NCP-ArchPreview

Next Concept Prediction at Trillion-Token Scale

Paper authors: The Intern-NCP Team
Shanghai AI Lab; LUMIA Lab, Shanghai Jiao Tong University

arXiv:2609.10715v1 | 28 September 2026

Predict multi-token latent representations while retaining token generation.

Independent paper reading · 25–30 minutes + backup

Can internal representations become prediction targets?

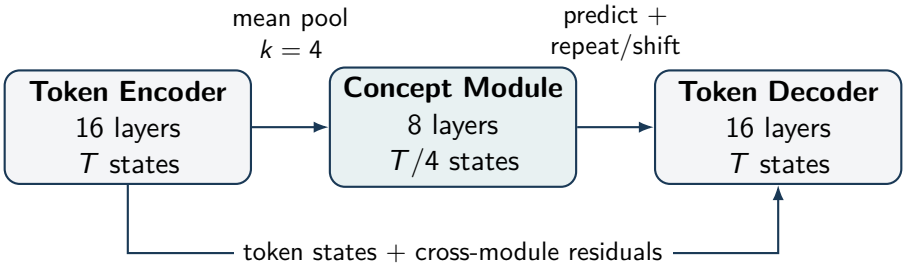
Research question

Does explicit prediction of future token-span representations improve large-scale autoregressive pretraining?

- NTP supervises individual tokens; longer-span structure is learned indirectly.
- NCP adds a predictive objective on a shorter, learned concept sequence.
- This report scales the **ConceptLM** approach to **8.94B** parameters and **5.73T** pretraining tokens, with NCP active from the start.

Paper, Secs. 1 and 6.2, PDF pp. 1–3, 22. NCP and product-quantized concepts originate in the cited ConceptLM work.

A compressed middle path feeds a full token decoder



- Only the Concept Module operates on the compressed sequence.
- IRC routes information across depths; CRC connects the three modules.
- The model still generates one token at a time.

Teaching schematic based on Fig. 2, PDF p. 5; configuration: Table 10, p. 34.

Product quantization creates a compact concept vocabulary

$$\mathbf{c}_m = \frac{1}{k}\sum_{i=1}^{k} \mathbf{h}_{(m-1)k+i} \quad \longrightarrow \quad (\mathbf{c}_m^1, \ldots, \mathbf{c}_m^S)$$

$$n_m^s = \arg\min_n \|\mathbf{c}_m^s - \mathbf{e}_n^s\|_2^2, \qquad \mathbf{d}_m^s = \mathbf{e}_{n_m^s}^s$$

- Hidden width $d = 4096$; $k = 4$ contextual token states per concept.
- $S = 32$ segments; each has $N = 128$ codewords of dimension 128.
- 128^{32} possible combinations require only $Nd = 524{,}288$ codebook parameters.
- Fixed token spans need not align with semantic boundaries.

Paper, Eqs. 1–6, PDF pp. 4–5; Table 10, p. 34. Parameter count is calculated from the reported configuration.

Prediction uses soft codebook mixtures

$$\mathbf{u}_m = \mathrm{ConceptModule}(\mathbf{c}_{<m}), \qquad \boldsymbol{\pi}_m^s = \mathrm{softmax}\big(\mathrm{Head}_s(\mathbf{u}_m)\big)$$

$$\hat{\mathbf{c}}_m^s = \sum_{n=1}^{N} \pi_{m,n}^s \mathbf{e}_n^s, \qquad \hat{\mathbf{c}}_m = \mathrm{concat}_s(\hat{\mathbf{c}}_m^s)$$

What discrete concept prediction means here

A finite codebook constrains the prediction, but the output is a differentiable weighted vector. Training uses continuous-target MSE.

- No argmax or sampled codeword on the feedback path.
- Each segment lies in its codebook's convex hull (from the softmax formula).

Paper, Eqs. 7–10 and 22, PDF pp. 6, 9. Convex-hull interpretation follows directly from the equations.

A causal shift prevents access to the target token span

$$\mathbf{b}_t = \begin{cases} \mathbf{0}, & 1 \le t < k, \\ \hat{\mathbf{c}}_{\lfloor (t-k)/k \rfloor + 2}, & k \le t < T, \end{cases} \qquad \tilde{\mathbf{h}}_t = \mathbf{h}_t + \mathbf{b}_t$$

Input position t	Available history	Signal	Predicts
1–3	incomplete chunk	0	x_2–x_4
4–7	$\mathbf{c}_1$	$\hat{\mathbf{c}}_2$	x_5–x_8
8–11	$\mathbf{c}_{1:2}$	$\hat{\mathbf{c}}_3$	x_9–x_{12}

At $t = 4$, predict the next chunk from tokens 1–4; do not inject its true encoding.

