UNIFIED SEMANTIC STUDENT MODELING WITH LEARNER-STATE TRAJECTORIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Student modeling infers a learner's evolving knowledge and error patterns from interaction traces. Standard knowledge tracing and cognitive diagnosis mostly use identifiers, correctness bits, and skill tags, so they record whether a learner was wrong but not what the error means. Language models can read questions and written responses, yet they rarely maintain an explicit learner state that is reusable across tasks and causally used when generating predictions. We introduce the *Unified Semantic Student Model (USSM)*, a task-conditioned generator that compresses semantic interaction histories into a layerwise learner-state trajectory at a dedicated state token. A late-fusion attention mask forces every history-dependent prediction to pass through this state, allowing one representation to support correctness prediction, written-response prediction, misconception diagnosis, future-response prediction, and error-persistence prediction. Counterfactual recombination encourages rule-level state learning, and state transplantation tests whether the state is used rather than merely readable. Across FoundationalASSIST, DBE-KT22, XES3G5M, Eedi, and MalRule, one USSM remains competitive with task-specific specialists. Unified training improves semantic accuracy on Eedi-Mining misconception diagnosis from 0.445 to 0.517. Under candidate support, transplanting a donor state raises donor-directed errors from 0.168 to 0.411. After no-candidate training, transplanted states reproduce trained error rules on new numerical instances. These results show that USSM learns an explicit semantic learner state that is reusable across heterogeneous student-modeling tasks and can causally steer generation toward persistent learner-specific error patterns.

1 INTRODUCTION

Student modeling aims to infer a learner's evolving state from interaction history and use it to predict future behavior. Most existing approaches, including knowledge tracing, cognitive diagnosis, and item response theory, rely primarily on structured signals such as student and item identifiers, correctness labels, and skill tags (Corbett & Anderson, 1994; Piech et al., 2015; Ghosh et al., 2020; Wang et al., 2020; Lord, 1980). Such representations capture whether a learner is wrong but largely discard what the error means. For example, the response $1/2 + 1/3 = 2/5$ reveals a recognizable error procedure, while an ID-and-bit trace records only an incorrect event. Large language models can directly use the semantic content of questions and student responses, but existing LLM-based approaches are often prompt-based or task-specific and do not establish whether a single representation can maintain a student-specific semantic state across time and support heterogeneous prediction targets (Worden et al., 2026; Bhattacharyya et al., 2026; Hooshyar et al., 2025). This motivates a central question: can one semantic learner state summarize past behavior and be reused for correctness prediction, written responses, misconception diagnosis, future behavior, and error persistence?

A semantic representation alone, however, is not sufficient for student modeling. A hidden representation may contain information about a learner's misconception without the generator actually using that information when making predictions. A meaningful learner state should therefore satisfy two requirements: it should summarize learner-specific information across tasks, and changes to the state should systematically affect learner-conditioned predictions.

We introduce the *Unified Semantic Student Model (USSM)* to realize this idea. USSM uses a single task-conditioned generator for correctness prediction, written-response prediction, mis-

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.

