target benchmark using only readily available statistics, providing a zero-shot signal for mixture selection [28]. Hierarchical latent-variable models on leaderboard matrices recover shared factors that explain co-movement of benchmark scores across models, offering a data-driven map of "skills" without interventional training runs [11]. Our estimator follows this observational spirit but focuses on predicting performance under counterfactual mixtures by leveraging the fraction of tokens in pre-training data from a specific domain.

Evidence for data synergy. Multiple empirical studies report positive transfer between code and mathematical reasoning. Continued pretraining on math+code corpora improves math benchmarks beyond either domain alone [1, 18]. Controlled ablations further indicate that injecting code during pretraining (rather than only SFT) yields broader reasoning gains with minimal negative transfer [19]. These findings motivate sparse, domain-specific parameters that can capture synergy between conceptually related domains.

Appendix B. Related Work

Scaling laws. A line of work shows that LLM loss follows smooth power laws as model capacity and data grow. Early results characterize scaling trends across parameters, data, and compute [13], while subsequent analyses refine the compute–data trade-off and argue that many models were under-trained for their size and recommend that token counts scale roughly with parameters to remain compute-optimal [9]. We treat these as the domain-agnostic baseline and study differences induced by heterogeneous data mixtures.

Domain-aware scaling and mixture optimization. Beyond total token count, several works show that *which* tokens matter. Data pruning and curation can "beat" naive power-law scaling by shifting the effective constants and exponents in favorable ways [26]. Mixture-optimization approaches explicitly treat domain composition as a control variable: AutoScale automates scaling-law fitting and uses the fitted laws to recommend data-mixture and broader training-design choices with minimal additional training [12]. RegMix frames mixture selection as a regression problem with regularization to stabilize estimates under limited supervision [17]. More recently, "data mixing laws" relate performance to mixture weights and document nonlinear returns, including phase-transition-like effects as specific domains are increased [31]. Our formulation contributes to this line by modeling domain effects via the empirical domain frequencies $u_{i,k}$ (fraction of tokens from domain k in the dataset used for model i), which allows the data term's exponent to vary for different domains and compositions, rather than only with D.

Observational inference of skills and benchmark structure. Orthogonal to controlled pretraining, a complementary literature uses observational variation across many (model, task) points to infer latent capability structure and transfer patterns. Perplexity-correlation methods identify promising upstream corpora for a target benchmark using only readily available statistics, providing a zero-shot signal for mixture selection [28]. Hierarchical latent-variable models on leaderboard matrices recover shared factors that explain co-movement of benchmark scores across models, offering a data-driven map of "skills" without interventional training runs [11]. Our estimator follows this observational spirit but focuses on predicting performance under counterfactual mixtures by leveraging the fraction of tokens in pre-training data from a specific domain.

Evidence for data synergy. Multiple empirical studies report positive transfer between code and mathematical reasoning. Continued pretraining on math+code corpora improves math benchmarks beyond either domain alone [1, 18]. Controlled ablations further indicate that injecting code during pretraining (rather than only SFT) yields broader reasoning gains with minimal negative transfer [19]. These findings motivate sparse, domain-specific parameters that can capture synergy between conceptually related domains.

Appendix C. Additional Details

C.1. First-order Domain→ Benchmark Synergy

De-confounding global and domain effects. To decouple the aggregate data-scaling coefficient β_j from domain synergies $\gamma_{j,\cdot}$, we impose the mean-zero (orthogonality) constraint $\mathbf{1}^\top \gamma_{j,\cdot} = 0$. Let $P_\perp := I - \frac{1}{K}\mathbf{1}\mathbf{1}^\top$ be the projector onto the subspace orthogonal to $\mathbf{1}$. We enforce this either by *reparameterization* $\gamma_{j,\cdot} = P_\perp \eta_{j,\cdot}$ with free $\eta_{j,\cdot}$. Using $z_{i,k} = u_{i,k} \log(u_{i,k} d_i)$ and $H(u_i) = -\sum_k u_{i,k} \log u_{i,k}$, recall $\mathbf{1}^\top z_i = \log d_i - H(u_i)$. Without the constraint, the data term

$$(\beta_j \mathbf{1} + \gamma_{j,\cdot})^\top z_i + \beta_j H(u_i)$$

lets the mean of $\gamma_{j,\cdot}$ shift the effective coefficient on $\log d_i$ and spuriously couple to $H(u_i)$. Enforcing $\mathbf{1}^\top \gamma_{j,\cdot} = 0$ removes this confounding and reduces the data term to

$$D_{i,j} = \beta_j \log d_i + \gamma_{j,\cdot}^\top z_i,$$

so β_j is identifiable as the aggregate token-scaling coefficient while $\gamma_{j,\cdot}$ captures only domain-specific deviations. We apply sparsity and shrinkage penalties to the projected coefficients $P_\perp \gamma_{j,\cdot}$.

