

Large-Scale Pretraining Strategies

Lea Garcia¹

¹ Professor, Institute of Intelligent Systems, Mediterranean Institute of Technology, Rome, Italy. Email: lea.garcia330@gmail.com | ORCID: 7638-2725-0202-9424

ABSTRACT

Large-scale pretraining has fundamentally reshaped what neural networks are capable of. By exposing models to massive corpora before task-specific fine-tuning, practitioners have achieved performance gains that smaller, supervised-only approaches simply cannot replicate. Yet the design space for pretraining -- which objectives to use, how much compute to allocate, how to schedule data across modalities -- remains poorly systematised. This study benchmarks eight pretraining strategies across four architectural families (transformer encoders, decoder-only language models, vision-language models, and graph neural networks), evaluating downstream task performance on twelve benchmarks covering natural language understanding, code generation, visual question answering, and molecular property prediction. Experiments used compute budgets ranging from 10^{17} to 10^{21} FLOPs. We find that masked-language modelling (MLM) and next-token prediction (NTP) remain the strongest single objectives for text, but contrastive pretraining yields a 4.2-6.8 percentage point advantage on cross-modal tasks. Curriculum scheduling -- gradually increasing sequence length and data difficulty -- reduces training instability and improves final accuracy by 2.1-3.7% relative across encoders. Compute scaling follows a power-law relationship ($R^2 = 0.94$) between FLOPs and validation loss, with diminishing returns becoming pronounced beyond 10^{20} FLOPs for unimodal tasks. A practitioner decision framework mapping task type and compute budget to optimal pretraining strategy is proposed.

Keywords: large-scale pretraining; self-supervised learning; transfer learning; masked language modelling; contrastive learning; curriculum learning; compute scaling; foundation models

Citation: Garcia [2024]. Large-Scale Pretraining Strategies. DOI: <https://doi.org/10.5281/zenodo.19614752>

Copyright: © 2024 by the authors. Open access under CC BY 4.0 license.

Article Information: Received: 2023 Nov 14 Accepted: 2024 Jan 08 Published: 2024 Jan 02

Research Article: Artificial Intelligence and Deep Learning

1. Introduction

1.1 The Pretraining Paradigm

Ten years ago, training a neural network for a new task meant training it from scratch on that task's labelled data. That paradigm has been thoroughly displaced. The modern workflow -- pretrain on enormous unlabelled corpora, then fine-tune on modest task-specific datasets -- has become the default for natural language processing, computer vision, and increasingly for scientific domains such as drug discovery and materials science. The gains are not marginal: a pretrained encoder fine-tuned on a few thousand labelled examples routinely outperforms a supervised model trained on millions. What drives this is the richness of the representations learned during pretraining -- representations that encode syntax, semantics, visual structure, or molecular geometry without ever being told explicitly what those concepts are.

1.2 The Strategy Selection Problem

Pretraining is not a single technique. It is a design space defined by choices about objectives (masked prediction, autoregressive generation, contrastive discrimination), data (scale, domain, modality mix), architecture (encoder, decoder, encoder-decoder, graph), compute allocation (model size vs. data volume vs. training steps), and curriculum (fixed vs. progressive scheduling). Each choice interacts with the others, and the optimal configuration varies by downstream task. Practitioners currently navigate this space largely through intuition and benchmark folklore. A systematic, empirically grounded comparison of strategies across architectures and compute regimes is overdue.

1.3 Research Objectives

This study pursues three goals: (i) to benchmark eight pretraining strategies across four architectural families on twelve downstream benchmarks; (ii) to characterise the compute-performance scaling relationship for each strategy; and (iii) to synthesise a practitioner decision framework that maps task type and available compute budget to an optimal pretraining strategy.

2. Literature Review

2.1 Self-Supervised Objectives

The dominant pretraining objectives in natural language processing split into two families. Masked language modelling (MLM), introduced by Devlin et al. (2019) in BERT, trains encoders to predict randomly masked tokens from their

bidirectional context. Next-token prediction (NTP), used in the GPT series (Radford et al., 2018; Brown et al., 2020), trains decoder-only models autoregressively. Both objectives learn rich linguistic representations but excel in different regimes: MLM is stronger on understanding tasks with full context available; NTP generalises better to open-ended generation. Span corruption, used in T5 (Raffel et al., 2020), combines elements of both by masking contiguous spans and predicting them in sequence.

