TAIL-CALIBRATED LOSS-DISTRIBUTION SHAPING FOR REASONING FINE-TUNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Curricula based on sample ordering or selection regulate which examples are presented during training, but do not directly specify the relative weighting of token losses within each response. We propose Tail-Calibrated Loss-Distribution Shaping (TCLD), which adapts response-token loss aggregation using feedback from the current training-loss distribution. TCLD replaces arithmetic averaging with a power-mean objective whose parameter p controls relative token gradient coefficients and recovers standard cross-entropy (CE) at $p = 1$. A Hill-tail controller calibrates $p \in (0, 1)$ at observation-window boundaries using no-gradient loss observations, reducing the relative coefficients of higher-loss tokens without a downstream-validation-accuracy search over p. On a heterogeneous five-source reasoning pool, TCLD improves Macro- and Micro-averaged maj@3 accuracy over random-shuffle training (RS) by 2.74 and 2.02 percentage points, respectively, across three matched seeds with Qwen3-1.7B-Base, and outperforms the evaluated sample-level curriculum baselines in aggregate accuracy. At the 25% optimizer-step checkpoint, TCLD exceeds the full-budget RS baseline in mean Macro accuracy. Source-wise diagnostics show heterogeneous sensitivity to fixed powers and the competitive performance of automatic calibration. Gains over RS also extend to Qwen3-0.6B and SmolLM2-1.7B. Single-seed experiments on six ordered two-source streams show order-dependent gains and per-source trade-offs relative to RS. TCLD requires no changes to the model architecture or inference procedure.

1 INTRODUCTION

Standard supervised fine-tuning (SFT) repeatedly samples examples and minimizes the arithmetic mean of their response-token cross-entropies. With sufficient optimization, this uniform procedure is intended to move the model toward the empirical target distribution. Curriculum learning (CL) adds progression: easy-to-hard schedules can first expose coarse regularities and then introduce harder cases, while self-paced learning, prioritized sampling, and data selection instead change inclusion or training opportunities according to their own criteria (Bengio et al., 2009; Kumar et al., 2010; Katharopoulos & Fleuret, 2018; Mindermann et al., 2022). These methods are not all strict easy-to-hard orderings, but they primarily intervene at the sample level, where a *sample* denotes a complete prompt–response training pair and "sample-level" identifies the unit of curriculum intervention; static curricula fix the rule in advance, whereas dynamic curricula may update it from the current model state.

Reasoning responses contain heterogeneity below the example level. Within one response, routine formatting, an uncertain derivation, a decisive answer token, and a possibly unreliable annotation can occupy very different regions of the teacher-forced loss distribution. Scheduling complete samples does not directly specify how token losses within each response are weighted relative to one another. This motivates a complementary question: rather than changing which example is presented, can training adapt the *loss geometry* through which the supervised tokens of an already presented response influence the update? A useful mechanism must describe relative loss structure without declaring every high-loss token to be either noise or valuable difficulty.

TCLD answers this question with a controller–actuator design. Its actuator replaces arithmetic response-token averaging with a power mean whose parameter p continuously controls relative token gradient coefficients: $p = 1$ recovers ordinary response-token cross-entropy, $p < 1$ reduces

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.

the relative coefficient of a higher-loss token, and $p > 1$ increases it. Its controller observes the current response-token loss distribution in a no-gradient window, summarizes upper-tail shape, and maps the tracked statistic to the power used for that window's optimization. In the primary mixed-pool setting, TCLD retains randomized sample order and adapts the aggregation of response-token losses using distributional feedback. The complete sequence is: partition the training stream into observation windows, measure current token losses, calibrate p, hold it fixed while training the window's mini-batches, and repeat for the next window. Figure 1 summarizes this controller–actuator cycle.

The specific strategy examined in this paper uses upper-tail severity to control compression. When extreme losses are more prominent relative to the rest of the observed distribution, the controller moves p farther below one and reduces their relative coefficients; when the fitted tail is lighter, it moves the objective toward ordinary CE. This is a testable design choice rather than a consequence that heavy tails make compression universally optimal. The controller studied here outputs only $p \in (0, 1)$, although the power-mean objective itself is defined for all $p > 0$. Because loss severity does not reveal whether an upper-tail token is noise or useful difficulty, the method shapes influence without assigning either semantic interpretation. TCLD is not parameter-free: it specifically avoids selecting p by a downstream-validation-accuracy sweep. We evaluate this distribution-feedback curriculum through a matched heterogeneous-pool comparison, source-wise fixed-power diagnostics, transfer across model configurations, observation-window sensitivity, and ordered two-source streams in which the loss distribution changes with the data source.

Our contributions are threefold: (i) we formulate power-mean response-token aggregation as an interpretable loss-shaping axis complementary to example ordering; (ii) we introduce a Hill-tail controller that calibrates this axis from window-level loss statistics without a downstream-accuracy-based power search or hand-set power clipping; and (iii) matched mixed-pool experiments demonstrate gains over RS and the evaluated sample-level curricula, with further evidence of competitive automatic calibration and transfer across the tested model scales and families, and ordered-source experiments show order-dependent benefits over SFT with arithmetic cross-entropy ($p = 1$) under the same source order.

2 Related Work

Sample-level curricula and data selection. Curriculum, self-paced, teacher–student, and difficulty-based schedules modify example order, inclusion, or pace (Bengio et al., 2009; Kumar et al., 2010; Matiisen et al., 2019; Zhou et al., 2020). Perplexity has also served as a model-based curriculum signal: CLIMB studies unigram- and self-perplexity-based data curricula for language-model pre-training, while CAMPUS uses current-model perplexity to schedule candidate sub-curricula during instruction tuning (Diehl Martinez et al., 2023; Li et al., 2025). Our experiments include a static curriculum that orders training samples by their initial response-token perplexity. Prioritized sampling increases exposure to informative examples (Katharopoulos & Fleuret, 2018); SuperLoss dynamically downweights high-loss examples through sample-level confidence (Castells et al., 2020); and RHO-LOSS uses learner and holdout-model losses to select by estimated reducible loss rather than raw difficulty (Mindermann et al., 2022). Data pruning asks which examples are needed at all (Sorscher et al., 2022). These methods primarily control a sample's training opportunity or sample-level contribution. TCLD instead changes how response-token losses are aggregated after a sample enters a training batch.

Token-level curricula and loss reweighting. Prior work progressively exposes target subsequences, adapts objectives using token frequency, or reweights difficult generation targets (Liang et al., 2021; Gu et al., 2020; Jiang et al., 2020). Such interventions can be static or state dependent and use position, frequency, confidence, or token-specific difficulty to decide exposure or weight. TCLD instead summarizes the current response-token loss distribution into one window-level state and uses it to reshape aggregation coefficients without changing the input sequence or response mask. This combination, rather than token-level intervention by itself, is the contribution studied here.

Generalized means and robust objectives. Generalized means provide nonlinear aggregators with the arithmetic mean at $p = 1$ (Hardy et al., 1952; Bullen, 2003); ordered, top-k, and robust objectives reshape losses in other ways (Fan et al., 2017; Barron, 2019). Focal loss downweights well-classified

examples to focus on harder classification cases (Lin et al., 2017), whereas SuperLoss suppresses high-loss samples. TCLD instead induces token-level coefficient ratios $(\ell_i/\ell_j)^{p-1}$ and combines this aggregator with current-model tail observation and window-level calibration. We do not claim an experimental advantage over objectives not evaluated here.

Tail estimation and adaptive calibration. The Hill estimator is a classical scale-invariant statistic for regularly varying upper tails (Hill, 1975; Resnick, 2007). Exponential averaging and stochastic approximation motivate tracked states under noisy observations and drift (Polyak & Juditsky, 1992; Borkar, 2008). We use upper-tail estimation as a calibration signal rather than as a claim that finite token-loss samples follow an exact Pareto law. The chosen tail fractions, robust aggregation, state update, and mapping are specified in Section 3.2.

3 Tail-Calibrated Loss-Distribution

Sample-level curricula typically use estimated difficulty to adjust the order, inclusion, or training opportunities of complete samples, and dynamic variants may revise these decisions as the model changes. TCLD uses model feedback to adapt response-token loss aggregation. In the primary mixed-pool setting, it retains randomized sample order and changes how the losses within each training batch contribute to optimization.

Figure 1 summarizes the organization of one training cycle. The shuffled training pool is organized into mega-batches, each of which also serves as an observation window. We represent the k-th mega-batch as the ordered collection $\mathcal{B}_k = (\mathcal{C}_{k,1}, \ldots, \mathcal{C}_{k,J_k})$, where $\mathcal{C}_{k,b} \in \mathcal{B}_k$ is its b-th ordinary training batch. The controller observes the examples and valid response-token losses across all members of $\mathcal{B}_k$, while each member computes its own training objective for backpropagation and parameter updates. In the primary experiment, 53,475 randomly permuted examples form four approximately equal mega-batches of about 13.4k examples, with remainder examples retained. Randomization reduces systematic composition bias but does not make each finite mega-batch identical to the full pool. Appendix N specifies the adopted configuration, and Appendix L examines sensitivity to the number of observation windows. The ordered-stream experiments preserve source phases and shuffle only within each phase (Section 4.4).