C.2. Second-order Pre-Training Data Synergy

Scaling–law interpretation (effective tokens). Define the benchmark-specific *effective tokens*

$$\log D^{\text{eff}}_{i,j} := \log d_i \;+\; \sum_{k<k'} (\gamma_{j,k} + \gamma_{j,k'})\, \sigma_{kk'}\, s_{i,kk'}, \quad D^{\text{eff}}_{i,j} = d_i \exp\Big(\sum_{k<k'} (\gamma_{j,k} + \gamma_{j,k'})\, \sigma_{kk'}\, s_{i,kk'}\Big),$$

so that

$$\exp(\Phi_{i,j}) = \exp\big(b_j - \gamma_{j,\cdot}^{\top} z_i\big)\ \big(D^{\text{eff}}_{i,j}\big)^{-\beta_j}.$$

In a Chinchilla-style view,

$$L_{i,j}(N_i, d_i) \;\approx\; L_{\infty,j} + A_j N_i^{-\alpha_j} + \underbrace{\widetilde{B}_{i,j}}_{=\exp(b_j - \gamma_{j,\cdot}^{\top} z_i)} \big(D^{\text{eff}}_{i,j}\big)^{-\beta_j},$$

and for small interactions, $\big(D^{\text{eff}}_{i,j}\big)^{-\beta_j} \approx d_i^{-\beta_j}\,\big[1 - \beta_j \sum_{k<k'} (\gamma_{j,k} + \gamma_{j,k'})\, \sigma_{kk'}\ \operatorname{softmin}_{\tau}(\bar{z}_{i,k}, \bar{z}_{i,k'})\big]$, which makes explicit the benchmark-gated "bonus tokens" contributed by co-occurence through the universal Σ.

Appendix D. Training and Uncertainty

Loss and optimizer. We fit all parameters by minimizing a single Huber$_\delta$ risk [10] over the LSE scaffold (Section 2):

$$\min_{\Theta} \sum_{j=1}^{n} \sum_{i=1}^{m} \text{Huber}_{\delta}\Big(\text{LSE}(e_j,\ a_j - \alpha_j s_i,\ \Phi_{i,j}) - l_{i,j}\Big) \;+\; \lambda_1 \sum_j \|\gamma_{j,\cdot}\|_1 \;+\; \lambda_2 \sum_j \|\gamma_{j,\cdot}\|_2^2,$$

where $s_i = \log N_i$ and $\Phi_{i,j}$ is either from the *domain-benchmark* data term (i.e., $\Sigma = 0$) or the *pretraining data synergy* term in Eq. (3). We optimize with full-batch L-BFGS [22] The pretraining domain synergy matrix Σ is parameterized on the upper triangle only ($\sigma_{kk'} = \sigma_{k'k}$, $\sigma_{kk} = 0$). In all reported runs we use $\delta = 0.5$, $\tau = 0.1$, $\lambda_1 = 0.01$, and $\lambda_2 = 10^{-5}$. For the synergy model we first optimize $(e_j, a_j, b_j, \alpha_j, \beta_j, \gamma_{j,\cdot})$ with Σ fixed to 0 and then continuing optimizing all parameters (including Σ) jointly.

Uncertainty via bootstrap. *first-order fits.* We estimate variability by resampling 80% of models (without replacement) $B = 50$ times and refitting; we report percentile intervals for $(e_j, a_j, b_j, \alpha_j, \beta_j, \gamma_{j,\cdot})$.

second-order fits. We warm-start with training on all models, and then we cluster bootstrap over model families: each replicate resamples families with replacement, retains all domains per sampled model, and refits the full objective from the warm start. With $B = 50$ replicates we report 90% percentile intervals for (α_j, β_j), the per-domain effects $\gamma_{j,k}$, and the shared pairwise coefficients $\sigma_{kk'}$.

Appendix E. Observational Data: Models and Data

We estimate composition effects from *observational variation* across open-weight models and their publicly documented pre-training mixtures, rather than training models. This provides variation in both scale and composition, and enables identifying of domain-specific returns using our framework. Throughout we denote by $u_{i,k}$ the fraction of tokens from training domain k in the dataset used for model i.