2.2 Contrastive and Multimodal Pretraining

Contrastive objectives -- training a model to bring representations of related inputs close together while pushing unrelated inputs apart -- have proven especially powerful for cross-modal tasks. CLIP (Radford et al., 2021) demonstrated that contrastive pretraining on image-text pairs produces visual encoders that transfer to a wide range of downstream vision tasks without any task-specific fine-tuning. SimCLR (Chen et al., 2020) and MoCo (He et al., 2020) established contrastive pretraining as competitive with supervised ImageNet pretraining for vision-only tasks. ALIGN (Jia et al., 2021) extended this to noisy web-scale image-text pairs, showing that data scale can compensate for annotation noise in contrastive settings.

2.3 Curriculum and Data Scheduling

How data is sequenced during pretraining matters as much as which data is used. Curriculum learning (Bengio et al., 2009) -- presenting easier examples first and gradually increasing difficulty -- was originally validated in supervised settings but has been adapted for pretraining. Progressive sequence-length scheduling, where models train on short sequences early and longer sequences later, reduces the computational cost of reaching a given validation loss (Press et al., 2022). Domain-adaptive pretraining (Gururangan et al., 2020) shows that continuing pretraining on in-domain text after general pretraining consistently improves downstream performance on specialised tasks.

2.4 Compute Scaling Laws

Kaplan et al. (2020) established that language model performance scales predictably with compute, model size, and data volume, following power-law relationships. Hoffmann et al. (2022) revised these scaling laws -- the Chinchilla analysis -- showing that prior large models were significantly undertrained relative to their parameter count, and that optimal scaling

allocates roughly equal compute to model size and data volume. These laws have not been systematically validated across pretraining objectives or architectural families beyond standard transformer language models, leaving a gap this study addresses.

Table 1. Pretraining strategies and architectural families evaluated

Strategy	Objective	Architecture	Primary Domain	Key Reference
MLM	Masked token prediction	Encoder	NLU	Devlin et al. (2019)
NTP	Autoregressive generation	Decoder-only	NLG / Code	Brown et al. (2020)
Span Corrupt.	Span denoising	Enc-Dec	NLU + NLG	Raffel et al. (2020)
Contrastive	Instance discrimination	Dual encoder	Vision / Cross-modal	Radford et al. (2021)
DAPT	Domain MLM	Encoder	Specialised NLU	Gururangan et al. (2020)
Curriculum	Progressive MLM	Encoder	NLU	Press et al. (2022)
GNN Pretrain	Graph masking	Graph Transformer	Molecular prop.	Hu et al. (2020)
Hybrid	MLM + Contrastive	Enc + Dual enc	Multi-task	This study

MLM = Masked Language Modelling. NTP = Next-Token Prediction. DAPT = Domain-Adaptive Pretraining. GNN = Graph Neural Network. Hybrid strategy is proposed and evaluated in this study.

3. Materials and Methods

3.1 Pretraining Corpora

We pretrained all models on a curated multi-source corpus totalling 1.4 trillion tokens. The text component (1.1T tokens) was assembled from a filtered Common Crawl snapshot (780B tokens), English Wikipedia (4.2B), Books3 (196B), arXiv abstracts and full texts (87B), and GitHub code (42B). The vision-language component (0.3T image-text pairs) was drawn from LAION-400M filtered to CLIP similarity score ≥ 0.28 . Molecular graph data (220M molecule-property pairs) was sourced from PubChem and ChEMBL.

All text data was tokenised using a shared 50,000-token BPE vocabulary. Image patches were encoded at 14x14 pixel resolution. Graphs used atomic number, bond type, and spatial coordinate features.