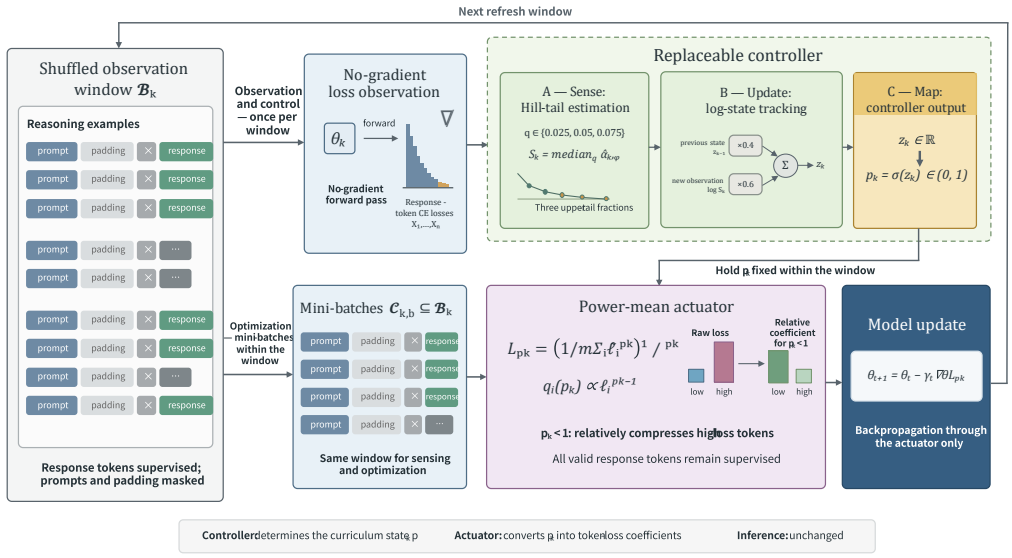

Figure 1: TCLD workflow for one mega-batch/observation window. A no-gradient pass supplies response-token losses to the replaceable Sense–Update–Map controller. Its detached output p_k is fixed while the same examples are divided into ordinary training batches, whose losses are recomputed for power-mean optimization. The updated model observes the next mega-batch and refreshes the controller.

Within an ordinary training batch, valid response-token losses can differ substantially. TCLD replaces their arithmetic mean with a power-mean objective, shown as the actuator in Figure 1, while prompt

and padding positions remain masked. Its parameter p controls the relative aggregation coefficients. At $p = 1$, the objective recovers ordinary response-token CE. Values $p < 1$ reduce the coefficient of a higher-loss token relative to a lower-loss token, whereas $p > 1$ increases it. Section 3.1 defines this objective and derives the corresponding gradient coefficients. The remaining challenge is to choose p from the current training state without a downstream-validation-accuracy sweep.

The upper path in Figure 1 addresses this challenge through a Sense–Update–Map controller. Before optimizing mega-batch $\mathcal{B}_k$, the model θ_{t_k} performs a no-gradient pass over that mega-batch to observe its valid response-token losses. Sense summarizes their upper-tail structure using Hill estimates, Update smooths the resulting statistic in log space, and Map converts the tracked state into a detached power $p_k \in (0, 1)$ (Section 3.2), where the mapping associates a smaller tracked tail index with a smaller p_k. A smaller p_k more strongly reduces the relative coefficients of higher-loss tokens. Observation losses are used only for calibration. During optimization, each ordinary batch recomputes its losses and updates the model while p_k remains fixed. Because model updates change the loss distribution, TCLD observes the next mega-batch with the updated model and refreshes the controller state. Section 3.3 discusses this moving objective. Section 4.3 compares automatic calibration with fixed-power alternatives, and Section 4.4 evaluates the strategy under source changes.

3.1 Power-Mean Objective and Token Coefficients

Ordinary response-token CE assigns equal aggregation coefficients to all valid positions before their individual gradients are considered. To expose a single controllable loss-geometry axis, let $\mathcal{A}(\mathcal{C})$ be the valid response-token positions in training mini-batch $\mathcal{C}$, let $m(\mathcal{C}) = |\mathcal{A}(\mathcal{C})|$, and let $\ell_i(\theta) > 0$ be token i's teacher-forced cross-entropy under model parameters θ. TCLD replaces their arithmetic mean with

$$L_p(\mathcal{C}; \theta) = \left[\frac{1}{m(\mathcal{C})} \sum_{i \in \mathcal{A}(\mathcal{C})} \ell_i(\theta)^p \right]^{1/p}, \qquad p > 0. \tag{1}$$

At $p = 1$, Eq. (1) recovers standard response-token CE. Values $p < 1$ and $p > 1$ respectively compress and amplify higher losses relative to lower ones. The numerical floor used for stable fractional powers is an implementation detail reported in Appendix A.

For an ordinary-batch member $\mathcal{C}_{k,b} \in \mathcal{B}_k$, TCLD uses the instance $L_{p_k}(\mathcal{C}_{k,b}; \theta)$ of Eq. (1). Hence $m_{k,b} = m(\mathcal{C}_{k,b})$ is the number of valid response-token positions in that batch, and both the sum and normalization are batch-local rather than mega-batch-wide. If optimization step t processes $\mathcal{C}_t = \mathcal{C}_{k(t),b(t)}$, then $p_t = p_{k(t)}$ and its objective is the same function $L_{p_t}(\mathcal{C}_t; \theta_t)$.

Proposition 3.1 (Relative coefficient geometry)**.** *For positive active-token losses and $p > 0$, differentiation of Eq. (1) gives*

$$\nabla_\theta L_p(\mathcal{C}; \theta) = \frac{L_p(\mathcal{C}; \theta)^{1-p}}{m(\mathcal{C})} \sum_{i \in \mathcal{A}(\mathcal{C})} \ell_i^{p-1} \nabla_\theta \ell_i. \tag{2}$$

The induced normalized coefficient measure is

$$q_i(p; \mathcal{C}) = \frac{\ell_i^{p-1}}{\sum_{r \in \mathcal{A}(\mathcal{C})} \ell_r^{p-1}}, \qquad \frac{q_i(p; \mathcal{C})}{q_j(p; \mathcal{C})} = \left(\frac{\ell_i}{\ell_j} \right)^{p-1}. \tag{3}$$

Eq. (3) gives the actuator a direct interpretation. At $p = 1$, active token losses have equal relative coefficients; decreasing p makes that coefficient curve fall more steeply with loss, while increasing p rotates it toward higher-loss tokens. This statement concerns relative coefficients, not semantic value or a vanishing complete parameter gradient. The implemented objective also retains the common factor in Eq. (2), which is determined by the current ordinary batch. The experiments therefore evaluate the complete power-mean actuator rather than an isolated normalized reweighting rule; this common scaling is not, in general, identical to changing AdamW's learning rate. Appendix B derives Eqs. (2)–(3) and gives the exact decomposition into relative token allocation and a batch-dependent common gradient scale. The remaining question is how to choose the actuator state without a downstream-accuracy sweep.

3.2 AUTOMATIC CALIBRATION OF p FROM TOKEN-LOSS TAIL STATISTICS

The controller uses upper-tail shape as a scale-invariant measure of the relative spread of extreme losses. We use $\alpha \in (0, \infty)$ to denote the population upper-tail index, with smaller values corresponding to a heavier upper tail and larger values to a lighter one. At window k, a no-gradient pass with parameters θ_{t_k} collects finite positive losses $X_{k,1}, \ldots, X_{k,n_k}$ from valid response tokens across the ordinary-batch members of $\mathcal{B}_k$. Let $X_{k,(1)} \le \cdots \le X_{k,(n_k)}$ denote their order statistics. For each tail fraction $q \in \mathcal{Q} = \{0.025, 0.05, 0.075\}$, the implementation sets $r_{k,q} = \max\{50, \lfloor q n_k \rfloor\}$ and, when the component is valid, computes the Hill estimate of α for the current window and the threshold determined by q:

$$\widehat{\alpha}_{k,q} = \left[\frac{1}{r_{k,q}} \sum_{s=1}^{r_{k,q}} \log \frac{X_{k,(n_k-s+1)}}{X_{k,(n_k-r_{k,q})}} \right]^{-1}. \tag{4}$$

Smaller fitted indices indicate a heavier upper tail. Classical Hill consistency assumes an intermediate number of upper-order statistics tending to infinity while its fraction tends to zero (Hill, 1975; Resnick, 2007); our finite fractions instead use Hill estimates as scale-invariant control signals. Three nearby fractions expose threshold sensitivity, and their median reduces sensitivity to an atypical component when multiple usable estimates agree:

$$S_k = \mathsf{Sense}(\mathcal{B}_k; \theta_{t_k}) = \operatorname{median}\{\widehat{\alpha}_{k,.025}, \widehat{\alpha}_{k,.05}, \widehat{\alpha}_{k,.075}\}, \tag{5}$$

with invalid components omitted. With one usable component, that estimate alone determines S_k; with fewer than three usable components, the usual three-component median protection does not apply. For each realized finite observation window, finite positive losses give a finite window maximum and every valid Hill threshold is strictly positive. Let $\mathcal{Q}_k^{\mathrm{valid}} \subseteq \mathcal{Q}$ denote the tail fractions for which $\widehat{\alpha}_{k,q}$ is well defined and passes the validity checks described above. Consequently, provided that $\mathcal{Q}_k^{\mathrm{valid}}$ is nonempty,

$$S_k \ge b_k := \min_{q \in \mathcal{Q}_k^{\mathrm{valid}}} \left[\log\left(\frac{\max_{1 \le i \le n_k} X_{k,i}}{X_{k,(n_k-r_{k,q})}} \right) \right]^{-1} > 0.$$

Here, b_k is a positive lower bound determined by the maximum observed loss and the thresholds of the valid Hill components in the current window. Appendix C provides its derivation and discusses the finite-window median property and dependence caveat; numerical validity and fallback rules are collected in Appendix A.

For a valid S_k, the controller tracks the log statistic and maps the resulting state to a power:

$$z_k = \beta z_{k-1} + (1-\beta) \log S_k, \qquad p_k = \sigma(z_k) = \frac{e^{z_k}}{1 + e^{z_k}}, \tag{6}$$

where $\beta = 0.4$ is the weight assigned to the previous log-state z_{k-1}, and S_k is the median of the valid Hill estimates in the current window. Tracking $\log S_k$ treats proportional changes in the estimated tail index symmetrically. Define the tracked effective tail index as $\widetilde{\alpha}_k = e^{z_k}$; then Eq. (6) is equivalently $p_k = \widetilde{\alpha}_k / (1 + \widetilde{\alpha}_k)$. The first valid statistic directly initializes $z_k = \log S_k$, whereas subsequent values of $\widetilde{\alpha}_k$ incorporate the controller history. The sigmoid turns this tracked state into the curriculum knob without a hand-set clip: for example, $\widetilde{\alpha}_k = 2$ gives $p_k = 2/3$, while $\widetilde{\alpha}_k = 10$ gives $p_k = 10/11$.