Model families Our panel spans six open-weight families with heterogeneous scales and training mixtures: `GPT-Neo/J/NeoX` [24] (125M–20B; 5 checkpoints), `Pythia` [2] (70M–12B; 8 checkpoints), `DataDecide` [20] (150M–1B; `30` checkpoints across Dolma variants with systematic ablations: *no-code*, *no-flan*, *no-math-code*, *no-reddit*), `OLMo` [7] (1B–13B; 5 checkpoints), `OpenLLaMA` [6] (3B–13B; 5 checkpoints), and `RedPajama-INCITE` [29] (3B–7B; 2 checkpoints). The DataDecide ablations provide controlled composition shifts that are especially informative for informing counterfactual domain effects. We map each subset in each model's pretraining data to one of the following domains: books, code, encyclopedia, legal, math, Q&A, Science, and Web.

Benchmarks and normalization. We evaluate on eight benchmarks chosen to roughly cover our domains: mathematical reasoning (`GSM8K` [5]), code generation (`HumanEval` [3]), science/general-knowledge MC (`ARC-Easy`[4]), commonsense inference (`HellaSwag` [33]), broad-context cloze (`LAMBADA` [23]), professional legal knowledge (`MMLU-Pro-Plus, Law` [27]), open-domain QA (`Natural Questions` [14]), and encyclopedia (`WikiText` [21]). To compare different evaluation metrics, we convert accuracy-/pass@1-type scores to error ($1-$success), and apply a per-task rank transform to obtain the pseudo-loss used for fitting.

Evaluation. In most benchmarks we do not directly observe direct loss $l_{i,j}$ but a task metric $m_{i,j}$ (accuracy, pass@1, etc.). We map metrics to a *pseudo log-loss* via a domain-specific monotone transform g_j (e.g., rank–Gaussian (inverse normal) transform $g_j(m) = \Phi^{-1}\big(\hat{F}_j(m)\big)$, where $\hat{F}_j$ is the empirical CDF of metric values on D_j and Φ^{-1} is the standard normal quantile function):
$l_{i,j} = g_j(m_{i,j}) \ \propto\ \log L_{i,j}$, so different metrics lie on a common (log-loss) scale.

Appendix F. Additional Results

F.1. Comparison Results

We also assess how predictive the model is on held-out samples. Figure 3 compares our domain-aware estimator with a domain-agnostic ($\gamma_{i,j} = 0$) baseline. Across a number of benchmarks, the synergy-aware models show better loss on residuals of held-out examples, on these datasets.

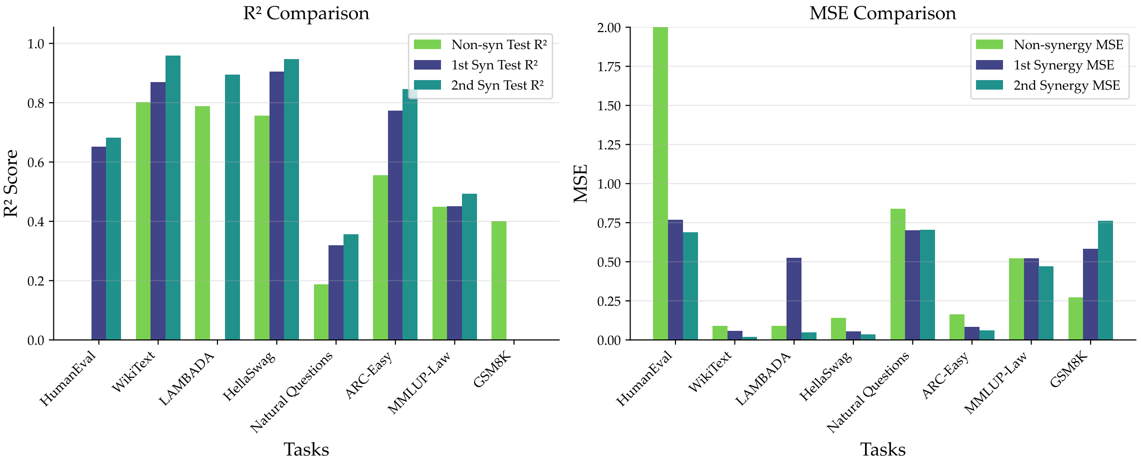


Figure 4: Comparison of performance between first- and second-order synergy-aware model vs. non-synergy-aware model

