# SUPERVISION MECHANISMS IN IMPLICIT CHAIN-OF-THOUGHT: TARGETS, ALLOCATION, AND SPACE

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

We study supervision mechanisms in implicit chain-of-thought models along three dimensions: the quantities to align (Targets), the updates to supervise directly (Allocation), and the dimension and directions of the alignment space (Space). Mathematical analysis reveals how different supervision targets organize reasoning errors: local update targets constrain discrepancies in individual state changes, whereas cumulative update targets aggregate these discrepancies across steps. Controlled comparisons in a shared recurrent student show that, with local update targets, extending direct supervision to intermediate updates in the answer subspace further improves reasoning accuracy. The choice of supervision space also matters: the answer subspace outperforms full-space alignment and rank-matched random subspaces, highlighting the role of direction selection beyond dimensionality reduction. Based on these findings, we introduce Answer-guided Local Update Supervision (ALUS), which aligns teacher and student changes between consecutive hidden states at every valid update. Alignment uses a fixed subspace derived from output-weight rows for training-answer tokens and adds no inference-time modules. ALUS reaches $41.17 \pm 0.20\%$ on GPT-2/GSM8K, 8.31 percentage points above no alignment and 4.88 above the strongest adapted supervision objective within the shared student. The benefits of all-valid supervision and the answer subspace persist with six updates; experiments with Pythia-1B and nonnumeric ProsQA provide further evidence for the effectiveness of ALUS.

## 1 INTRODUCTION

Explicit chain-of-thought (CoT) enables language models to work through a problem by generating intermediate reasoning text before an answer (Wei et al., 2022). Expressing this computation as natural-language tokens increases generation length and inference cost. Implicit CoT reduces reliance on intermediate text by internalizing reasoning in hidden states through distillation or continuous hidden-state feedback (Deng et al., 2023; Hao et al., 2025). This shift changes the supervision problem: explicit rationales provide token-level prediction targets, but there is no directly observed label for what each reasoning state should represent. The training problem is therefore to use explicit reasoning traces to supervise these hidden states effectively.

Existing methods obtain supervision from teacher representations, including pre-answer hidden states, compressed key–value caches, and low-rank trajectory statistics (Shen et al., 2025; Kuzina et al., 2026; Solgi et al., 2026), or connect hidden states to textual reasoning through step reconstruction and span alignment (Wei et al., 2026; Chu et al., 2026).

Recent work has begun to explain how supervision affects learning in latent reasoning. Chen et al. (2026) relate supervision design to the reasoning information recoverable from hidden states. In looped language models, Sharma & Vu (2026) identify a complementary issue: normalized readouts can hide hidden-state scale changes from the immediate prediction loss, even with supervision at every loop.

The reasoning information supplied by supervision does not, by itself, determine the learning signals it produces. We therefore study supervision mechanisms within direct teacher–student alignment, asking: *how does the design of supervision affect learning signals and reasoning performance?*



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.

We study three supervision choices: **Targets** defines the quantities matched between teacher and student, **Allocation** determines which updates receive direct supervision, and **Space** specifies the dimension and directions of the supervision space. We combine mathematical analysis with empirical comparisons in a shared recurrent student, keeping student computation fixed. Our contributions are:

1. **Analysis of supervision mechanisms.** We study Targets, Allocation, and Space within the shared student. Mathematical analysis explains how local-update and cumulative-update targets organize errors across reasoning steps. Controlled comparisons show that directly supervising intermediate local updates in the answer subspace improves accuracy, and that selecting answer-related directions can improve on full-space and rank-matched random alignment.
2. **ALUS training method.** We introduce Answer-guided Local Update Supervision (ALUS), which implements the supervision choices supported by our analysis. ALUS aligns the teacher's and student's changes between consecutive hidden states in an answer subspace constructed from output weights for the training answer vocabulary. It applies direct supervision at every valid update. Projection acts only in the alignment loss, adding no inference-time modules.
3. **Empirical evaluation across reasoning settings.** ALUS reaches $41.17 \pm 0.20\%$ on GPT-2/GSM8K, 4.88 percentage points above the strongest adapted supervision objective within the shared student. Six-update experiments support the benefits of all-valid supervision and the answer subspace; Pythia-1B/GSM8K extends the comparisons of all three supervision choices to a second backbone, while GPT-2/ProsQA supports the effectiveness of ALUS on a nonnumeric reasoning task.

## 2 Related Work

**Internalizing textual reasoning.** Explicit reasoning traces can guide the transition to implicit computation during training. Early work transfers textual CoT into an implicit student through knowledge distillation (Deng et al., 2023); stepwise internalization instead progressively removes reasoning tokens during fine-tuning (Deng et al., 2024). Coconut replaces explicit reasoning steps with continuous hidden-state feedback over a training curriculum (Hao et al., 2025). RiM similarly uses reasoning-step prediction to train fixed memory blocks before shifting to iterative answer refinement (Aichberger & Hochreiter, 2026). These curricula use textual supervision to establish internal computation before reducing dependence on intermediate text.

**Distilling teacher representations.** A key choice in representation distillation is how to summarize a longer teacher trace for the student. CODI aligns states immediately before answer generation (Shen et al., 2025), whereas CCoT predicts selected intermediate hidden states (Cheng & Van Durme, 2024) and KaVa distills compressed key–value caches (Kuzina et al., 2026). LoRi transfers aggregate trajectory structure by matching first- and second-order statistics in teacher-derived low-rank subspaces, together with an answer-boundary anchor (Solgi et al., 2026). Related feature-distillation work also considers which representation directions to retain: Flex-KD selects teacher coordinates by output sensitivity to accommodate a narrower student (Saadi & Wang, 2025). ALUS aligns individual teacher–student updates in a fixed answer subspace while retaining the student's full hidden dimension.

**Textual and semantic supervision.** Latent states can also be supervised through the reasoning content they express. SIM-CoT reconstructs the text associated with each latent step using an auxiliary decoder (Wei et al., 2026), while LOTUS supervises latent positions with reasoning-token prediction through the language model's output head (Fan et al., 2026). Semantic matching provides a related alternative to text generation: SemCoT compares implicit and textual reasoning through a contrastively trained sentence encoder (He et al., 2025), and SPOT aligns pause states with teacher spans after mapping both to probability-weighted token embeddings through frozen vocabulary parameters (Chu et al., 2026). ALUS also uses output weights to guide supervision, but derives a fixed linear subspace from training-answer tokens for matching state updates.

**Understanding supervision mechanisms.** Chen et al. (2026) compare matching latent states to mean reasoning-step embeddings with reconstructing the steps as text. Their space axis separates

geometric from generative supervision; our Space axis selects the dimension and directions of the matching subspace. They also study progressive rationale-suffix prediction and use a shared reconstruction probe on frozen states to relate recoverable reasoning information to performance. In looped language models, Sharma & Vu (2026) show how normalized readouts can leave hidden-state scale invisible to the immediate prediction loss despite supervision at every loop. We examine how alignment targets, their placement, and the supervision subspace affect learning within a shared recurrent student.

## 3 Study Design: Targets, Allocation, and Space

Our comparisons use a shared recurrent student and vary its teacher–student alignment objective. A frozen teacher provides states from explicit reasoning text, while the student advances through continuous hidden-state feedback. All implicit students in the main comparisons share two training objectives: answer prediction after recurrence and reference-text prediction in a separate teacher-forced branch of the same student. We vary the alignment term along Targets, Allocation, and Space while keeping student computation fixed (Figure 1). Section 3.4 combines these choices in ALUS and gives the complete training objective.

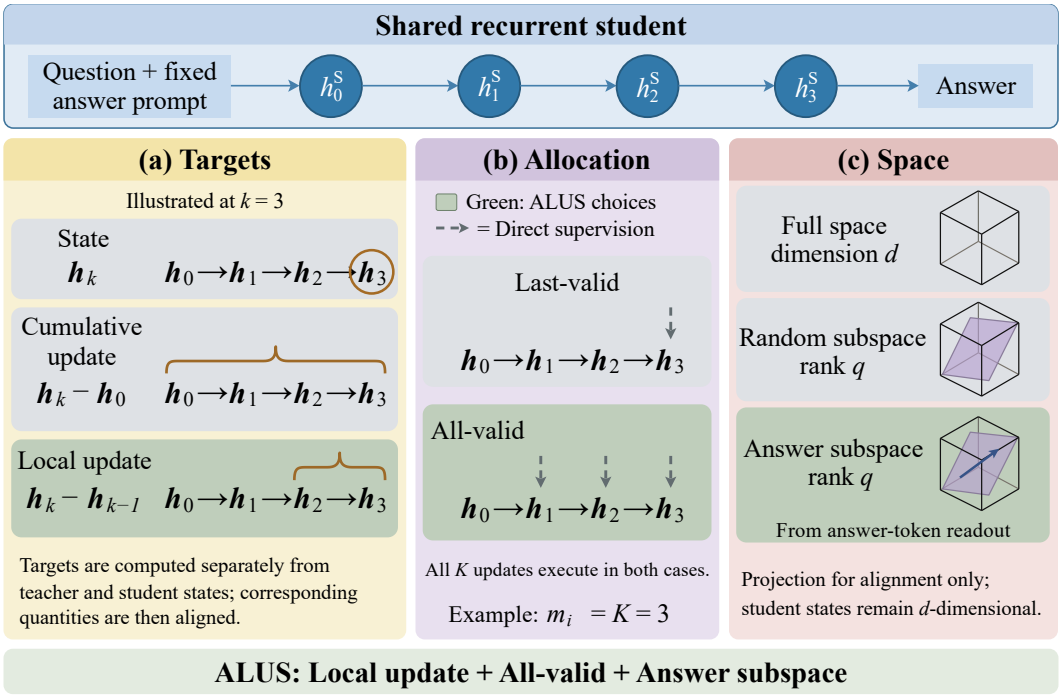



Figure 1: **Supervision choices under a shared recurrent student.** Targets selects state, cumulative-update, or local-update quantities; Allocation selects all valid updates or only the last valid update for direct supervision; Space selects the dimension and directions used for alignment. ALUS combines local-update targets, all-valid supervision, and the answer subspace. All configurations in a comparison execute the same $K$ student updates and retain recurrent backpropagation. Projection acts only in the alignment loss; student states retain their full dimension.