The emitted p_k is detached from autograd and inserted into the already-defined objective $L_{p_k}(\mathcal{C}_{k,b}; \theta)$. One observation defines one mega-batch-level curriculum state, so p_k is held fixed for all ordinary-batch members of $\mathcal{B}_k$. At the next mega-batch, the updated model supplies a new loss distribution and the controller refreshes the state. A heavier fitted tail (smaller S_k) maps to stronger relative compression, whereas a lighter tail moves the objective toward ordinary averaging. The map is monotone, bounded in $(0, 1)$, and has no hand-set power clip. Conditional tracking and mapping bounds are deferred to Appendix D; the mapping is a studied policy, not an accuracy-optimality theorem.

3.3 SCOPE OF DYNAMIC CALIBRATION

Let $\mathcal{C}_t = \mathcal{C}_{k(t),b(t)}$ denote the mini-batch at optimization step t. Window-wise calibration produces the moving objective $L_{p_{k(t)}}(\mathcal{C}_t; \theta_t)$. This is generally not equivalent to training with the arithmetic mean of the realized powers, for two reasons: the objective is nonlinear in p, and each update changes

later model states. Appendix E formalizes this path dependence and gives a conditional objective-drift bound. Section 4.4 then tests whether such updating is useful relative to the fixed $p = 1$ state when the data source changes.

The controller's scope is distributional rather than semantic. Two windows can have the same loss multiset, yet assign the upper-tail values to corrupted labels in one case and to rare decisive reasoning steps in the other. A loss-only controller cannot distinguish them. A formal counterexample appears in Appendix F. Thus TCLD tests a severity-driven allocation policy without claiming universal utility optimality.

TCLD incurs no inference cost; its main training overhead is one no-gradient observation pass per mega-batch. Appendix N states the cost assumptions and a matched timing example, while Appendix A gives pseudocode and safeguards. Learning curves are reported per optimizer step, not as equal-FLOP or wall-clock acceleration.

4 EXPERIMENTS

We evaluate TCLD on a heterogeneous mixed training pool, then use fixed-power and observation-window diagnostics to examine its calibration behavior. Ordered two-source streams provide a complementary setting in which the training distribution changes at a known boundary, and matched runs with two additional model configurations test whether the observed gains are confined to the primary model. These studies use distinct protocols where stated and are not pooled into a single statistical comparison.

4.1 EXPERIMENTAL SETUP

Datasets. The mixed training pool contains GSM8K, MathQA, MATH, AQuA-RAT, and SVAMP (Cobbe et al., 2021; Amini et al., 2019; Hendrycks et al., 2021; Ling et al., 2017; Patel et al., 2021). It deliberately combines natural-language calculations, multiple-choice rationales, competition-level derivations, and short arithmetic chains. Supervision is response-only rather than answer-only: prompt labels are masked, while rationale, derivation, final-answer, and terminal response positions remain supervised.

The preprocessing audit uses a maximum sequence length of 1,536 and removes an item rather than truncating a trailing answer. The retained pool contains 53,475 examples: 7,473 from GSM8K, 29,837 from MathQA, 7,465 from MATH, 8,000 from AQuA-RAT, and 700 from SVAMP. Its response and answer formats, evaluation splits, and response-length statistics are detailed in Appendix M.

Single-source experiments use their native training and evaluation partitions under the same response-mask contract. We additionally construct a seven-source diagnostic suite from the five mixed-pool sources, QASC, and EntailmentBank. The two-stage study uses ordered pairs drawn from EntailmentBank, QASC, GSM8K, and AQuA-RAT; examples are shuffled within a source phase but not across phases. Cross-source averages complement rather than replace per-source results because answer formats, data volumes, and response lengths differ.

Compared methods. The primary mixed-pool comparison covers four method families and five evaluated configurations: We call the shuffled supervised baseline *Random-shuffle supervised fine-tuning (RS)*.

1. **RS**: random-shuffle supervised fine-tuning with arithmetic response-token cross-entropy;
2. **Static CL**: examples are globally ordered once by a frozen difficulty score, with stable sample identifiers breaking ties; we report length-based and initial-PPL-based scores, and cycle the resulting order without refreshing difficulty;
3. **Dynamic CL**: model-dependent difficulty scores from the current response-token losses order examples within each observation window and are refreshed as the model changes, while optimization uses arithmetic response-token cross-entropy; and
4. **TCLD**: the proposed Hill-tail controller coupled to the power-mean actuator.

Fixed-power controls are analyzed separately in Section 4.3, where they diagnose calibration rather than serve as primary curriculum baselines.

Table 1: Endpoint true maj@3 accuracy (%) on the heterogeneous reasoning pool. Values are mean ± sample standard deviation over three matched seeds. Macro averages sources equally; Micro weights evaluation examples. Bold and underline mark the two largest displayed means; they do not imply statistical significance.

Method	GSM8K	MathQA	MATH-500	AQuA-RAT	SVAMP	Macro	Micro
RS	65.00±0.48	63.77±1.12	30.27±1.63	51.18±1.97	75.22±1.39	57.09±0.20	60.99±0.74
Static CL (length)	62.17±1.64	58.92±1.04	28.27±0.23	50.79±3.36	68.44±2.59	53.72±1.01	57.01±0.67
Static CL (initial PPL)	53.70±0.84	58.03±0.07	24.07±0.42	46.98±0.82	71.89±1.02	50.93±0.27	54.05±0.17
Dynamic CL	65.33±0.57	63.98±0.53	29.87±2.27	50.39±1.36	**76.33±0.58**	57.18±0.94	61.17±0.61
TCLD	**67.53±0.49**	**65.05±0.49**	**34.73±1.40**	**55.51±3.88**	**76.33±1.86**	**59.83±0.69**	**63.01±0.26**

Training and evaluation. All methods in the primary comparison share model initialization, mixed-pool data, response mask, optimizer schedule, maximum training budget, seeds, and generation budget. We report true majority-vote accuracy from three generations (maj@3). Macro is the unweighted mean over the five evaluation sources, while Micro weights each source by its number of evaluation items. Means and sample standard deviations use three matched seeds. Table 1 reports the common pre-specified endpoint evaluation for every method; no intermediate test result selects the checkpoint. The single-source calibration study instead uses a lower learning rate and an inner-validation CE rule before one endpoint test, and the ordered-stream study uses source-contiguous phases. Full settings and paired results are listed in the appendix.

4.2 Main Mixed-Pool Results and Optimization-Step Progress

TCLD improves over RS by 2.74 Macro and 2.02 Micro points and exceeds both static curricula and Dynamic CL in aggregate accuracy. Its mean accuracy is also higher than RS on each of the five evaluation sources, although the source-level margins vary. These results evaluate the implemented power-mean objective together with its controller; the relative-allocation and common-scale terms are not separated experimentally. Section 4.3 examines sensitivity to the power and the competitiveness of automatic calibration.

Figure 2 tracks test accuracy for RS, Dynamic CL, and TCLD at eight pre-specified checkpoints spanning 12.5%–100% of the shared optimizer-step budget. Static CL appears only in the endpoint comparison because intermediate evaluations were not collected. The curves compare progress per optimizer step; TCLD's additional observation passes incur computation beyond this shared budget. Exact values and normalized AULC are reported in Appendix K.

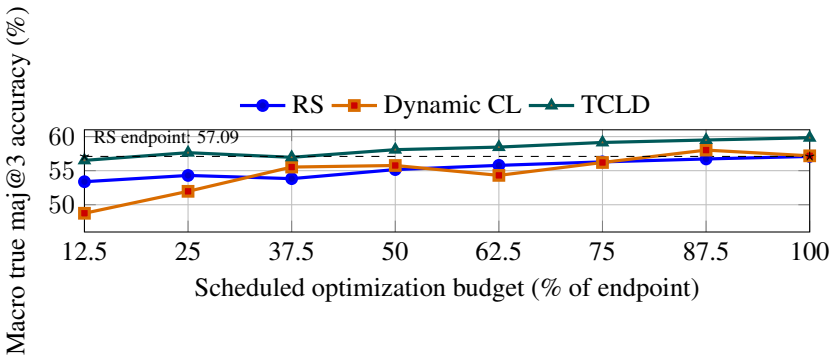

Figure 2: Three-seed mean Macro true maj@3 at common R1–R8 checkpoints. The dashed line marks the RS endpoint; observation passes are excluded from the horizontal axis.

TCLD first exceeds the RS endpoint Macro value at the 25% scheduled checkpoint, whereas Dynamic CL does so at 87.5%. This indicates earlier progress per optimizer update, not wall-clock or equal-FLOP acceleration.

4.3 Fixed-Power and Calibration Diagnostics

To characterize aggregation sensitivity, the single-source diagnostic covers the five constituents of the mixed pool plus QASC and EntailmentBank. Each source uses a lower learning rate of 4×10^{-5}



Figure 3: Source-wise sensitivity to token-loss aggregation. Curves show fixed-power changes relative to RS; gold rings mark the best fixed power and red diamonds report TCLD. The final panel is the equal-weight mean over seven sources. Results use seed 42 and characterize task dependence rather than statistical significance.

and compares RS, six fixed powers $p \in \{0.3, 0.5, 0.7, 0.9, 1.1, 1.3\}$, and TCLD. An inner validation split selects one checkpoint by standard response-token CE, after which that checkpoint receives a single true maj@3 evaluation. These seed-42 runs describe source dependence rather than statistical significance or an accuracy-optimality guarantee for the controller.