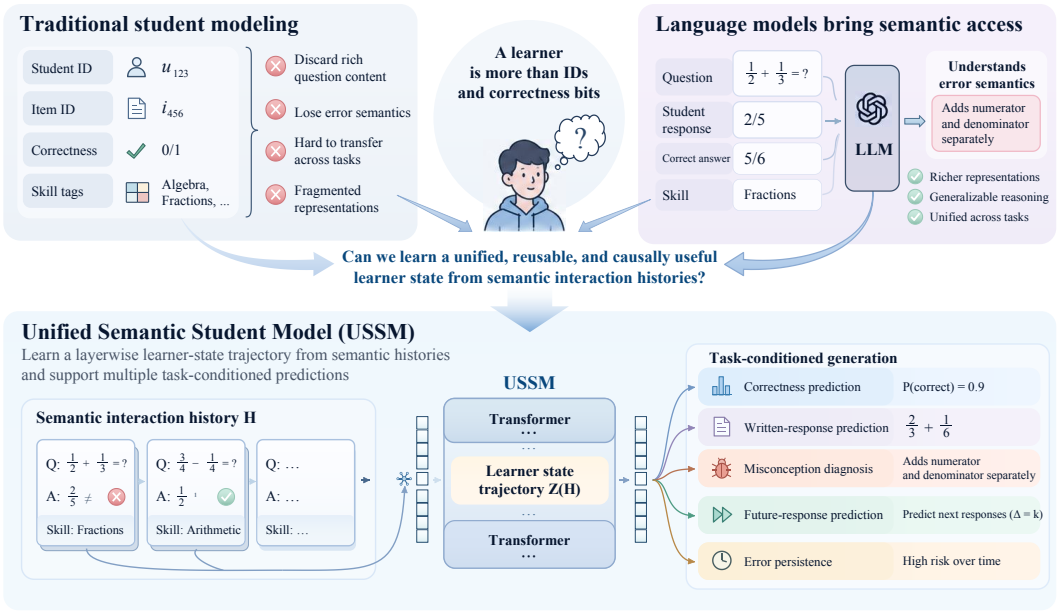


Figure 1: Motivation for USSM. Traditional student models compress learner interactions into sparse structured signals, losing semantic information about learner errors. Language models provide semantic access to questions and responses, motivating a unified learner state that summarizes interaction history, supports heterogeneous prediction tasks, and is causally validated through intervention.

conception diagnosis, future-response prediction, and error-persistence prediction. A dedicated <STUDENT_STATE> token reads the semantic interaction history, and its representations across transformer layers form a layerwise learner-state trajectory. The downstream task instruction is introduced only after this state is formed. A late-fusion attention mask prevents the query and generated answer from directly accessing the raw history, forcing history-dependent information to pass through the learner state.

To encourage the state to capture reusable error patterns rather than specific history–item pairs, we construct counterfactual examples that pair a learner history with new items governed by the same error rule. We then test whether the resulting state is functionally used through state transplantation. We keep the current item fixed, replace the recipient learner state with a donor state, and measure whether the generated response shifts toward the donor's characteristic error. We further remove answer candidates and test generation on new numerical instances, requiring the transplanted state to reproduce the underlying error procedure rather than select an exposed option.

Across a frozen 21,076-instance test set, a single USSM performs competitively with task-specific specialists across all five capabilities. Unified training improves semantic accuracy on Eedi-Mining misconception diagnosis from 0.445 to 0.517. State transplantation also increases donor-directed errors from 0.168 under a random-student state to 0.411 under the donor state, and transplanted states reproduce trained error rules on new numerical instances. Together, these results show that USSM learns a shared semantic learner state that supports heterogeneous prediction tasks and directly influences generation.

Our main contributions can be summarized as follows.

- We develop *USSM*, a unified student-modeling framework that uses a shared learner state to support five heterogeneous prediction tasks.
- We extend student modeling beyond structured predictions to semantic outputs including written responses, misconception diagnosis, and future learner behavior.

- We show that unified training enables positive cross-task transfer under matched target-task supervision.
- We validate the learned state through cross-template probing and controlled state transplantation, showing that it is both readable and functionally used during generation.

2 Related Work

2.1 Student Modeling

Classical student modeling can be grouped into psychometric, probabilistic, and neural sequence approaches. Item response theory (IRT) models learner ability and item difficulty (Rasch, 1960; Lord, 1980); Bayesian knowledge tracing (BKT) tracks latent skill mastery with a hidden Markov model (Corbett & Anderson, 1994); and performance factors analysis (PFA) predicts success from counts of prior successes and failures (Pavlik et al., 2009). Neural methods such as deep knowledge tracing (DKT), attentive knowledge tracing (AKT), and simpleKT replace hand-designed updates with recurrent or attention-based sequence models over item and correctness histories (Piech et al., 2015; Ghosh et al., 2020; Liu et al., 2023a), while pyKT and EduStudio standardize evaluation across such models (Liu et al., 2022; Wu et al., 2025). Cognitive-diagnosis models instead estimate proficiency over a knowledge-component (KC) structure, as in NeuralCDM (Wang et al., 2020). These approaches are effective for structured prediction, but their standard representations do not directly preserve the semantic content of the problem and the student's written error. Our *Unified Semantic Student Model (USSM)* treats that missing semantic content as part of the state-estimation problem itself.