**Teacher states.** Each example contains a question $x$, an answer $y$, and a rationale $r$. A CoT-SFT teacher with the student's backbone is frozen during student training. We partition a retained rationale prefix into at most $K$ nonempty spans ($K = 3$ by default), giving $0 \le m_i \le K$ valid spans for example $i$. At layer $\ell$, $h^T_{0,\ell}$ is the question-prefix state and $h^T_{k,\ell}$ is the final-token state of span $k$, including preceding context. Appendix A.3 specifies prefix construction and segmentation.

**Student states.** The student reads a prefix $\bar{x}$ containing the question and a fixed answer prompt, performs $K$ continuous updates, and then generates the answer. Let $h^S_{k,\ell}$ be its final-position state at

layer $\ell$ after update $k$, $L$ the final layer, $E_\theta$ the embedding map, and $H_{\theta,\ell}$ the layer output. Then

$$\begin{aligned} h^S_{0,\ell} &= H_{\theta,\ell}(E_\theta(\bar{x}))_{\text{last}}, \\ e_k &= h^S_{k-1,L}, \qquad h^S_{k,\ell} = H_{\theta,\ell}([E_\theta(\bar{x}), e_1, \ldots, e_k])_{\text{last}}. \end{aligned} \tag{1}$$

Each update feeds the preceding final-layer state back as a continuous input, retaining the question and earlier inputs (Hao et al., 2025; Wei et al., 2026). Thus $e_k$ denotes the recurrent input, whereas $h^S_{k,\ell}$ denotes a layer output. Updates share parameters and use no transition adapter. We align the final two layers, $\mathcal{S} = \{L-1, L\}$; implementation details are in Appendix A.

**Teacher–student pairing.** We pair the state at the end of teacher span $k$ with the student state after update $k$, for $1 \le k \le m_i$. Direct alignment applies only to these valid pairs; the student always executes all $K$ updates, even when fewer teacher spans are available.

Given this correspondence, we examine three choices in alignment supervision: what quantities to match (Targets), which updates receive direct supervision (Allocation), and in which representation space to compute the alignment loss (Space). We define each choice below; Appendix B.1 provides a unified mathematical analysis.

### 3.1 Targets

We distinguish state matching from update matching, and measure updates from either the initial or the preceding state. This gives three target quantities: state targets (State), cumulative-update targets (Cumulative), and local-update targets (Local). For either teacher or student, indexed by $a \in \{T, S\}$, define

$$\begin{aligned} t^{a,\text{state}}_{k,\ell} &= h^a_{k,\ell}, \\ t^{a,\text{cum}}_{k,\ell} &= h^a_{k,\ell} - h^a_{0,\ell}, \\ t^{a,\text{local}}_{k,\ell} &= h^a_{k,\ell} - h^a_{k-1,\ell} = \Delta h^a_{k,\ell}. \end{aligned} \tag{2}$$

State targets match boundary representations. Cumulative and local targets instead match changes, both removing a trajectory-wide constant offset. Comparing these two difference targets distinguishes the effect of using state changes from that of matching individual updates. A local teacher target describes how its representation changes as it reads the next reasoning span in context.

When all valid updates are supervised in the same space, exact local matching and exact cumulative matching are equivalent: cumulative errors are prefix sums of local errors, and differences between consecutive cumulative errors recover each local error. The objectives differ in how they aggregate deviations from matching. If $\varepsilon_k = \Delta h^S_k - \Delta h^T_k$, the difference between the student's and teacher's cumulative targets at step $k$ is $\sum_{j \le k} \varepsilon_j$: early errors enter multiple terms, and errors across updates can cancel or reinforce one another. Local targets compare adjacent changes directly while retaining recurrent backpropagation. Thus equivalent matching conditions need not yield equivalent learning signals. Appendix B.2 derives this distinction; Section 4.1 evaluates how the target choices affect task performance.

### 3.2 Allocation

Even a loss applied only at the last valid update can affect earlier states through recurrent backpropagation. Allocation tests whether direct targets at intermediate updates improve learning beyond these propagated gradients. We define all-valid and last-valid supervision for a fixed choice of Targets and Space, and compare them experimentally with Local Targets. Let $\mathcal{A} = \{(i,k) : 1 \le k \le m_i\}$ be the valid pairs in a batch and $N = \sum_i m_i$. The allocation weights are

$$w^{\text{all}}_{ik} = 1, \qquad w^{\text{last}}_{ik} = m_i \, \mathbf{1}[k = m_i], \qquad 1 \le k \le m_i. \tag{3}$$

For $N > 0$, both choices satisfy $\sum_k w_{ik} = m_i$, preserving each sample's total aggregation weight $m_i/N$. Last-valid supervision retains only the pair at $m_i$ from the all-valid objective. Given a per-pair, per-layer discrepancy $d_{ik\ell}$ for the selected Targets and Space, the alignment objective is

$$\mathcal{L}_{\text{align}}(w) = \frac{1}{N|\mathcal{S}|} \sum_{(i,k)\in\mathcal{A}} w_{ik} \sum_{\ell \in \mathcal{S}} d_{ik\ell}. \tag{4}$$

For two valid spans, all-valid and last-valid weights are $(1, 1)$ and $(0, 2)$, respectively. The student executes all $K = 3$ updates in both cases. This changes which updates receive direct targets while keeping recurrent computation fixed. A batch with no valid pairs contributes no alignment loss.

Across alignment comparisons, coefficients are calibrated before training to a common initial alignment-gradient scale within each experimental setting and then held fixed. Appendix A.4 gives the loss-reduction and calibration details.

### 3.3 SPACE

The dimension and directions of the supervision space determine which teacher–student differences enter the alignment loss. For any choice of Targets, let $z^{S/T}_{ik\ell} = R^\top t^{S/T}_{ik\ell}$ be the projected targets. Full-space alignment uses $R = I_d$; subspace alignment uses an orthonormal basis $R \in \mathbb{R}^{d\times q}$, where $d$ is the hidden-state dimension and $q$ is the subspace dimension. Student states retain their full dimension throughout training and inference; projection is used only to compute the alignment loss.

**Supervision spaces.** We compare full-space alignment, a rank-$q$ random subspace, and a rank-$q$ answer subspace derived from training-answer output-weight rows. An additional teacher-PCA comparison uses $q$ principal directions of teacher state changes. Rank-matched comparisons examine direction selection, while full-space comparisons change both dimension and directions.

**Answer subspace construction.** Let $W^0$ be the base-model output weights before student training. Under this fixed linear readout, a hidden-state change $\Delta h$ changes token $v$'s score by $W^0[v, :]\Delta h$, motivating supervision along directions that affect answer-token scores. We select the rows $W_{\text{ans}}$ corresponding to the aggregate training-answer vocabulary and subtract their mean row vector $\mu$. Centering removes the common mean of answer-token scores while preserving their pairwise differences. We then compute

$$A = \widetilde{W}_{\text{ans}} = W_{\text{ans}} - \mathbf{1}\mu^\top = U\Sigma V^\top, \qquad B = V_{:,1:q}, \quad q = 32. \tag{5}$$

The leading right singular vectors retain directions along which this fixed, centered readout is most sensitive to hidden-state changes. They define the answer-token subspace, or answer subspace for short, with $R = B$. Constructed once from the aggregate training-answer vocabulary, the basis is frozen and shared by all examples. Appendix A.3 details vocabulary construction, and Appendix B.4 derives the approximation property for this fixed readout. Section 4.3 tests how the choice of supervision space affects reasoning accuracy.

### 3.4 ALUS: ANSWER-GUIDED LOCAL UPDATE SUPERVISION

ALUS aligns each valid student update with the corresponding teacher update in the answer subspace. It uses local-update targets, all-valid Allocation ($w = w^{\text{all}}$), and the answer subspace ($R = B$). We compare the direction and magnitude of these projected updates and train the student jointly with the shared answer and reference-text objectives. Figure 2 shows the alignment and inference procedures.

**Direction and magnitude.** Matching direction alone leaves update magnitude unspecified. ALUS therefore uses cosine alignment for projected update directions and a log-norm loss for relative magnitudes; $\epsilon$ stabilizes the logarithms near zero. For each valid target pair and layer, let $z^{S/T} = B^\top \Delta h^{S/T}$, suppressing indices for clarity. We define

$$\begin{aligned} d_{\text{dir}}(z^S, z^T) &= 1 - \cos(z^S, z^T), \\ d_{\text{norm}}(z^S, z^T) &= \text{SmoothL1}\big(\log(\|z^S\| + \epsilon), \log(\|z^T\| + \epsilon)\big)\,. \end{aligned} \tag{6}$$

The main Targets, Allocation, and Space comparisons apply these discrepancies to the selected targets in their respective supervision spaces and aggregate them with the corresponding allocation weights. For ALUS, averaging each discrepancy over valid pairs and supervised layers, using Equation (4) with $w = w^{\text{all}}$, gives $\mathcal{L}_{\text{dir}}$ and $\mathcal{L}_{\text{norm}}$. Appendix B.5 relates these terms to projected error and a fixed-readout bound.