Paper, Eqs. 11–13, PDF p. 6. Table is a teaching expansion for $k = 4$. Concept-to-decoder CRC requires the same causal alignment.

Three losses jointly train the two prediction levels

$$\mathcal{L} = \mathcal{L}_{\mathrm{NTP}} + \alpha \mathcal{L}_{\mathrm{NCP}} + \beta \mathcal{L}_{\mathrm{VQ}},$$

$$\mathcal{L}_{\mathrm{NTP}} = -\frac{1}{T-1} \sum_{t=1}^{T-1} \log p(x_{t+1} \mid \tilde{\mathbf{h}}_{\leq t}),$$

$$\mathcal{L}_{\mathrm{NCP}} = \frac{1}{M-1} \sum_{m=2}^{M} \|\hat{\mathbf{c}}_m - \mathrm{sg}(\mathbf{c}_m)\|_2^2,$$

$$\mathcal{L}_{\mathrm{VQ}} = \frac{1}{MS} \sum_{m,s} \|\, \mathrm{sg}(\mathbf{c}_m^s) - \mathbf{d}_m^s \|_2^2.$$

- VQ fits codewords; sg blocks target-side gradients.
- NCP updates the encoder via past concepts; NTP trains both paths.

Paper, Eqs. 21–24, PDF pp. 8–9. Main-text convention shown; Algorithm 1 differs by a factor of $1/S$ in NCP normalization (backup).

Residual routing selects information across depth and scale

IRC: within a module

$$\mathbf{H}_{\ell+1} = \sum_j w_{\ell,j} \mathbf{X}_{\ell,j}$$

- Candidates are states from earlier depths.
- Weights depend on the current block output.
- Unnormalized, potentially signed.
- Initially selects the latest state.

CRC: across modules

$$\mathbf{T}_{\text{out}} = \mathbf{T} + \mathbf{D} \odot \sum_j \alpha_j \, \mathrm{LN}(\mathbf{S}_j)$$

- Source depths aligned by pooling/repetition.
- Target-conditioned softmax weights.
- Learned per-dimension scale $\mathbf{D}$.
- Encoder→Concept/Decoder; Concept→Decoder.

Paper, Eqs. 14–20, PDF pp. 7–8. Equations abbreviated for presentation; see Chinese notes for full notation.

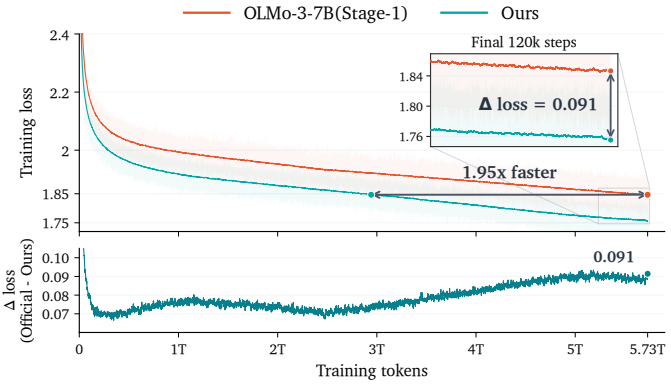
The evidence comes from several distinct experimental regimes

Question	Model / budget	Evidence
Large-scale utility	8.94B; 5.73T + 100B tokens	Table 1
Capacity vs. computation	First 200B tokens	Fig. 3; Table 2
Residual design	1B; 150B tokens	Table 3
Compute-optimal scaling	10^{19}–10^{20} FLOPs	Figs. 4, 10
Domain adaptation	Stage-1; Full / LoRA / VQ	Tables 4–7
Complementarity with MTP	3B; 150B tokens	Table 8
Speculative drafting	1.1B drafter; matched training	Table 9

Overall AVG: 26 task scores, equally weighted. BPB is evaluated separately.

Paper, Secs. 4–5, PDF pp. 10–21; aggregation: Appendix E, pp. 41–42.