2.2 Error Diagnosis

Semantic error modeling has a separate lineage centered on bugs, misconceptions, and executable malrules. Brown and Burton model errors as procedures that generate observed wrong answers, and VanLehn studies stable misconceptions underlying such procedures (Brown & Burton, 1978; VanLehn, 1990). More recent benchmarks such as Eedi-Mining formulate misconception identification as retrieval over a closed expert taxonomy (King et al., 2024; Wang et al., 2021), whereas MalRule provides executable rules and cross-template traces that permit controlled tests of whether an error procedure generalizes to a new surface (Chen et al., 2026). These datasets make error semantics measurable, but they do not by themselves provide a unified longitudinal learner state that also predicts correctness, concrete responses, future behavior, and persistence. We use Eedi for real semantic diagnosis and MalRule for controlled state-learning and causal analyses rather than treating either as a replacement for longitudinal student modeling.

2.3 Language Models

Large language models (LLMs) have recently been applied to student traces in three broad ways: prompting, unified generation, and student simulation. FoundationalASSIST and related evaluations prompt frontier models to predict future correctness or responses and find that specialized knowledge-tracing systems remain competitive or stronger on conventional targets (Worden et al., 2026; Bhattacharyya et al., 2026). UniEDU, Thinking-KT, and CIKT extend language-model interfaces toward multi-output prediction, feedback, recommendation, or textual profiles (Chu et al., 2025; Lee et al., 2026; Li et al., 2025), while CSM trains a model to simulate students carrying named algebra misconceptions (Sonkar et al., 2024). Centaur provides a broader precedent for fine-tuning a language model on trial-by-trial human behavior and predicting the same individual's next choice (Binz et al., 2025). Unlike these lines, our focus is not only multi-task output generation but the construction and causal validation of an explicit semantic learner state through which all history-dependent predictions must pass.

3 Method

The *Unified Semantic Student Model (USSM)* represents a learner from the semantic content of past interactions and uses the resulting learner state to support multiple student-modeling tasks. As illus-

