Learning Cost and Token Use with Controlled Literary Chinese: A Paired State-Tracking Study

Anonymous authors
Paper under double-blind review

Abstract

We measure learning cost and generation token use separately: shorter examples need not reduce the cost of learning. We compare controlled Literary Chinese (WY) and controlled English (EN) in 80 pre-specified pairs of from-scratch Transformer classifiers across three scales on a fixed 720-item training inventory. Certification requires at least 75% inventory accuracy throughout evaluations spanning 7,200 presentations, charging training through certification. All 80 pairs attain it with lower WY costs; scales are analyzed separately. The 40 primary 10M pairs yield a mean cost difference of -0.7447 S (approximate paired-t 95% interval [-0.7811, -0.7084] S), where S is the scale-specific training-FLOPs unit: descriptive mean paired savings are 50.38%, with similar repetitions. A development-informed supplement trains 20 paired 11.24M-parameter generators. At equal exposure, WY uses 62.12% fewer training tokens; complete-output test accuracy is WY 97.86%, EN 95.03%. At nominal 16-million-token budgets, accuracy is WY 98.09%, EN 95.15%, with 2.64 times as many WY presentations. Input-plus-generated-output tokens per request, including visible steps and errors, are then 61.98% lower. One training-unseen layout accounts arithmetically for 39–43% of generator quality gaps in a post-hoc audit. Classifier transfer is poor, and generator evaluation data were development-reused. The evidence quantifies acquisition cost, checkpoint quality, and visible-output length for these task-specific packages, without isolating language effects or establishing matched-test-quality deployment savings.

1 Introduction

Literary Chinese and its tradition of continuous, unpunctuated writing motivate an economical, interpretable representation. We study learning costs and generation token use in contemporary-authored controlled packages. Figure L1 maps the classifier study and generator supplement; neither tests a universal replacement for English.

Our primary question is whether lower per-presentation cost survives learning to a shared quality requirement. We measure acquisition cost, repetitions, and higher minima (Section 3). The generator supplement separately asks how quality and token use compare at equal exposure and matched nominal training-token budgets (Section 4.6).

The 720 state-tracking problems pair contemporary-authored, mixed-script WY with rule-generated EN. Each pair comprises two classifier runs with identical initial parameters, trained and evaluated on the same inventory. The rendering/tokenizer packages share a compute-indexed policy; this does not isolate language-intrinsic effects.

Dense evaluations identify the first window meeting a fixed persistence rule, charging all training through its end. Actual scores, trajectories, and higher minima distinguish this common observed floor from equal endpoint quality or convergence.

The two studies yield three linked contributions:

1. **Lower acquisition cost at a shared criterion.** All 40 primary 10M pairs favor WY: mean paired savings are 50.38% in training FLOPs and 50.66% in tokens, with similar repetitions and positive savings at four higher pre-specified minima. Separate 1M/100M samples are supportive (Table 2).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

2. **Generator quality under two training controls.** Twenty development-informed pairs use 62.12% fewer WY training tokens at equal exposure, with higher mean complete-output test accuracy at both exposure and nominal-token checkpoints (Table 5).

3. **Fewer request tokens, including visible steps.** At the nominal-token checkpoint, input plus all generated output uses 61.98% fewer WY tokens, including errors. Step-token allocation bounds are reported within this total, not as additional savings (Appendix J).

The generator supplement has its own development-informed protocol. Its 20 fresh seed pairs neither enlarge the primary sample nor inherit its confirmatory status.

Classifier forward costs are lower, but transfer to separate challenges is poor (Sections 4.4–4.5). The generator extension (Section 4.6) adds emitted steps and held-out configurations under different tokenizers and a shallower task. It complements, rather than repairs or replaces, the classifier results.

Measurement map. The generator supplement is development-informed: curves use dev512; endpoints use test1024.

Study and comparison	Matched condition and measurement
Classifiers: acquisition (Table 2)	Shared accuracy criterion: training cost to certification
Generators: 176 passes (Table 5)	Equal presentations: training tokens and test quality
Generators: nominal 16M (Table 5)	Matched nominal tokens: test quality, not token savings
Generators: use (Table J1)	Same requests: input plus all output; steps included

2 Related work

Compact linguistic representations have precedents: Al-shaibani and Ahmad evaluate consonant-only and masked-vowel English, including vocabulary-dependent size and epoch time; Si et al. shorten Chinese sequences using pronunciation or glyph encodings before subword tokenization. We instead compare complete controlled sentence renderings. (Al-shaibani & Ahmad, 2023); (Si et al., 2023)

Compression need not improve quality. Goldman et al. compare BPE compression and performance under different English/Turkish training corpora; Schmidt et al. find that shortest segmentation does not consistently predict downstream accuracy. Limisiewicz et al. study compression-dependent compute optimality across scales and languages, including inflated English. These motivate joint quality/resource measurement. (Goldman et al., 2024); (Schmidt et al., 2024); (Limisiewicz et al., 2026)

Liu et al.'s SuperBPE learns subwords before cross-whitespace merges; matched-size, vocabulary, and FLOPs experiments connect compact segmentation to downstream quality and inference cost. Our comparison changes both controlled rendering and its cross-word BPE encoding. (Liu et al., 2025) Petrov et al. and Ahia et al. document cross-language tokenization and access-cost disparities; they do not measure our from-scratch acquisition contrast. (Petrov et al., 2023); (Ahia et al., 2023)

State tracking supports semantic pairing, not identification of a learned algorithm. Kim and Schuster track objects in boxes via move/put/remove operations; Li et al. analyze permutation composition and held-out sequence lengths. Our full-content swaps and binary queries instead measure fixed-inventory acquisition costs. (Kim & Schuster, 2023); (Li et al., 2025)

Language choice is also studied in prompts to existing models. In a preliminary Chinese/English coding-agent study, Ren et al. report lower task success for Chinese and no systematic success-adjusted efficiency gain, with raw token effects differing by model. Li's exploratory LangMatch project compares Literary Chinese, modern Chinese, and English prompts and labels its findings preliminary. These are relevant public antecedents; our from-scratch setting measures a different intervention. (Ren et al., 2026); (Li, 2026)

Our contribution is a paired acquisition-cost test, not compression or language choice itself. Persistent accuracy and repetition counts connect encoding to learning cost; generator checkpoints control exposure or token budget. This is not a head-to-head comparison with the systems above. Muen-

nighoff et al. study diminishing returns from repeated training data; we count repetitions rather than fit a scaling law. (Muennighoff et al., 2023)

3 Primary classifier experiment

3.1 Population and representations

Both conditions encode the same frozen 720 canonical problems and binary labels. Models train on this inventory, and acquisition observations evaluate this same inventory. Historical subset names do not create a held-out test set. Statistical replication is across initialization seeds, conditional on the shared inventory and training policy, not across independently sampled tasks or human speakers.

The two renderers and tokenizers are inherited unchanged. WY uses continuous text without punctuation or whitespace; EN retains conventional capitalization, punctuation, and interword spaces. Unpunctuated classical texts are also documented by (Tang et al., 2021), cited here as background rather than the source of our motivation. WY is contemporary-authored, not a historical transcription. The classifier tokenizers each have 492 entries but different units and segmentations. Canonical pairing matches meanings, not orthography. This complete-package comparison neither isolates punctuation removal as the cause of savings nor shows that English requires these marks to be learnable.

Both tokenizers use byte-level byte-pair encoding (BPE), 492 entries, no normalization, and no regex pre-tokenization; merges may span word boundaries. The short sequences reflect this tokenizer and the rendering, not a general-purpose multilingual tokenizer. Appendix E gives exact true/false texts and lengths, tokenizer settings, and the full state transitions. These examples verify the task semantics, not which algorithm the classifiers learned.

The frozen encodings (Appendix B) use 7,934 WY versus 15,920 EN tokens per complete inventory pass, including beginning- and end-of-sequence markers (BOS/EOS): a descriptive reduction of 50.16%. Lengths range from 4–20 and 7–38 tokens respectively. The inventory has 328 true and 392 false labels, so a constant majority-label prediction attains 54.44%. Compression per pass and measured cost to quality are different quantities; their similar magnitudes here do not by themselves establish that WY learns from fewer item presentations.

The inherited inventory contains 716 distinct queries under within-swap endpoint-order normalization. Under that normalization, 96 query identities (100 inventory rows) also occur in the 16,384-row tokenizer-construction corpus. Tokenizer exposure is not supervised classifier training, but the inputs are not semantically independent of that corpus. Rows remain unchanged; Appendix K preserves the diagnostic and its incomplete equivalence coverage.

The inventory, renderers, encodings, model, optimizer, cyclic item sequence, accumulation, and original learning-rate schedule were inherited from development. Earlier development curves informed the persistence span. For the present confirmation, these choices, the new initialization seeds, and the analysis were fixed before runs began. Earlier experimental outcomes are kept separate and cannot replace missing confirmation runs.

3.2 Models, cost policy, and fixed horizon

We use architecture-matched bidirectional Transformer classifiers with a two-class head (Table 1). Each initialization seed defines one WY/EN pair: the 40 primary 10M pairs therefore comprise 80 training runs, not 80 independent pairs.

Table 1. Architectures and frozen training settings.

Scale	Params.	Layers	Width	Heads	FFN width	Items / update	Peak learning rate	Pairs
1M	1,031,970	4	136	4	544	16	0.0006	20
10M	10,150,178	6	368	8	1,472	32	0.0003	40
100M	100,648,706	12	832	16	3,328	64	0.00015	20

The optimizer is FP32 AdamW with betas (0.9, 0.95), epsilon 1e-8, and weight decay 0.1. Training processes unpadded single-item microbatches in the same deterministic cyclic inventory order; accumulated gradients produce optimizer updates. The primary scale is 10M; the other scales are analyzed separately and cannot substitute for the primary evidence.

The shared learning-rate policy is indexed by cumulative training FLOPs: linear warmup to S (a scale-specific FLOPs unit, not seconds), then linear decay toward one tenth of peak at 20S. We retain that policy when ending runs at 4S, rather than compressing the schedule. Because per-item costs differ, learning rates at equal optimizer updates can differ. Thus the measured contrast includes package-dependent learning-rate trajectories under this common policy; it is not a length-only causal effect.

The fixed scale units are 3,048,664,697,856 FLOPs at 1M, 28,582,662,379,008 at 10M, and 258,856,688,982,784 at 100M. Every model trains through the first completed optimizer update crossing 4S, regardless of its scores. No score-triggered extension, replacement seed, or early success stop is permitted. A technical failure stops new admissions and remains visible rather than being assigned a fabricated learning cost.

Training FLOPs are declared operator-equivalent costs at actual sequence lengths, including forward/backward and optimizer arithmetic. They exclude evaluation, serialization, rendering, tokenizer construction, and input preparation. Tokens count actual encodings, not characters or bytes. Independent analysis reconstructs these accounts from update traces and frozen code. Hardware instructions, time, energy, and total financial expense are not measured.

3.3 Common observed quality

Let $A(t)$ count correct predictions among the 720 items at a saved evaluation, and $E(t)$ count cumulative training-item presentations. One presentation processes one item; repeats count again, so 7,200 presentations equal ten inventory passes in item count, not 7,200 distinct problems. The primary minimum is $q = 540$ (75%), 20.56 percentage points above the 54.44% majority-label baseline. Recognition is the earliest saved window end t with $A \geq q$ at every saved evaluation in the window and a span in E of at least 7,200. We charge all training through t, not just to the first threshold crossing. This observed persistence does not guarantee passing at unsampled updates or later times.

We evaluate at initialization, every five optimizer updates, and the final update. Initialization is a diagnostic and cannot start a qualifying passing window. Nominal intervals therefore correspond to 80, 160, and 320 item presentations at the three scales. The discrete grid can overshoot the required span; it is not retrospectively interpolated to a favorable fractional update. We save actual endpoint scores, window scores, and full curves rather than treating a shared lower bound as equal realized accuracy.

Pre-specified supportive minima are 576, 612, 648, and 684 correct answers (80%, 85%, 90%, and 95%). Shorter spans of 1,440 and 3,600 item presentations are sensitivity analyses, not alternative headline endpoints. A candidate score-resolution tolerance in the plan is not an exclusion or success rule. No pair is removed because its two endpoint scores differ.

3.4 Estimand and uncertainty

For each training run, restricted cost R is the cumulative FLOPs through recognition, capped at $4S$; a separate indicator records whether the criterion was attained within this budget. A run that does not attain it receives the cap only for this statistical summary, not as an observed cost of successful learning. Technical missingness is not budget nonattainment and prevents complete primary inference.

For pair i, $D_i = (R_{\mathrm{WY},i} - R_{\mathrm{EN},i})/S$. The primary target is its expected value across the prescribed initialization-seed procedure at 10M, conditional on this inventory and protocol; we estimate it with the mean of 40 paired differences and a pre-fixed approximate paired-t 95% interval. Sensitivities use a paired percentile bootstrap (20,000 resamples, seed 20260913), median, signs, and leave-one-pair-out estimates. Intervals quantify seed variability, not task-population uncertainty or continuous crossing times. They are scale-specific, unpooled, and not simultaneous family-wise intervals.

For jointly attained pairs, descriptive savings are $100 \times (1 - C_{\mathrm{WY},i}/C_{\mathrm{EN},i})$, where C is cumulative training cost through recognition, in FLOPs or tokens for the respective summary. We average these per-pair percentages, with EN cost as the denominator in each pair. This differs from a ratio of aggregate mean costs. When joint attainment is incomplete, percentages are conditional summaries and are reported with their coverage and original planned N.