Stage-1 reaches the baseline loss with 51.3% of tokens



Final loss gap: 0.091 **Reported token-convergence ratio: 1.95×**

Paper, Fig. 1(a), PDF p. 2; Sec. 4.2.2, p. 10. Same training data; main models are not parameter-matched. This is not measured wall-clock speedup.

Downstream gains shrink after mid-training

Score	Stage-1 delta (pp)	Stage-2 delta (pp)
Overall AVG	**+2.45**	**+0.59**
MMLU aggregate	+2.58	+1.82
GSM8K	+5.99	+3.34
Math domain AVG	+3.75	+1.76
Code domain AVG	+2.64	−0.65
MC-Non-STEM AVG	+4.63	+0.91
HumanEval	+4.28	−3.69

- Stage-1 overall: 46.59 → 49.04; Stage-2: 56.98 → 57.57.
- Lower mixture loss does not guarantee better performance in every domain.

Paper, Table 1, PDF p. 11. Deltas are NCP minus released OLMo-3-7B; pp = percentage points.

Parameter and compute controls address different explanations

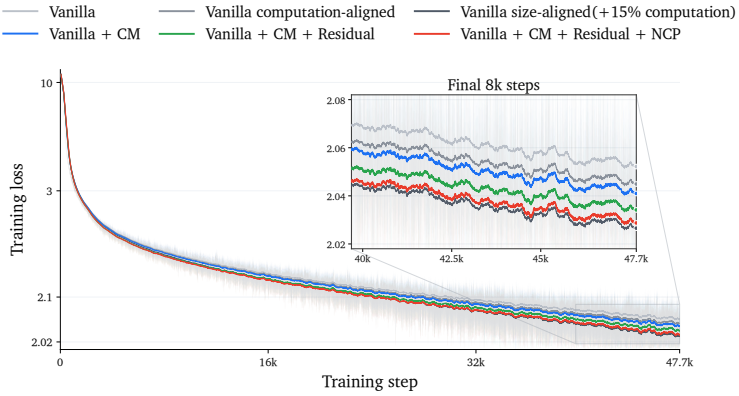
Configuration	Parameters	Computation
Vanilla	$32P_{\mathrm{blk}}$	$32F_{\mathrm{blk}}$
Compute-aligned Vanilla	$34P_{\mathrm{blk}}$	$34F_{\mathrm{blk}}$
Size-aligned Vanilla	$40P_{\mathrm{blk}}$	$40F_{\mathrm{blk}}$
NCP-ArchPreview	$40P_{\mathrm{blk}}$	$34F_{\mathrm{blk}}$

$$F_{\mathrm{NCP}} \approx (16 + 8/4 + 16)F_{\mathrm{blk}} = 34F_{\mathrm{blk}}$$

- Eight concept layers add capacity at about one-quarter sequence length.
- $34/40 = 85\%$: approximate compute relative to the size-aligned model.

Paper, Table 2 and Sec. 4.3, PDF p. 13. Small components are omitted in this block accounting; comparisons cover the first 200B tokens.

Concept layers, routing, and NCP improve loss progressively



Full NCP beats the compute-aligned baseline and approaches the size-aligned one.

Paper, Fig. 3, PDF p. 12. Same recipe, first 200B tokens. Sequential additions do not establish independent, additive effects at every scale.

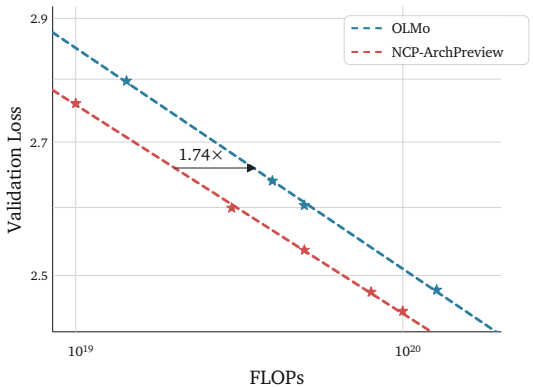
IRC + CRC gives the best loss in the 1B routing study

Variant	LM loss ↓	Extra FLOPs
No hierarchical residuals	2.2588	0.000%
IRC + CRC	**2.2265**	+0.051%
IRC only	2.2315	+0.024%
IRC + input-level cross-module links	2.2292	+0.024%
IRC + all-stage softmax	2.2295	+0.024%
Block AttnRes ($S = 4$) + cross-module links	2.2409	+0.026%

- 150B training tokens; loss averaged over the final 200 logged steps.
- Analytical FLOPs exclude memory traffic and several routing overheads.