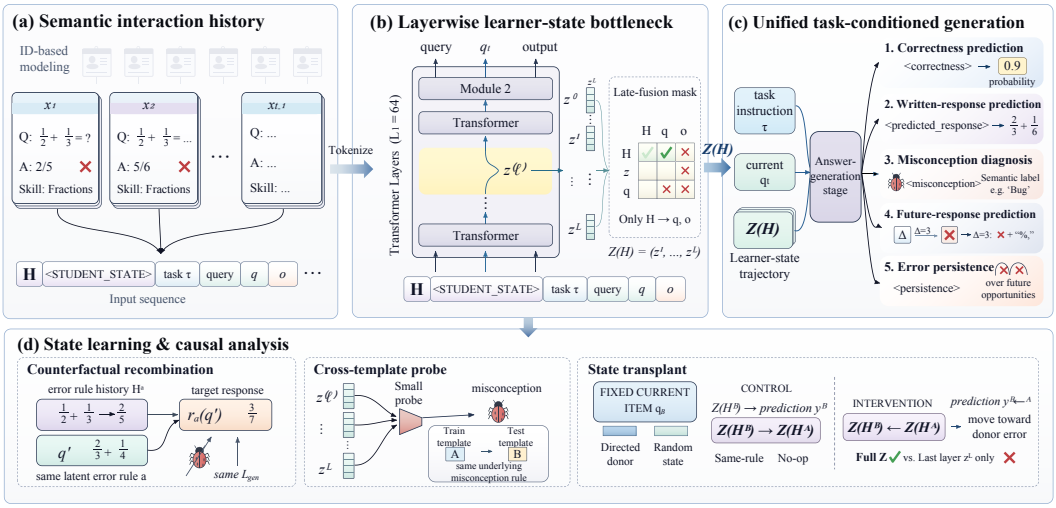


Figure 2: Overview of USSM. (a) Semantic interaction histories retain question, response, answer, and skill information. (b) A dedicated <STUDENT_STATE> token compresses the history into a layerwise learner-state trajectory under a late-fusion attention bottleneck; the task instruction is placed after the state token. (c) The resulting state is shared across five task-conditioned prediction tasks. (d) Counterfactual recombination, cross-template probing, and state transplantation test whether the learned state captures reusable error patterns and is functionally used during generation.

trated in Figure 2, USSM encodes semantic interaction histories into an explicit layerwise learner-state trajectory and uses this trajectory as the shared interface between past learner behavior and task-conditioned generation. The explicit state also provides a controllable representation for analyzing what the model has learned and whether that information is actually used during prediction.

3.1 Semantic Interaction History

As shown in Figure 2(a), USSM represents each student interaction using its semantic content rather than reducing it to an item identifier and a correctness signal. Let $H_t = (x_1, \ldots, x_{t-1})$ denote the observed interaction history before step t, where each interaction x_i may contain the question text, the student's written or selected response, the correct answer, the correctness label, and available knowledge-component (KC) information.

For prediction at step t, the model receives a task instruction τ and a query q_t, with the desired output denoted by o_t. The instruction specifies the student-modeling task, while the query identifies the current item or prediction target. USSM first maps the history to an explicit learner state $Z_t = E_\theta(H_t)$, where E_θ denotes the history encoding process implemented by the model with parameters θ. The learner state is extracted independently of the downstream task instruction; τ is introduced only after state formation. Prediction is then modeled as $p_\theta(o_t \mid \tau, q_t, Z_t)$. This formulation separates learner-specific information from both the task request and the current query, and provides an explicit representation that can be reused across prediction tasks and directly manipulated during intervention.

3.2 Layerwise Learner-State Bottleneck

Figure 2(b) illustrates the learner-state bottleneck. USSM inserts a dedicated <STUDENT_STATE> token after the semantic interaction history and retains its representation at every transformer layer. Let $z^{(\ell)} = h^{(\ell)}_{\text{state}} \in \mathbb{R}^d$ denote the state-token representation after layer ℓ. The learner state is the layerwise trajectory

$$Z(H) = \left(z^{(1)}, z^{(2)}, \ldots, z^{(L)}\right) \in \mathbb{R}^{L \times d}, \tag{1}$$

where L is the number of transformer layers. For Qwen3-32B, $L = 64$ and $d = 5120$. Unlike a single final-layer vector, this trajectory can be restored at the corresponding transformer layers during history-free generation.

USSM couples this trajectory with the late-fusion attention pattern in Figure 2(b). Because the sequence is $[H, \texttt{<STUDENT_STATE>}, \tau, q, o]$, causal attention lets the state token attend to the interaction history but not to τ. The generated output can attend to the task instruction, the current query, and the learner state, but not to the raw history. Thus, all history-dependent information must flow through $H \rightarrow Z(H) \rightarrow o$, while the task and query contribute through $\tau, q \rightarrow o$. During training, the history is present to construct $Z(H)$. During history-free evaluation and intervention, the history is removed and the captured $z^{(\ell)}(H)$ is injected at the corresponding state position of each transformer layer. This makes $Z(H)$ the sole, task-independent interface between past learner behavior and subsequent generation.

3.3 Unified Task-Conditioned Generation

As shown in Figure 2(c), USSM uses the same layerwise learner state for five student-modeling tasks: correctness prediction, written-response prediction, misconception diagnosis, future-response prediction, and error persistence. The task instruction τ specifies the requested capability, the query q_t defines the current prediction target, and $Z(H_t)$ provides learner-specific information. All tasks therefore share the same backbone, state representation, and generation mechanism. The output formats are summarized in Appendix D.