4 Verified results

All 160 training runs completed without missingness or technical invalidity: 20 pairs (40 runs) at 1M, 40 pairs (80 runs) at 10M, and 20 pairs (40 runs) at 100M. Independent raw-record recomputation covers accuracy, update-level tokens and FLOPs, evaluation-grid completeness, stopping updates, initialization pairing, and exit records. The disjoint 10M shards cover all 40 planned pairs. Scales are reported separately without pooling.

Table 2. Costs to the primary requirement. Brackets: pointwise approximate 95% t intervals. The 10M cost difference is primary; percentage intervals are descriptive.

Scale	Joint / planned	Mean difference / S	Approx. 95% interval	Mean FLOPs savings	Mean token savings
1M	20/20	-1.0691	[-1.2727, -0.8654]	50.10% [44.92, 55.27]	50.15% [44.98, 55.32]
10M, primary	40/40	-0.7447	[-0.7811, -0.7084]	50.38% [48.73, 52.02]	50.66% [49.02, 52.29]
100M	20/20	-1.0618	[-1.1494, -0.9741]	60.87% [58.13, 63.61]	61.14% [58.42, 63.86]

All 40 primary paired differences favor WY (Table 2; learning curves in Figure 1). The bootstrap interval is [-0.7814, -0.7112] S, the median difference is -0.7359 S, and every leave-one-pair-out t interval excludes zero on the same side. These sensitivities do not replace the primary analysis. All 20 separately analyzed 1M differences also favor WY.

At 10M, all 40 pairs jointly attain every pre-specified higher minimum with the same persistence requirement. Mean FLOPs savings at 80/85/90/95% are 48.43/45.59/44.77/46.25%. Thus the observed contrast is not confined to the 75% readout (Figure 2). These are correlated supportive readouts, not independent confirmations or additional primary hypotheses.

All 20 differences at the primary criterion at 100M favor WY. The bootstrap interval is [-1.1437, -0.9836] S, the median difference is -1.0383 S, and all leave-one-pair-out t intervals remain below zero. All 20 pairs jointly attain 80/85/90/95% with mean FLOPs savings of 58.94/57.86/51.97/49.13%. The 95% minimum saving has a pointwise interval of [45.71, 52.55]%. These supportive results do not add primary hypotheses or form a simultaneous confidence band.

At 1M, joint attainment at 75/80/85/90/95% is 20/20/17/12/1 out of 20 pairs. Remaining higher-threshold pairs attain only on the WY side within the cap. In particular, the 95% conditional percentage describes one jointly successful pair and cannot stand for the full sample or support a sampling interval.

4.1 Common minimum versus exact quality

Certification matches an observed floor, not budgets or exact scores. At each model's own certification point, mean 10M scores are 625.175/720 (WY) and 630.400/720 (EN); the largest paired gap is 74 answers. These are not same-token test scores. Later dips and unequal follow-up exposure preclude causal stability claims (Appendix K). Section 4.4 separately reports scores after continued training to 4S.

4.2 Window sensitivity

The pre-specified 1,440- and 3,600-presentation windows retain the favorable direction at 75% in every pair (Appendices C and K). At 10M, mean paired restricted differences are -0.6044S and -0.6681S, versus -0.7447S for the primary window. A post-result diagnostic finds 50.46% mean

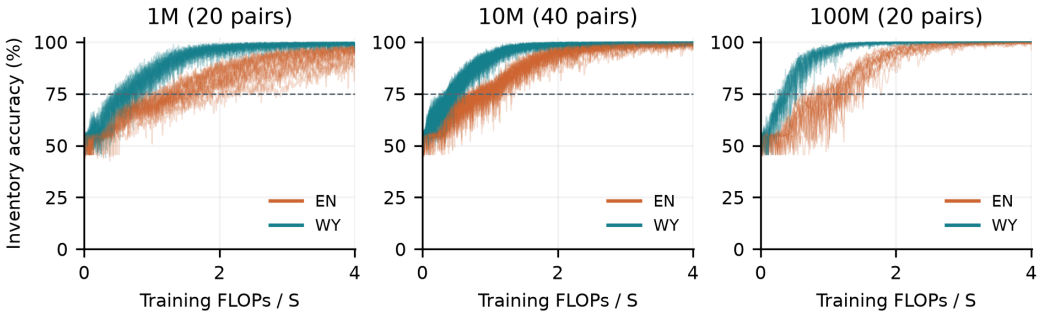

Figure 1. All saved, unsmoothed learning curves; no favorable-seed selection. WY denotes controlled Literary Chinese and EN controlled English. The dashed line marks 75% inventory accuracy. The horizontal axis uses each scale's own FLOPs unit S, not common absolute compute or time. Segments connect observations, not measured intermediate scores.

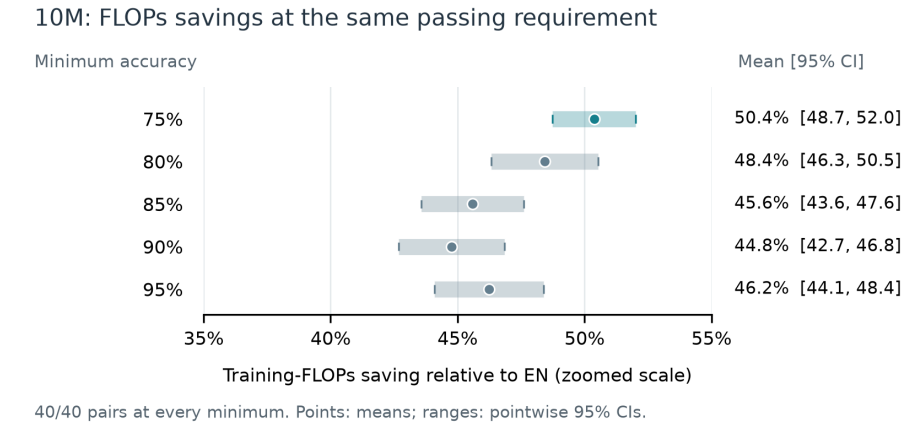

Figure 2. At each pre-specified minimum, points show mean paired training-FLOPs savings and horizontal ranges show pointwise approximate 95% t confidence intervals. All 40 pairs jointly attain all five minima under the unchanged 7,200-presentation passing-span rule. These are intervals for the mean, not run-to-run ranges or simultaneous bands. The zoomed axis is labelled explicitly. Percentage intervals are descriptive and do not replace the primary restricted-cost-difference analysis. All-scale readouts and incomplete coverage are retained in Appendix C.

FLOPs savings at accepted-window starts versus 50.38% at charged ends (Appendix H); the contrast is not created solely by the certification span. Window starts are retrospective diagnostics, not replacement endpoints.

4.3 Presentations versus tokens

At 10M recognition, mean presentation counts are 31,244 WY versus 31,644 EN; mean per-pair savings are 0.9165% (18 pairs fewer WY presentations, two tied, 20 more). Token ratio factors into presentation-count ratio and tokens-per-presentation ratio: cheaper presentations retain their advantage without halving repetitions. At 100M, most pairs use fewer WY presentations and mean per-pair savings are 21.95%. At 1M, mean per-pair savings are -0.0622%; this average of ratios is not the ratio of aggregate counts and does not imply a larger WY aggregate. The decomposition (Appendices G and K) is descriptive, not length-only causal attribution.

4.4 Descriptive forward-inference accounting

Post-result deterministic accounting measures one unpadded, batch-one classifier forward pass per inventory item, not runtime or a new prediction test. At encoded length T (including BOS/EOS), cost is aT^2+bT+c. Appendix K preserves the coefficients, operator checks, and exclusions.

Table 3. Deterministic forward costs for one pass over the fixed inventory.

Scale	WY forward FLOPs over all 720 items	EN forward FLOPs over all 720 items	Reduction in inventory-total forward FLOPs
1M	14,503,877,372	29,572,461,248	50.9548%
10M	156,389,965,068	315,694,032,096	50.4615%
100M	1,589,203,842,812	3,197,341,846,208	50.2961%

Table 3 uses inventory-total ratios, not mean paired acquisition-cost ratios. Input tokens are 7,934 WY versus 15,920 EN per pass (50.1633% fewer). Costs depend on architecture and encoded shapes, not learned weights; these deterministic totals have no seed-level interval and say nothing about unfamiliar-item accuracy.

After continued training to the fixed 4S run end (Section 3.2), recomputed mean WY/EN accuracies on the 720 inventory items are 98.39%/94.38% at 1M, 99.62%/98.81% at 10M, and 99.99%/99.40% at 100M. This is approximately matched training FLOPs, not exposure: WY sees the same inventory about 1.99–2.01 times as often. Table K1 gives seed ranges. These final inventory scores differ from the earlier certification-time scores (Section 4.1); neither is held-out quality. All final models are included without checkpoint selection, exact-score matching, or removing errors from the cost account.

These classifiers emit two logits, not textual steps. Autoregressive generation is evaluated separately in Section 4.6. Padding, batching, caching, and alternative tokenizers can change use costs; neither elapsed-time nor price savings are inferred.

4.5 Supplementary challenge evaluation: cheaper processing, poor transfer

All 160 final 4S classifiers evaluate 512 pre-existing records without retraining, tokenizer refitting, or checkpoint selection: 320 named challenges and 192 synthetic diagnostics. All 81,920 classifier predictions (160 models $\times$ 512 records) are present, and a training-logit replay verifies checkpoint loading. These are constructed challenges, not random deployment samples or guaranteed historically untouched data; Appendix I and Appendix K retain provenance, overlap, evaluation settings, and seed-bootstrap details.

Table 4. Final-checkpoint accuracy on the 320 named challenge records; paired-seed bootstrap intervals, conditional on fixed items.

Scale	WY accuracy %	EN accuracy %	WY balanced accuracy %	EN balanced accuracy %	WY-EN accuracy, pp (pointwise 95% interval)
1M	52.11	50.56	50.10	48.81	1.55 [-0.58, 3.69]
10M	51.68	51.05	49.85	48.59	0.63 [-0.66, 1.91]
100M	51.86	50.91	49.14	48.99	0.95 [-0.67, 2.52]

On the 320 named challenges, both packages approach the 50% balanced-accuracy baseline (Table 4); majority-label prediction yields 61.25% ordinary accuracy. Appendix I retains all groups: pointwise, unadjusted intervals favor EN on the 100M focal-item challenge and all three scales of the separate 192-record synthetic diagnostic.

The same records use 4,030 WY versus 6,720 EN tokens (40.03% fewer), with 40.88%, 40.35%, and 40.17% lower forward FLOPs across scales. Cheaper processing with poor accuracy is not a demonstrated useful inference advantage. Only final checkpoints are evaluated, so this is not cost to a held-out quality target.

A source-only diagnostic finds training-unseen input-token IDs in 304/320 WY and 296/320 EN challenge records. Their embeddings received no task-data gradient. This is a plausible contributor, not a proven sole cause or grounds for removing failures; no model, tokenizer, or item was changed.

4.6 Unified generator supplement: equal exposure, equal nominal tokens, and output cost

Twenty fresh pairs of 11,237,376-parameter autoregressive generators solve zero/one-swap BOX problems with paired initial tensors and canonical example order. Train/dev/test have 1,024/512/1,024 records, configuration-disjoint within this batch. Earlier development informed the task, tokenizers, output format, and budget and reused this evaluation inventory. Thus, held-out configurations here are not a newly blind benchmark. Realized tokenizer vocabularies differ (EN 359, WY 512); both decoders retain 512 output slots. Appendix J gives settings and lineage.

Each trajectory saves checkpoints at 176 full training passes (180,224 presentations; 2,816 updates) and the first complete batch reaching 16 million nonpadding input-plus-target tokens. The pass count was development-informed; stopping is budget-based, not accuracy-based. Dev512 is observed every nominal 250,000 tokens and at the exact exposure point; both test checkpoints are evaluated after training ends. Token-indexed learning rates differ in phase at equal exposure, so comparisons concern whole representation/encoding/training packages.

Equal exposure. All 40 runs complete; all 20 pairs enter both analyses. At 176 passes, fixed encodings yield 6,037,328 WY versus 15,936,448 EN training tokens (62.12% fewer, identical across pairs). Test complete-output accuracy is WY 97.86%, EN 95.03% (Table 5). Logical training matrix-operation FLOPs are approximately 65.23% lower, an accounting estimate (formula and totals in Table J4), not measured runtime. This is an alternative cost account, not an additional 65.23% saving to add to 62.12%. Figure 3 marks equal exposure on the development curves.