Paper, Table 3 and Sec. 4.4, PDF pp. 14–15. Small FLOPs overhead is not a measured runtime or memory overhead.

Scaling fits favor NCP within the tested compute budgets



- Search model/data allocation and hyperparameters per budget.
- Compare best validation loss.
- Author-reported efficiency: **1.74**×.

$$C = F_{\text{tok}} D$$

Analytical compute, not elapsed time.

Paper, Fig. 4, PDF p. 15; Appendix D, pp. 38–40. Extrapolation beyond the measured scaling ladder is not established.

Capability proxies reveal weaknesses hidden by mixture loss

Stage-2 recipe	V1	V2	V3
Function-generation NLL ↓	**0.7330**	0.7501	0.7598
HumanEval ↑	**45.60**	42.19	39.96
Competition-math NLL ↓	**0.5742**	0.5949	0.6075
MATH-500 ↑	**43.74**	41.66	37.21

- Fixed teacher trajectories; source-balanced capability scores.
- Paper reports lower training loss from V1 to V3, but weaker capabilities.
- Only three matched-budget recipes: evidence for screening, not a universal predictor.

Paper, Sec. 4.2.4, PDF p. 12; Appendix C, Table 11, p. 36. These auxiliary evaluation values must not be substituted into Table 1.

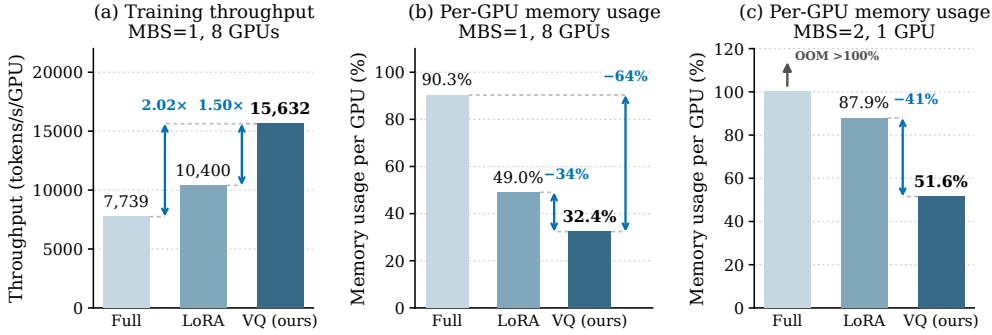
The 17M adaptation interface offers a domain-dependent tradeoff

Domain	Full gain	LoRA gain	VQ gain	VQ general Δ
Code AVG	−1.61	+1.34	**+2.65**	−0.42
Math AVG	**+6.16**	+3.15	+4.27	+0.39
TriviaQA EM	**+18.00**	+17.48	+9.19	+0.03

- VQ updates existing **codebooks and prediction heads**, about 17M parameters.
- LoRA also trains 17M parameters; Full trains the entire model.
- VQ is strongest on this code average, but freezing the backbone does not eliminate forgetting.

Paper, Tables 4–7, PDF pp. 17–19. All gains are percentage points relative to each NCP Stage-1 adaptation baseline. General AVG uses five tasks.

VQ adaptation is faster and uses less memory in this setup



- At MBS 1 on 8 GPUs: VQ reaches 15,632 tokens/s/GPU.
- 1.50× LoRA throughput; 2.02× Full throughput.
- GPU type/capacity is unspecified; percentages are not absolute GB.

Paper, Fig. 7, PDF p. 19. The right panel uses a different setup: MBS 2 on one GPU. These are adaptation measurements.

NCP and MTP are complementary in a 3B controlled study

Configuration	Mean LM loss ↓	FLOPs (10^{21})
OLMo-3-3B + Residual	2.3805	3.245
OLMo-3-3B + Residual + MTP	2.3739	3.750
NCP-ArchPreview	2.3707	3.227
NCP-ArchPreview + MTP	**2.3664**	3.731

- Residual-aligned baselines; first 150B tokens; one additional MTP layer.
- MTP predicts two positions ahead, with loss weight 0.3.
- Loss averages the **entire training interval**, excluding auxiliary losses.