**Joint training.** The alignment terms supplement answer cross-entropy (CE) $\mathcal{L}_{\text{ans}}$ after recurrence and reference-text cross-entropy $\mathcal{L}_{\text{ref}}$ on rationale tokens. The reference-text branch uses teacher

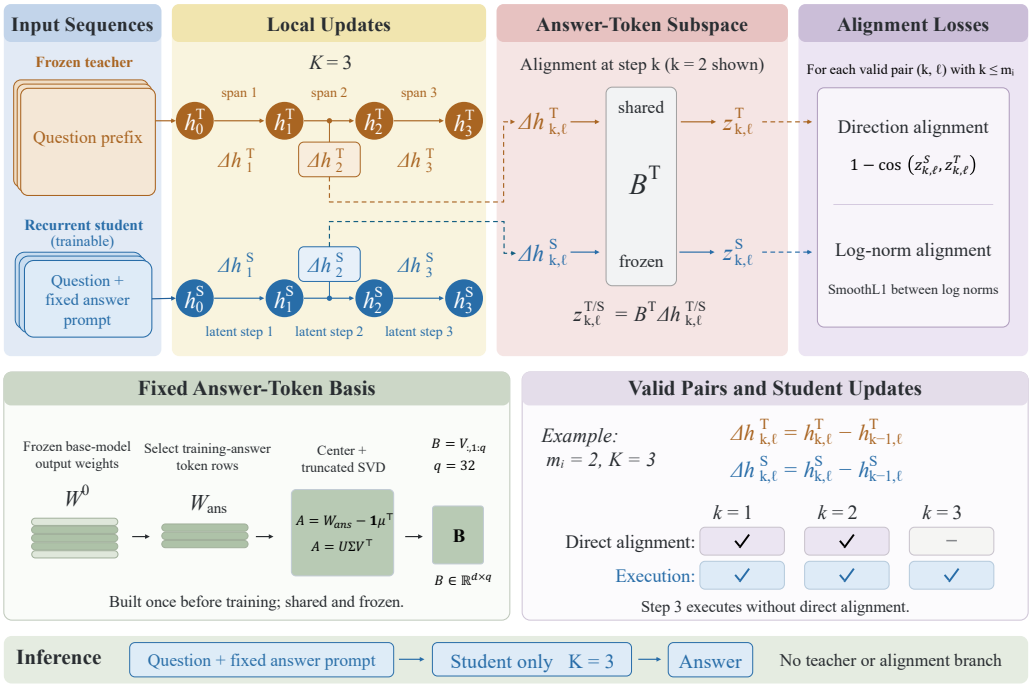



Figure 2: **ALUS training and inference.** Local updates from the frozen teacher and student are projected through the same fixed answer basis $B$, constructed before training from centered training-answer output-weight rows. Direction and log-norm losses align every valid pair ($1 \le k \le m_i$) at layers $L-1$ and $L$; the detail illustrates $k = 2$. In the separate validity example, $m_i = 2$ and $K = 3$: only the first two updates receive direct alignment, while all three execute. Training also includes answer CE and reference-text CE in a separate branch of the same student. Inference retains only question encoding, $K$ student updates, and answer decoding.

forcing and updates the same student parameters. The complete training objective is

$$\mathcal{L} = \mathcal{L}_{\text{ans}} + \mathcal{L}_{\text{ref}} + \lambda_{\text{dir}}\mathcal{L}_{\text{dir}} + \lambda_{\text{norm}}\mathcal{L}_{\text{norm}}. \tag{7}$$

Loss coefficients and numerical settings are specified in Appendix A.4.

**Inference.** Only question encoding, $K$ student updates, and answer decoding are retained. The teacher, reference-text branch, and alignment projection are used only during training.

## 4 Experiments: Targets, Allocation, and Space

We examine how Targets, Allocation, and Space affect reasoning accuracy within a shared student, then compare ALUS with other supervision objectives. Further experiments extend the analysis to longer update sequences, a second backbone, and nonnumeric answers.

**Experimental setup.** The main comparisons use GPT-2 (124M), rank-128 LoRA, and $K = 3$ updates, trained on GSM8K-Aug and evaluated on the 1,319 GSM8K test questions with greedy decoding (Radford et al., 2019; Hu et al., 2022; Deng et al., 2023; Cobbe et al., 2021). We report means and sample standard deviations (SD) over three student seeds; accuracy differences are paired by seed and reported in percentage points (pp). Sections 4.1–4.3 use ALUS—Local Targets, all-valid Allocation, and the answer subspace—as a common reference. Conditions shared across Tables 1–4 reuse the same results. Training, loss calibration, and evaluation details are given in Appendix A; settings for the additional experiments are given in Appendix D.

### 4.1 TARGETS: STATES, CUMULATIVE UPDATES, AND LOCAL UPDATES

We first compare State, Cumulative, and Local Targets with all-valid Allocation and the answer subspace fixed. Cumulative and Local share offset invariance and exact matching conditions in this setting, but the former compares changes accumulated from the initial state, while the latter compares changes between consecutive states (Section 3.1).

Table 1: **Targets on GPT-2/GSM8K.** Allocation is all-valid and Space is the answer subspace. Accuracy is mean ± sample SD over three seeds. Offset invariance refers to $h_k \mapsto h_k + b$.

| Targets | Target quantity | Offset invariant | Accuracy (%) |
|---|---|---|---|
| State | $h_k$ | No | $38.97 \pm 0.67$ |
| Cumulative update | $h_k - h_0$ | Yes | $39.17 \pm 0.88$ |
| **Local update (ALUS)** | $h_k - h_{k-1}$ | Yes | $\mathbf{41.17} \pm 0.20$ |

Local-update targets achieve $41.17 \pm 0.20\%$, exceeding State by 2.20 pp and Cumulative by 2.00 pp, with positive paired gains in all three seeds (Table 1). The advantage over Cumulative indicates that how a supervision objective aggregates errors across reasoning steps can affect the final reasoning accuracy.

Local-update supervision also pairs each teacher target with a specific student update. We test this pairing by cyclically shifting the targets among valid positions, keeping the teacher-update vectors unchanged and applying the same initial-gradient calibration procedure. Accuracy drops by 7.38 pp to $33.79 \pm 0.64\%$. In this setting, pairing teacher and student updates in their original step order improves the effectiveness of local-update supervision. Table C1 gives the per-seed results.

### 4.2 ALLOCATION: DIRECT SUPERVISION ACROSS UPDATES

We test whether supervising all valid updates improves on supervising only the last valid update. With Local Targets fixed, we compare these two Allocation choices in the answer subspace and a rank-matched random subspace. Both retain the full sequence of student updates; allocation weights are defined in Section 3.2.

Table 2: **Allocation in two supervision spaces on GPT-2/GSM8K.** Targets are local updates. Accuracy is mean ± sample SD over three seeds. Gains are seed-paired differences computed before rounding.

| Allocation | Accuracy (%) | |
|---|---|---|
| | **Answer subspace** | **Rank-matched random** |
| Last-valid | $33.03 \pm 1.41$ | $34.02 \pm 0.46$ |
| **All-valid** | $\mathbf{41.17} \pm 0.20$ | $35.28 \pm 0.95$ |
| Gain (pp): all-valid − last-valid | $+8.14$ | $+1.26$ |

Compared with last-valid supervision, all-valid supervision improves accuracy by 8.14 pp in the answer subspace and 1.26 pp in the random subspace (Table 2). Providing direct teacher targets to intermediate updates improves reasoning accuracy even though earlier updates already receive gradients through recurrence. This gain is larger in the answer subspace. Appendix C.2 provides per-seed results.

### 4.3 SPACE: DIMENSION AND DIRECTION SELECTION

We examine whether supervision needs the full hidden space and which directions to use at a fixed dimension. With Local Targets and all-valid Allocation fixed, we compare the full hidden space, a rank-32 random subspace, and the rank-32 answer subspace.

The answer subspace improves accuracy over full-space alignment by 7.23 pp (Table 3). Thus, concentrating supervision on answer-related directions is more effective than supervising the full

Table 3: **Space on GPT-2/GSM8K.** Targets are local updates and Allocation is all-valid. All conditions use the direction/log-norm objective and valid-pair aggregation. Accuracy is mean ± sample SD over three seeds.

| Space | Dimension | Accuracy (%) |
|---|---|---|
| Full space | $d$ | $33.94 \pm 1.21$ |
| Random subspace | 32 | $35.28 \pm 0.95$ |
| Teacher PCA subspace | 32 | $34.39 \pm 0.38$ |
| **Answer subspace (ALUS)** | 32 | $\mathbf{41.17} \pm 0.20$ |

hidden space in this setting. At the same dimension of 32, the answer subspace also outperforms the random subspace by 5.89 pp. This rank-matched comparison shows that the choice of directions matters beyond the number of supervised dimensions.

Teacher PCA retains directions of greatest variance in teacher updates. With the same rank, direction/log-norm objective, and valid-pair aggregation as Table 3, it achieves $34.39 \pm 0.38\%$. A rank-32 basis from frequency-matched non-answer vocabulary reaches $37.53 \pm 0.89\%$ (Table C3). The answer subspace outperforms both alternatives, supporting its use over teacher-variance-based and non-answer vocabulary projections in this setting.

### 4.4 ALUS and Other Supervision Objectives

We compare ALUS with no alignment and adapted supervision objectives in the same three-update student. No alignment retains answer and reference-text CE and removes only the alignment terms. All alignment objectives use the calibration in Appendix A.4; Appendix A.5 specifies their adaptations to this student.