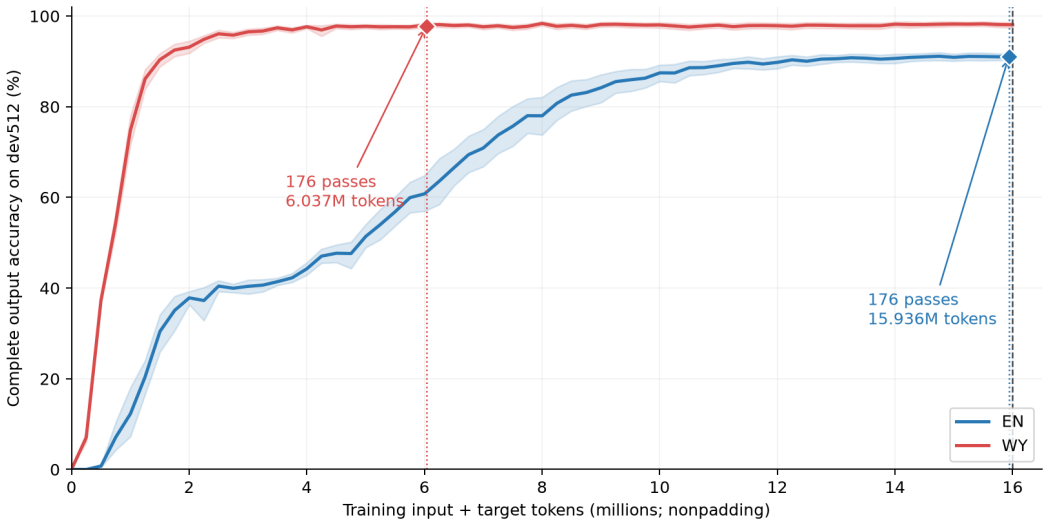

Figure 3. Mean dev512 complete-output accuracy, all 20 pairs. Diamonds mark 176 passes; the right boundary marks nominal 16M tokens. Coordinates use actual-token means at scheduled observations; lines connect unsmoothed measurements. Shading is pointwise 95% seed-bootstrap intervals, not a simultaneous band. Test1024 scores are separate (Table 5); Figure J2 shows every pair. EN dev maximum 93.75% coincides with the 32 four-in-one failures (post-hoc diagnostic, Table J5).

Matched nominal token budget. Complete-batch overshoot is reported in Table 5. Accuracy is WY 98.09%, EN 95.15%; WY receives 477,632 versus 180,992 EN presentations (2.64 times as many), not a training-token saving. Both quality contrasts use Bonferroni-adjusted paired-seed intervals. Input-plus-generation means are 33.614 WY versus 88.406 EN tokens: mean paired reduction 61.98% (descriptive 95% interval [61.97%, 61.98%]), including malformed and erroneous outputs. This interval describes output-length variation across seeds on fixed inputs, not new-task uncertainty or matched-test-quality deployment savings.

Layout concentration. A post-hoc raw-output audit identifies 16 test items sharing one initial configuration: all four items in box D, a layout/list length absent from training. Every EN seed has exactly these 16 unsegmented outputs at both checkpoints. This subgroup contributes arithmetically 43.28%/38.94% of the exposure/token quality gaps. On the remaining 1,008 items, WY leads by

1.63/1.82 points, with seven/three reverse pairs. Table J5 retains both strata and all-item results. This is not an isolated test of linguistic length generalization.

Table 5. All 20 pairs: test accuracy (%) and training counts (EN / WY; M tokens, k presentations). Differences are percentage points; 97.5% paired bootstrap intervals are Bonferroni-adjusted for the two quality contrasts (approximation detailed in Appendix J). Rows match exposure or nominal tokens, not both; only row 1 saves training tokens. One pair favors EN at each endpoint (Figure J1). Differences use unrounded scores.

Checkpoint	Tokens (M)	Presentations (k)	EN	WY	WY minus EN
176 passes	15.936 / 6.037	180.224 / 180.224	95.03	97.86	+2.83 [+1.62, +4.18]
Nominal 16M	16.004 / 16.000	180.992 / 477.632	95.15	98.09	+2.93 [+1.77, +4.13]

Visible steps. Nominal-budget step-token reductions have allocation bounds of 81.76–82.08%, not confidence intervals. At this checkpoint, unsegmented outputs number 320/20,480 EN and zero WY; at equal exposure, 320 EN and one WY. All remain in quality and total cost. Successful segmentation does not imply correct answers. Steps are already included in request tokens and cannot be added as extra savings. These measurements establish neither hidden reasoning nor a causal benefit of emitting steps (Appendix J).

5 Discussion and limitations

The studies connect acquisition efficiency with generator quality and request-token accounting. Lower cost survives primary 10M acquisition with similar repetitions; generators separate equal-exposure training economy, nominal-token quality, and shorter emitted outputs. Both compare packages, not linguistic causes.

Matched tokens do not match FLOPs; steps belong within request cost. Certification savings are not full-run savings: classifiers continue to 4S.

General-language claims require evidence on expressive capability, real-corpus scaling, conversion overhead, and other tokenizers, tasks, and deployed models.

Classifier acquisition permits inventory memorization; rendering, tokenization, and compute-indexed learning rates co-vary. Persistence was development-informed; caps censor harder criteria, and costs exclude preparation/monitoring. Poor challenge transfer with token-ID support mismatch limits generalization. Nonattainments remain visible; runtime and price savings are unestablished.

Fresh generator seeds do not undo development reuse of the evaluation inventory. Intervals capture fixed-inventory seed variation, not task selection. Rendering, vocabulary, schedule, and layout support remain entangled; Table J5 is post hoc. Without a no-step control, trace benefits are not causal. Neither quality interval establishes ±3-point equivalence. Appendix J retains adverse history.

6 Conclusion

The primary 10M pairs average 50.38% acquisition-FLOPs savings with similar repetitions. In the development-informed generator supplement, WY uses 62.12% fewer training tokens at equal exposure and has higher mean test quality at both checkpoints. At nominal-token parity, WY receives 2.64 times the presentations; request tokens are 61.98% lower, including visible steps and errors. Step savings are part of this total. A training-unseen layout contributes 39–43% of quality gaps arithmetically, each endpoint has a reverse pair, and evaluation data were development-reused. These task-specific package benefits establish neither language-intrinsic effects, hidden reasoning, nor matched-test-quality deployment savings.

Reproducibility statement

The evidence/replay component contains classifier surfaces, tokenizers and the frozen plan, generator inputs and tokenizers, all 81,920 generator test predictions (40 models $\times$ 2 checkpoints $\times$

1,024 items), run summaries, statistical-replay code, and primary reference analyses. It excludes full training sources, logs, optimizer states, and checkpoints, which are retained separately. Appendix F distinguishes the included checks from full-archive recomputation; Appendices A, E, J, and K specify costs, encodings, settings, and source fingerprints. The component is not a complete training-reproduction release.

For the unified supplement, retained evidence comprises executed sources and inputs, paired initialization hashes, both checkpoint identities, update ledgers, raw generated outputs, and an independent recount. Appendix J reports settings, coverage, all pairs, and development lineage. The research archive is not yet a public release.

All 45 pre-specified scale/quality/span combinations, including nonattainment, are reported in the appendices. Source-overlap diagnostics are identified separately and are not additional confirmation samples.

AI USE STATEMENT

The research concept, decisions about scope and representation, sample review, resource provision, and insistence on inspecting curves and comparing a common quality requirement were author contributions. AI systems assisted implementation, analysis, interpretation of results, critical review, translation, and drafting; they are not human authors or independent human scientific votes. No outside human annotators or reviewers were recruited for the experiments. Author-side checks do not establish population-level linguistic naturalness.

AI assistance also covered protocol refinement, implementation of synthetic-data generation and bilingual rendering, literature checks, and figures. Task instances were produced by deterministic programs, not accepted solely on an AI judgment. Semantic replay and separate numerical recomputation checked correctness. The human author remains responsible for the content.

REFERENCES

Orevaoghene Ahia, Sachin Kumar, Hila Gonen, Jungo Kasai, David Mortensen, Noah Smith, and Yulia Tsvetkov. Do all languages cost the same? tokenization in the era of commercial language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 9904–9923, 2023. URL https://aclanthology.org/2023.emnlp-main.614/.

Maged S. Al-shaibani and Irfan Ahmad. Consonant is all you need: a compact representation of English text for efficient NLP. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 11578–11588, 2023. doi: 10.18653/v1/2023.findings-emnlp.775. URL https://aclanthology.org/2023.findings-emnlp.775/.

Omer Goldman, Avi Caciularu, Matan Eyal, Kris Cao, Idan Szpektor, and Reut Tsarfaty. Unpacking tokenization: Evaluating text compression and its correlation with model performance. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 2274–2286, 2024. doi: 10.18653/v1/2024.findings-acl.134. URL https://aclanthology.org/2024.findings-acl.134/.

Najoung Kim and Sebastian Schuster. Entity tracking in language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3835–3855, 2023. URL https://aclanthology.org/2023.acl-long.213/.

Nayoung Lee, Kartik Sreenivasan, Jason D. Lee, Kangwook Lee, and Dimitris Papailiopoulos. Teaching arithmetic to small transformers. *arXiv preprint arXiv:2307.03381*, 2023. URL https://arxiv.org/abs/2307.03381. Code: https://github.com/lee-ny/teaching_arithmetic.

Belinda Z. Li, Zifan Carl Guo, and Jacob Andreas. (how) do language models track state? In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 34429–34452, 2025. URL https://proceedings.mlr.press/v267/li25r.html.

Leonard Li. LangMatch. Project page and public repository, 2026. URL https://lsamc.website/projects/langmatch/. Exploratory project report; accessed September 14, 2026.

Tomasz Limisiewicz, Artidoro Pagnoni, Srini Iyer, Mike Lewis, Sachin Mehta, Alisa Liu, Margaret Li, Gargi Ghosh, and Luke Zettlemoyer. Compute optimal tokenization. *arXiv preprint arXiv:2605.01188*, 2026. URL https://arxiv.org/abs/2605.01188v2. Version 2.

Alisa Liu, Jonathan Hayase, Valentin Hofmann, Sewoong Oh, Noah A. Smith, and Yejin Choi. Superbpe: Space travel for language models. In *Second Conference on Language Modeling*, 2025. URL https://superbpe.github.io/.

Niklas Muennighoff, Alexander M. Rush, Boaz Barak, Teven Le Scao, Aleksandra Piktus, Nouamane Tazi, Sampo Pyysalo, Thomas Wolf, and Colin Raffel. Scaling data-constrained language models. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://arxiv.org/abs/2305.16264.

Aleksandar Petrov, Emanuele La Malfa, Philip Torr, and Adel Bibi. Language model tokenizers introduce unfairness between languages. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/74bb24dca8334adce292883b4b651eda-Abstract-Conference.html.

Simiao Ren, Xingyu Shen, Yuchen Zhou, Dennis (Tsang) Ng, and Ankit Raj. Chinese language is not more efficient than English in vibe coding: A preliminary study on token cost and problem-solving rate. *arXiv preprint arXiv:2604.14210*, 2026. URL https://arxiv.org/abs/2604.14210v1. Version 1.

Craig W Schmidt, Varshini Reddy, Haoran Zhang, Alec Alameddine, Omri Uzan, Yuval Pinter, and Chris Tanner. Tokenization is more than compression. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 678–702, 2024. doi: 10.18653/v1/2024.emnlp-main.40. URL https://aclanthology.org/2024.emnlp-main.40/.

Chenglei Si, Zhengyan Zhang, Yingfa Chen, Fanchao Qi, Xiaozhi Wang, Zhiyuan Liu, Yasheng Wang, Qun Liu, and Maosong Sun. Sub-character tokenization for Chinese pretrained language models. *Transactions of the Association for Computational Linguistics*, 11:469–487, 2023. doi: 10.1162/tacl_a_00560. URL https://aclanthology.org/2023.tacl-1.28/.

Xuemei Tang, Qi Su, Jun Wang, Yuhang Chen, and Hao Yang. Automatic traditional ancient chinese texts segmentation and punctuation based on pre-training language model. In *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pp. 678–688. Chinese Information Processing Society of China, 2021. URL https://aclanthology.org/2021.ccl-1.61/. In Chinese.

A Exact training-arithmetic accounting

For one unpadded microbatch of T input tokens, including BOS/EOS, $m(T) = aT^2 + bT + c$. This is an algebraic reduction of the executed operator rules, not a fit to observed costs.

Table A1. Coefficients and optimizer-related additions in the training-FLOPs account.

Scale	a	b	c	Extra per warmup update	Extra per decay update
1M	6,688	5,398,506	-5,685	33,023,800	33,023,803
10M	26,976	58,789,770	-24,995	487,209,640	487,209,643
100M	121,728	599,358,322	-121,437	8,051,898,584	8,051,898,587

For L layers, width d, h heads, and FFN width $4d$: $a = L(12d + 10h)$; $b = L(72d^2 + 126d + 4 - 2h) + 23d + 2$; $c = L(1 - 13d) + 10d + 23$.

The extra optimizer-update term is $(A + 16)P + 14K + k$, where A is accumulation, P parameter count, K parameter-tensor count, and $k = 4$ in warmup or $k = 7$ in decay. It is charged once per optimizer update, in addition to all A microbatch costs. Warmup applies when pre-update cumulative cost is at most S. Negative polynomial constants do not imply negative costs.

The algebra matches the bound accounting module at every supported integer length 2–512 for each scale and in both update phases. This checks presentation against the declared operator convention, not against hardware instruction counts. Multiply-accumulate counts as two operations; elementary scalar/unary arithmetic counts as one; comparisons, indexing, copies and zero fills count as zero. Evaluation, saving, rendering, tokenizer construction and tokenization are outside this training-cost boundary.

B Frozen encoded inventory

Table B1. Frozen inventory lengths in tokens, including BOS/EOS.

Condition	Rows	Tokens per inventory pass	Minimum length	Maximum length
WY	720	7934	4	20
EN	720	15920	7	38

The inventory has 328 true and 392 false labels. A majority-negative prediction scores 54.44%. Per-pass token reduction is 50.1633%; this is descriptive compression, not a measured learning-speed or cost-to-quality saving. All three scale releases contain byte-identical encoded inventories.