Each training instance is serialized as $[H, \texttt{<STUDENT_STATE>}, \tau, q, o]$, and only the target tokens in $o = (o_1, \ldots, o_m)$ contribute to the training objective. We optimize the target-token negative log-likelihood

$$\mathcal{L}_{\text{gen}} = -\sum_{j=1}^{m} \log p_\theta \left(o_j \mid o_{<j}, \tau, q, Z(H) \right), \tag{2}$$

where o_j denotes the j-th target token and $o_{<j}$ denotes the target tokens preceding it. The model parameters θ are shared across all five tasks. The task instruction, history, state token, and query receive no direct language-modeling loss.

The unified checkpoint uses no distillation, contrastive objective, or state-swap objective. Misconception descriptions appear only as targets of the diagnosis task. We first train Qwen3-32B using controlled MalRule examples, including the counterfactual recombinations described below. We then train the model jointly on the real educational tasks summarized in Appendix B. MalRule provides controlled supervision and mechanistic tests, while its results are kept separate from claims based on real student logs.

3.4 State Learning and Causal Analysis

Figure 2(d) examines whether the learner state captures reusable error patterns and whether those patterns are functionally used during generation.

The attention bottleneck determines where historical information flows, but does not guarantee that the state captures the underlying error procedure. We therefore construct counterfactual recombinations from MalRule. Given a history H^a generated by error rule a and a new item q', we form

$$\tilde{x}^{\,a} = \big(H^a,\ q',\ r_a(q')\big), \tag{3}$$

where $r_a(q')$ is the response obtained by applying rule a to q'. Because the new item is not taken from the original history, predicting $r_a(q')$ encourages $Z(H^a)$ to preserve the underlying error rule across changes in surface form. We further train a linear probe on one template and evaluate it on another to test whether this rule information is readable from the learner state.

Readability alone does not establish functional use. We therefore keep the recipient query q^B fixed and replace its learner state with a donor state,

$$Z(H^B) \leftarrow Z(H^A), \qquad y^{B \leftarrow A} \sim p_\theta\big(y \mid \tau, q^B, Z(H^A)\big)\,. \tag{4}$$

If the output moves toward the donor's characteristic error, the transplanted state is directly influencing generation. Under candidate support, we quantify this effect as

$$D_{\text{transfer}} = P\big(y = e_A \mid q^B, Z(H^A)\big) - P\big(y = e_A \mid q^B, Z(H^R)\big)\,, \tag{5}$$

where e_A is the donor-specific error and H^R is a random control history. Noise, same-rule, and no-op controls distinguish donor-specific steering from generic perturbation. We further remove answer candidates and evaluate free generation on new numerical instances, requiring the transplanted state to reproduce the underlying error procedure rather than select an exposed option.

4 Experiments

We evaluate USSM from four complementary perspectives: unified generality, semantic capability, cross-task transfer, and causal state use.