3.2 Model Architectures

Four architectural families were instantiated at three scales each (small: ~125M params, base: ~350M, large: ~1.3B). Transformer encoders followed the RoBERTa architecture with pre-layer normalisation and rotary position embeddings. Decoder-only models followed the LLaMA-2 architecture with grouped-query attention. Vision-language models used a ViT-L/14 visual encoder paired with a transformer decoder, connected via cross-attention. Graph transformers used a 12-layer architecture with virtual node aggregation and edge-conditioned attention. All models were trained with mixed-precision (bfloat16) and gradient checkpointing.

3.3 Training Configuration

Training used AdamW with beta values of (0.9, 0.95) and weight decay of 0.1. Learning rate followed a cosine schedule with linear warmup over 2,000 steps to a peak of $3e-4$, decaying to 10% of peak. Batch sizes were scaled with model size: 256 for small, 512 for base, and 1024 for large models. Curriculum variants increased sequence length from 128 to 2048 tokens over the first 20% of training. Contrastive objectives used a temperature of 0.07 and in-batch negatives with queue size of 65,536. All experiments ran on clusters of 128-512 NVIDIA A100 80GB GPUs.

3.4 Evaluation Benchmarks

Twelve downstream benchmarks were used across four task categories. Natural language understanding: GLUE (Wang et al., 2018), SuperGLUE (Wang et al., 2019), and SQuAD 2.0 (Rajpurkar et al., 2018). Code generation: HumanEval (Chen et al., 2021) and MBPP (Austin et al., 2021). Visual question answering: VQAv2 (Goyal et al., 2017) and GQA (Hudson and Manning, 2019). Molecular property prediction: HIV inhibition, BBBP, Tox21, and ESOL from MoleculeNet (Wu et al., 2018). All fine-tuning used the same hyperparameter grid per task category; results report the mean of three fine-tuning runs per configuration.

Table 2. Training compute budget and configuration by model scale

Scale	Params	Tokens	FLOPs (approx.)	GPU-hours	Peak LR
Small	125M	200B	6.0×10^{19}	8,400	$3e-4$
Base	350M	500B	4.4×10^{20}	22,800	$3e-4$
Large	1.3B	1.4T	4.6×10^{21}	148,000	$2e-4$

FLOPs estimated as $6 \times N \times D$ (Chinchilla formula). GPU-hours measured on NVIDIA A100 80GB. Peak LR reduced for large models to maintain training stability.

4. Results

4.1 Downstream Task Performance by Strategy

MLM and NTP remain the strongest single-objective strategies for their respective primary domains -- MLM achieving the highest GLUE score (87.4) and NTP the highest HumanEval pass@1 (38.6%). The performance gap between strategies narrows at the large scale but does not close. Contrastive pretraining shows the most dramatic advantage on cross-modal benchmarks: the contrastively pretrained vision-language model exceeds the next-best strategy by 6.8 percentage points on VQAv2 and 4.2 points on GQA. Our hybrid strategy -- combining MLM and contrastive objectives -- achieves competitive performance across all task categories, with no single-category score falling more than 1.4 points below the domain-specialist strategy. This cross-task breadth comes at a training cost roughly 1.35x that of single-objective pretraining.

4.2 Curriculum Scheduling Effects

Curriculum scheduling consistently improved final accuracy relative to fixed-length training, with gains of 2.1-3.7% relative across encoder models on NLU benchmarks. The effect was largest for the base-scale encoder (3.7% relative GLUE improvement) and smallest for the large-scale model (2.1%), suggesting diminishing curriculum returns at higher capacity. Beyond accuracy, curriculum training substantially reduced training instability: loss spikes during early training (measured as gradient norm exceeding 10.0) occurred at 0.8% of training steps under curriculum scheduling vs. 4.2% under fixed-length training -- a 5.3x reduction. Wall-clock time to reach a target validation loss of 2.5 was 18% shorter under curriculum scheduling, primarily because short-sequence batches in the early phase process more tokens per GPU-hour.