The best grid point ranges from $p = 0.3$ on QASC to $p \in \{1.1, 1.3\}$ on SVAMP. TCLD reaches 62.73% on the equal-weight mean, versus 59.72% for RS, 62.16% for the strongest shared fixed power, and 64.00% for a post-hoc per-source oracle. It ranks within the top two on five of seven sources. This supports competitive calibration without per-source downstream-accuracy search, not oracle power prediction; Appendix H gives the absolute values.

We then compare TCLD with matched fixed powers under the primary mixed-pool protocol.

Table 2: Matched mixed-pool calibration diagnostic (endpoint true maj@3, %). Values are mean ± sample standard deviation over three seeds.

Objective	Macro	Micro
Fixed $p = 0.3$	58.26±0.69	61.81±0.42
Fixed $p = 0.7$	59.04±0.62	62.67±0.67
Fixed $p = 1.3$	52.98±0.97	56.82±0.78
TCLD	**59.83±0.69**	**63.01±0.26**

The fixed-power results show sensitivity to the aggregation parameter, with different sources favoring different values of p. In the matched mixed-pool comparison, TCLD exceeds the fixed $p = 0.7$ reference by 0.79 Macro and 0.34 Micro percentage points. However, the three-seed paired confidence intervals include zero (Appendix G), so the evidence does not establish a statistically reliable advantage over this reference. Together, the source-wise and mixed-pool results support automatic calibration as a competitive alternative to selecting p through downstream-accuracy sweeps.

The descriptive observation-window analysis in Appendix L shows similar aggregate performance with 2, 8, and 16 windows, with no setting consistently best across sources. Refresh frequency therefore does not improve endpoint accuracy monotonically, reflecting a source-dependent trade-off between state freshness and finite-window estimation noise.

4.4 Two-Stage Non-Stationary Streams

The ordered-stream experiment evaluates the feedback loop under an explicit source change. Each stream has two ordered, internally shuffled phases. SFT keeps $p_t \equiv 1$, whereas Online TCLD updates $p_t = p_{k(t)}$ from window losses. Within each direction, both methods share the Qwen3-1.7B-Base

Table 3: Final accuracy on ordered two-source streams (%). E=EntailmentBank, Q=QASC, G=GSM8K, and A=AQuA-RAT. Macro averages sources; Micro weights test items; deltas are Online TCLD minus SFT (percentage points).

Stream	SFT Macro	TCLD Macro	Δ	SFT Micro	TCLD Micro	Δ
E→G	73.36	**75.38**	+2.01	66.20	**68.92**	+2.72
G→E	65.62	**65.72**	+0.10	62.68	**65.67**	+2.99
Q→G	66.86	**69.50**	+2.64	65.97	**69.00**	+3.03
G→Q	68.15	**69.41**	+1.25	68.06	**69.31**	+1.25
A→Q	60.35	**61.03**	+0.69	66.02	**67.54**	+1.53
Q→A	56.87	**63.43**	+6.56	63.47	**68.39**	+4.92

initialization, seed 42, source order, budget, and pre-specified endpoint; final true maj@3 evaluates both sources. The comparison tests online distribution-conditioned powers against ordinary CE under matched conditions, rather than serving as a general continual-learning benchmark. Protocol details are in Appendix I.

Online TCLD has higher final Macro and Micro point estimates in all six orders, although the 0.10-point Macro difference for G→E is effectively a tie. Both source accuracies increase in E→G, Q→G, G→Q, and Q→A; G→E and A→Q trade performance between sources, and Q→A has the largest aggregate gain. The controller state changes with order: across E→G it decreases from 0.7086 to 0.6475, whereas across G→E it increases from 0.6024 to 0.7122; the full trajectories and phase boundaries are in Appendix I. These observations support responsiveness of the state and improved endpoint point estimates relative to fixed $p = 1$. They do not isolate a particular transition as the unique cause of the gain, and the six orders are paired conditions rather than independent seeds.

4.5 Transfer across Model Scales and Families

Table 4 reports matched RS–TCLD comparisons within Qwen3-0.6B and SmolLM2-1.7B. The results test transfer beyond the primary model configuration, but the limited set of sub-2B models does not establish a scaling law. Appendix J gives per-source results.

Table 4: Cross-model mixed-pool endpoint true maj@3 (%), reported as mean ± sample standard deviation over three matched seeds.

Model	RS Macro	TCLD Macro	Δ M.	RS Micro	TCLD Micro	Δ M.
Qwen3-0.6B	43.36 ± 0.53	$\mathbf{47.20 \pm 0.66}$	+3.84	47.26 ± 0.53	$\mathbf{49.66 \pm 0.33}$	+2.41
SmolLM2-1.7B	34.48 ± 0.49	$\mathbf{36.33 \pm 0.14}$	+1.86	37.04 ± 0.43	$\mathbf{38.83 \pm 0.47}$	+1.79

Within each model, TCLD improves both Macro and Micro means over its matched RS control. The larger Macro gain on Qwen3-0.6B and the positive SmolLM2 result show that the effect is not confined to the primary Qwen3-1.7B run, while the limited set of sub-2B configurations does not support broader claims about model scale.

5 Limitations

The controller observes loss severity rather than semantics and emits only $p \in (0, 1)$, so it cannot identify useful hard tokens or explore amplification. Source-wise and stream diagnostics use one seed, the math-heavy study covers only sub-2B models, and finite-window Hill estimates are calibration signals rather than tail guarantees. TCLD retains controller choices and observation cost despite avoiding a downstream-accuracy search over p. Because its objective couples relative allocation to common gradient scaling, the gains belong to the complete actuator; order-dependent stream effects do not imply that every refresh helps every source.

6 Conclusion

TCLD calibrates response-token loss aggregation from the current training-loss distribution. It improves mixed-pool accuracy over RS and the evaluated curricula, remains competitive under source-dependent fixed-power sensitivity, and transfers to two additional model configurations. Six ordered streams also yield higher final Macro and Micro point estimates than SFT, with order-dependent trade-offs. Future work can add utility-aware signals and broader natural shifts.

Reproducibility Statement

Implementation code, experiment configurations, and evaluation scripts are available in the anonymous supplementary repository: https://anonymous.4open.science/r/MuxxduLRJxXWBxmWVpTo.

AI Use Statement

Generative AI tools were used to assist with language editing and stylistic polishing, and for literature retrieval and discovery. All AI-assisted writing and retrieved references were reviewed by the authors.

References

Aida Amini, Saadia Gabriel, Peter Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. Mathqa: Towards interpretable math word problem solving with operation-based formalisms. In *Proceedings of NAACL-HLT*, pp. 2357–2367, 2019.

Jonathan T. Barron. A general and adaptive robust loss function. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4331–4339, 2019.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th International Conference on Machine Learning*, pp. 41–48, 2009.

Vivek S. Borkar. *Stochastic Approximation: A Dynamical Systems Viewpoint*. Cambridge University Press, 2008.

Peter S. Bullen. *Handbook of Means and Their Inequalities*. Kluwer Academic Publishers, 2003.

Thibault Castells, Philippe Weinzaepfel, and Jérôme Revaud. Superloss: A generic loss for robust curriculum learning. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, et al. Training verifiers to solve math word problems. In *Advances in Neural Information Processing Systems*, 2021.

Richard Diehl Martinez, Zébulon Goriely, Hope McGovern, Christopher Davis, Andrew Caines, Paula Buttery, and Lisa Beinborn. CLIMB – curriculum learning for infant-inspired model building. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pp. 112–127, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.conll-babylm.10. URL https://aclanthology.org/2023.conll-babylm.10/.

Yanbo Fan, Siwei Lyu, Yiming Ying, and Baogang Hu. Learning with average top-k loss. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Shuhao Gu, Jinchao Zhang, Fandong Meng, Yang Feng, Wanying Xie, Jie Zhou, and Dong Yu. Token-level adaptive training for neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pp. 1035–1046, 2020. doi: 10.18653/v1/2020.emnlp-main.76.

Godfrey Harold Hardy, John Edensor Littlewood, and George Pólya. *Inequalities*. Cambridge University Press, 1952.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. In J. Vanschoren and S. Yeung (eds.), *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/file/be83ab3ecd0db773eb2dc1b0a17836a1-Paper-round2.pdf.

Bruce M. Hill. A simple general approach to inference about the tail of a distribution. *The Annals of Statistics*, 3 (5):1163–1174, 1975. doi: 10.1214/aos/1176343247.

Shaojie Jiang, Thomas Wolf, Christof Monz, and Maarten de Rijke. TLDR: Token loss dynamic reweighting for reducing repetitive utterance generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 2259–2265, 2020. doi: 10.18653/v1/2020.acl-main.203.

Angelos Katharopoulos and François Fleuret. Not all samples are created equal: Deep learning with importance sampling. In *International Conference on Machine Learning*, 2018.

M. Pawan Kumar, Benjamin Packer, and Daphne Koller. Self-paced learning for latent variable models. In *Advances in Neural Information Processing Systems*, 2010.

Yangning Li, Tingwei Lu, Yinghui Li, Yankai Chen, Wei-Chieh Huang, Wenhao Jiang, Hui Wang, Hai-Tao Zheng, and Philip S. Yu. Teaching according to talents! instruction tuning LLMs with competence-aware curriculum learning. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 11724–11741, Suzhou, China, November 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.findings-emnlp.629. URL https://aclanthology.org/2025.findings-emnlp.629/.

Chen Liang, Haoming Jiang, Xiaodong Liu, Pengcheng He, Weizhu Chen, Jianfeng Gao, and Tuo Zhao. Token-wise curriculum learning for neural machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 3658–3670, 2021. doi: 10.18653/v1/2021.findings-emnlp.310.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2980–2988, 2017.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In Regina Barzilay and Min-Yen Kan (eds.), *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 158–167, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1015. URL https://aclanthology.org/P17-1015/.