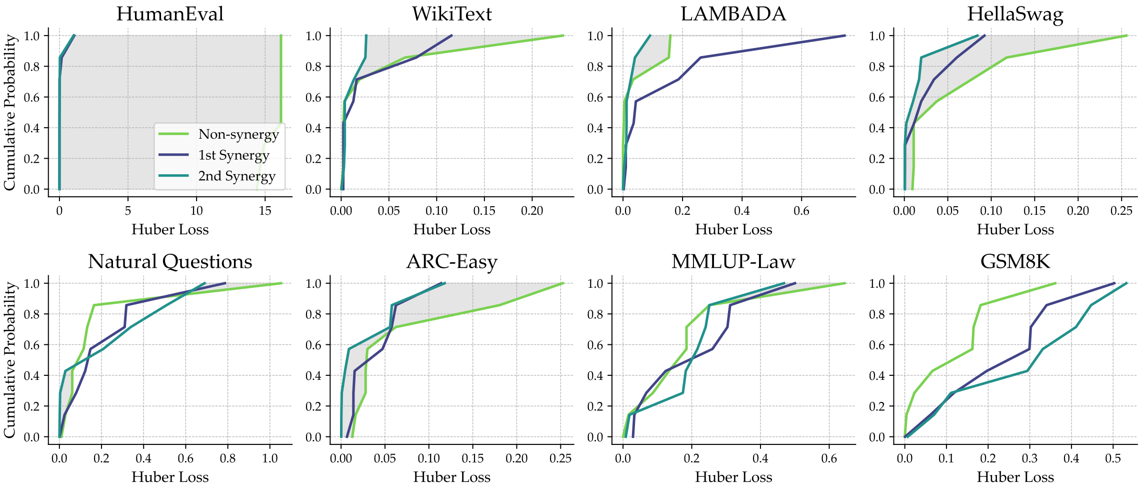


Figure 5: Comparison of test loss between first- and second-order synergy-aware model vs. non-synergy-aware model

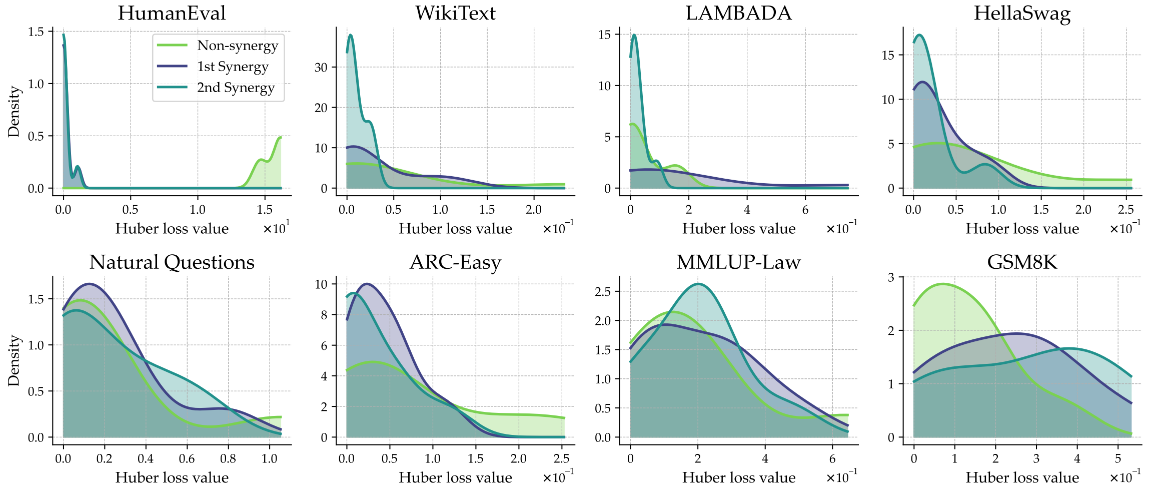
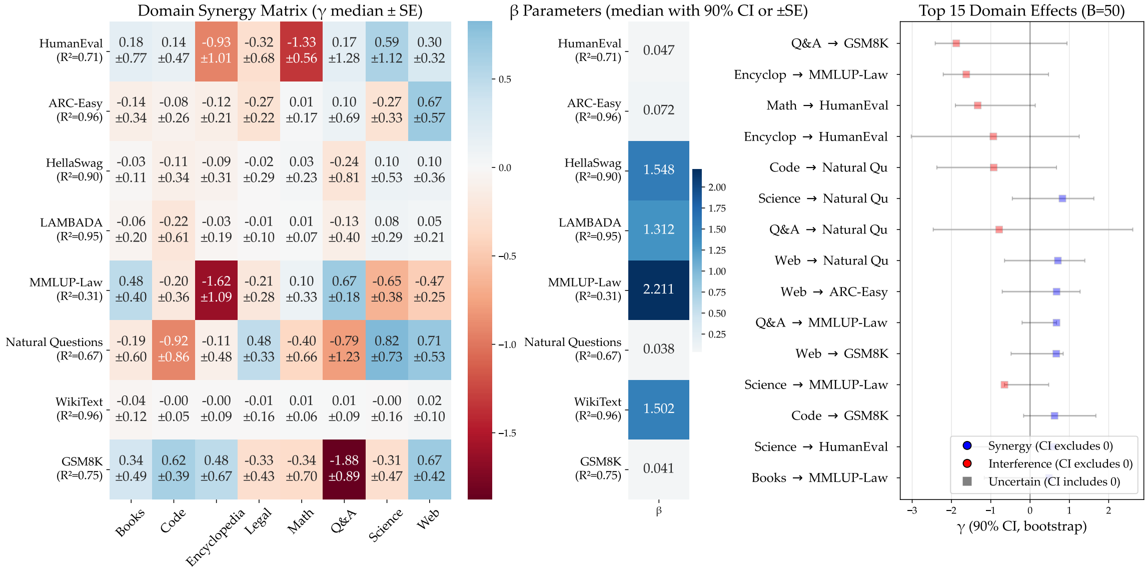