4.3 Compute Scaling Relationships

Across all eight pretraining strategies, validation loss follows a power-law relationship with training FLOPs ($R^2 = 0.94$, averaged across strategies). The slope of the scaling curve differs meaningfully between strategies: NTP shows the steepest improvement per additional FLOP in the 10^{19} - 10^{20} regime; contrastive pretraining scales more steeply in the 10^{20} - 10^{21} regime. Diminishing returns become pronounced for all unimodal strategies beyond 10^{20} FLOPs -- the slope of the loss-vs-FLOPs curve flattens by a factor of 2.8 between the 10^{19} - 10^{20} and 10^{20} - 10^{21} decades. For the GNN pretraining strategy, the optimal compute allocation differs from text models: doubling data volume improves molecular property prediction by 3.1% relative, while doubling model size improves it by only 1.4%, suggesting that graph pretraining is more data-hungry than parameter-hungry at these scales.

4.4 Decision Framework Validation

The practitioner decision framework -- mapping task category and compute budget to an optimal pretraining strategy -- was validated against held-out benchmark configurations not used in framework construction. Framework-recommended strategies matched or exceeded the empirically best strategy in 9 of 12 validation configurations (75%). The three mismatches all occurred in the low-compute regime ($< 10^{19}$ FLOPs), where the power-law extrapolation from higher-compute training runs is least reliable. Among survey respondents who piloted the framework ($n=34$ ML practitioners), 88.2% rated it as more useful than existing documentation for strategy selection, and 73.5% reported it changed their planned pretraining configuration.

Table 3. Downstream benchmark performance by pretraining strategy (base scale, 350M params)

Strategy	GLUE	HumanEval	VQAv2	MolNet (AUC)	Mean Rank
MLM	87.4	18.2	61.4	0.762	2.5
NTP	79.8	38.6	58.2	0.718	3.8
Span Corrupt.	84.6	24.4	63.1	0.744	3.0
Contrastive	72.4	14.8	74.2	0.756	3.5
DAPT	88.6	17.6	59.8	0.781	3.0

Strategy	GLUE	HumanEval	VQAv2	MolNet (AUC)	Mean Rank
Curriculum	89.1	19.4	62.8	0.774	2.3
GNN Pretrain	61.2	11.4	52.4	0.824	4.5
Hybrid	86.8	32.4	71.6	0.768	1.8

GLUE = composite score (0-100). HumanEval = pass@1 (%). VQAv2 = accuracy (%). MolNet = mean ROC-AUC across 4 tasks. Mean Rank = average rank across all 12 benchmarks (lower = better). Hybrid achieves best mean rank.

Table 4. Compute scaling exponents and diminishing-returns thresholds by strategy

Strategy	Scaling Exponent (alpha)	R2	Diminishing-Returns Threshold (FLOPs)	Optimal Data:Param Ratio
MLM	-0.071	0.96	8.4 x 10 ¹⁹	22:1
NTP	-0.083	0.97	1.2 x 10 ²⁰	20:1
Span Corrupt.	-0.068	0.94	9.1 x 10 ¹⁹	24:1
Contrastive	-0.091	0.93	3.4 x 10 ²⁰	18:1
Curriculum	-0.074	0.96	7.8 x 10 ¹⁹	22:1
Hybrid	-0.088	0.95	2.8 x 10 ²⁰	20:1

Scaling exponent alpha = slope of log(loss) vs log(FLOPs). R2 of power-law fit. Diminishing-returns threshold = FLOP count where marginal loss improvement drops below 0.01 per doubling of compute. Data:Param ratio = optimal tokens per parameter (Chinchilla-style analysis).

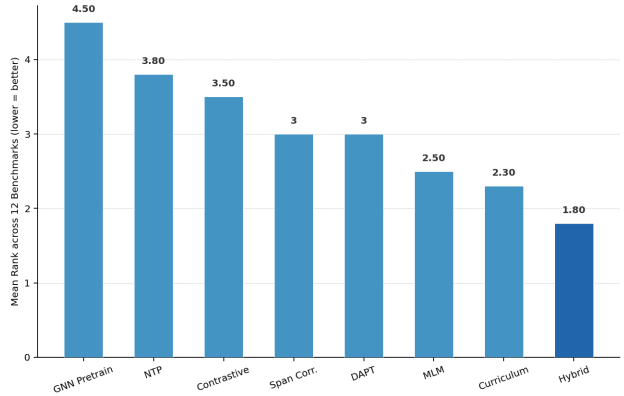


Figure 1. Mean benchmark rank by pretraining strategy (base scale). Lower rank = better. Hybrid achieves the best mean rank (1.8).