All 720 rows are training and acquisition-observation rows. Historical role labels do not create a test split. Matching these rows does not establish semantic independence of the historical tokenizer corpus; see the manuscript provenance limitation.

C Every pre-specified quality/span readout

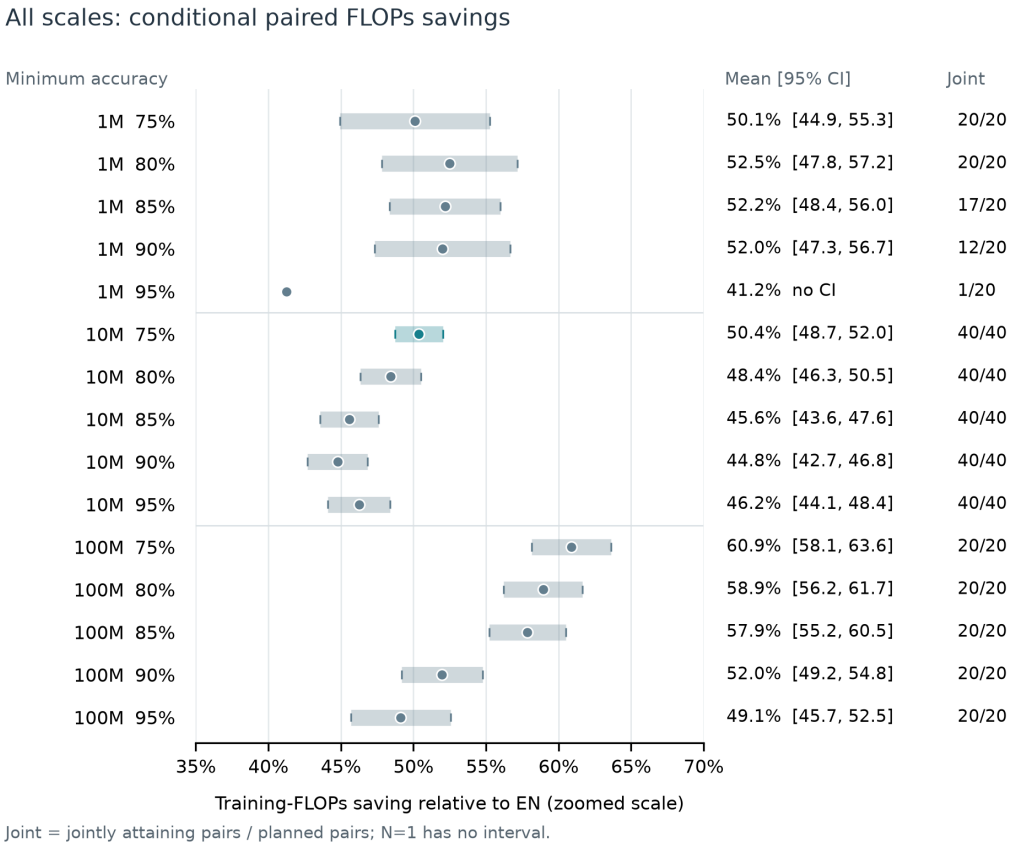

Figure C1. The point-and-interval display of Figure 2 for all three scales at the unchanged 7,200-presentation span and all pre-specified minima. Percentages are conditional on joint attainment, with jointly attained/planned counts printed on every row. Only one 1M pair jointly attains 95%, so no interval is estimated for its conditional percentage; the table's restricted-cost difference uses all 20 pairs and retains its interval. All other attainment states remain in the complete tables below. Scales are not pooled; pointwise intervals are not simultaneous bands.

Difference is restricted WY-EN cost divided by S. Percent savings are means of per-pair ratios among jointly attained pairs only. Nonattainment and technical incompleteness are distinct. The intervals below are pointwise approximate paired-t intervals, not a simultaneous band. Only 10M at 75%/7200 is primary; all other cells remain supportive or sensitivity analyses.

C.1 1M READOUTS

Table C1. Part 1/6: 1M; Attainment.

Minimum %	Span visits	Joint / planned	WY only	EN only	Neither	Incomplete
75	7200	20/20	0	0	0	0
75	1440	20/20	0	0	0	0
75	3600	20/20	0	0	0	0
80	7200	20/20	0	0	0	0
80	1440	20/20	0	0	0	0
80	3600	20/20	0	0	0	0
85	7200	17/20	3	0	0	0
85	1440	20/20	0	0	0	0
85	3600	18/20	2	0	0	0
90	7200	12/20	8	0	0	0
90	1440	15/20	5	0	0	0
90	3600	15/20	5	0	0	0
95	7200	1/20	19	0	0	0
95	1440	9/20	11	0	0	0
95	3600	4/20	16	0	0	0

Table C1. Part 2/6: 1M; Costs; savings condition on joint attainment.

Minimum %	Span visits	Difference / S	95% interval	FLOPs saving %	Token saving %
75	7200	-1.0691	[-1.2727, -0.8654]	50.0953	50.1496
75	1440	-0.9585	[-1.1098, -0.8072]	55.8439	55.8920
75	3600	-1.0010	[-1.1904, -0.8115]	52.4945	52.5461
80	7200	-1.3904	[-1.6426, -1.1381]	52.4939	52.5457
80	1440	-1.1808	[-1.4122, -0.9494]	54.6334	54.6828
80	3600	-1.2581	[-1.4862, -1.0300]	52.6494	52.7010
85	7200	-1.7008	[-1.9592, -1.4425]	52.1764	52.2284
85	1440	-1.4138	[-1.6696, -1.1580]	53.9102	53.9603
85	3600	-1.5799	[-1.8673, -1.2924]	51.9287	51.9810
90	7200	-1.9525	[-2.1408, -1.7642]	51.9986	52.0508
90	1440	-1.8074	[-2.0734, -1.5414]	52.9331	52.9843
90	3600	-1.8852	[-2.1468, -1.6235]	52.1985	52.2505
95	7200	-1.6303	[-1.7607, -1.4999]	41.2358	41.2997
95	1440	-2.0269	[-2.1646, -1.8892]	52.4493	52.5011
95	3600	-1.9469	[-2.0516, -1.8422]	50.0282	50.0826

C.2 10M READOUTS

Table C1. Part 3/6: 10M; Attainment.

Minimum %	Span visits	Joint / planned	WY only	EN only	Neither	Incomplete
75	7200	40/40	0	0	0	0
75	1440	40/40	0	0	0	0
75	3600	40/40	0	0	0	0
80	7200	40/40	0	0	0	0
80	1440	40/40	0	0	0	0
80	3600	40/40	0	0	0	0
85	7200	40/40	0	0	0	0
85	1440	40/40	0	0	0	0
85	3600	40/40	0	0	0	0
90	7200	40/40	0	0	0	0
90	1440	40/40	0	0	0	0
90	3600	40/40	0	0	0	0
95	7200	40/40	0	0	0	0
95	1440	40/40	0	0	0	0
95	3600	40/40	0	0	0	0

Table C1. Part 4/6: 10M; Costs; savings condition on joint attainment.

Minimum %	Span visits	Difference / S	95% interval	FLOPs saving %	Token saving %
75	7200	-0.7447	[-0.7811, -0.7084]	50.3769	50.6551
75	1440	-0.6044	[-0.6453, -0.5636]	53.5447	53.8052
75	3600	-0.6681	[-0.7087, -0.6275]	51.3845	51.6572
80	7200	-0.8139	[-0.8654, -0.7624]	48.4285	48.7176
80	1440	-0.6667	[-0.7063, -0.6270]	51.1672	51.4410
80	3600	-0.7516	[-0.7922, -0.7109]	50.3761	50.6545
85	7200	-0.8648	[-0.9260, -0.8035]	45.5852	45.8902
85	1440	-0.7403	[-0.7872, -0.6935]	48.7335	49.0209
85	3600	-0.7871	[-0.8410, -0.7332]	46.1611	46.4631
90	7200	-1.0122	[-1.1008, -0.9236]	44.7667	45.0763
90	1440	-0.8358	[-0.9030, -0.7687]	45.9356	46.2387
90	3600	-0.9036	[-0.9843, -0.8230]	44.4371	44.7486
95	7200	-1.3729	[-1.4905, -1.2552]	46.2459	46.5472
95	1440	-1.0228	[-1.1087, -0.9369]	44.2059	44.5186
95	3600	-1.2304	[-1.3343, -1.1265]	46.4413	46.7415

C.3 100M readouts

Table C1. Part 5/6: 100M; Attainment.

Minimum %	Span visits	Joint / planned	WY only	EN only	Neither	Incomplete
75	7200	20/20	0	0	0	0
75	1440	20/20	0	0	0	0
75	3600	20/20	0	0	0	0
80	7200	20/20	0	0	0	0
80	1440	20/20	0	0	0	0
80	3600	20/20	0	0	0	0
85	7200	20/20	0	0	0	0
85	1440	20/20	0	0	0	0
85	3600	20/20	0	0	0	0
90	7200	20/20	0	0	0	0
90	1440	20/20	0	0	0	0
90	3600	20/20	0	0	0	0
95	7200	20/20	0	0	0	0
95	1440	20/20	0	0	0	0
95	3600	20/20	0	0	0	0

Table C1. Part 6/6: 100M; Costs; savings condition on joint attainment.

Minimum %	Span visits	Difference / S	95% interval	FLOPs saving %	Token saving %
75	7200	-1.0618	[-1.1494, -0.9741]	60.8729	61.1371
75	1440	-0.8236	[-0.9182, -0.7290]	63.9280	64.1716
75	3600	-0.9842	[-1.0717, -0.8967]	63.6618	63.9070
80	7200	-1.0855	[-1.1705, -1.0006]	58.9417	59.2188
80	1440	-0.9264	[-0.9843, -0.8685]	63.7416	63.9865
80	3600	-1.0022	[-1.0851, -0.9193]	60.9449	61.2085
85	7200	-1.1589	[-1.2401, -1.0776]	57.8642	58.1485
85	1440	-0.9675	[-1.0652, -0.8697]	60.2219	60.4904
85	3600	-1.0625	[-1.1444, -0.9806]	58.9022	59.1795
90	7200	-1.1523	[-1.2591, -1.0455]	51.9686	52.2927
90	1440	-1.0772	[-1.1629, -0.9915]	58.3718	58.6527
90	3600	-1.1080	[-1.1901, -1.0258]	55.4801	55.7804
95	7200	-1.3057	[-1.4640, -1.1475]	49.1272	49.4704
95	1440	-1.1362	[-1.2518, -1.0207]	51.5662	51.8930
95	3600	-1.1967	[-1.3370, -1.0565]	49.6836	50.0230

D Reproduction of this supplement

Input independent-analysis SHA-256: `dafd7e8a3a33df1a2e4d41b6009c8f8982e52981fc7c24291924c5a7c5046f2c`. Frozen plan SHA-256: `8e9320634cd767cde4209c18ac888ae64dcd35df00aeb872f87a9d11e32791a5`. Bound accounting source SHA-256: `4ae7132b482bc3ecd4c93e9975742cfe6ec2d3cddb2cd8a1881a1b9f360dce57`.

Run `python -B build_supplement.py --package package --analysis <analysis-directory>/analysis.json --out <analysis-directory>/supplement.md` from the preparation directory. It checks release identities and input bytes, checks the algebra, and presents all existing readouts without changing the frozen analyzer. Full raw-record recomputation uses analyze_confirmation_v2.py, not this presentation script. Unrecovered scales remain incomplete; they are not silently dropped or inferred from process completion.

E Actual paired examples and tokenizer settings

The following are frozen inventory rows 86 and 87 (zero-based), not newly authored examples. They are a short true/false illustration with four boxes and three swaps, selected for exposition rather than model outcomes or maximal compression. They do not represent the full distribution of problem difficulty. Historical CALIBRATION labels in these source rows do not create a held-out split in the present training run.

E.1 Inventory row 86: label true

WY (7 tokens including BOS/EOS):

> 初A盒有W物B盒空C盒空D盒空乃AB互易之乃AC互易之乃AD互易之问W物今在B盒乎

EN (13 tokens including BOS/EOS):

> Initially, box A contains item W. Box B is empty. Box C is empty. Box D is empty. Then, boxes A and B swap all their contents. Then, boxes A and C swap all their contents. Then, boxes A and D swap all their contents. Is item W now in box B?

E.2 Inventory row 87: label false

WY (7 tokens including BOS/EOS):

> 初A盒有W物B盒空C盒空D盒空乃AB互易之乃AC互易之乃AD互易之问W物今在D盒乎

EN (13 tokens including BOS/EOS):

> Initially, box A contains item W. Box B is empty. Box C is empty. Box D is empty. Then, boxes A and B swap all their contents. Then, boxes A and C swap all their contents. Then, boxes A and D swap all their contents. Is item W now in box D?

In both rows W starts in A, moves to B at the first swap, and stays there while empty A is exchanged with C and D. Asking whether W is in B is true; asking whether it is in D is false. A transition irrelevant to the queried item still appears in both representations. This example is not evidence of the algorithm learned by a model.

WY tokenizer SHA-256: `eb9b47b21b4ad4858d5bffdcd8a2c12f7b06cbd56131fde490482497271ace01`.

EN tokenizer SHA-256: `56c11730640e06e309ebcd8bff82dd570ca105e2c05afaf41d98b9d262eae83b`.