Table 4: **Adapted supervision objectives on GPT-2/GSM8K.** Implicit conditions share the three-update student, training data and number of training updates, and answer/reference-text CE. Each adapted objective is implemented in this student (Appendix A.5); explicit CoT is a separate reference. Accuracy is mean ± sample SD over three seeds.

| Method | Supervision signal | Accuracy (%) |
|---|---|---|
| Explicit CoT | Rationale and answer tokens | $46.25 \pm 0.86$ |
| No alignment | Answer and reference-text CE only | $32.85 \pm 1.36$ |
| CODI adaptation | Pre-answer hidden state | $34.52 \pm 2.85$ |
| LoRi adaptation | Trajectory statistics and answer anchor | $34.62 \pm 0.70$ |
| SIM-CoT adaptation | Step-level text reconstruction | $36.29 \pm 0.92$ |
| SPOT adaptation | State-to-span alignment | $34.12 \pm 1.22$ |
| KaVa adaptation | Compressed teacher key/value cache | $33.94 \pm 1.62$ |
| **ALUS** | Local updates in the answer subspace | $\mathbf{41.17} \pm 0.20$ |

ALUS achieves $41.17 \pm 0.20\%$, the highest mean accuracy among the implicit conditions in Table 4. It outperforms no alignment by 8.31 pp and the strongest adapted objective, SIM-CoT, by 4.88 pp.

**Loss components.** Direction-only alignment reaches $39.83 \pm 1.16\%$ and log-norm-only alignment $33.26 \pm 1.35\%$, compared with $41.17 \pm 0.20\%$ for their combination in ALUS (Table C5). The combined objective has the highest accuracy in all three seeds. Without reference-text CE, ALUS alignment still improves accuracy in all three seeds, by 9.02 pp on average (Appendix C.5).

### 4.5 Different Reasoning Lengths, Backbones, and Tasks

We extend the three-update GPT-2/GSM8K study to longer update sequences, a second backbone, and nonnumeric reasoning. Table 5 summarizes gains over no alignment.

Table 5: **ALUS versus no alignment across reasoning settings.** Accuracy is averaged over three seeds; gains are computed before rounding. Each pair shares the student, training data and number of training updates, and answer/reference-text CE. Appendix D provides complete per-seed results.

| Backbone | Task | $K$ | Mean accuracy (%) | | Gain (pp) |
|---|---|---|---|---|---|
| | | | No alignment | **ALUS** | |
| GPT-2 | GSM8K | 6 | 32.04 | **41.55** | $+9.50$ |
| Pythia-1B | GSM8K | 3 | 22.95 | **26.54** | $+3.59$ |
| GPT-2 | ProsQA | 3 | 81.47 | **89.53** | $+8.07$ |

**Six-update comparison.** With $K = 6$ student updates and up to six teacher spans, Local, State, and Cumulative targets reach $41.55 \pm 1.63\%$, $41.98 \pm 1.49\%$, and $41.70 \pm 0.30\%$, respectively. Their mean accuracies are close. With Local Targets, all-valid supervision adds $9.45 \pm 1.80$ pp in the answer subspace, while the answer subspace adds $7.81 \pm 3.21$ pp over rank-matched random alignment under all-valid supervision. Both gains are positive in all three seeds. ALUS exceeds no alignment by $9.50 \pm 1.71$ pp. Appendix D.1 gives all seven conditions and paired comparisons.

**A second backbone.** On GSM8K with Pythia-1B (Biderman et al., 2023), all four Targets, Allocation, and Space contrasts in Table 6 are positive in all three seeds. ALUS also improves over no alignment by 3.59 pp (Table 5). Appendix D.2 provides the complete per-seed results.

Table 6: **Accuracy gains from supervision choices on GSM8K across backbones at $K = 3$.** Entries are mean $\pm$ sample SD of seed-paired accuracy differences over three seeds. Targets comparisons fix all-valid Allocation and the answer subspace; Allocation fixes local-update Targets and the answer subspace; Space fixes local-update Targets and all-valid Allocation.

| Choice | Contrast | Accuracy gain (pp) | |
|---|---|---|---|
| | | **GPT-2** | **Pythia-1B** |
| Targets | Local $-$ State | $2.20 \pm 0.61$ | $2.22 \pm 0.19$ |
| | Local $-$ Cumulative | $2.00 \pm 0.83$ | $1.97 \pm 1.62$ |
| Allocation | All-valid $-$ last-valid | $8.14 \pm 1.44$ | $1.47 \pm 0.44$ |
| Space | Answer $-$ random | $5.89 \pm 1.01$ | $2.25 \pm 0.27$ |

**Nonnumeric reasoning.** ProsQA (Hao et al., 2025) tests reasoning about an entity's membership in named concepts. We construct the answer subspace from concept-answer tokens in the training set and retain the three-update GPT-2 student. ALUS has the highest mean accuracy among the seven supervision conditions and improves over no alignment by $8.07 \pm 3.30$ pp, with positive gains in all three seeds (Table 5). Appendix D.3 provides the protocol and all per-seed comparisons.

**Numeric transfer.** Without further training, ALUS improves over no alignment by 7.87 pp on SVAMP, 6.88 pp on ASDiv, and 22.03 pp on MultiArith. These evaluations use examples with one normalizable numeric answer; Appendix D.4 reports the subsets and results.

## 5 Discussion and Conclusion

We study Targets, Allocation, and Space within a shared recurrent student. Local and Cumulative targets can share exact matching conditions yet aggregate errors differently, and Local targets perform better in the three-update GSM8K comparisons. With Local Targets, directly supervising intermediate updates improves accuracy in the answer subspace. Rank-matched comparisons further show that selecting supervision directions matters beyond dimensionality reduction.

ALUS combines local-update targets, all-valid supervision, and an answer subspace without adding inference-time modules. Longer update sequences, a second backbone, and nonnumeric reasoning extend the evidence beyond the main GPT-2/GSM8K setting. Together, these results establish supervision design as a practical way to improve implicit reasoning while retaining the same student architecture and recurrent computation within each comparison.

### Reproducibility Statement

Sections 3–4 specify computation, training, and selection. Appendices provide per-seed results, paired controls, numeric transfer, timing, and mathematical proofs. Appendix A documents text construction, loss reduction, calibration, and scoring.

### Statement on AI Use

OpenAI Codex was used for literature review and retrieval, analysis of experimental results, manuscript editing, and English–Chinese translation. The authors are responsible for verifying references, interpreting experimental results, and the final content of the manuscript.

## References

Lukas Aichberger and Sepp Hochreiter. Unlocking the working memory of large language models for latent reasoning. arXiv preprint, 2026. URL https://arxiv.org/abs/2605.30343.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, Usvsn Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar Van Der Wal. Pythia: A suite for analyzing large language models across training and scaling. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 2397–2430. PMLR, 2023. URL https://proceedings.mlr.press/v202/biderman23a.html.

Zefan Cai, Wen Xiao, Hanshi Sun, Cheng Luo, Yikai Zhang, Ke Wan, Yucheng Li, Yeyang Zhou, Li-Wen Chang, Jiuxiang Gu, Zhen Dong, Animashree Anandkumar, Abedelkadir Asi, and Junjie Hu. R-KV: Redundancy-aware KV cache compression for reasoning models. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 60980–61005. Curran Associates, Inc., 2025. doi: 10.52202/085713-2038. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/57e0358aed314784cad14a26e1eba642-Paper-Conference.pdf.

Xinghao Chen, Chak Tou Leong, Wenjin Guo, Jian Wang, Wenjie Li, and Xiaoyu Shen. What makes effective supervision in latent chain-of-thought: An information-theoretic analysis. arXiv preprint, 2026. URL https://arxiv.org/abs/2606.20075.

Jeffrey Cheng and Benjamin Van Durme. Compressed chain of thought: Efficient reasoning through dense representations. arXiv preprint, 2024. URL https://arxiv.org/abs/2412.13171.

Yunlong Chu, Minglai Shao, Yuhang Liu, Bing Hao, Yumeng Lin, Jialu Wang, and Ruijie Wang. SPOT: Span-level pause-of-thought for efficient and interpretable latent reasoning in large language models. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, KDD '26, pp. 675–686. Association for Computing Machinery, 2026. doi: 10.1145/3770855.3818055. URL https://doi.org/10.1145/3770855.3818055.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. arXiv preprint, 2021. URL https://arxiv.org/abs/2110.14168.

Yuntian Deng, Kiran Prasad, Roland Fernandez, Paul Smolensky, Vishrav Chaudhary, and Stuart Shieber. Implicit chain of thought reasoning via knowledge distillation. arXiv preprint, 2023. URL https://arxiv.org/abs/2311.01460.

Yuntian Deng, Yejin Choi, and Stuart Shieber. From explicit CoT to implicit CoT: Learning to internalize CoT step by step. arXiv preprint, 2024. URL https://arxiv.org/abs/2405.14838.

Ying Fan, Anej Svete, and Kangwook Lee. Bridging the gap between latent and explicit reasoning with looped transformers. arXiv preprint, 2026. URL https://arxiv.org/abs/2606.31779.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. Training large language models to reason in a continuous latent space. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=Itxz7S4Ip3.

Yinhan He, Wendy Zheng, Yaochen Zhu, Zaiyi Zheng, Lin Su, Sriram Vasudevan, Qi Guo, Liangjie Hong, and Jundong Li. SemCoT: Accelerating chain-of-thought reasoning through semantically-aligned implicit tokens. In *Advances in Neural Information Processing Systems*, 2025. URL https://papers.neurips.cc/paper_files/paper/2025/hash/3ddbd473456a57e3cafb1ee51ddf8ff6-Abstract-Conference.html.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2106.09685.

Anna Kuzina, Maciej Pióro, and Babak Ehteshami Bejnordi. KaVa: Latent reasoning via compressed KV-cache distillation. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/d2ca35069eb6e9cbd2a37bf90ba9091c-Paper-Conference.pdf.

Shen-yun Miao, Chao-Chun Liang, and Keh-Yih Su. A diverse corpus for evaluating and developing English math word problem solvers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 975–984, 2020. doi: 10.18653/v1/2020.acl-main.92.