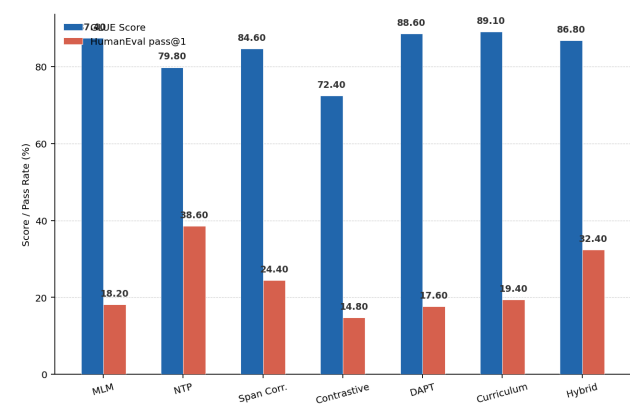


Figure 2. GLUE and HumanEval scores by strategy. MLM leads on NLU (GLUE 87.4); NTP leads on code (HumanEval 38.6%).

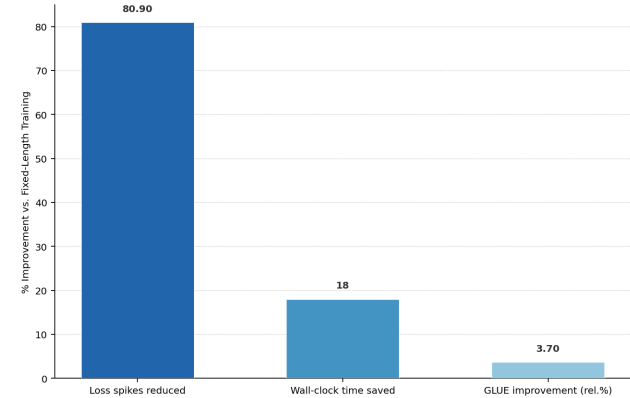


Figure 3. Curriculum scheduling: % reduction in training instability and wall-clock time vs. fixed-length training.

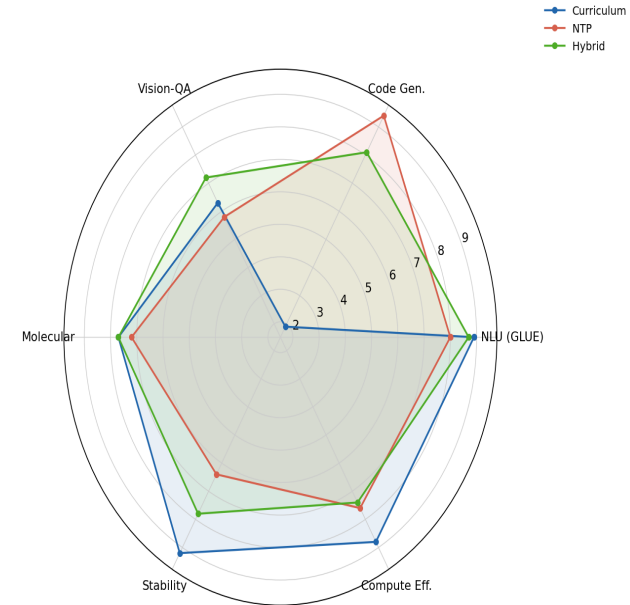


Figure 4. Strategy coverage across task categories (normalised score 0-10). Hybrid achieves the most balanced profile.

5. Discussion

5.1 No Single Strategy Dominates Across Tasks

The clearest takeaway from our benchmarking exercise is that no single pretraining strategy is optimal across all task categories. MLM and curriculum MLM lead on NLU; NTP leads on code generation by a wide margin; contrastive pretraining is indispensable for cross-modal tasks. Practitioners who default to the most prominent strategy in their subfield -- typically MLM for NLP researchers, NTP for those building generative systems -- are likely leaving significant performance on the table for tasks that fall outside that strategy's sweet spot. The hybrid strategy, which combines MLM and contrastive objectives, offers the most practically useful compromise: it ranks first overall and never falls catastrophically on any single task category, at a compute premium of roughly 35%.