Figure 6: Comparison of test loss between first- and second-order synergy-aware model vs. non-synergy-aware model

F.2. Second-order Pre-training Synergy

We provide γ and β values for the second-order synergy model.


Domain Synergy Matrix (γ median ± SE)
HumanEval (R²=0.71): 0.18 ±0.77, 0.14 ±0.47, -0.93 ±1.01, -0.32 ±0.68, -1.33 ±0.56, 0.17 ±1.28, 0.59 ±1.12, 0.30 ±0.32
ARC-Easy (R²=0.96): -0.14 ±0.34, -0.08 ±0.26, -0.12 ±0.21, -0.27 ±0.22, 0.01 ±0.17, 0.10 ±0.69, -0.27 ±0.33, 0.67 ±0.57
HellaSwag (R²=0.90): -0.03 ±0.11, -0.11 ±0.34, -0.09 ±0.31, -0.02 ±0.29, 0.03 ±0.23, -0.24 ±0.81, 0.10 ±0.53, 0.10 ±0.36
LAMBADA (R²=0.95): -0.06 ±0.20, -0.22 ±0.61, -0.03 ±0.19, -0.01 ±0.10, 0.01 ±0.07, -0.13 ±0.40, 0.08 ±0.29, 0.05 ±0.21
MMLUP-Law (R²=0.31): 0.48 ±0.40, -0.20 ±0.36, -1.62 ±1.09, -0.21 ±0.28, 0.10 ±0.33, 0.67 ±0.18, -0.65 ±0.38, -0.47 ±0.25
Natural Questions (R²=0.67): -0.19 ±0.60, -0.92 ±0.86, -0.11 ±0.48, 0.48 ±0.33, -0.40 ±0.66, -0.79 ±1.23, 0.82 ±0.73, 0.71 ±0.53
WikiText (R²=0.96): -0.04 ±0.12, -0.00 ±0.05, -0.00 ±0.09, -0.01 ±0.16, 0.01 ±0.06, 0.01 ±0.09, -0.00 ±0.16, 0.02 ±0.10
GSM8K (R²=0.75): 0.34 ±0.49, 0.62 ±0.39, 0.48 ±0.67, -0.33 ±0.43, -0.34 ±0.70, -1.88 ±0.89, -0.31 ±0.47, 0.67 ±0.42
Books
Code
Encyclopedia
Legal
Math
Q&A
Science
Web
β Parameters (median with 90% CI or ±SE)
HumanEval (R²=0.71) 0.047
ARC-Easy (R²=0.96) 0.072
HellaSwag (R²=0.90) 1.548
LAMBADA (R²=0.95) 1.312
MMLUP-Law (R²=0.31) 2.211
Natural Questions (R²=0.67) 0.038
WikiText (R²=0.96) 1.502
GSM8K (R²=0.75) 0.041
β
Top 15 Domain Effects (B=50)
Q&A → GSM8K
Encyclop → MMLUP-Law
Math → HumanEval
Encyclop → HumanEval
Code → Natural Qu
Science → Natural Qu
Q&A → Natural Qu
Web → Natural Qu
Web → ARC-Easy
Q&A → MMLUP-Law
Web → GSM8K
Science → MMLUP-Law
Code → GSM8K
Science → HumanEval
Books → MMLUP-Law
● Synergy (CI excludes 0)
● Interference (CI excludes 0)
■ Uncertain (CI includes 0)
γ (90% CI, bootstrap)