Both serialized tokenizers are ByteLevel BPE with 492 vocabulary entries, no normalizer, no truncation or padding, and deterministic merges (dropout is null). The pre-tokenizer has add_prefix_space=false and use_regex=false. Consequently, merges can span whitespace and conventional word boundaries; these results must not be assumed to transfer to an off-the-shelf word-boundary tokenizer. Special IDs are PAD=0, BOS=1, EOS=2 and UNK=3; the input template adds BOS/EOS. Encodings and exact surface bytes, rather than a new tokenizer fit, are used for reproducing training.

The paired-surface file is identified by basename `FORMAL720.paired.jsonl` and SHA-256 `333b8d9931d41d533ae2919962847657bd63789c237056d643ac670f287f7a7a`. Tokenizer file identities are given above. These content identifiers refer to the retained research archive; they are not a claim that the evidence/replay component includes the full raw archive. These surfaces, both tokenizers, and the frozen confirmation plan are also included under neutral filenames in the evidence/replay component.

Table E1. Post-hoc encoding-length check on the same 720 frozen classifier input surfaces, without normalization, BOS/EOS, padding, labels, or outputs. Public-tokenizer counts use tiktoken 0.13.0 encode_ordinary; frozen BPE uses tokenizers 0.22.2. UTF-8 is a byte count, not a token measure.

Encoding / unit	EN total	WY total	Reduction (%)
Frozen BPE / tokens	14,480	6,494	55.15
cl100k_base / tokens	81,244	59,804	26.39
o200k_base / tokens	81,244	45,060	44.54
UTF-8 / bytes	321,692	117,984	63.32

Adding the actual classifier BOS/EOS adds 1,440 tokens to each frozen-BPE total: EN 15,920, WY 7,934 (50.16% reduction). This differs from the no-special-token row. Public-tokenizer reductions establish shorter encodings on these inputs only, not learning cost or quality with those tokenizers; no models were retrained.

F Offline analysis and figure commands

The following primary-analysis commands require the separately retained original analysis sources and raw archives; they cannot run from the evidence/replay component alone. Run from the primary preparation directory. Set RESULTS_1M, RESULTS_10M_A, RESULTS_10M_B, and RESULTS_100M to the recovered 1M, two disjoint 10M shards, and 100M result directories, respectively; the variables are archive locations, not host addresses. The first command recomputes all three scales from four raw result roots, using a new output directory. The next two regenerate the numerical supplement (A-G only, not the authored H-K) and diagnostic plots. The final command regenerates Figures 1, 2, and C1 from the retained, hash-checked `analysis_20260915_all_verified/analysis.json` into its `figures_20260916` subdirectory, not from `analysis_reader_check`. Appendix I gives challenge-analysis commands.

```
python -B analyze_confirmation_v2.py --package package
--plan-sha256
8e9320634cd767cde4209c18ac888ae64dcd35df00aeb872f87a9d11e32791a5
--results "$RESULTS_1M" "$RESULTS_10M_A" "$RESULTS_10M_B"
"$RESULTS_100M" --out analysis_reader_check

python -B build_supplement.py --package package --analysis
analysis_reader_check/analysis.json --out
analysis_reader_check/supplement.md

python -B plot_confirmation_results.py --analysis
analysis_reader_check/analysis.json --plan
package/confirmation_plan.json --out analysis_reader_check/figures

python -B plot_paper_intervals_20260916.py
```

RESULTS_100M is the fully recovered 100M queue. Do not substitute terminal status for raw observations. The analyzer checks disjoint task identities, planned seeds, source identity, raw-logit scores, update costs, observation completeness, initialization pairing and process exits. Inspect every run's status and all scale denominators, not only a zero command exit code.

The local numerical/plotting environment used for this report was Python 3.12.3, NumPy 2.4.6, SciPy 1.17.1 and Matplotlib 3.11.0. Analysis requires NumPy/SciPy but not CUDA; plotting requires Matplotlib. Source tokenization replay used tokenizers 0.22.2. These are analysis/replay versions, not a claim that the training processes used this local environment. Training software and device settings are retained separately in each run's launch.json.

F.1 Evidence/replay component

From the extracted component root, run `python -B verify_evidence.py` and `python -B generator/replay_statistics.py` with Python and NumPy. The first checks the SHA256 manifest, joins all 81,920 raw generator test predictions (40 models $\times$ 2 checkpoints $\times$ 1,024 items) by canonical question ID, and recomputes full-output correctness and layout counts. It does not rerun inference or validate training ledgers. The second reproduces the original generator statistics from all 40 independently recounted run summaries and requires exact agreement with the retained reference. Generator operational-path fields are omitted; 160 classifier operational strings are replaced by explicit withholding markers. Numerical fields and raw prediction bytes are unchanged. The statistical routine differs from the archived analyzer only by neutral host labels and removal of plotting. Primary analyses are supplied as reference outputs, not a fresh raw-record recomputation. Full training reproduction and complete primary/generator ledger or checkpoint checks require the separately retained original sources and archives; they are not supplied by this component. The README gives these limitations. The earlier internal source archive is not a submission attachment.

G Item presentations versus tokens: descriptive accounting

This post-result accounting explanation uses the already fixed 75%/7200 recognition endpoints. It is not another primary test or a new endpoint. Here V denotes cumulative training-item presentations including repeats and the complete passing window, not unique examples, and T denotes cumulative input tokens. For each jointly attained pair the exact identity is $T_{\mathrm{WY}}/T_{\mathrm{EN}} = (V_{\mathrm{WY}}/V_{\mathrm{EN}}) \times [(T_{\mathrm{WY}}/V_{\mathrm{WY}})/(T_{\mathrm{EN}}/V_{\mathrm{EN}})]$. This separates how many item presentations were processed from how many tokens each presentation averaged; it is not a causal mediation analysis.

Table G1. Training-item presentations at primary recognition; post-result descriptive accounting.

Scale	Joint/planned	Mean WY visits	Mean EN visits	Mean per-pair visit saving %	Median visit saving %	WY fewer / equal / more visits
1M	20/20	48764.00	50364.00	-0.0622	2.2911	11/0/9
10M	40/40	31244.00	31644.00	0.9165	-0.9804	18/2/20
100M	20/20	25824.00	33376.00	21.9482	22.5512	19/0/1

Positive visit savings mean fewer presentations; negative values mean more. Means of per-pair percentages differ from ratios of aggregate means. Averaging the multiplicative identity does not permit multiplying separate marginal means or adding percent savings. A token-cost advantage is not automatically evidence of fewer training examples, faster wall-clock training, or a language-intrinsic learning advantage. The FLOPs-anchored learning-rate schedule and representation-specific segmentation remain part of the compared packages. No confidence interval or new hypothesis test is attached to this post-result visit decomposition.

H Does the passing window create the cost contrast?

This is a post-result accounting diagnostic, not a new test, endpoint, or selection rule. The main endpoint remains the charged end of the earliest window satisfying the pre-specified 75%/7200 persistence rule. For every pair, we compare its already recorded first observed 75% crossing, the beginning of its eventually qualifying window, and the end of that window. The latter two points use later observations to identify a qualifying sequence; its start is not an independently certified stopping time.

Table H1. Savings at three trajectory landmarks; only the window end is the primary endpoint.

Scale	Pairs	Mean per-pair FLOPs saving at first observed crossing	At accepted window start	At charged window end
1M	20	57.7207%	49.8030%	50.0953%
10M	40	52.3001%	50.4624%	50.3769%
100M	20	61.9658%	63.9302%	60.8729%

All pairs have lower WY costs at all three recorded points. At 10M, the accepted window's incremental cost represents a mean 23.25% of WY endpoint cost and 22.84% of EN endpoint cost. Its contribution to the aggregate EN-minus-WY cost difference is 22.44%. These are distinct denominators: a share of a cost difference is not a share of either condition's cost or a causal mediation estimate.

The approximate halving at 10M is therefore not created solely by appending the confirmation window. The window still matters to the estimand and magnitude: at 100M, the corresponding mean saving changes from 63.93% at the accepted starts to 60.87% at the charged ends. First crossing, retrospective window start, and certified window end must not be substituted for one another or interpreted as convergence. The original pre-specified shorter-window analyses remain separate.

Reproduction: `python -B explain_passing_window_cost.py --out <new-output-file.json>` from the preparation directory. The script checks the unchanged analysis SHA-256, reads all planned pairs, and asserts valid within-cap ordering. Output retained here: `passing_window_cost_explanation.json`. The calculation was independently reproduced during the manuscript review. No training, sampling rule, threshold, or main estimate was modified.

I Post-result fixed-checkpoint challenge evaluation

This evaluation was specified after the confirmation results were known and before these new predictions. It neither changes the original primary test nor constitutes a newly sealed prospective test population. All 160 final, fixed-budget checkpoints were retained: 20/40/20 pairs at 1M/10M/100M. No fine-tuning, new tokenizer fit, early-checkpoint search, or selection by predictions was performed. Each model evaluated all 512 pre-existing guard records. All evaluations completed, with zero failed or missing runs; an independent reader recomputed 81,920 decisions from raw logits and cross-checked each against the streaming records. The first 16 fixed training records replayed exactly against the saved final logits for every model. This validates the loaded evaluation path; it is not evidence that the new records have familiar token support.

I.1 Source identities and overlap

The original renderer/parser was reused without wording changes. WY and EN sentences parse to the same canonical sample; labels were checked by an independent box-content swap simulator. The four fixed groups are shown below. Query overlap normalizes unordered endpoints of each swap. Structure overlap additionally removes all box/item naming and query choice while retaining initial occupancy and the ordered swap sequence. It does not remove every possible algorithmic equivalence. Overlapping records are retained with flags, not covertly described as novel.

Table I1. Challenge-source composition, label baselines, and overlap counts.

Group	Rows	Majority accuracy %	Training-query overlap	Training-structure overlap	Tokenizer-query overlap	Tokenizer-structure overlap
Successor heldout	144	75.00	0	0	0	36
Empty/multi boundary	32	50.00	0	0	0	16
Focal-item shortcut challenge	144	50.00	0	8	0	40
Synthetic matrix	192	67.71	5	21	50	99

The first three groups together contain 320 records, with 124 true and 196 false labels. They have no training-query or complete-world overlap. Their combined majority-class baseline is 61.25%; balanced accuracy uses the mean of true-class and false-class recall and has a 50% constant-prediction baseline. A pre-fixed sensitivity excludes the eight training-structure equivalents, retaining 312 records. These groups were designed as boundary and counterexample checks, not as a random representative sample from natural-language deployment. They are untrained by the classifier, but not guaranteed free of tokenizer-construction exposure or earlier scientific inspection.

I.2 All source-group results

Scores average over all 20/40/20 model pairs, respectively. Intervals resample paired initialization seeds, conditional on the fixed rows, using 10,000 percentile-bootstrap resamples and NumPy default_rng seed 20260915. They are pointwise descriptive intervals, not a simultaneous band or new primary hypothesis test. Correlated problem rows are not treated as independent replications. The full per-seed scores, balanced-accuracy intervals, and exact depth/box/item/label strata are retained in the supplementary analysis JSON.

Table I2. All source-group accuracies; pointwise paired-seed bootstrap intervals conditional on fixed rows.

Group	Scale	WY accuracy %	EN accuracy %	WY balanced accuracy %	EN balanced accuracy %	Accuracy difference pp [95% interval]
Successor heldout	1M	54.24	49.86	52.18	47.04	4.38 [0.14, 8.54]
Successor heldout	10M	54.27	52.14	52.62	47.42	2.14 [-0.16, 4.39]
Successor heldout	100M	55.90	51.15	52.73	47.57	4.76 [1.22, 8.19]
Empty/multi boundary	1M	49.69	50.16	49.69	50.16	-0.47 [-5.47, 4.69]
Empty/multi boundary	10M	48.52	47.19	48.52	47.19	1.33 [-1.25, 4.06]
Empty/multi boundary	100M	47.66	45.62	47.66	45.62	2.03 [-1.88, 5.94]
Focal-item shortcut challenge	1M	50.52	51.35	50.52	51.35	-0.83 [-2.12, 0.59]
Focal-item shortcut challenge	10M	49.79	50.82	49.79	50.82	-1.02 [-2.33, 0.28]
Focal-item shortcut challenge	100M	48.75	51.84	48.75	51.84	-3.09 [-4.69, -1.39]
Synthetic matrix	1M	49.24	52.45	46.83	50.61	-3.20 [-6.20, -0.13]
Synthetic matrix	10M	47.98	52.85	45.32	51.54	-4.87 [-6.85, -2.79]
Synthetic matrix	100M	47.37	50.57	44.32	49.90	-3.20 [-5.83, -0.68]

Neither package achieves strong performance on these challenge sets. Some pointwise contrasts favor WY and others EN, including reverse contrasts at 100M in the focal-item group and throughout the synthetic matrix. We do not select favorable groups, equate non-significance with equivalence, or interpret a small relative difference near chance as useful task competence.

The structure-novel 312-record sensitivity gives WY/EN accuracies of 52.31%/50.58% at 1M, 51.75%/51.06% at 10M, and 52.00%/50.90% at 100M. Paired accuracy differences and pointwise intervals are 1.73 [-0.50, 3.97], 0.70 [-0.65, 2.02], and 1.11 [-0.51, 2.69] percentage points. Excluding the eight renamed structures therefore does not change the qualitative conclusion.