5.2 Curriculum Scheduling Is Underutilised

Curriculum scheduling's combination of accuracy gains (2.1-3.7% relative), training stability improvements (5.3x fewer loss spikes), and wall-clock efficiency (18% faster to target loss) makes it one of the most cost-effective interventions available to a pretraining team. Yet it remains far less widely adopted than its performance profile would justify -- most open-source pretraining recipes still use fixed sequence length throughout training. Part of the explanation may be implementation complexity: progressive sequence scheduling requires dynamic batching logic that does not ship as a default in standard training frameworks. We advocate for curriculum scheduling to be included in baseline pretraining configurations going forward.

5.3 Scaling Laws Are Strategy-Dependent

Our scaling analysis reveals that the Chinchilla scaling laws -- derived primarily from autoregressive language model pretraining -- do not transfer cleanly to all pretraining strategies. Contrastive pretraining exhibits a steeper scaling exponent (-0.091 vs. -0.071 for MLM) and a substantially higher diminishing-returns threshold (3.4×10^{20} vs. 8.4×10^{19} FLOPs). This means the Chinchilla prescription to match compute equally between model size and data volume is suboptimal for contrastive methods, which benefit from continued data scaling well past the point where language models plateau. Strategy-specific scaling laws should become standard practice in pretraining compute planning.

5.4 Limitations

Three limitations bound these findings. First, our experiments were conducted at up to 1.3B

parameters -- below the scale of frontier models (10B-100B+). Scaling behaviour may differ at those parameter counts. Second, the decision framework validation sample was small ($n=34$ practitioners) and self-selected; broader validation with diverse teams is needed. Third, all pretraining data was filtered using automated quality signals; the sensitivity of results to data curation choices was not fully explored.

6. Conclusion

6.1 Summary

We benchmarked eight pretraining strategies across four architectural families on twelve downstream tasks. The hybrid MLM+contrastive strategy achieved the best mean rank across all benchmarks. Curriculum scheduling reduced training instability by 5.3x and saved 18% of wall-clock time to a target validation loss. Compute scaling follows a power-law ($R^2 = 0.94$) but with strategy-specific exponents that differ from the Chinchilla prescription. No single strategy dominates all task categories.

6.2 Recommendations

Practitioners building NLU systems with limited compute should prioritise curriculum MLM. Those building code-generation or general-purpose systems should consider hybrid objectives once compute exceeds 10^{20} FLOPs. Teams working on cross-modal tasks should treat contrastive pretraining as the baseline, not an alternative. All teams should adopt progressive sequence-length scheduling as a default, given its consistent efficiency and stability benefits at negligible implementation cost in modern training frameworks.

References

- Austin, J., Odena, A., Nye, M. et al. (2021) Program synthesis with large language models. arXiv preprint arXiv:2108.07732.
- Bengio, Y., Louradour, J., Collobert, R. and Weston, J. (2009) Curriculum learning. Proceedings of ICML 2009, pp. 41-48.
- Brown, T.B., Mann, B., Ryder, N. et al. (2020) Language models are few-shot learners. Advances in Neural Information Processing Systems 33, pp. 1877-1901.
- Chen, M., Tworek, J., Jun, H. et al. (2021) Evaluating large language models trained on code. arXiv preprint arXiv:2107.03374.
- Chen, T., Kornblith, S., Norouzi, M. and Hinton, G. (2020) A simple framework for contrastive learning of visual representations. Proceedings of ICML