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. doi: 10.18653/v1/2021.naacl-main.168.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. Technical report, OpenAI, 2019. URL https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

Subhro Roy and Dan Roth. Solving general arithmetic word problems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1743–1752, 2015. doi: 10.18653/v1/D15-1202.

Khouloud Saadi and Di Wang. What should feature distillation transfer in LLMs? A task-tangent geometry view. arXiv preprint, 2025. URL https://arxiv.org/abs/2507.10155.

Rituraj Sharma and Tu Vu. Dense supervision is not enough: The readout blind spot in looped language models. arXiv preprint, 2026. URL https://arxiv.org/abs/2606.24898.

Zhenyi Shen, Hanqi Yan, Linhai Zhang, Zhanghao Hu, Yali Du, and Yulan He. CODI: Compressing chain-of-thought into continuous space via self-distillation. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 677–693, Suzhou, China, November 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.emnlp-main.36. URL https://aclanthology.org/2025.emnlp-main.36/.

Ryan Solgi, Jiayi Tian, and Zheng Zhang. LoRi: Low-rank distillation for implicit reasoning. arXiv preprint, 2026. URL https://arxiv.org/abs/2606.05315.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, 2022. URL https://papers.nips.cc/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html.

Xilin Wei, Xiaoran Liu, Yuhang Zang, Xiaoyi Dong, Yuhang Cao, Jiaqi Wang, Xipeng Qiu, and Dahua Lin. SIM-CoT: Supervised implicit chain-of-thought. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/5d087955ee13fe9a7402eedec879b9c3-Paper-Conference.pdf.

## APPENDIX OVERVIEW

The appendices provide implementation details, the mathematical analysis supporting Section 3, and complete tables for the main supervision comparisons. Supplementary evaluations follow.

- Appendix A: experimental settings, teacher targets, and objective adaptations.
- Appendix B: matching conditions, error aggregation, and supervision-space analysis.
- Appendix C: GPT-2 results organized by Targets, Allocation, Space, and ALUS comparisons.
- Appendix D: longer reasoning, a second backbone, ProsQA, numeric transfer, and inference cost.

## A EXPERIMENTAL AND IMPLEMENTATION DETAILS

### A.1 DATA AND EVALUATION

GSM8K-Aug contains 385,620 records. After trimming question text, we group identical questions, remove two conflicting-answer records, and split with seed 11 into 347,056 training and 38,562 validation records. This groups question text rather than original pre-augmentation source questions. We select checkpoints by numeric accuracy on a fixed 2,048-example validation subset. Final evaluation uses all 1,319 GSM8K test questions, raw prompts, greedy decoding, a 128-token cap, and normalized numeric exact match. GPT-2 validation uses a 16-token cap; Pythia-1B uses 128.

The validation-based checkpoint rule and held-out test evaluation apply throughout unless a task-specific protocol is stated. No-alignment students retain answer and reference-text CE and set both alignment weights to zero.

### A.2 STUDENT TRAINING

Table A1 summarizes GSM8K training. GPT-2 applies LoRA to attention and feed-forward projections. Each backbone has its own frozen CoT-SFT teacher and answer basis; paired conditions share initialization and teacher. Dropout masks are reused along recurrent trajectories. Reference-text CE uses text with the final reasoning step removed. Student training uses seeds 11–13. Pythia-1B uses LoRA scaling coefficient 128 and the same continuous-input rule as GPT-2, with the first input given by the final-layer prefix state.

Table A1: **Student training settings.** GSM8K defaults use three updates; the six-update and ProsQA settings are detailed in Appendix D.

| Setting | GPT-2 | Pythia-1B |
|---|---|---|
| Training epochs | 40 | 40 |
| Effective batch size | 256 | 256 |
| LoRA rank | 128 | 64 |
| Backbone/LoRA learning rate | $3 \times 10^{-3}$ | $2 \times 10^{-4}$ |
| Default updates | $K = 3$ | $K = 3$ |
| Supervised layers | $L-1, L$ | $L-1, L$ |
| Answer-basis rank | 32 | 32 |
| Student seeds | 11–13 | 11–13 |

Student continuous inputs use consecutive position IDs after the unpadded prefix: for prefix length $n$, $e_k$ occupies zero-based position $n + k - 1$. Position IDs count valid tokens, so left padding does not shift them.

### A.3 TEACHER PREFIXES AND BASIS CONSTRUCTION

Teacher rationales are truncated immediately before the earliest recognized numeric surface form of the training answer, preferring longer matches at tied locations. Integer forms include plain,

comma-separated, and limited decimal variants (e.g., 1200, 1,200, 1200.0, 1200.00); nonintegers use floating-point string forms. Adjacent-digit boundaries exclude 12 within 120 but can match 12 at the start of 12.5. If no match is found, we retain the rationale; questions and answer labels remain unchanged. This is a literal-matching rule, so semantic clues may remain.

Truncation precedes segmentation. When at least $K$ steps remain, we merge them evenly into $K$ spans; shorter rationales are divided by whitespace tokens. Empty trailing spans reuse the preceding boundary and are masked out. At $K = 3$, the numbers of training examples with 0/1/2/3 valid spans are 78/73,259/126,252/147,467; empty-prefix examples still contribute both CE objectives. The same retained prefixes supply SIM-CoT reconstruction labels and teacher PCA.

Tokenizing training answers both with and without a leading space yields 1,598 GPT-2 tokens and 1,836 Pythia-1B tokens. We center the corresponding output-weight rows at initialization and take the leading 32 right singular vectors. The basis is frozen and shared across examples, using training-answer vocabulary only.

### A.4 Loss Reduction and Calibration

ALUS, the Targets comparisons, the loss-component ablations, and the Allocation–Space comparisons with local-update targets average losses over layers and valid sample/span pairs (`active_slot_mean`). Thus samples with more valid spans have greater alignment weight. Empty-prefix samples contribute no alignment. Pair averaging is performed within each device microbatch before gradient accumulation and synchronization.

The main ALUS setting uses direction/log-norm coefficients $(4.0, 0.8)$, log stabilizer $10^{-6}$, cosine norm floor $10^{-8}$, and SmoothL1 threshold 1. A batch without valid pairs has zero alignment loss. Coefficients are development settings; test scores select neither coefficients nor checkpoints.

For GPT-2 comparisons, alignment gradients are measured before training on 2,048 fixed training examples: eight batches of 256, each emulated by four microbatches of 64 with distributed averaging. Each condition is rescaled by $g_{\mathrm{ref}}/g_c$, where $g_c$ is its median alignment-gradient norm, and the coefficients are then frozen. The Local-Targets Allocation–Space comparison uses $g_{\mathrm{ref}} = 0.3166423187$. Table A2 lists the fixed coefficients for both backbones. Answer and reference-text CE retain weight 1.

Table A2: **Alignment coefficients for local-update targets.** Each pair is shared across seeds 11–13 and fixed during training. The columns give direction and log-norm weights; their ratio is $5 : 1$.

| **Backbone / Allocation / Space** | $\lambda_{\mathrm{dir}}$ | $\lambda_{\mathrm{norm}}$ |
|---|---|---|
| GPT-2 / all-valid / answer | 4.0 | 0.8 |
| GPT-2 / all-valid / random | 3.369537 | $3.369537/5$ |
| GPT-2 / last-valid / answer | 2.283545 | $2.283545/5$ |
| GPT-2 / last-valid / random | 1.239005 | $1.239005/5$ |
| Pythia-1B / all-valid / answer | 4.0 | 0.8 |
| Pythia-1B / all-valid / random | 1.986049 | 0.397210 |
| Pythia-1B / last-valid / answer | 1.687351 | 0.337470 |
| Pythia-1B / last-valid / random | 2.238675 | 0.447735 |

For the Pythia-1B Targets comparison, Local retains coefficients $(4.0, 0.8)$; State and Cumulative are calibrated separately for each seed against the Local initial alignment-gradient norm, with coefficients fixed thereafter.

Initial-gradient calibration applies to all reported alignment comparisons, including cyclic-order and vocabulary-source controls and adapted supervision objectives. Each objective retains its specified loss form and uses coefficients fixed after calibration. The factorials also preserve each sample's aggregation weight across Allocation choices. No-alignment conditions omit the alignment objective.

### A.5 Baseline Adaptations

All adaptations retain the common recurrent student, frozen teacher, training data and number of training updates, reference-text CE, and checkpoint-selection rule. They compare supervision objectives within the shared student; the training FLOPs need not be equal.

**CODI.** We use teacher distillation at a single pre-answer boundary.

**LoRi.** We match trajectory means, covariance, and an end anchor in a common rank-32 teacher PCA space in place of the full Tucker decomposition.

**SIM-CoT.** We reconstruct retained spans with a separate 124,439,808-parameter GPT-2 decoder, initialized from the same base model with tied embeddings and no decoder LoRA. Its learning rate is $10^{-4}$, selected on seed 11 from $\{10^{-4}, 3 \times 10^{-4}, 3 \times 10^{-3}\}$ by validation accuracy among stable candidates, then fixed across seeds. Stability requires finite losses/gradients and final-five-epoch mean validation accuracy at least 80% of the run's best. The decoder is removed at inference.

**SPOT.** We map each student state and its corresponding teacher-span state set to probability-weighted embeddings using frozen vocabulary parameters. Uniform teacher mass and one student state per span reduce transport to mean normalized squared distance. The base alignment weight is 1 before initial-gradient calibration (Appendix A.4). Entropy regularization is 0.0025, and each teacher span retains at most 256 tokens. The adaptation omits SPOT's hybrid text/pause architecture, SpanDrop, and rejection-sampled fine-tuning.