I.3 Costs and post-result input-support diagnostic

Over the 320 named challenge records, WY uses 4,030 and EN 6,720 input tokens including BOS/EOS (40.0298% fewer). Applying the frozen forward-only operator account gives 40.8792%, 40.3499%, and 40.1723% fewer FLOPs at the three scales. The independent reader matches the polynomial to the original operator lowering at every encoded length present. Incorrectly answered records remain in the cost totals. These shape-accounting reductions do not establish equivalent useful accuracy, elapsed-time savings, or cost to a held-out accuracy target.

A diagnostic conducted after observing the challenge results counts input support without making new predictions. The training inventory uses 74 WY and 53 EN distinct token IDs, including BOS/EOS, within the respective 492-entry vocabularies. In the named challenge, 304/320 WY and 296/320 EN records contain valid tokenizer IDs absent from classifier training. These account for 1,371/4,030 WY and 1,016/6,720 EN token occurrences. No UNK is introduced: valid learned BPE units can nevertheless be absent from the classifier's training inputs. Their input embeddings receive no task-data gradient during that training, although optimizer weight decay can still affect the parameters. This support mismatch is a plausible contributor to failure, not proof of its sole cause. We neither repair the tokenizer nor select a supported-only test set after viewing scores. The original inventory result remains unchanged, and the transfer limitation remains visible.

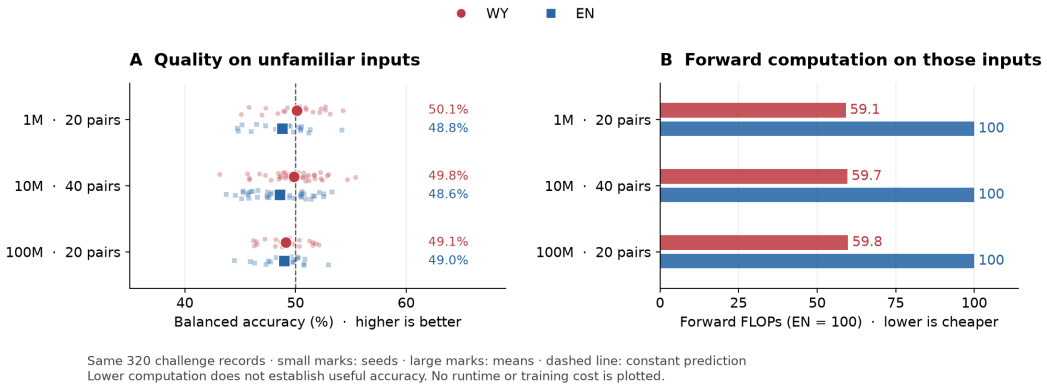

Figure I1. All 160 final classifiers on the same 320 named challenge records. Left: balanced accuracy (equal weight for true and false labels); small marks are individual seeds, large marks are means. The 50% line is the constant-prediction baseline. Right: deterministic forward FLOPs for these inputs, indexed to EN=100 separately at each scale; this is neither runtime nor training cost. WY costs about 60 units where EN costs 100, but both have poor challenge accuracy. Pointwise paired-seed intervals and ordinary accuracy remain in Table 4; no confidence interval is drawn here.

I.4 Reproduction

The retained directory is `work/Paper1_COMMON_QUALITY_CONFIRMATION_PREP_20260913/unseen_supplement_20260915`. The fixed plan SHA-256 is `007520ddeff2597ac9a8acf42308dd0adc529d9bfa34a023e693b0ca155dd203`; it records all input, runner, model-source, tokenizer, and checkpoint identities before the new predictions. `inputs.json`, `source_overlap.json`, all raw `predictions`, `analysis.json`, and `post_result_token_coverage.json` are retained. This separate analysis must not replace the original 80-pair training-analysis JSON. The A-G table generator does not regenerate the authored H–K extensions; retain this full supplement alongside independently recomputed tables rather than overwriting it.

From the preparation directory, `python -B analyze_unseen.py` independently recomputes the supplementary scores from raw logits, and `python -B plot_unseen.py` renders diagnostic quality/cost views of the same analysis, not the presentation layout of Figure I1. Actual prediction execution uses `python -B evaluate_unseen.py run`: one local GPU, one checkpoint at a time, FP32, unpadded batch one, TF32 off, evaluation mode, and no optimizer update. Its PyTorch allocator is capped at 8% of GPU memory. Every loaded checkpoint was checked against its recovered SHA-256. The run took about 409 seconds on the local RTX 5090, but this end-to-end audit time includes file hashing/loading and is not a paired inference-latency experiment or a paper speed claim. Local versions were PyTorch 2.8.0.dev20250621+cu128 and tokenizers 0.22.2; the environment is retained with each prediction record.

J Unified generator supplement

Twenty paired seeds, 9223101–9223120, were fixed before this batch; ten pairs ran on each of two four-GPU hosts, with each pair on the same host. All 40 runs completed normally under a 7,200-second per-model limit including evaluation. No seed replacement, score-triggered extension, or accuracy-based stopping was used. The sample size reflects a bounded replication budget, not a guaranteed power calculation or a conference requirement. The main classifier evidence and prior experiments remain unchanged and are not pooled with this batch.

The complete space of four items independently assigned to four boxes contains 256 initial configurations. A fixed shuffle assigns 128/64/64 configurations to training/development/test. All events and queries derived from a configuration stay in that partition. Each record has zero or one full-content box swap. True and false queries are paired, giving four equally sized depth-by-truth cells in each split. The sampled train/dev/test sets contain 128/63/64 distinct initial configurations in 1,024/512/1,024 records, respectively; one allocated dev configuration is not sampled; 128 training records form a familiar-item diagnostic, not another held-out test. Current split initial configurations and exact question identities are disjoint.

The train/dev/test inventory and renderers are inherited from development. Test records are excluded from this batch's gradient updates and tokenizer fitting, but the same test inventory was evaluated in earlier experiments. In an earlier overlap audit, 113/1,024 test records matched question identities and 166 matched group identities in preceding development bundles. Those limited overlap counts do not erase the later reuse of the entire test inventory. The present estimates describe fresh initialization seeds on a development-reused task, not prospectively unseen task performance.

Each condition uses one fixed rendering system. WY inputs and outputs are unpunctuated controlled Literary Chinese with fixed ASCII identifiers; EN retains spaces, capitalization, and punctuation. The target describes the queried item's initial and subsequent location, names the queried box, compares the final and queried boxes, and gives the answer. It is teacher-provided during supervised training, but evaluation supplies only the question. Greedy decoding is unconstrained, stops on EOS, and has a 512-token output cap; gold steps are not supplied at evaluation. Correctness is independently checked by parsing the question, replaying the box states, and checking the generated trace, comparison, answer, and termination. Full correctness requires the fixed output form, not arbitrary semantically equivalent paraphrases.

Each language fits one byte-level BPE tokenizer with a 512-token cap on its training strings only. The realized vocabularies contain 359 EN and 512 WY entries; the common 512-slot architecture is not a matched realized vocabulary. Input encoding splits around each fixed box/item identifier before BPE application, symmetrically in both languages; no characters or spaces are added to the visible input. Target encoding separates the answer span to support the fixed answer-weighted objective. All encodings round-trip to the original strings. This differs from the inherited 492-token classifier tokenizer and prevents treating the supplement as a tokenizer-controlled replication or a length-only intervention.

The decoder uses six layers, six heads, width 384, context 1,024, vocabulary capacity 512, bias enabled, and dropout 0.2, totaling 11,237,376 trainable parameters. Batch size is 64 with microbatches of eight. Paired initial tensors and canonical sample order are identical. FP32 training disables TF32. The objective is half the mean next-token cross-entropy over the full nonpadding sequence (including the prompt) and half the mean answer-token cross-entropy. Input identifier-boundary encoding and target answer-span separation are fixed symmetrically; exact encodings and padded shapes are logged.

The decoder adapts the MIT-licensed (Lee et al., 2023), itself based on nanoGPT; no pretrained weights or arithmetic training examples are used (Lee et al., 2023). AdamW uses betas (0.9, 0.99), epsilon 1e-8, weight decay 0.1 on eligible linear weights, and gradient-norm clipping at 1.0. Learning rate warms up over 1,066,667 training tokens to 0.001, then follows token-indexed cosine decay toward 0.0001 at 16 million. Equal-exposure checkpoints therefore do not match learning-rate phase. This differs from both the main classifier FLOPs-indexed policy and the earlier generator update-indexed policy.

Training records every update and evaluates dev512 at initialization and the first complete batch crossing each 250,000-token grid point. The extra 180,224-presentation checkpoint is saved sepa-

rately. Both final test1024 evaluations and familiar128 diagnostics occur after training ends, from their respective saved weights. Exact nominal-budget overshoot is retained; it is less than one permitted 16,000-token batch. There are 65 scheduled token observations plus the exact exposure observation per model. Intermediate weights need not all remain stored: raw predictions and ledgers establish the measured curves, while both named endpoint weights are preserved.

The two named full-output test contrasts use paired-seed percentile bootstrap intervals with 100,000 resamples (seed 9223000). Each two-sided interval has nominal 97.5% coverage, Bonferroni-adjusted for this two-contrast family (at least 95% family coverage under the interval approximation). Cost summaries average per-pair percentage reductions, with descriptive 95% bootstrap intervals, rather than a ratio of aggregate means. The unit is 20 paired training seeds, not 20,480 independent requests. Four-cell, answer-only, state, and familiar-item results are descriptive. All intervals condition on the fixed reused inventory and do not account for development selection. The optional ±3-point author margin is not a venue standard; the full intervals do not support equivalence.

Request cost counts actual encoded input and every generated ID, including EOS and malformed outputs. Step bounds allocate unsegmented body tokens either zero or all to steps, propagate both extremes through each pair, then average the paired reductions. The bounds describe allocation uncertainty on observed text, not sampling confidence. Same-exposure segmentation failures are 320/20,480 EN and 1/20,480 WY; matched-token failures are 320/20,480 EN and zero WY. No request reaches the generation cap without EOS. Illegal special-symbol outputs remain errors. Table J1 reports both ledgers without dropping failed responses.

Table J1. All-request token ledgers. Request reduction intervals are descriptive 95% paired-seed bootstrap intervals; step ranges are allocation bounds, not confidence intervals. Request intervals condition on the fixed input inventory.

Checkpoint	EN/request	WY/request	Request reduction % [CI]	Step reduction % [bounds]
176 passes	88.4069	33.6114	61.9810 [61.9770, 61.9856]	[81.7712, 82.0873]
Nominal 16M	88.4057	33.6141	61.9774 [61.9729, 61.9819]	[81.7611, 82.0751]

At equal exposure, WY has higher complete-output accuracy in 18 pairs, ties one, and is lower in one. At the nominal token budget it is higher in 19 pairs and lower in one. Seed 9223114 is reverse at the latter endpoint (97.46% EN, 95.61% WY); its WY depth-one false/true cells score 91.41%/93.75%. Both languages score 100% on familiar128 at the nominal-budget endpoint, so familiar examples do not establish a quality advantage there. Descriptive five-observation, four-cell 90% dev attainment is 20/20 WY versus 9/20 EN; no run stops on this diagnostic, and no crossing is imputed for the remaining EN runs.

Development lineage matters. The earlier 20-pair, accuracy-stopped generator study certified 16/20 WY and 10/20 EN within its update cap; unmatched-budget endpoint test means were 94.10% WY and 95.02% EN (difference -0.923 points, descriptive 95% interval [-2.278, 0.433]). Its answer-only difference was -0.967 [-1.880, -0.054]. Those adverse or inconclusive results are preserved, not relabeled as the current batch. Earlier matched-token capacity diagnostics also performed well for the smaller English model and do not demonstrate WY-specific parameter compression. The present fixed-budget trajectories answer a different, explicitly development-informed comparison; old runs neither augment N nor replace any new pair. Full earlier artifacts remain in the research archive rather than reproducing every development plot here.

The archived batch retains the frozen executed sources and inputs, initialization and checkpoint hashes, update/token/shape accounts, raw generated outputs, failure records, and one independent semantic/accounting replay. The replay verifies both checkpoints, all scheduled observations, paired initialization, stopping, and full per-request output coverage. SHA-256 identifies bytes and detects changes; it is not by itself evidence of valid controls, unbiased selection, semantic correctness, or generalizability. Training logical matrix-operation estimates exclude evaluation, preparation, and hardware overhead; request tokens are not measured latency, energy, or commercial charges.

Table J2. Every paired test outcome (% complete output); no seed is excluded.

Seed	Exposure EN	Exposure WY	Token EN	Token WY
9223101	97.66	97.66	97.27	98.44
9223102	96.00	97.56	95.41	97.56
9223103	92.09	99.32	92.48	98.14
9223104	95.51	97.27	95.21	98.54
9223105	97.85	98.34	97.85	99.12
9223106	97.17	98.14	96.78	97.07
9223107	92.58	96.29	93.16	97.66
9223108	92.97	99.12	93.36	98.73
9223109	94.04	98.44	94.43	99.90
9223110	91.02	96.00	91.70	97.17
9223111	95.61	96.39	95.41	96.29
9223112	97.17	98.54	97.36	99.12
9223113	96.48	99.12	96.39	99.51
9223114	97.36	96.97	97.46	95.61
9223115	93.55	95.51	93.95	95.70
9223116	97.07	98.54	97.17	98.93
9223117	91.89	99.51	92.97	100.00
9223118	97.66	97.85	98.05	98.83
9223119	96.39	98.44	96.19	98.14
9223120	90.53	98.24	90.43	97.27

Table J3. Descriptive test means (%): answer/state diagnostics and complete-output accuracy in each depth/truth cell. These are not extra confirmatory endpoints.