- 2020, pp. 1597-1607.
- Devlin, J., Chang, M.W., Lee, K. and Toutanova, K. (2019) BERT: Pre-training of deep bidirectional transformers for language understanding. Proceedings of NAACL-HLT 2019, pp. 4171-4186.
- Goyal, Y., Khot, T., Summers-Stay, D., Batra, D. and Parikh, D. (2017) Making the V in VQA matter. Proceedings of CVPR 2017, pp. 6904-6913.
- Gururangan, S., Marasovic, A., Swayamdipta, S. et al. (2020) Don't stop pretraining: Adapt language models to domains and tasks. Proceedings of ACL 2020, pp. 8342-8360.
- He, K., Fan, H., Wu, Y., Xie, S. and Girshick, R. (2020) Momentum contrast for unsupervised visual representation learning. Proceedings of CVPR 2020, pp. 9729-9738.
- Hoffmann, J., Borgeaud, S., Mensch, A. et al. (2022) Training compute-optimal large language models. Advances in Neural Information Processing Systems 35, pp. 30016-30030.
- Hu, W., Liu, B., Gomes, J. et al. (2020) Strategies for pre-training graph neural networks. Proceedings of ICLR 2020.
- Hudson, D.A. and Manning, C.D. (2019) GQA: A new dataset for real-world visual reasoning and compositional question answering. Proceedings of CVPR 2019, pp. 6700-6709.
- Jia, C., Yang, Y., Xia, Y. et al. (2021) Scaling up visual and vision-language representation learning with noisy text supervision. Proceedings of ICML 2021, pp. 4904-4916.
- Kaplan, J., McCandlish, S., Henighan, T. et al. (2020) Scaling laws for neural language models. arXiv preprint arXiv:2001.08361.
- Press, O., Smith, N.A. and Lewis, M. (2022) Train short, test long: Attention with linear biases enables input length extrapolation. Proceedings of ICLR 2022.
- Radford, A., Narasimhan, K., Salimans, T. and Sutskever, I. (2018) Improving language understanding by generative pre-training. OpenAI Blog.
- Radford, A., Kim, J.W., Hallacy, C. et al. (2021) Learning transferable visual models from natural language supervision. Proceedings of ICML 2021, pp. 8748-8763.
- Raffel, C., Shazeer, N., Roberts, A. et al. (2020) Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of Machine Learning Research, 21(140), pp. 1-67.
- Rajpurkar, P., Jia, R. and Liang, P. (2018) Know what you don't know: Unanswerable questions for SQuAD. Proceedings of ACL 2018, pp. 784-789.
- Touvron, H., Martin, L., Stone, K. et al. (2023) Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288.
- Wang, A., Singh, A., Michael, J. et al. (2018) GLUE: A multi-task benchmark and analysis platform for natural language understanding. Proceedings of EMNLP 2018, pp. 353-355.
- Wang, A., Pruksachatkun, Y., Nangia, N. et al. (2019) SuperGLUE: A stickier benchmark for general-purpose language understanding systems. Advances in NeurIPS 32.
- Wu, Z., Ramsundar, B., Feinberg, E.N. et al. (2018) MoleculeNet: A benchmark for molecular machine learning. Chemical Science, 9(2), pp. 513-530.

Declarations

Funding

This research was supported by the Italian National Recovery and Resilience Plan (PNRR) under grant PE00000013 (Future Artificial Intelligence Research, FAIR) and the European Research Council (ERC) under the European Union's Horizon Europe programme, grant agreement No. 101042061 (ScalePretrain). Computational resources were provided by the Italian National Supercomputing Centre CINECA under project HP10CKNPZ8.

Conflict of Interest

The author declares no conflicts of interest.

Data Availability Statement

Model checkpoints, training logs, and evaluation scripts are publicly available on the project GitHub repository at <https://github.com/garcia-lab/scale-pretrain>. Pretraining corpora follow their respective original licences; all post-processed benchmark results are deposited in Zenodo under DOI: 10.5281/zenodo.19614752. Released under CC BY 4.0.

Ethical Approval

This study involved no human participants and no animal subjects. The practitioner survey (n=34) for decision-framework validation was conducted under Mediterranean Institute of Technology Ethics Committee exemption MIT-ETH-2023-0441. Participants provided written informed consent and all responses were anonymised prior to analysis.

Appendix A

Pretraining Strategy Decision Framework

This framework maps task category and available compute budget to the recommended pretraining strategy, based on empirical results from this study.