Tambet Matiisen, Avital Oliver, Taco Cohen, and John Schulman. Teacher–student curriculum learning. *IEEE Transactions on Neural Networks and Learning Systems*, 31(9):3732–3740, 2019.

Sören Mindermann, Jan M. Brauner, Muhammed T. Razzak, Mrinank Sharma, Andreas Kirsch, Winnie Xu, Benedikt Höltgen, Aidan N. Gomez, Adrien Morisot, Sebastian Farquhar, and Yarin Gal. Prioritized training on points that are learnable, worth learning, and not yet learnt. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 15630–15649, 2022.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. Are nlp models really able to solve simple math word problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2080–2094, 2021.

Boris T. Polyak and Anatoli B. Juditsky. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, 1992.

Qwen Team. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Sidney I. Resnick. *Heavy-Tail Phenomena: Probabilistic and Statistical Modeling*. Springer, 2007. doi: 10.1007/978-0-387-45024-7.

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

Tianyi Zhou, Shengjie Wang, and Jeff A. Bilmes. Curriculum learning by dynamic instance hardness. In *Advances in Neural Information Processing Systems*, 2020.

A Implementation Contract

The current implementation satisfies the following contract.

1. Prompt positions are assigned label -100; all supplied response positions, including rationale and final answer, remain labeled.

2. Controller observations are raw finite positive response-token cross-entropies collected in evaluation mode without gradients.

3. The history weight is the coefficient on the previous log-index state. The studied setting uses 0.4 history and 0.6 new observation.

4. Valid Hill outputs are mapped by a stable sigmoid with no power clipping. Invalid later updates reuse history; an invalid first update uses $p = 1$.

5. The emitted full-precision power is detached and held fixed while the mega-batch/observation window is trained as ordinary batches. Observation losses calibrate p_k only; every training batch recomputes its losses for backpropagation.

6. Fractional-power training evaluates $\widetilde{\ell}_i = \max\{\ell_i, \epsilon\}$ with $\epsilon = 10^{-3}$ for numerical stability. Values below the floor are raised to 10^{-3} rather than deleted or set to zero, and they remain in the valid-token count. With the ordinary clamp implementation, however, their direct derivative with respect to the raw loss is zero. The explicit $p = 1$ control bypasses this floor and matches arithmetic response-token CE.

Algorithm 1 TCLD with the power-mean actuator

Require: training stream, ordinary-batch size B, mega-batch/observation-window size M, fractions $\mathcal{Q}$

initialize controller state as invalid and training fallback $p \leftarrow 1$
for each shuffled mega-batch $\mathcal{B}_k = (\mathcal{C}_{k,1}, \ldots, \mathcal{C}_{k,J_k})$ containing at most M examples **do**
    Sense: observe $\mathcal{B}_k$ and collect response-token CE losses without gradients
    compute components in Eq. (4) only when $r_{k,q} < n_k$ and ratios are valid
    if at least one component is valid **then**
        set S_k to their median (or the sole estimate when only one is usable)
        Update: initialize or update z_k by Eq. (6)
        Map: set detached $p_k \leftarrow \sigma(z_k)$
    else
        reuse the previous valid state, or retain $p_k = 1$ if none exists
    end if
    use the observed losses only for calibration
    for each ordinary-batch member $\mathcal{C}_{k,b} \in \mathcal{B}_k$ with $|\mathcal{C}_{k,b}| \leq B$ **do**
        recompute token losses and $L_{p_k}(\mathcal{C}_{k,b}; \theta)$ with detached p_k
        backpropagate $\nabla_\theta L_{p_k}(\mathcal{C}_{k,b}; \theta)$
        update θ with the optimizer
    end for
end for

B Power-Mean Derivations

Let

$$A_p = \frac{1}{m} \sum_i \ell_i^p, \qquad L_p = A_p^{1/p},$$

for $p > 0$ and positive active-token losses. Since $A_p = L_p^p$, the chain rule gives

$$\nabla_\theta L_p = \frac{1}{p} A_p^{1/p-1} \nabla_\theta A_p \tag{7}$$

$$= \frac{1}{p} A_p^{1/p-1} \left(\frac{1}{m} \sum_i p \ell_i^{p-1} \nabla_\theta \ell_i \right) \tag{8}$$

$$= \frac{L_p^{1-p}}{m} \sum_i \ell_i^{p-1} \nabla_\theta \ell_i. \tag{9}$$

Thus, the full scalar coefficient multiplying the gradient of token i is

$$a_i(p) = \frac{L_p^{1-p}}{m} \ell_i^{p-1}.$$

The factor L_p^{1-p}/m is shared by all active tokens in the current training batch, whereas ℓ_i^{p-1} is token-specific. Normalizing these scalar coefficients across tokens removes the shared factor:

$$q_i(p) = \frac{a_i(p)}{\sum_r a_r(p)} = \frac{\ell_i^{p-1}}{\sum_r \ell_r^{p-1}}.$$

Consequently,

$$\frac{q_i(p)}{q_j(p)} = \left(\frac{\ell_i}{\ell_j} \right)^{p-1},$$

which gives Eq. (3). Equivalently,

$$\log \frac{q_i}{q_j} = (p-1)(\log \ell_i - \log \ell_j), \qquad \frac{\partial}{\partial p} \log \frac{q_i}{q_j} = \log \frac{\ell_i}{\ell_j}.$$

These normalized coefficients describe relative scalar allocation; they do not account for the magnitude or direction of $\nabla_\theta \ell_i$ and therefore are not fractions of the complete parameter gradient. To decompose the gradient into relative token allocation and a batch-dependent common gradient scale, define

$$Z_p = \frac{1}{m}\sum_i \ell_i^{p-1}, \qquad w_i = \frac{\ell_i^{p-1}}{Z_p}, \qquad c_p = L_p^{1-p} Z_p.$$

By this definition, $w_i = m q_i$. Thus q_i sums to one, whereas w_i has mean one; the two normalizations encode the same relative allocation. Moreover,

$$\nabla_\theta L_p = c_p \left(\frac{1}{m} \sum_i w_i \nabla_\theta \ell_i \right).$$

The normalized weights w_i isolate relative token allocation, whereas c_p is a batch-dependent common gradient scale. A relative-only diagnostic would detach w_i and optimize $m^{-1}\sum_i w_i \ell_i$; a scale-only diagnostic would detach c_p and multiply ordinary mean CE by it. These controls are not part of the present result set, so the paper attributes performance to the complete power-mean objective rather than to either component alone.

When the floor is inactive, common rescaling $\ell_i \mapsto c\ell_i$, $c > 0$, leaves every coefficient ratio unchanged. When the floor is active, that invariance need not hold. For $p \neq 1$, the implemented value is $\widetilde{\ell}_i = \max\{\ell_i, \epsilon\}$ and

$$\frac{\partial \widetilde{\ell}_i}{\partial \ell_i} = \begin{cases} 0, & \ell_i < \epsilon, \\ 1, & \ell_i > \epsilon, \end{cases}$$

with the framework's clamp subgradient convention at equality. Thus a below-floor term remains in the aggregate and token count but has no direct gradient path through its raw loss. For the explicit $p = 1$ branch, raw arithmetic CE bypasses the clamp.

C Hill Estimation and Robust Aggregation

For an exact Pareto tail above threshold u, the log exceedances $\log(X/u)$ are exponential with mean $1/\alpha$, motivating Eq. (4). Under the classical intermediate-order conditions $r_{k,q} \to \infty$ and $r_{k,q}/n_k \to 0$, the Hill estimator is consistent for a regularly varying tail (Hill, 1975; Resnick, 2007). The implementation instead uses finite preset fractions and a minimum count, so a finite-sample error guarantee requires additional assumptions and is not asserted here.

Because Eq. (4) uses log ratios,

$$\widehat{\alpha}(cX_{k,1}, \ldots, cX_{k,n_k}) = \widehat{\alpha}(X_{k,1}, \ldots, X_{k,n_k}), \qquad c > 0,$$

whenever the same component remains valid. A component is usable only when $r_{k,q} < n_k$, its threshold and selected values are finite and positive, and its mean log ratio is finite and strictly positive. These checks protect the computation but do not validate the tail model. To make the positivity statement explicit, let

$$M_k = \max_i X_{k,i} < \infty, \qquad u_{k,q} = X_{k,(n_k - r_{k,q})} > 0$$

for a valid, nondegenerate component. Since every selected upper-order loss is at most M_k, Eq. (4) gives

$$\widehat{\alpha}_{k,q} \geq \left[\log\left(\frac{M_k}{u_{k,q}} \right) \right]^{-1} > 0.$$

Thus the usable Hill estimates and their median S_k are strictly positive on every realized finite window. This lower bound depends on the observed maximum and threshold; it is not a uniform bound over unrestricted model parameters.

Lemma C.1 (Median protection)**.** *If at least two of three usable Hill estimates lie in* $[\alpha - \delta, \alpha + \delta]$*, their median lies in the same interval.*

To prove the lemma, sort the usable estimates. If two of three lie in the stated interval, the middle order statistic cannot lie below both or above both. Dependence among the values does not affect this deterministic statement, but it matters for probabilistic uncertainty quantification. Nearby fractions expose some threshold sensitivity, while their shared sample prevents interpreting them as independent replications. The fallback after zero usable components is a training-state rule, not a consequence of Hill estimation.

D Log-Index Tracking and Mapping

For $\alpha > 0$, the map $h(\alpha) = \alpha/(1+\alpha)$ satisfies

$$0 < h(\alpha) < 1, \qquad h'(\alpha) = \frac{1}{(1+\alpha)^2} > 0, \qquad h(\alpha) < \alpha.$$

These properties follow directly from $\alpha > 0$: the denominator exceeds the numerator, differentiation gives the displayed positive derivative, and division by $1 + \alpha > 1$ gives $h(\alpha) < \alpha$. The first two properties establish the bounded range and monotonicity used by the controller. They do not establish that this map is optimal for downstream accuracy.