**KaVa.** We add KV SmoothL1 with a base weight of 1 before initial-gradient calibration (Appendix A.4). We use R-KV selection (Cai et al., 2025) as adapted for distillation by Kuzina et al. (2026): each layer/head selects up to three teacher-prefix positions using 0.1 attention importance plus 0.9 key diversity, sorts them, and matches student slots. Targets exclude question/answer/padding positions, while importance uses training-answer queries including EOS. Loss averages valid elements, keys/values, and the final two layers without standard-deviation normalization. Empty prefixes contribute no KV loss. The adaptation omits Jacobi decoding, projection modules, and shared-model self-distillation.

## B Mathematical Analysis

We analyze target matching and error aggregation, the answer subspace under a fixed readout, and the losses used in training. Squared error provides an analytic reference; the final subsection connects the direction/log-norm objective to projected error.

### B.1 A Unified Representation of Supervision

We use a squared-error form to analyze the roles of the three supervision choices. For one sample and layer, write $m = m_i$ and suppress the sample and layer indices. Stack the teacher or student boundary states as rows of $H^{T/S} \in \mathbb{R}^{(m+1)\times d}$. Let $D_{\text{state}}$, $D_{\text{cum}}$, and $D_{\text{local}}$ map this stack to rows $h_k$, $h_k - h_0$, and $h_k - h_{k-1}$, respectively. With allocation weights $W = \text{diag}(w_1, \ldots, w_m)$ and a supervision basis $R$, the squared alignment loss is

$$\mathcal{L}_2(D, W, R) = \left\| W^{1/2} \left[ DH^S - DH^T \right] R \right\|_F^2 . \tag{8}$$

Targets selects the quantities compared through the temporal operator $D$, Allocation weights their positions through $W$, and Space selects dimension and representation directions through $R$. This expression separates the temporal transformation, allocation weights, and representation directions for analysis. The training objectives in Section 3.4 use direction and log-norm discrepancies in place of squared error.

### B.2 Equivalent Matching Conditions and Different Error Aggregation

Local and cumulative targets can specify the same exact match while assigning different penalties to deviations from it. The following expressions make this distinction explicit when all valid updates use the same supervision space. Let $R$ be the common orthonormal supervision basis,

$u_k = R^\top \Delta h_k^S$, $v_k = R^\top \Delta h_k^T$, and row $k$ of $E$ be $(u_k - v_k)^\top$. Cumulative targets are $\sum_{j \le k} u_j$ and $\sum_{j \le k} v_j$. Under squared Euclidean error,

$$L_{\text{loc}} = \|E\|_F^2, \qquad L_{\text{cum}} = \|CE\|_F^2 = \operatorname{tr}(E^\top G E), \qquad G_{ij} = m - \max(i,j) + 1, \tag{9}$$

where $C_{ij} = \mathbf{1}[j \le i]$ and $G = C^\top C$. Entry $G_{ij}$ counts cumulative targets that contain both errors. The local quadratic form is the identity. Gradients with respect to independent error coordinates are $2E$ and $2GE$. A scalar coefficient rescales but cannot remove off-diagonal coupling, so initial-gradient norm matching does not equate error weights or directions. Parameter gradients additionally depend on the state-to-parameter Jacobians.

These equations describe squared error. For an actual per-pair loss $\ell$, the local objective is $\sum_k \ell(u_k, v_k)$ and the cumulative objective is $\sum_k \ell(\sum_{j \le k} u_j, \sum_{j \le k} v_j)$. Where differentiable, the derivative with respect to independent $u_j$ in the cumulative objective receives contributions from all $k \ge j$, while the local objective has only its direct $j$th term. Recurrent states still depend on previous states, so local objectives retain cross-step backpropagation. Nonlinear direction/log-norm losses are not fully represented by the quadratic form. The empirical comparisons evaluate how these different ways of aggregating error affect reasoning accuracy.

**Equivalent exact matching, different penalties.** The cumulative map $C$ is lower triangular with unit diagonal and is invertible. Hence $CE = 0$ if and only if $E = 0$: with all valid updates and a common projection, exact cumulative and local matching impose the same condition on a pair of trajectories. This statement concerns equality of projected targets, not a zero-loss characterization of stabilized cosine losses. Away from equality, the objectives can rank errors differently. For two updates and any nonzero vector $a$, local squared error is $2\|a\|^2$ for both $E = (a, -a)^\top$ and $E = (a, a)^\top$. Their cumulative squared errors are $\|a\|^2$ and $5\|a\|^2$, respectively. Cancellation and reinforcement thus change the penalty even though no target information is lost by the invertible cumulative map. This example establishes a property of error aggregation, not which objective yields higher task accuracy.

### B.3 Proposition 1: Invariance of Projected Local Updates

This proposition identifies the trajectory changes that leave projected local targets unchanged. It describes the supervised quantities rather than arbitrary perturbations to the network. For one sample and layer, let row $k$ of $H \in \mathbb{R}^{(m+1)\times d}$ be $h_k^\top$. Row $k$ of $D \in \mathbb{R}^{m \times (m+1)}$ has $-1, 1$ at columns $k-1, k$ and zeros elsewhere. For $R^\top R = I_q$ and $P = RR^\top$, define $\mathcal{T}(H) = DHR$. For any $b \in \mathbb{R}^d$ and $N \in \mathbb{R}^{(m+1)\times d}$ satisfying $NR = 0$,

$$\mathcal{T}(H + \mathbf{1}b^\top + N) = \mathcal{T}(H). \tag{10}$$

Conversely, $\mathcal{T}(H') = \mathcal{T}(H)$ if and only if $H' - H = \mathbf{1}b^\top + N$ for some $b, N$ satisfying $NR = 0$.

**Proof.** The forward direction follows from $D\mathbf{1} = 0$ and $NR = 0$. For the converse, write $X = H' - H$. Since $DXR = 0$, every row of $XR$ equals some $c^\top$. Set $b = Rc$ and $N = X - \mathbf{1}b^\top$; then $NR = 0$. □ Cumulative Targets also cancel trajectory-wide constant offsets. Step-dependent offsets change a local target by $R^\top(b_k - b_{k-1})$; the invariance concerns supervision representations rather than arbitrary network perturbations.

### B.4 Proposition 2: Error in a Fixed Answer Readout

The answer basis retains directions of a fixed linear readout. We first bound the accumulated readout error, then state the rank-constrained approximation property of the basis. Let $A = \widetilde{W}_{\text{ans}}$, with leading right singular vectors $B$ and descending singular values $\sigma_j$. Take $\sigma_{q+1} = 0$ when $q$ covers the rank of $A$.

For one sample and layer, write $P = BB^\top$ and let $\varepsilon_k = \Delta h_k^S - \Delta h_k^T$, $\alpha_k = \|B^\top \varepsilon_k\|_2$, and $\beta_k = \|(I - P)\varepsilon_k\|_2$. Orthogonality of retained and omitted SVD directions after applying $A$ yields

$$\left\| A[(h_m^S - h_0^S) - (h_m^T - h_0^T)] \right\|_2 \le \sqrt{\sigma_1(A)^2 \Big( \sum_{k=1}^m \alpha_k \Big)^2 + \sigma_{q+1}(A)^2 \Big( \sum_{k=1}^m \beta_k \Big)^2}. \tag{11}$$

This bound retains omitted-space error and concerns a frozen linear readout, rather than full nonlinear decoding.

**Proof.** For $s = \sum_k \varepsilon_k$, SVD orthogonality gives $\|As\|^2 = \|APs\|^2 + \|A(I-P)s\|^2$. Bound these components by $\sigma_1^2\|B^\top s\|^2$ and $\sigma_{q+1}^2\|(I-P)s\|^2$, apply the triangle inequality to each sum, and use $s = (h_m^S - h_0^S) - (h_m^T - h_0^T)$. □

**Fixed-readout approximation criterion.** A hidden-state change $\Delta h$ changes the fixed linear score of token $v$ by $W^0[v,:]\Delta h$. Centering the answer-token weight rows removes the common mean of these scores and preserves their pairwise differences. With $P = BB^\top$, the truncated SVD satisfies

$$\min_{Q^2=Q=Q^\top,\,\mathrm{rank}(Q)=q} \|A(I-Q)\|_2 = \|A(I-P)\|_2 = \sigma_{q+1}(A). \tag{12}$$

This minimizes the worst omitted readout change over unit-length hidden changes for the specified frozen matrix and rank.

**Worst omitted direction.** Taking $Q = P$ gives $\|A(I-P)\|_2 = \sigma_{q+1}$, an attainable upper bound on the minimum in Equation (12). Conversely, for any admissible $Q$, if $q < \mathrm{rank}(A)$, the span of the first $q+1$ right singular vectors has a nonzero intersection with $\ker Q$. For a unit vector $v$ there, $\|A(I-Q)v\|_2 = \|Av\|_2 \geq \sigma_{q+1}$. Otherwise the lower bound is zero. Thus the minimum is $\sigma_{q+1}$ and is attained by $P$.

Centering preserves pairwise answer-token score differences. Their error is at most $\sqrt{2}$ times Equation (11); the criterion does not optimize trained-model accuracy or non-answer-token competition.