Metric/cell	Exposure EN	Exposure WY	Token EN	Token WY
Answer only	97.13	98.78	97.20	98.90
State trace	95.97	98.02	96.10	98.21
depth0_false	95.70	98.38	95.92	98.61
depth0_true	96.60	99.20	96.68	98.93
depth1_false	93.46	96.11	93.65	96.97
depth1_true	94.36	97.75	94.36	97.83

Table J4. Equal-exposure training accounting at 176 passes: means over all 20 seeds per condition. PFLOPs denotes 10^{15} logical matrix operations, not measured hardware work. Mean paired WY reduction is 65.232%; it is computed per pair, not from the rounded table totals.

Condition	Matrix-operation estimate (PFLOPs)	Padded positions (million)
EN	1.213794	17.946362
WY	0.422012	6.407316

Table J4 is reconstructed from the update-level padded microbatch lengths using $F_{\text{micro}} = 3b[L(24td^2 + 4t^2d) + 2tdV]$. Here $b = 8$, $L = 6$, $d = 384$, and $V = 512$; t is the maximum shifted input-plus-target length in that microbatch. Multiply-add counts as two operations, and the factor 3 approximates forward plus backward work. Eight microbatches form each 64-example update; costs sum over all updates through 176 passes. This matrix-only convention excludes non-matrix operators, optimizer arithmetic, evaluation, and preparation. Unlike the broader classifier account in Appendix A, it is not an operator-complete training total. Linear and quadratic length terms and padding explain why its reduction need not equal the nonpadding-token reduction; neither is a runtime measurement.

Table J5. Post-hoc initial-layout diagnostic of complete-output test accuracy: means (%) over all 20 seeds per condition. All 1,024 items remain the original endpoints; strata do not replace them. Differences are WY minus EN in percentage points.

Checkpoint	Stratum	Items	EN	WY	Difference
Equal exposure	All	1024	95.029	97.861	+2.832
Equal exposure	Four in D	16	0.000	78.438	+78.438
Equal exposure	Other	1008	96.538	98.170	+1.632
Nominal tokens	All	1024	95.151	98.086	+2.935
Nominal tokens	Four in D	16	0.000	73.125	+73.125
Nominal tokens	Other	1008	96.662	98.482	+1.820

The training inventory contains no four-in-one initial state (0/1,024); dev contains 32/512 (boxes A, B, or C), and test contains 16/1,024 (box D). Joining raw generations by question ID confirms that every EN seed produces exactly these 16 unsegmented test outputs at both checkpoints. Full-output correctness on them is zero, but answer-only correctness is 185/320 at equal exposure and 183/320 at nominal tokens. WY full-output correctness is 251/320 and 234/320, respectively. Thus the 43.28%/38.94% arithmetic shares are observed subgroup contributions, not causal fractions. On the other 1,008 items, WY win/tie/loss counts are 13/0/7 and 16/1/3. Descriptive paired-seed bootstrap 97.5% intervals for their differences are [0.407, 2.986] and [0.670, 3.001] points (100,000 resamples, seed 9223000); these post-hoc intervals do not adjust for selecting this subgroup after seeing results. They are not additional confirmatory tests.

The four-item EN input list introduces a list form absent from training, whereas WY extends its repeated conjunction. However, list form, box identity, tokenizer support, and training history are not independently varied. Malformed EN outputs include following the wrong item and merging or omitting comparison clauses; this is not merely a missing reporting label. All EN endpoint dev outputs for the 32 four-in-one records are also fully incorrect; the observed EN dev maximum is 480/512 (93.75%). This is a property of these runs, not a theoretical accuracy ceiling. Aggregate curves (Figures 3, J2, and L1) therefore should not be read as uniform within-layout superiority. No rows were deleted, no test examples were added to training, and no new model was fitted for this diagnostic.

Table J6. Complete initial-occupancy partition: post-hoc descriptive complete-output accuracy (%) across all 20 paired seeds. Occupancy is the maximum number of items initially in any box. W/T/L counts WY higher/equal/lower than EN. No subgroup is a new confirmatory endpoint.

Checkpoint	Occupancy	Train n	Test n	EN	WY	Gap (pp)	W/T/L
Exposure	1	82	128	90.312	98.164	+7.852	18/0/2
Exposure	2	772	672	97.835	98.504	+0.670	10/1/9
Exposure	3	170	208	96.178	97.091	+0.913	8/1/11
Exposure	4	0	16	0.000	78.438	+78.438	20/0/0
Tokens	1	82	128	90.430	98.672	+8.242	19/1/0
Tokens	2	772	672	97.976	99.100	+1.124	13/2/5
Tokens	3	170	208	96.250	96.370	+0.120	8/1/11
Tokens	4	0	16	0.000	73.125	+73.125	20/0/0

Table J6 shows that the two strata with the largest mean gaps are rare or absent in training (maximum occupancy one: 82/1,024; four: 0/1,024). The three-item-maximum stratum has 208 items: mean gaps are +0.913 points at equal exposure and +0.120 at nominal tokens, with 11/20 pairs favoring EN at both checkpoints. Mean signs therefore do not imply majority-pair or uniformly large advantages. The maximum-one stratum contributes larger gaps (+7.852/+8.242 points; 128 items). These slices, alongside depth, truth, queried-item, and queried-box checks, are exploratory descriptions on the same reused inventory, not independent replications. All original aggregate results and sample counts are unchanged.

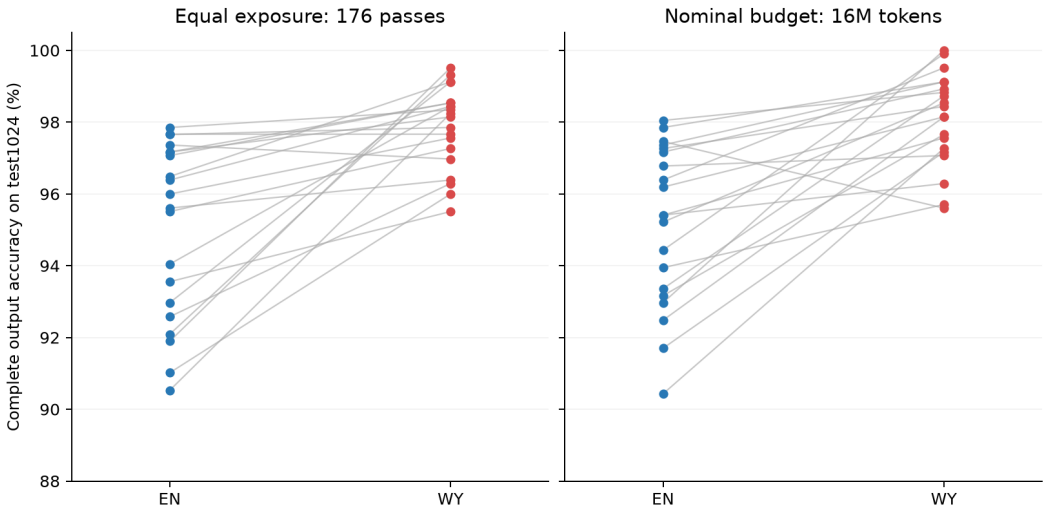

Figure J1. All 20 paired test1024 complete-output scores at both checkpoints. Blue is EN and red WY; gray segments link paired seeds, not training trajectories. The score axis is zoomed. At equal exposure, mean accuracy is WY 97.86% versus EN 95.03%, with 62.12% fewer WY training tokens. At nominal 16M tokens, means are WY 98.09% versus EN 95.15%; WY receives 2.64 times the presentations, not fewer training tokens. Paired quality differences are +2.83/+2.93 percentage points (intervals in Table 5, differences from unrounded scores). Each endpoint includes one reverse pair. A training-unseen four-in-one layout contributes 43.28%/38.94% of these gaps arithmetically; remaining-item gaps are +1.63/+1.82 points (post-hoc Table J5). These are task-specific package comparisons, not isolated language effects.

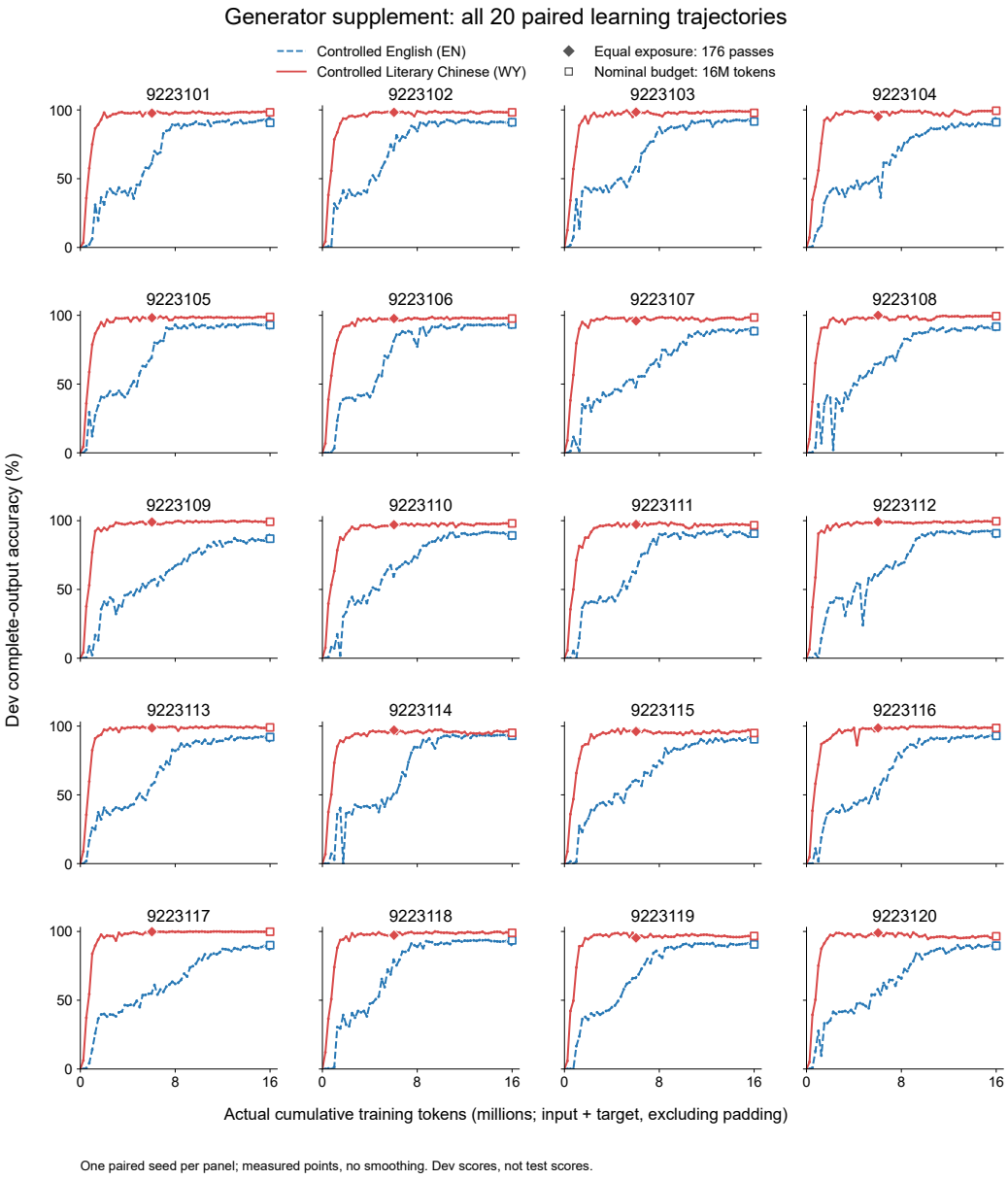

Figure J2. All 20 paired generator trajectories on dev512. Each panel is one initialization seed; blue dashed lines are EN and red solid lines WY. Points are the 65 scheduled observations at actual cumulative training tokens; connecting lines are not smoothed or interpolated evaluations. Diamonds are the extra exact 176-pass observations; open squares are the first full batch reaching the nominal 16M-token budget. Individual curves have no mean-confidence shading. Axes are shared across panels. These are development scores, not the test scores in Figure J1; no new threshold or stopping rule is introduced. The EN dev maximum of 93.75% coincides with failure on the 32 four-in-one dev records; see the post-hoc layout diagnostic in Table J5. Equal-exposure diamonds correspond to 6.037M WY versus 15.936M EN training tokens (62.12% fewer WY tokens); the budget endpoints are both nominally 16M. Separate test1024 endpoint means appear beside Figure J1, not on these dev curves.

K Detailed classifier endpoint and sensitivity diagnostics

K.1 What the common minimum does and does not equalize

At the 10M recognition endpoints, mean scores are 625.175/720 for WY and 630.400/720 for EN. The mean absolute paired score gap is 18.975 answers; the largest gap is 74 answers (10.28 percentage points). The 1M means are 609.50 (WY) and 587.45 (EN), with a mean absolute gap of 26.25 and maximum 74. Near-equal aggregate means at 10M therefore do not establish within-pair score equivalence. The conclusion is lower cost to a shared observed lower-bound criterion, not exactly matched final scores, matched asymptotic capability, or greater intelligence.

After primary recognition, 1M has two later below-threshold observations in two WY runs and four in three EN runs, over 36,649 versus 11,769 follow-up observations. At 10M no later below-75% observations occur in the recorded follow-up, with 35,069 WY and 13,569 EN observations. Follow-up exposure differs, and even absence of an observed dip is not a proof of permanent stability. We neither interpret raw dip counts nor exposure-normalized rates as a causal stability comparison.