Let $a_k = \log \alpha_k$ be an ideal log-index path and write its observation as $\widehat{a}_k = a_k + \xi_k$. Repeated substitution in the state update gives

$$z_k - a_k = \beta^k (z_0 - a_0) + (1-\beta) \sum_{j=1}^{k} \beta^{k-j} \xi_j + \sum_{j=1}^{k} \beta^{k-j+1} (a_{j-1} - a_j). \tag{10}$$

Since $\sup_z \sigma'(z) = 1/4$,

$$|p_k - h(\alpha_k)| \le \frac{1}{4} |z_k - a_k|. \tag{11}$$

The first sum is propagated sensing error, and the second captures lag under a changing ideal state. For a compact conditional summary, let $e_k = z_k - a_k$ and substitute $\widehat{a}_k = a_k + \xi_k$ into the recursion:

$$e_k = \beta e_{k-1} + (1-\beta)\xi_k + \beta(a_{k-1} - a_k).$$

Repeated substitution gives Eq. (10). If $\mathbb{E}|\xi_k| \le \varepsilon_a$ and $|a_k - a_{k-1}| \le \Delta_a$, then

$$\limsup_{k \to \infty} \mathbb{E}|z_k - a_k| \le \varepsilon_a + \frac{\beta}{1-\beta} \Delta_a,$$

and the sigmoid's $1/4$ Lipschitz constant yields the corresponding power-state bound. The result describes a bias–lag trade-off but does not select a universally optimal observation-window size or refresh rate. The mapping error inherits this bound only through the sigmoid's Lipschitz property. Its range $(0, 1)$ enforces one-sided compression; it supplies no evidence about whether amplification would be useful. A bidirectional map would require an additional utility-aware state and a separate stability analysis.

E Dynamic Objective Paths

For a fixed mini-batch and parameter state, write

$$g_t(p; \theta) = \nabla_\theta L_p(\mathcal{C}_t; \theta).$$

Even at fixed θ, the generalized mean of unequal positive losses is nonlinear in p. Consequently, for $\bar{p} = T^{-1} \sum_{t=1}^{T} p_t$, a same-state comparison generally gives

$$\sum_{t=1}^{T} g_t(p_t; \theta_t) \neq \sum_{t=1}^{T} g_t(\bar{p}; \theta_t). \tag{12}$$

Equation (12) deliberately evaluates both sides at the states θ_t reached by the dynamic run; it is not an identity between the dynamic trajectory and a separately trained fixed-$\bar{p}$ trajectory. In an actual fixed-power run, the first differing update can also change subsequent model and optimizer states, so later gradients are evaluated at different parameters. A fixed value near the mean dynamic trajectory is therefore a useful diagnostic, but neither equality nor superiority follows from the mean alone.

The following bound only controls how much the scalar objective changes when the power changes while the mini-batch and parameter state are held fixed. Let $I = [p_{\min}, p_{\max}] \subset (0, 1)$ contain $p_0, \ldots, p_T$, and suppose the finitely many active losses evaluated at the stated pairs $(\mathcal{C}_t, \theta_t)$ are positive and finite. With

$$A_t(p) = \frac{1}{m_t} \sum_i \ell_{t,i}^p, \qquad L_t(p) = A_t(p)^{1/p},$$

direct differentiation gives

$$\frac{\partial L_t(p)}{\partial p} = L_t(p)\left[-\frac{\log A_t(p)}{p^2} + \frac{1}{p}\frac{\sum_i \ell_{t,i}^p \log \ell_{t,i}}{\sum_i \ell_{t,i}^p}\right].$$

This derivative is continuous on the compact interval I for every t. Because the run contains finitely many steps,

$$K = \max_{1\le t\le T} \sup_{p\in I} \left|\frac{\partial L_t(p)}{\partial p}\right| < \infty.$$

Applying the mean-value theorem separately to $L_t(p_t)$ and $L_t(p_{t-1})$, and then summing, yields

$$\sum_{t=1}^{T} \left|L_{p_t}(\mathcal{C}_t;\theta_t) - L_{p_{t-1}}(\mathcal{C}_t;\theta_t)\right| \le K \sum_{t=1}^{T} |p_t - p_{t-1}|. \tag{13}$$

This same-state scalar bound describes variation induced by the moving controller state; it is not a convergence result, does not compare the parameter trajectories of two training runs, and does not imply improved downstream utility. Extending the power interval to zero, or seeking a uniform bound over an unrestricted family of losses and parameter states, requires additional control and is not covered.

F Severity–Utility Counterexample

Consider two corpora whose teacher-forced loss multisets are identical. In corpus A, the largest losses are attached to corrupted response suffixes. In corpus B, the same values are attached to correct, rare proof steps. Every distribution-only tail controller, including one using an exact Hill estimator, receives the same observation and emits the same power. Yet an external utility criterion can prefer compression in A and preservation or amplification in B. This construction establishes the severity–utility identifiability boundary and shows why improved tail estimation cannot remove it.

G Matched Static-Power Diagnostic

Table 5 reports paired endpoint differences between TCLD and the matched fixed $p = 0.7$ diagnostic. Positive values favor TCLD. The two-sided 95% t intervals are $[-1.25, 2.82]$ Macro and $[-1.95, 2.64]$ Micro. With only three seeds these intervals are descriptive and do not establish a statistically reliable difference.

Table 5: Paired mixed-pool endpoint comparison (true maj@3, %). Δ is TCLD minus fixed $p = 0.7$.

Seed	Fixed Macro	TCLD Macro	Δ Macro	Fixed Micro	TCLD Micro	Δ Micro
42	59.67	60.35	$+0.68$	63.38	62.75	-0.63
100	59.03	59.05	$+0.02$	62.56	63.01	$+0.45$
340	58.44	60.09	$+1.65$	62.06	63.27	$+1.21$
Mean $\pm$ s.d.	—	—	$+0.78\pm0.82$	—	—	$+0.34\pm0.92$

H Source-Wise Absolute Results

Table 6 provides the absolute accuracies underlying Figure 3. The fixed-power grid is diagnostic: the column-wise mean asks whether one globally fixed geometry transfers across sources, whereas the bold entry in each row is a post-hoc source-specific oracle and is not an implementable shared baseline.

Table 6: Single-source true maj@3 accuracy (%) under the low-learning-rate protocol. All entries use seed 42 and a checkpoint selected without test accuracy. Bold denotes the highest observed accuracy and underline denotes the second highest; ties are preserved. The formatting does not imply statistical significance. "Mean" gives the equal-weight average over the seven sources.

Source	RS	$p = 0.3$	$p = 0.5$	$p = 0.7$	$p = 0.9$	$p = 1.1$	$p = 1.3$	TCLD
EntailmentBank	64.71	62.57	67.38	67.38	<u>71.12</u>	67.38	60.96	**72.73**
QASC	74.51	**75.38**	73.54	74.41	73.00	73.33	73.87	<u>75.16</u>
SVAMP	73.00	71.67	71.67	<u>73.67</u>	73.33	**74.33**	**74.33**	69.00
AQuA-RAT	51.18	<u>54.72</u>	53.15	50.79	**56.69**	51.57	48.82	54.33
MATH-500	29.00	33.40	**36.40**	33.00	32.40	32.40	25.00	<u>35.20</u>
GSM8K	64.29	68.08	68.84	**69.90**	66.94	67.10	63.08	<u>69.07</u>
MathQA	61.37	61.81	**64.15**	63.18	61.07	58.29	55.78	<u>63.62</u>
Mean	59.72	61.09	62.16	61.76	<u>62.08</u>	60.63	57.41	**62.73**

I Two-Stage Stream Details

All six directions use Qwen3-1.7B-Base, seed 42, two source-local training epochs per phase, deterministic within-source shuffling, and no cross-source shuffle. Each phase is divided into four observation windows. SFT and Online TCLD share the same ordered stream and pre-specified terminal checkpoint within a direction; Online TCLD records eight controller states, including the initialization from the first observed window, with no fallback or implicit clipping. EG and GE use 2,356 EntailmentBank training rows and 2,700 GSM8K rows, ending their first phases at steps 589 and 675, respectively, and both end at step 1,264. QG and GQ use 6,152 QASC rows and 2,700 GSM8K rows; their phase boundaries are $1{,}538 \to 2{,}213$ and $675 \to 2{,}213$. AQ and QA use the corresponding AQuA-RAT and QASC source-local phases. Each aggregate below is computed from the displayed endpoint counts and evaluation denominators.

Table 7: Per-source terminal true maj@3 results for the two-stage streams. Accuracy is computed from the displayed correct count and denominator.

Stream	Source	N	SFT		Online TCLD	
			Correct	Acc.	Correct	Acc.
E→G	EntailmentBank	187	155	82.89	157	83.96
	GSM8K	1319	842	63.84	881	66.79
G→E	GSM8K	1319	814	61.71	866	65.66
	EntailmentBank	187	130	69.52	123	65.78
Q→G	QASC	926	666	71.92	670	72.35
	GSM8K	1319	815	61.79	879	66.64
G→Q	GSM8K	1319	892	67.63	908	68.84
	QASC	926	636	68.68	648	69.98
A→Q	AQuA-RAT	254	128	50.39	126	49.61
	QASC	926	651	70.30	671	72.46
Q→A	QASC	926	634	68.47	668	72.14
	AQuA-RAT	254	115	45.28	139	54.72

Macro is the arithmetic mean of the two source accuracies, whereas Micro is $100(c_1 + c_2)/(n_1 + n_2)$. We compute all entries in Table 3 from the unrounded ratios above and round only the displayed percentages.

Table 8: Online TCLD power states at the eight observation windows of each ordered stream. Values are rounded to four decimals for display.