### B.5 How the Implemented Losses Bound Projected Error

To connect the analysis to ALUS training, we bound projected vector error using the direction and log-norm discrepancies, including their numerical stabilization constants. For one valid pair let $u = B^\top \Delta h^S$, $v = B^\top \Delta h^T$, $a = \|u\|_2$, $b = \|v\|_2$, and $M = \max(a,b)$. The implementation uses cosine floor $\delta = 10^{-8}$ and log stabilizer $\epsilon = 10^{-6}$. Define

$$\begin{aligned} d_\delta &= 1 - \frac{u^\top v}{\max(a,\delta)\max(b,\delta)}, \\ \eta &= |\log(a+\epsilon) - \log(b+\epsilon)|, \qquad n = \mathrm{SmoothL1}(\eta, 0). \end{aligned} \tag{13}$$

For threshold-1 SmoothL1, $\eta = \psi(n)$ with $\psi(n) = \sqrt{2n}$ if $n \leq 1/2$ and $\psi(n) = n + 1/2$ otherwise. Including zero vectors,

$$\|u - v\|_2 \leq \sqrt{(M+\epsilon)^2\psi(n)^2 + 2\max(M,\delta)^2 d_\delta}. \tag{14}$$

**Proof.** Assume without loss of generality $a \geq b$. Then $a - b = (a+\epsilon)(1 - \exp(-\eta)) \leq (M+\epsilon)\eta$. With $C = \max(a,\delta)\max(b,\delta)$,

$$\|u - v\|_2^2 = (a-b)^2 + 2(Cd_\delta + ab - C) \leq (a-b)^2 + 2Cd_\delta. \tag{15}$$

Use $C \leq \max(M,\delta)^2$. If both norms exceed $\delta$, the direction term reduces to the ordinary cosine discrepancy. □

Substitution of this bound for each $\alpha_k$ connects the implemented local losses to the fixed-readout bound in Equation (11). Dependence on projected norms, omitted-space errors, and singular-value tails remains, so small average training loss alone gives no uniform per-example guarantee.

## C GPT-2 Supervision Results

The following tables report results for student seeds 11–13. SD denotes the sample standard deviation; paired differences use the same seed. Repeated ALUS references reuse the same results. Compact condition labels use State, Cumulative, and Local for the target quantities; all-valid and last-valid for Allocation; and answer, random, and full space for Space. Captions specify any fixed choices.

### C.1 Targets and Teacher–Student Pairing

Table C1 compares target quantities with all-valid Allocation and the answer subspace. The cyclic-shift condition preserves the teacher update vectors while changing which student update each vector supervises.

Table C1: **Targets and teacher–student pairing on GPT-2/GSM8K.** Accuracy is in %, with all-valid Allocation and the answer subspace. All alignment conditions use initial-gradient calibration.

| Condition | Seed 11 | Seed 12 | Seed 13 | Mean ± SD |
|---|---|---|---|---|
| ALUS | 40.94 | 41.24 | 41.32 | $41.17 \pm 0.20$ |
| State | 38.74 | 38.44 | 39.73 | $38.97 \pm 0.67$ |
| Cumulative | 38.67 | 40.18 | 38.67 | $39.17 \pm 0.88$ |
| Cyclic teacher-target shift | 33.81 | 33.13 | 34.42 | $33.79 \pm 0.64$ |

### C.2 Direct Target Allocation

Table C2 reports correct counts for Local Targets under each Allocation and Space choice. Last-valid Allocation retains the same last valid update and each sample's total aggregation weight; the all-valid/answer cell is ALUS.

Table C2: Per-seed correct counts for the Allocation and Space comparisons with Local Targets, including the full-space control. Seed columns give correct counts out of 1,319; the final column gives mean accuracy and sample standard deviation in %.

| Space / Allocation | Correct (11) | Correct (12) | Correct (13) | Accuracy (%) |
|---|---|---|---|---|
| Answer / all-valid | 540 | 544 | 545 | $41.17 \pm 0.20$ |
| Random / all-valid | 471 | 451 | 474 | $35.28 \pm 0.95$ |
| Answer / last-valid | 434 | 455 | 418 | $33.03 \pm 1.41$ |
| Random / last-valid | 450 | 454 | 442 | $34.02 \pm 0.46$ |
| Full space / all-valid | 430 | 452 | 461 | $33.94 \pm 1.21$ |

### C.3 Supervision-Space Comparisons

**Supervision bases under the main objective.** Table C3 gives per-seed results for the teacher-PCA comparison in Table 3 and a frequency-matched non-answer vocabulary control under the full direction/log-norm objective.

Table C3: **Supervision bases under the ALUS objective.** Accuracy is in % for seeds 11–13. These controls retain local updates, all-valid supervision, direction/log-norm alignment, and valid-pair averaging. All bases have rank 32.

| Condition | Seed 11 | Seed 12 | Seed 13 | Mean ± SD |
|---|---|---|---|---|
| ALUS | 40.94 | 41.24 | 41.32 | $41.17 \pm 0.20$ |
| Teacher PCA basis | 34.80 | 34.34 | 34.04 | $34.39 \pm 0.38$ |
| Frequency-matched non-answer basis | 36.54 | 37.76 | 38.29 | $37.53 \pm 0.89$ |

### C.4 Adapted Supervision Objectives

Table C4 gives the per-seed results for the shared-student objective comparison, including direct-answer training as an additional reference. Appendix A.5 specifies each objective adaptation.

Table C4: Per-seed GSM8K accuracy (%) for the supervision objectives in Table 4, with direct-answer training as an additional reference. Means and sample SDs use seeds 11–13.

| Method | Seed 11 | Seed 12 | Seed 13 | Mean ± SD |
|---|---|---|---|---|
| Explicit CoT | 45.26 | 46.85 | 46.63 | 46.25 ±0.86 |
| No alignment | 34.34 | 31.69 | 32.52 | 32.85 ±1.36 |
| CODI adaptation | 36.32 | 36.01 | 31.24 | 34.52 ±2.85 |
| LoRi adaptation | 35.41 | 34.42 | 34.04 | 34.62 ±0.70 |
| SIM-CoT adaptation | 35.25 | 37.00 | 36.62 | 36.29 ±0.92 |
| SPOT adaptation | 33.74 | 35.48 | 33.13 | 34.12 ±1.22 |
| KaVa adaptation | 35.71 | 32.52 | 33.59 | 33.94 ±1.62 |
| ALUS | 40.94 | 41.24 | 41.32 | 41.17 ±0.20 |
| Direct answer | 19.18 | 19.26 | 18.27 | 18.90 ±0.55 |

### C.5 ALUS Loss Components

**Alignment components.** Table C5 separates the direction and log-norm terms while retaining the ALUS target, allocation, and space. Single-component losses use the calibration in Appendix A.4.

Table C5: **ALUS loss components on GPT-2/GSM8K.** Accuracy is in % for seeds 11–13. Single-component conditions use initial-gradient calibration; Targets, Allocation, and Space are unchanged.

| Condition | Seed 11 | Seed 12 | Seed 13 | Mean ± SD |
|---|---|---|---|---|
| ALUS | 40.94 | 41.24 | 41.32 | 41.17 ± 0.20 |
| Direction only | 38.82 | 39.58 | 41.09 | 39.83 ± 1.16 |
| Log-norm only | 32.68 | 32.30 | 34.80 | 33.26 ± 1.35 |

**Reference-text CE.** Table C6 crosses ALUS alignment with reference-text CE. For a fixed alignment choice, paired conditions differ only in the reference-text CE weight; all active alignment conditions use the same truncated teacher prefixes.

Table C6: Matched $2 \times 2$ auxiliary-objective ablation on GPT-2/GSM8K. All arms share the student, data, optimization, validation selection, and decoding protocol. Alignment-on uses Local Targets, all-valid Allocation, the answer subspace, and the teacher-prefix rule in Appendix A.3. Accuracy is mean ± sample SD over seeds 11–13.

| Reference-text CE | Alignment | Seed 11 | Seed 12 | Seed 13 | Mean ± SD |
|---|---|---|---|---|---|
| Off | Off | 29.19 | 27.29 | 17.13 | 24.54 ±6.48 |
| Off | On | 33.28 | 33.43 | 33.97 | 33.56 ±0.36 |
| On | Off | 34.34 | 31.69 | 32.52 | 32.85 ±1.36 |
| On | On | 40.94 | 41.24 | 41.32 | 41.17 ±0.20 |

Alignment improves accuracy by $9.02 \pm 6.84$ pp without reference-text CE and $8.31 \pm 1.54$ pp with it. Reference-text CE improves accuracy by $8.31 \pm 6.14$ pp without alignment and $7.61 \pm 0.23$ pp with it. Each of these paired gains is positive in all three seeds. The paired interaction is $-0.71 \pm 6.36$ pp.

## D Additional Evaluations

### D.1 Six-Update Evaluation

We extend GPT-2/GSM8K to $K = 6$ student updates and up to six teacher spans. The six supervision conditions and no alignment use seeds 11–13, 40 epochs, and effective batch size 256 (eight workers with 32 examples each, without gradient accumulation). Evaluation follows Appendix A.1,

with test batch size 64. This setting expands the recurrent update budget and teacher segmentation together.

Table D1 gives all per-seed accuracies. Table D2 reports the paired comparisons: all-valid Allocation in the answer subspace, answer versus random subspaces under all-valid Allocation, and ALUS versus no alignment each improve accuracy in all three seeds. The three Targets have similar mean accuracies in this setting, so the six-update results support the Allocation and Space findings rather than a uniform ordering of Targets.