Paper, Sec. 5.2 and Table 8, PDF pp. 19–20. This does not establish a downstream gain for the full 8.94B model with MTP.

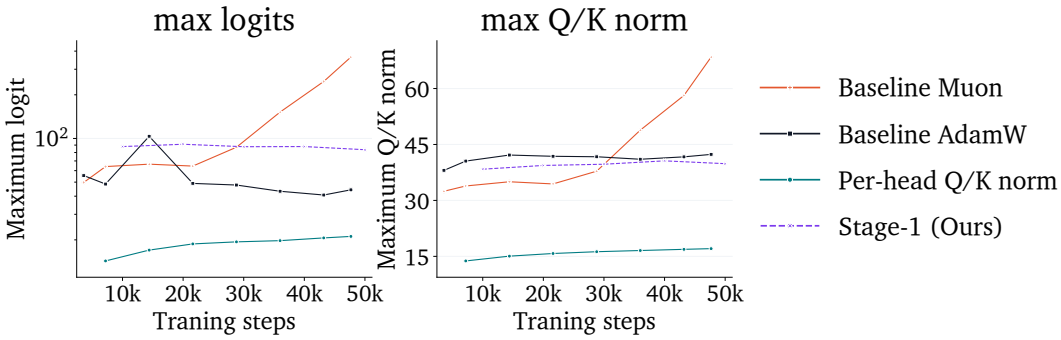
Concept conditioning increases speculative acceptance length

Task	Baseline MAL	+Concept MAL	Reported gain
GSM8K	6.351	6.537	+2.93%
MATH	6.105	6.240	+2.22%
HumanEval	5.432	5.845	+7.59%
MBPP	5.844	6.099	+4.37%
Macro average	**5.933**	**6.180**	**+4.17%**

- 1.1B drafter, +0.04M parameters; 16-token proposal horizon.
- MAL counts committed tokens, including target correction/bonus when present.
- End-to-end latency improvement is not reported.

Paper, Table 9 and Eqs. 26–27, PDF p. 21. Relative gains are as reported; rounded values yield small last-digit differences.

Per-head Q/K normalization stabilizes the Muon ablation



- Layer-wise normalization permits head-level norm outliers.
- Per-head normalization suppresses logit and Q/K-norm growth here.
- **Main results still use layer-wise Q/K normalization.**

Paper, Sec. 4.6 and Fig. 6, PDF pp. 16–17. Optimizer/normalization interaction is the authors' interpretation of the controlled comparison.

The strongest result is scalable joint token–concept learning

1. **Feasibility at scale.**
   NCP works from the start of 8.94B / 5.73T pretraining while retaining token-level autoregressive generation.
2. **Useful optimization and interfaces.**
   Controlled losses, lightweight adaptation, and speculative acceptance support several uses of the learned concept pathway.
3. **The evidence has clear limits.**
   Smaller Stage-2 gains; no long-context training; no full-budget matched downstream study; no demonstrated semantic interpretability.

Synthesis of Secs. 4–8, PDF pp. 10–23. Distinguish tokens-to-loss, analytical FLOPs, adaptation throughput, and acceptance length.

Paper, materials, and scope of this reading

The Intern-NCP Team.
NCP-ArchPreview Technical Report: Moving towards Latent Space Language Models through Next Concept Prediction.

arXiv:2609.10715v1 Submitted 9 September 2026; read 28 September 2026.

- Read author TeX, all substantive sections and appendices; checked PDF figures/tables.
- Figures are copied from the versioned author sources.
- Architecture schematic and interpretations are for this presentation.
- No training reproduction or implementation audit was performed.

Author-listed resources: model collection; LMDeploy; evaluation code. Source links recorded from the paper; repositories were not audited.

Backup: the NCP loss normalization needs implementation review

Main text, Eq. 22:

$$\mathcal{L}_{\mathrm{NCP}}^{\mathrm{text}} = \frac{1}{M-1} \sum_{m=2}^{M} \|\hat{\mathbf{c}}_m - \mathrm{sg}(\mathbf{c}_m)\|_2^2$$

Algorithm 1, line 21:

$$\mathcal{L}_{\mathrm{NCP}}^{\mathrm{alg}} = \frac{1}{(M-1)S} \sum_{m=2}^{M} \sum_{s=1}^{S} \|\hat{\mathbf{c}}_m^s - \mathrm{sg}(\mathbf{c}_m^s)\|_2^2 = \frac{1}{S} \mathcal{L}_{\mathrm{NCP}}^{\mathrm{text}}$$

- With standard squared norms and concatenation, the difference is $S = 32$.
- The factor can be absorbed into α, but α, β are not specified numerically.
- Inference feedback, incomplete chunks, and draft-state notation also need code-level verification.