Stream	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8
E→G	.7086	.7095	.7083	.7066	.6762	.6621	.6489	.6475
G→E	.6024	.6338	.6379	.6458	.6879	.7053	.7097	.7122
Q→G	.8320	.7653	.7188	.7011	.6501	.6507	.6452	.6456
G→Q	.5989	.6439	.6501	.6485	.8085	.7569	.7319	.7212
A→Q	.7370	.7416	.7419	.7392	.7961	.7491	.7191	.7119
Q→A	.8320	.7649	.7178	.7007	.7211	.7369	.7417	.7425

The change between the fourth and fifth entries coincides with the source-phase transition in each run. Each state is computed from the mega-batch just observed and is then held fixed for that mega-batch's ordinary-batch updates; it does not use validation accuracy or future-source data. The endpoint comparison does not rely on intermediate accuracy trajectories.

J Full Cross-Model Results

The compact aggregate view is reproduced in Table 9; the following tables provide the per-source values. Bold denotes the higher endpoint within each model comparison; the corresponding deltas are reported in the final row.

Table 9: Cross-model mixed-pool endpoint summary (true maj@3, %). Values are mean ± sample standard deviation over three matched seeds.

Model	RS Macro	TCLD Macro	Δ Macro	RS Micro	TCLD Micro	Δ Micro
Qwen3-0.6B	43.36±0.53	**47.20±0.66**	+3.84	47.26±0.53	**49.66±0.33**	+2.41
SmolLM2-1.7B	34.48±0.49	**36.33±0.14**	+1.86	37.04±0.43	**38.83±0.47**	+1.79

Table 10: Mixed-pool true maj@3 accuracy (%) with Qwen3-0.6B-Base.

Method	GSM8K	MathQA	MATH-500	AQuA-RAT	SVAMP	Macro	Micro
RS	47.79	51.22	17.47	33.99	66.33	43.36±0.53	47.26±0.53
TCLD	**50.32**	**52.60**	**23.47**	**41.08**	**68.56**	**47.20±0.66**	**49.66±0.33**
Δ	+2.53	+1.38	+6.00	+7.09	+2.23	+3.84	+2.41

Table 11: Mixed-pool true maj@3 accuracy (%) with SmolLM2-1.7B-Base.

Method	GSM8K	MathQA	MATH-500	AQuA-RAT	SVAMP	Macro	Micro
RS	33.38	41.76	10.47	32.55	54.22	34.48±0.49	37.04±0.43
TCLD	**35.58**	**43.32**	**12.60**	**33.73**	**56.44**	**36.33±0.14**	**38.83±0.47**
Δ	+2.20	+1.55	+2.13	+1.18	+2.22	+1.86	+1.79

K Learning-Curve Numerical Results

Table 12 records the numerical values behind Figure 2. The horizontal coordinate is the scheduled optimization budget normalized by the common endpoint; AULC is the normalized trapezoidal area under the three-seed mean Macro curve. These values are optimization-step efficiency measures and do not account for the additional no-gradient observation passes used by TCLD.

Table 12: Mixed-pool learning-curve values. Entries are three-seed mean Macro true maj@3 accuracy (%).

Method	R1	R2	R3	R4	R5	R6	R7	R8	AULC	Target budget (%)
RS	53.38	54.30	53.83	55.16	55.78	56.29	56.72	57.09	48.41	100.0
Dynamic CL	48.75	51.97	55.53	55.76	54.31	56.20	58.02	57.18	48.09	87.5
TCLD	56.51	57.64	56.96	58.08	58.46	59.14	59.50	59.83	50.99	25.0

L Controller and Refresh Diagnostics

The primary Qwen3-1.7B controller records contain repeated Hill estimates, log-state updates, and detached powers for every observation window. Across seeds 42, 100, and 340, the mean recorded power is approximately 0.688, the final powers lie in 0.663–0.665, and no fallback or clipping event is recorded. The controller therefore remains on the compression side while tracking a slowly changing loss distribution; Table 13 separately evaluates how the refresh granularity affects endpoint performance.

Observation-window granularity. Table 13 compares three single-source schedules on the seven-source suite used in Figure 3. Aggregate accuracy is nearly flat: the Macro range is only 0.24 points and no schedule dominates source-wise. More frequent refresh is therefore not monotonically better; window granularity jointly changes state freshness and the sample size of each tail estimate, and its effect interacts with the source distribution. The corresponding power ranges are narrow for several sources but substantially wider for QASC and SVAMP, without a monotonic relation to endpoint accuracy. These endpoint-only seed-42 runs are a sensitivity audit rather than part of the matched three-seed main comparison. The two- and eight-window runs were executed on A800, whereas the sixteen-window run was executed on A100, so the table is descriptive and is not used for a wall-clock or hardware-efficiency claim.

Table 13: Observation-window sensitivity on the seven-source single-source suite (true maj@3 accuracy, %). "Windows" denotes the number of observation windows over the run. Macro excludes StrategyQA, which is outside the paper's diagnostic suite.

Windows	EntailmentBank	QASC	SVAMP	AQuA-RAT	MATH-500	GSM8K	MathQA	Macro
2	68.98	74.51	72.67	54.72	33.80	70.13	63.28	62.59
8	72.19	73.87	74.67	51.18	35.20	69.14	63.58	**62.83**
16	**72.73**	**75.16**	69.00	54.33	**35.20**	69.07	**63.62**	62.73

M MIXED-POOL RESPONSE-LENGTH AUDIT

Table 14 records the composition and supervision formats of the mixed pool. Table 15 and Figure 4 provide the corresponding response-side statistics. Together they document the heterogeneity of the supervised responses; they are not themselves evidence that a particular token should be emphasized or compressed.

Table 14: Composition of the heterogeneous reasoning pool after length filtering. MATH-500 is used for MATH evaluation.

Source	Train N	Supervised response	Answer type	Evaluation split (N)
GSM8K	7,473	natural-language calculation	open numeric	test (1,319)
MathQA	29,837	multiple-choice rationale	choice a–e	test (2,985)
MATH	7,465	competition-level derivation	boxed numeric/symbolic	MATH-500 (500)
AQuA-RAT	8,000	algebraic choice rationale	choice A–E	validation (254)
SVAMP	700	short arithmetic chain	open numeric	validation (300)

Table 15: Response-token composition after the training length filter. Counts are median/mean/P90 under the Qwen3-1.7B-Base tokenizer; "Resp./full" is the mean supervised fraction of each example.

Source	Response tokens (median/mean/P90)	Resp./full
GSM8K	112 / 122.8 / 196	66.2%
MathQA	79 / 92.1 / 161	49.9%
MATH	162 / 222.7 / 471.6	71.9%
AQuA-RAT	72 / 86.5 / 155	47.2%
SVAMP	23 / 24.2 / 30	36.3%

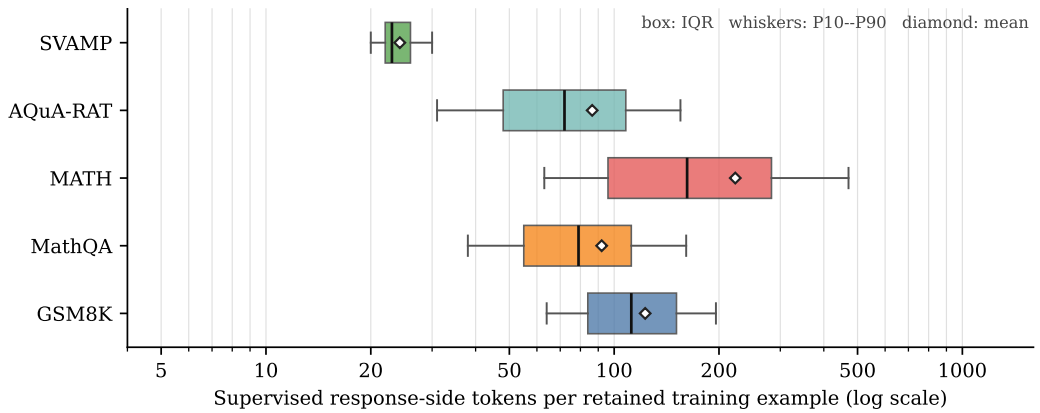

Figure 4: Response-token distributions in the retained mixed pool. Boxes show interquartile ranges, whiskers P10–P90, and diamonds means on a logarithmic horizontal axis.

N Experimental Settings

Primary mixed-pool protocol. The primary mixed-pool experiments use Qwen3-1.7B-Base (Qwen Team, 2025) and a training pool of 53,475 examples from GSM8K, MathQA, MATH, AQuA-RAT, and SVAMP. Examples are filtered to a maximum training sequence length of 1,536. Overlength examples are removed rather than retaining a truncated target response. Supervision is applied only to response-token positions, while prompt and padding positions are masked.

The primary experiments are conducted on an A100 GPU using FP32 LoRA fine-tuning. The LoRA rank is 16, the scaling factor is 32, and the dropout rate is 0.05. Adapters are applied to all attention projections (`q_proj`, `k_proj`, `v_proj`, and `o_proj`) and MLP projections (`gate_proj`, `up_proj`, and `down_proj`). The ordinary per-device training batch size is four, gradient accumulation is one, and the learning rate is 8×10^{-5} with cosine decay and a 2% warm-up. The matched random seeds are 42, 100, and 340.

Within each matched seed, the compared methods use the same Base checkpoint, matched LoRA initialization, training-example multiset, response-token mask, optimization schedule, training budget, and evaluation protocol. The sample-level curriculum methods intentionally change the example-exposure order. The reported primary result is the eighth and final pre-specified maj@3 evaluation checkpoint. Training continues to the common maximum optimization budget, but no later generation evaluation is used to select or replace the reported result.

Observation windows and TCLD controller. In each complete traversal of the primary training pool, the 53,475 randomly permuted examples are divided into $K = 4$ approximately equal observation windows, each containing about 13.4k examples, with remainder examples retained in the final window. The shared training budget covers approximately two complete traversals, yielding about eight window-level controller updates over the complete run. A recorded controller history may additionally include an initialization-state entry; such an entry does not correspond to an additional trained window.