At 100M, mean recognition scores are 655.8 (WY) and 656.5 (EN), but the mean absolute paired gap is 29.5 answers and the maximum is 116 (16.11 percentage points). The shared minimum therefore still does not guarantee score equivalence within each pair. WY has one later below-75% observation among 7,966 follow-up observations; EN has none among 2,714. The same unequal-exposure and observation limitations apply. Reporting the higher-minimum results addresses dependence on the chosen minimum, not exact matching of all endpoint scores.

K.2 Pre-specified shorter-window sensitivity

Both conditions jointly attain the 75% minimum in every recovered pair under both shorter windows. The direction remains the same, but magnitude depends on the window definition. These readouts do not justify choosing a different primary span after observing the confirmation.

At 10M, the mean paired restricted differences are -0.6044 S [-0.6453, -0.5636] for the 1,440-presentation span and -0.6681 S [-0.7087, -0.6275] for 3,600, compared with -0.7447 S [-0.7811, -0.7084] for the primary 7,200 span. Intervals are pointwise approximate 95% intervals. Appendix C provides every scale/minimum/span cell, including all shorter-window values and attainment counts.

A separate post-result accounting diagnostic asks how much of the contrast accrues during the accepted 7,200-presentation window. At 10M, mean per-pair FLOPs savings are 50.46% at the retrospectively identified window starts and 50.38% at the charged ends. Thus the approximately halved primary cost is not created solely by appending the confirmation window. Window-start costs are not valid substitutes for our pre-fixed endpoint: the subsequent observations are needed to identify a qualifying window. Appendix H reports all three scales, first observed crossings, and the window's cost contribution without changing the primary analysis.

K.3 Descriptive separation of item presentations and token cost

A post-result decomposition separates presentations from tokens per presentation at primary recognition (Appendix G). At 10M, mean WY/EN presentation counts are 31,244/31,644; mean per-pair savings are 0.9165% (18 pairs fewer, two equal, 20 more). At 1M, means are 48,764/50,364, with -0.0622% mean per-pair savings (11 fewer, nine more). Ratios of aggregate means differ from averaged pair ratios. These counts do not support halving the presentations required, despite approximately halved token and FLOPs costs.

For each pair, total-token ratio equals presentation-count ratio times mean-tokens-per-presentation ratio. The observed cost contrast is thus consistent with much shorter encoded presentations and broadly similar presentation requirements, rather than a uniform reduction in repetitions. This identity is not a causal attribution: segmentation, wording, and the cost-anchored learning-rate schedule still vary as parts of the packages. No new inferential claim or endpoint is introduced by this decomposition.

At 100M, the corresponding mean presentation counts are 25,824 WY versus 33,376 EN; the mean per-pair presentation saving is 21.95%, with 19 pairs requiring fewer WY presentations and one

more. Thus this scale combines shorter encodings with fewer presentations in most pairs, unlike the approximately unchanged presentation counts at 1M and 10M. The descriptive 21.95% presentation saving must not be conflated with the 61.14% token saving or interpreted as a language-intrinsic learning effect.

K.4 Retained inference and challenge methods

At the author's request, post-result deterministic accounting covers one unpadded, batch-one classifier forward pass per inventory item. It is not a pre-specified test, new prediction experiment, or runtime measurement. The bound module's non-loss forward operators exclude loss, backward propagation, and optimizer updates. Per-item cost is $aT^2 + bT + c$, where T is encoded input length including BOS/EOS; independent expansion matches the operator sum at every supported integer length 2–512.

For 1M, 10M, and 100M, the coefficients (a, b, c) are $(2256, 1801186, 550)$, $(9072, 19603338, 1480)$, and $(40896, 199816306, 3342)$, respectively. Table 3 uses ratios of fixed inventory totals, not the mean paired cost-to-quality ratios in the main analysis. Input tokens are 7,934 WY versus 15,920 EN per pass at every scale (50.1633% reduction). These costs depend on architecture and encoded shapes, not on learned weights; final inventory accuracy is reported separately in Table K1. There is no seed-level confidence interval for these deterministic totals and no claim about accuracy on unfamiliar items.

Table K1. Final classifier inventory accuracy (%) at the first complete update crossing 4S. All planned models and all 720 training-inventory items are retained. Ranges are seed minima and maxima, not confidence intervals. Training FLOPs are approximately matched, but WY receives about 1.99–2.01 times the EN presentations. These are not certification-time or held-out scores.

Scale	Models per condition	WY mean	WY range	EN mean	EN range
1M	20	98.389	95.000–99.722	94.375	86.250–97.917
10M	40	99.625	98.472–100.000	98.809	96.250–100.000
100M	20	99.993	99.861–100.000	99.403	96.111–100.000

The classifier outputs two logits, not a textual reasoning trace. This deterministic account does not cover autoregressive generation, runtime, or price; the separate generator study is in Section 4.6. Padding, batching, caching, and other tokenizers can alter deployment costs. `account_classifier_inference.py` checks the frozen identities and uses all encoded rows without selection. A separate post-result prompt exploration with an existing model remains outside the present evidence and is not pooled with either experiment.

After confirmation, all 160 final 4S checkpoints evaluated 512 pre-existing paired records: 320 named challenges and a 192-record synthetic diagnostic. Compute budgets are approximately matched, not repetitions: WY sees roughly twice as many training-item presentations. No retraining, tokenizer refitting, checkpoint search, or prediction-based selection occurs. Independent semantic and label replay is described in Appendix I.

These are averages over every planned final model. The 320 records contain 124 true and 196 false labels: always predicting false scores 61.25%, above both packages' mean accuracy. Balanced accuracy, the mean of the two class recalls, has a constant-prediction baseline of 50% and is also near that baseline. Removing the eight training-structure equivalents does not materially change the pattern. Results are heterogeneous across source groups, with reverse differences retained in Appendix I rather than selecting only favorable groups.

A subsequent source-only diagnostic finds that 304/320 WY and 296/320 EN records contain valid tokenizer IDs absent from classifier training: their input embeddings received no task-data gradient. This support mismatch is a plausible contributor, not a proven sole cause or grounds for removing records. No tokenizer, model, or item set is changed. The challenges expose a limitation, not failure on every possible unseen in-distribution problem.

Under swap-endpoint normalization, the 320 challenges have no training-query or complete-world overlap; a pre-fixed sensitivity removes eight renamed structural equivalents. The synthetic subset

has five training-query overlaps. Neither source is a random deployment sample or guaranteed free of earlier inspection; Appendix I details provenance and tokenizer exposure.

Evaluation uses the original model/tokenizers, FP32, and unpadded batch one. Raw logits, all strata, accuracy, balanced accuracy, and costs are retained. Intervals bootstrap paired seeds conditional on fixed rows; they are descriptive, not new confirmatory tests.

All 81,920 classifier predictions (160 models $\times$ 512 records) are present. A 16-item training-logit replay matches every checkpoint. Both packages transfer poorly to the 320 named challenges (Table 4).

Earlier null and reverse results remain intact. This confirmation uses independent seeds and a newly pre-fixed measurement policy; it does not revise the earlier scientific question. Intervals excluding zero do not expand its fixed-inventory scope.

K.5 Retained encoding and statistical detail

Both tokenizers use byte-level byte-pair encoding (BPE) with 492 entries and no normalization. Regex pre-tokenization is disabled, so learned merges may span conventional word boundaries. The very short sequences in Appendix E therefore reflect this particular tokenizer as well as the written representation, not token counts under a general-purpose multilingual tokenizer. Appendix E prints byte-checked true/false examples, their exact encoded lengths, and the serialized tokenizer settings. In the illustrated pair of questions, the queried item moves from A to B, while subsequent swaps involve empty boxes; both systems explicitly retain those transitions. This illustrates the task semantics, not proof that the classifiers learned state tracking rather than inventory-specific regularities.

Tokenizer construction is inherited, not a fresh corpus-independent intervention. The source diagnostic retains 716 distinct queries among 720 rows after normalizing only the order of endpoints within simultaneous swaps. Under this same normalization, 96 query identities, covering 100 inventory rows, also occur in the 16,384-row tokenizer-construction corpus. This diagnostic concerns the byte-identical inherited inputs, not old experimental outcomes. Tokenizer-only exposure is not supervised classifier training, but record-ID separation must not be described as semantic independence. The normalization also does not exhaust all structural equivalences. We preserve these rows and report the limitation rather than removing them after seeing results.

Training FLOPs are the declared operator-equivalent accounting at actual sequence lengths, including forward/backward training work and optimizer-related arithmetic. They exclude evaluation, serialization, rendering, tokenizer construction, and input preparation. They are not a hardware instruction count, elapsed time, energy measurement, or total financial expense. Tokens likewise denote actual encoded training tokens, not characters or UTF-8 bytes. Independent analysis recomputes costs from update traces and the frozen operator implementation rather than accepting runtime summary totals alone.

For pair i, $D_i = (R_{\mathrm{WY},i} - R_{\mathrm{EN},i})/S$. The primary target is the expected paired restricted-cost difference over the prescribed initialization-seed procedure at 10M, conditional on this fixed inventory and training protocol; its estimate is the mean of the 40 observed D_i values. The pre-fixed approximate paired-t 95% interval is primary. Complementary sensitivity checks use a paired percentile bootstrap with 20,000 resamples and seed 20260913, median and sign information, and leave-one-pair-out analysis. The two interval procedures need not give identical endpoints. These intervals address seed variability conditional on the inventory and protocol; they do not quantify uncertainty over all possible tasks or continuous threshold-crossing times. Scale-specific intervals are not pooled or simultaneous family-wise intervals.

The frozen plan, executable entry points, per-scale sources, inputs, seeds, launch records, update logs, observation logits, and terminal records are retained. Separate reader code recomputes the reported numbers. The current local result source is `analysis.json`, SHA-256 `dafd7e8a3a33df1a2e4d41b6009c8f8982e52981fc7c24291924c5a7c5046f2c`; the plan is `8e9320634cd767cde4209c18ac888ae64dcd35df00aeb872f87a9d11e32791a5`. The release is not yet public.

The archived source-overlap check `Paper1_BOX_DENSE_PAPER_20260912/probe_feasibility.json` is diagnostic only, not an additional confirmation sample.

L Visual guide to the two studies

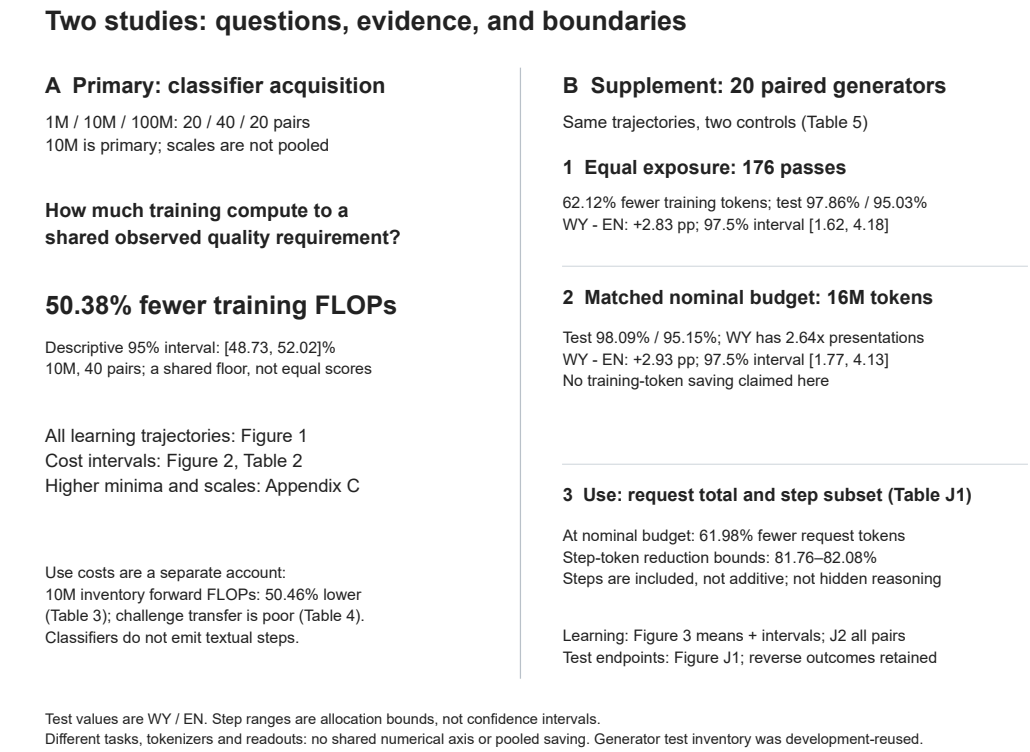

Figure L1. Reading map of the existing evidence, not a pooled analysis. Panel A summarizes the primary 10M classifier result and separates its deterministic forward account and poor challenge transfer. Panel B locates both controls and the request/step account in the 20-pair generator supplement. Test percentages are WY / EN. The primary percentage interval is descriptive; Table 2 retains the primary paired cost-difference inference. The two generator quality intervals have 97.5% nominal coverage each; step ranges are allocation bounds, not confidence intervals. Request reductions include erroneous outputs, and the step subset must not be added to the total. See Tables 2–5 and J1 for definitions and full precision. The studies differ in architecture, task, tokenizer and readout; their percentages are not interchangeable or additive. Generator quality gaps are layout-concentrated (Table J5), not uniform across configurations.