Table D1: **Six-update GPT-2/GSM8K results.** The student executes six updates, with up to six valid teacher spans. Entries are test accuracies (%); the final column gives mean ± sample SD over seeds 11–13. All seven conditions retain answer and reference-text CE.

| Condition | Seed 11 | Seed 12 | Seed 13 | Mean ± SD |
|---|---|---|---|---|
| No alignment | 32.15 | 32.07 | 31.92 | $32.04 \pm 0.12$ |
| Local / all-valid / answer (ALUS) | 41.47 | 39.95 | 43.21 | $41.55 \pm 1.63$ |
| Local / last-valid / answer | 32.98 | 31.61 | 31.69 | $32.10 \pm 0.77$ |
| Local / all-valid / random | 36.09 | 33.36 | 31.77 | $33.74 \pm 2.19$ |
| Local / last-valid / random | 31.69 | 33.06 | 31.84 | $32.20 \pm 0.75$ |
| State / all-valid / answer | 41.40 | 43.67 | 40.86 | $41.98 \pm 1.49$ |
| Cumulative / all-valid / answer | 41.70 | 42.00 | 41.40 | $41.70 \pm 0.30$ |

Table D2: **Paired comparisons in the six-update setting.** Entries are mean ± sample SD of seed-paired accuracy differences (pp). Targets comparisons fix all-valid/answer; Allocation and Space comparisons fix Local Targets. The interaction is the all-minus-last gain in the answer subspace minus the corresponding gain in the random subspace.

| Contrast | Mean ± SD (pp) |
|---|---|
| Local − State | $-0.43 \pm 3.06$ |
| Local − Cumulative | $-0.15 \pm 1.93$ |
| All-valid − last-valid (answer) | $+9.45 \pm 1.80$ |
| All-valid − last-valid (random) | $+1.54 \pm 2.48$ |
| Answer − random (all-valid) | $+7.81 \pm 3.21$ |
| Answer − random (last-valid) | $-0.10 \pm 1.37$ |
| Allocation–Space interaction | $+7.91 \pm 3.75$ |
| ALUS − no alignment | $+9.50 \pm 1.71$ |

### D.2 PYTHIA-1B

Table D3 combines the Targets, Allocation–Space, and no-alignment comparisons, listing the shared ALUS condition once. Each contrast is paired by seed. Tests use batch size 16 and the numeric-answer scoring in Appendix A.1; the coefficients in Appendix A.4 remain fixed during training.

With all-valid Allocation and the answer subspace, Local exceeds State by $2.22 \pm 0.19$ pp and Cumulative by $1.97 \pm 1.62$ pp in paired differences, positive in all three seeds. ALUS reaches $26.54 \pm 0.15\%$, compared with $22.95 \pm 0.36\%$ without alignment, a 3.59-pp difference in mean accuracy. In the table, the Allocation–Space interaction is the all-minus-last difference in the answer subspace minus its counterpart in the random subspace; differences are computed before display rounding.

### D.3 PROSQA

**Data and scoring.** We use the official ProsQA train/validation/test files released with Coconut (Hao et al., 2025): 17,886/300/500 examples, without regeneration or resplitting. Each question asks whether an entity belongs to one of two named concepts. The teacher is a separately trained GPT-2 CoT model selected on validation accuracy. Test evaluation uses greedy generation capped at 32

Table D3: **Complete Pythia-1B/GSM8K supervision comparison.** Accuracy (%) and paired differences (pp) use seeds 11–13; SD is the sample standard deviation. ALUS is the shared Local/all-valid/answer reference. Every condition retains answer and reference-text CE; no alignment sets the alignment weights to zero.

| Condition | Seed 11 | Seed 12 | Seed 13 | Mean $\pm$ SD |
|---|---|---|---|---|
| Local / all-valid / answer (ALUS) | 26.54 | 26.38 | 26.69 | $26.54 \pm 0.15$ |
| State / all-valid / answer | 24.11 | 24.18 | 24.64 | $24.31 \pm 0.29$ |
| Cumulative / all-valid / answer | 23.28 | 26.23 | 24.18 | $24.56 \pm 1.51$ |
| Local / all-valid / random | 24.26 | 23.88 | 24.72 | $24.29 \pm 0.42$ |
| Local / last-valid / answer | 24.56 | 25.17 | 25.47 | $25.07 \pm 0.46$ |
| Local / last-valid / random | 25.09 | 23.96 | 24.79 | $24.61 \pm 0.59$ |
| No alignment | 22.67 | 23.35 | 22.82 | $22.95 \pm 0.36$ |
| All-valid minus last-valid, answer (pp) | 1.97 | 1.21 | 1.21 | $1.47 \pm 0.44$ |
| All-valid minus last-valid, random (pp) | -0.83 | -0.08 | -0.08 | $-0.33 \pm 0.44$ |
| Allocation–Space interaction (pp) | 2.81 | 1.29 | 1.29 | $1.79 \pm 0.88$ |

tokens. The parser reads the last complete claim about the queried entity and accepts only one of the question's two concepts.

**Seven matched supervision conditions.** Table D4 crosses all-valid versus last-valid Allocation with rank-32 answer versus random subspaces under Local Targets. State and Cumulative Targets use all-valid/answer supervision; full-space alignment uses Local/all-valid supervision. The answer basis uses only training concept-answer tokens, and both rank-32 bases are frozen before training. All conditions share the teacher, official split, three recurrent updates, answer and reference-text CE, rank-128 LoRA, 40 epochs, and eight GPU workers with per-worker batch size 16 and no gradient accumulation. Alignment coefficients are calibrated on fixed training batches to match the initial auxiliary gradient scale. Each of the 21 checkpoints is selected by validation accuracy before evaluation on all 500 official test questions. The rank-matched answer–random comparison holds dimension fixed; the full-space comparison changes both dimension and directions.

Table D4: ProsQA matched Targets, Allocation, and Space comparisons. Entries are official-test accuracy (%) or paired differences (pp), each on the same 500 questions. All conditions use the same three-update GPT-2 student and frozen CoT teacher.

| Condition | Seed 11 | Seed 12 | Seed 13 | Mean |
|---|---|---|---|---|
| Local / all-valid / answer | 91.0 | 89.2 | 88.4 | 89.53 |
| Local / last-valid / answer | 86.8 | 88.2 | 84.8 | 86.60 |
| Local / all-valid / random | 86.6 | 89.6 | 89.0 | 88.40 |
| Local / last-valid / random | 89.8 | 88.2 | 85.6 | 87.87 |
| State / all-valid / answer | 88.6 | 90.4 | 86.4 | 88.47 |
| Cumulative / all-valid / answer | 86.4 | 88.4 | 87.8 | 87.53 |
| Local / all-valid / full space | 87.4 | 86.2 | 89.6 | 87.73 |
| All-valid − last-valid, answer (pp) | 4.2 | 1.0 | 3.6 | 2.93 |
| All-valid − last-valid, random (pp) | -3.2 | 1.4 | 3.4 | 0.53 |
| Allocation–Space interaction (pp) | 7.4 | -0.4 | 0.2 | 2.40 |

The interaction is $(\mathrm{all}_{\mathrm{answer}} - \mathrm{last}_{\mathrm{answer}}) - (\mathrm{all}_{\mathrm{random}} - \mathrm{last}_{\mathrm{random}})$ within each seed, with Local targets fixed in all four cells.

**Matched no-alignment comparison.** Table D5 compares the Local/all-valid/answer condition above with a no-alignment student using the same teacher-derived reference text, student initialization, data, recurrence, and training budget. Both answer and reference-text CE retain weight 1; the no-alignment arm sets both alignment coefficients to zero. Each seed uses its validation-selected checkpoint and is evaluated on the official 500-example test set with batch size 16 and a 32-token

generation cap. The mean accuracies are 89.53% for ALUS and 81.47% for no alignment, with a paired gain of $8.07 \pm 3.30$ pp, positive in all three seeds.

Table D5: **Matched no-alignment comparison on ProsQA.** Both conditions use the same three-update GPT-2 student and answer/reference-text objectives. Accuracies are percentages and gains are seed-paired differences in pp. ALUS reuses the Local/all-valid/answer row of Table D4; the paired gain is $8.07 \pm 3.30$ pp.

| Method | Seed 11 | Seed 12 | Seed 13 | Mean |
|---|---|---|---|---|
| No alignment | 86.6 | 80.2 | 77.6 | 81.47 |
| ALUS | 91.0 | 89.2 | 88.4 | 89.53 |
| Paired gain (pp) | +4.4 | +9.0 | +10.8 | +8.07 |

### D.4 Numeric Distribution Shift

Without further training, we evaluate examples with one normalizable numeric answer: 1,000/1,000 SVAMP (Patel et al., 2021), 2,098/2,305 ASDiv (Miao et al., 2020), and 174/174 in the fixed MultiArith test split (Roy & Roth, 2015). Table D6 reports the results. These tests change question distributions within numeric-answer tasks.

Table D6: Numeric distribution-shift tasks, evaluated without additional training. Accuracy (%) and differences (pp) are reported over three seeds.

| Dataset | ALUS | No alignment | Difference |
|---|---|---|---|
| SVAMP(1,000) | 39.80 ±0.95 | 31.93 ±0.74 | +7.87 |
| ASDiv(2,098) | 53.53 ±0.38 | 46.65 ±0.51 | +6.88 |
| MultiArith(174) | 84.48 ±2.63 | 62.45 ±1.76 | +22.03 |

### D.5 Inference Cost

ALUS adds no trainable alignment parameters; its frozen basis stores 24,576 coefficients for GPT-2 and 65,536 for Pythia-1B. In the main evaluation, ALUS and explicit CoT decode 1.24 and 27.05 tokens on average, at 41.17% and 46.25% accuracy. On PPU-ZW810E with PyTorch 2.9.0/BF16, optimized ALUS has batch-1 latency $97.34 \pm 0.31$ ms and batch-32 throughput $192.85 \pm 0.90$ examples/s across three seeds; explicit CoT seed 11 takes 482.69 ms and 23.38 examples/s. Batch-32 peak allocated memory is 0.522 versus 0.717 GB.

Timing covers all 1,319 questions, greedy decoding up to 128 tokens, five warm-up batches, and three synchronized repetitions; recurrence and decoding are included, input preparation and parsing excluded. ALUS recomputes the prefix without an answer KV cache. Timed batch-1 accuracies are $41.09 \pm 0.13\%$ for ALUS and 45.56% for explicit CoT seed 11; these separately scored generations do not replace main-evaluation results.