Paper, Eq. 22, PDF p. 9; Algorithm 1, p. 35; Sec. 2.3, p. 6; Eq. 26, p. 21.

Backup: architecture and optimizer configuration

Token / concept / token layers	16 / 8 / 16
Hidden / FFN width	4096 / 11008
Attention heads / KV groups	32 / 32
Vocabulary / context	100278 / 8192
Token local window	4096; full attention every fourth layer
RoPE base / activation	500000 / SwiGLU
Normalization	RMSNorm, $\epsilon = 10^{-6}$; layer-wise Q/K
Precision / dropout	BF16 / 0
PQ configuration	32 segments; 128 words; 128 dimensions
Optimizers	Moonlight Muon + AdamW for non-Muon weights
Default learning rate	6×10^{-5}; cosine schedule

Paper, Table 10, PDF p. 34; Sec. 3.6, p. 9. Tokenizer vocabulary and latent vocabulary are different objects.

Backup: multiple samples still estimate Pass@1

Benchmark	Shots	Samples	Metric
GSM8K	8	1	Pass@1
GSM-Symbolic	8	8	Pass@1
Minerva	4	1	Exact match
MATH-500	4	32	Pass@1
BigCodeBench	3	5	Execution pass@1
HumanEval	3	32	Execution pass@1
DS-1000	3	5	Execution pass@1
MBPP	3	32	Execution pass@1
MultiPL-E HumanEval / MBPP	0	32	Execution pass@1

Global seed: 42. Samples are completions per problem, not inference batch size.

Paper, Table 13 and Appendix E, PDF pp. 39–42. Overall AVG excludes BPB, intermediate averages, and the aggregate MMLU row.

Backup: proxy scores retain a vector of capabilities

$$\ell_m(i) = -\frac{1}{|T_i|} \sum_{j \in T_i} \log p_m(y_{ij} \mid x_i, y_{i,<j})$$

- 177,202 fixed teacher trajectories, 63 sources, about 100M suffix tokens.
- Average within each source, then across sources in each capability.
- Four free-form capability NLLs agree with downstream recipe ordering.
- HellaSwag uses correct-vs-best-distractor choice margin instead.

HellaSwag	V1	V2	V3
Margin ↑	0.09678	0.09932	0.10083
Accuracy ↑	66.40	67.30	67.25

Paper, Appendix C, Eq. 28 and Table 11, PDF pp. 34–38. V2/V3 are treated as a leading tier; no individual data-source causal attribution.

Backup: adaptation averages conceal important task differences

NCP adaptation metric	Stage-1	Full	LoRA	VQ
HumanEval (code adaptation)	31.27	43.27	36.49	33.94
MBPP+ (code adaptation)	29.63	7.14	21.96	31.22
Code AVG	30.04	28.43	31.38	32.69
Math AVG	30.56	36.72	33.71	34.83
TriviaQA EM	40.28	58.28	57.76	49.47

- Separate runs: Magicoder / Orca-Math / TriviaQA-RC.
- General AVG includes MMLU, ARC-E, HellaSwag, NQ Open, and SciQ.
- Adaptation checkpoints/evaluation values differ from the main results table.

Paper, Secs. 4.1 and 5.1, Tables 5–7, PDF pp. 10, 18–19. Appendix E's prose omits SciQ from its list; tables include it.

Backup: the scaling ladder searches model–data allocation

$C = F_{\mathrm{tok}} D$ rather than treating all parameters as active per token

Budget (FLOPs)	Best listed NCP loss	Encoder / Concept / Decoder
10^{19}	2.759	4 / 4 / 4
3×10^{19}	2.599	8 / 4 / 8
5×10^{19}	2.537	8 / 8 / 8
8×10^{19}	2.476	8 / 4 / 8
10^{20}	2.449	8 / 8 / 8

Search learning rate and batch size on a representative model, then refine learning rates for other allocations at the same budget.

Paper, Appendix D, PDF p. 38; Table 12, p. 40. Best rows selected from the reported points. FLOPs accounting excludes embeddings.