At the beginning of each observation window, TCLD performs a no-gradient forward pass, using an observation batch size of four, and collects the losses of valid response tokens across the window. These observed losses are used only to calibrate the controller. The same examples are subsequently divided into ordinary training batches of at most four examples. Each ordinary batch recomputes its token losses and its own power-mean objective for backpropagation and parameter updates. The calibrated power p_k is held fixed within the observation window, whereas the model parameters, current token losses, and their relative gradient coefficients continue to change during optimization. The next window is observed using the updated model.

The controller evaluates Hill components at upper-tail fractions $\{0.025, 0.05, 0.075\}$. A component is used only when at least 50 tail observations are available, and the usable estimates are aggregated

by their median. When fewer than three components are usable, the median is taken over the available components; a single usable component therefore determines the observation statistic by itself. The log-state update assigns weight 0.4 to the previous state and 0.6 to the current observation. Equation (6) then maps the tracked state to $p_k \in (0, 1)$. The resulting power is detached from the computation graph and is not explicitly clipped.

Across all reported TCLD experiments, fractional-power optimization uses the numerical floor

$$\widetilde{\ell}_i = \max\{\ell_i, 10^{-3}\}.$$

The floor does not remove a token, change the response-token mask, or exclude it from the token count. With the ordinary clamp implementation, however, the direct derivative through the raw loss is zero when $\ell_i < 10^{-3}$. The explicit $p = 1$ branch uses ordinary arithmetic response-token cross-entropy and bypasses the fractional-power floor. Further implementation details are provided in Appendix A.

Primary sample-level curriculum baselines. The RS baseline randomly shuffles the training examples and optimizes ordinary response-token cross-entropy. Static CL (length) computes the formatted effective sequence length once before training, globally sorts the complete pool from shorter to longer sequences, and uses the stable sample identifier to break ties. The resulting order is reused on subsequent traversals without rescoring, reshuffling, or paced subset expansion.

Static CL (initial PPL) applies the same frozen global-ordering protocol, replacing length with a frozen Base-model score. The implementation caches the mean cross-entropy over the valid response tokens of each example. Exponentiation is strictly monotone, so this cached score induces exactly the same ordering as response-token perplexity. Examples are ordered from lower to higher initial difficulty, and the order remains fixed throughout training.

Dynamic CL uses the same observation-window partition as its matched TCLD run. At the beginning of each window, it performs a no-gradient forward pass and assigns every example the mean cross-entropy over its valid response tokens. Examples within the observed window are sorted from lower to higher score, with stable sample identifiers used to break ties. The same window is then trained in the resulting order using ordinary response-token cross-entropy. Difficulty scores and the within-window ordering are refreshed at the next observation boundary. Dynamic CL therefore incurs an observation forward pass comparable in form to that of TCLD, while changing sample order rather than response-token loss aggregation.

Mixed-pool fixed-power references. The mixed-pool calibration comparison reports three completed, matched, three-seed fixed-power references. The setting $p = 0.3$ represents a relatively small power, $p = 0.7$ is close to the average control state produced by TCLD in the primary mixed-pool experiments, and $p = 1.3$ represents a relatively large power. These values provide low, intermediate, and high references for the power-mean geometry.

The other fixed-power values considered in the single-source sweep do not have complete matched three-seed mixed-pool results under the same protocol and are therefore not combined with these three references. No downstream accuracy from a fixed-power experiment is used to determine the online TCLD trajectory.

Single-source calibration diagnostics. The source-wise experiments in Figure 3 and Table 6 use Qwen3-1.7B-Base, seed 42, and a learning rate of 4×10^{-5}. They compare RS, fixed powers

$$p \in \{0.3, 0.5, 0.7, 0.9, 1.1, 1.3\},$$

and TCLD under the same response-only supervision rule.

The TCLD results used in the primary source-wise comparison were produced on A100 with approximately 16 observation windows over each complete source-specific training run. Because the source datasets differ in size and training budget, the number of examples and valid response tokens contained in an individual window is source dependent.

Each method completes its full source-specific optimization budget. Checkpoint selection uses only standard response-token cross-entropy on an inner-validation split. Let $v_{\min}$ denote the lowest recorded validation CE for a method. Checkpoints whose CE does not exceed $1.005 v_{\min}$ are treated

as approximately tied, and the earliest such checkpoint is selected. The selected checkpoint receives one true maj@3 evaluation on the held-out evaluation split. Neither validation accuracy nor test accuracy is used for checkpoint selection.

The refresh-frequency audit compares $K \in \{2, 8, 16\}$ observation windows over the complete source-specific run. The 2- and 8-window runs were conducted on A800 resources, whereas the approximately 16-window reference was conducted on A100. These results are used descriptively to study endpoint sensitivity to refresh granularity and are not used for cross-hardware timing comparisons.

Ordered two-source streams. The ordered-stream experiments use Qwen3-1.7B-Base, seed 42, learning rate 4×10^{-5}, ordinary training batch size four, gradient accumulation one, and maximum training sequence length 1,536. Each source contributes two locally shuffled training epochs and is divided into four observation windows. Examples are deterministically shuffled within a source phase, while examples from different phases are not mixed.

SFT keeps $p_t \equiv 1$ throughout the stream. Online TCLD applies the controller at the beginning of every observation window and holds the resulting p_k fixed while that window is trained. Within a stream direction, SFT and Online TCLD use the same source order, source-local training budget, prompt and response-mask convention, and pre-specified terminal evaluation. The same controller settings and the same $\epsilon = 10^{-3}$ numerical floor as in the primary TCLD protocol are used in all six stream directions.

The evaluated stream orders are EntailmentBank→GSM8K, GSM8K→EntailmentBank, QASC→GSM8K, GSM8K→QASC, AQuA-RAT→QASC, and QASC→AQuA-RAT. The EntailmentBank–GSM8K directions and QASC→GSM8K were executed on A100. GSM8K→QASC and the AQuA-RAT–QASC directions were executed on A800. Each direction is therefore interpreted as a paired within-stream SFT–Online TCLD comparison; absolute runtime is not compared across hardware. Exact source sizes, phase boundaries, endpoint counts, and controller trajectories are reported in Appendix I.

Cross-model experiments. The Qwen3-0.6B-Base and SmolLM2-1.7B experiments use the same mixed training pool, response-token masking rule, FP32 arithmetic, maximum training sequence length 1,536, ordinary batch size four, gradient accumulation one, learning rate 8×10^{-5}, cosine schedule, 2% warm-up, common optimization budget, and endpoint protocol as the primary experiment. Their TCLD runs use four observation windows per complete traversal of the mixed pool.

Both model configurations use LoRA rank 16, scaling factor 32, and dropout 0.05. The Qwen3-0.6B configuration differs from the primary Qwen3 configuration in the Base checkpoint. The Qwen3-0.6B comparison uses matched seeds 42, 100, and 340.

The SmolLM2-1.7B configuration applies adapters to the architecture's `q_proj`, `k_proj`, `v_proj`, `o_proj`, `gate_proj`, `up_proj`, and `down_proj` modules. It uses a distinct padding token rather than aliasing padding to EOS. The seed coverage of the reported SmolLM2 comparison is stated with the corresponding result table. These experiments support paired RS–TCLD comparisons within each model configuration rather than controlled performance comparisons between model architectures.

Generation and maj@3 scoring. For every evaluation item, the model generates three independently sampled reasoning paths. Generation uses `do_sample=true`, temperature 0.7, one beam, three returned sequences, and at most 1,024 newly generated tokens. EOS and padding token identifiers are supplied to generation. The evaluation function does not explicitly override top-p or top-k; these values are inherited from the saved model generation configuration. Evaluation prompts are tokenized with a maximum input length of 2,048, which is distinct from the 1,536-token training limit.

A task-specific parser extracts the final answer from each generated sequence. The predicted answer is the most frequent valid extracted answer among the three generations. If multiple valid answers have equal frequency, the first encountered tied answer is used. If none of the three generations yields a valid extracted answer, the item is scored as incorrect. Macro accuracy assigns equal weight to

each evaluation source, whereas Micro accuracy pools the correct counts and sample counts across sources.

Computation and timing. The training-cost decomposition is

$$C_{\mathrm{TCLD}} = C_{\mathrm{SFT}} + C_{\mathrm{observation}} + C_{\mathrm{controller}}.$$

The dominant additional term is the no-gradient forward pass performed at the beginning of each observation window. Sorting the observed upper-tail losses and updating the scalar controller state are secondary. If each training example is observed once in addition to a training forward–backward pass, and the costs of a forward and backward pass are denoted by F and B, respectively, the leading relative increment is approximately

$$\frac{F}{F+B}.$$

A value near one third additionally assumes $B \approx 2F$ and is not treated as a measured overhead for the reported LoRA experiments.

The three primary TCLD runs required approximately 28.0 hours on average on A100. This end-to-end measurement includes training, the eight scheduled maj@3 evaluations, checkpoint handling, and associated framework overhead. A same-scope timing record for the three primary RS runs was not recovered, so no primary mixed-pool TCLD–RS wall-clock overhead ratio is reported.

A paired timing record is available for the QASC→GSM8K stream. SFT required 4,081 seconds and Online TCLD required 4,454 seconds, corresponding to an approximately 9.1% increase in that run. Both measurements include training and terminal maj@3 evaluation. They therefore provide an end-to-end example under one experimental condition, rather than an estimate of controller-only cost or a universal TCLD overhead.

The learning curves in Figure 2 and Appendix K compare progress under a common optimizer-step budget and are not interpreted as equal-FLOP or wall-clock acceleration.

Experiment records and software environment. The experiment artifacts record resolved training and evaluation configurations, random seeds, checkpoint metadata, evaluation counts, observation-window settings, and controller histories. For TCLD runs, the recorded controller-update steps and resolved mega-batch settings document the realized observation-window schedule. The primary-run metadata reports Python 3.10.20, PyTorch 2.8.0+cu126, NumPy 2.2.6, and CUDA 12.6.