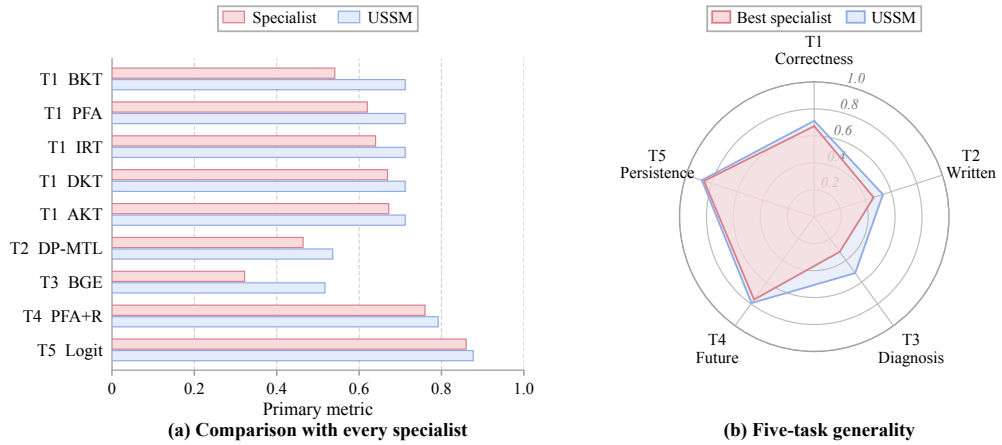
4.1 Experimental Setup

Datasets and splits. Across the five student-modeling tasks, filtering yields 241,331 training, 15,214 validation, and 21,076 test task instances. FoundationalASSIST is the primary real longitudinal dataset and provides item text, student responses, choices, and skill information (Worden et al., 2026). DBE-KT22 provides temporal multiple-choice interactions (Abdelrahman et al., 2022), while XES3G5M provides large-scale temporal correctness traces (Liu et al., 2023b). Eedi Response is used for profile-conditioned response prediction because its public version lacks timestamps (Wang et al., 2021). Eedi-Mining provides expert misconception labels (King et al., 2024), and MalRule provides executable error rules and cross-template traces for controlled mechanism experiments (Chen et al., 2026). Real longitudinal evaluations use student-disjoint splits, while MalRule test examples use surface templates that differ from those seen during training. Detailed task construction, dataset coverage, and split statistics are provided in Appendix B. The final one-epoch fit merges the original training and validation partitions into 256,545 task instances, while the frozen test set remains untouched.

Baselines and evaluation. For correctness prediction, we compare against Rasch item response theory (IRT) (Rasch, 1960), performance factors analysis (PFA) (Pavlik et al., 2009), Bayesian knowledge tracing (BKT) (Corbett & Anderson, 1994), deep knowledge tracing (DKT) (Piech et al., 2015), and attentive knowledge tracing (AKT) (Ghosh et al., 2020; Liu et al., 2022). Written-response prediction uses a single-task option-tracing baseline in the spirit of DP-MTL with λ=0 (An et al., 2022), while misconception diagnosis uses BGE-small retrieval over the Eedi taxonomy (Xiao et al., 2024). Future-response prediction and error persistence use corresponding task-specific baselines, and an untuned Qwen3-32B checkpoint serves as the base language-model reference. We report one primary metric per task. Misconception diagnosis on the held-out Eedi-Mining fold ($n = 894$) uses semantic accuracy (SA) from the frozen gpt6-astra LLM-as-judge for every reported model, including generative checkpoints and the BGE retrieval specialist (Appendix A.1). Correctness prediction and error persistence use exact accuracy; written-response and future-response prediction use LLM-judge accuracy. Cross-task transfer uses the same primary metric for each target task under matched supervision.

4.2 Unified Generality

Figure 3 asks whether one learner-state representation can support all five student-modeling tasks without a separate specialist for each target. Panel (a) compares USSM with every named task-specific method on the task primary metric; panel (b) summarizes the five-task envelope against the best specialist on each axis. All diagnosis scores use the same SA protocol. On Eedi-Mining ($n = 894$), SA_{BGE}=0.322, $\text{SA}_{\text{task-only}}$=0.445, and SA_{USSM}=0.517. The largest margin is misconception diagnosis against BGE-small retrieval; the smallest is error persistence against logistic persistence. Both are still positive.



(a) Comparison with every specialist **(b) Five-task generality**

Figure 3: Unified generality across five student-modeling tasks. (**a**) Comparison with every named specialist on the task primary metric (misconception diagnosis: SA throughout). (**b**) Five-task envelope: USSM versus the best specialist on each axis.

Table 1: Main results across five student-modeling tasks. (**a**) Comparison of the untuned base model, task-only training, and unified USSM using the same Qwen3-32B backbone. (**b**) Task-specific reference methods reported with their corresponding evaluation metrics. Misconception diagnosis uses SA (gpt6-astra) for all rows, including BGE retrieval.

(a) Unified model comparison

Method	Correctness Acc.	Written Judge Acc.	Diagnosis SA	Future Joint Acc.	Persistence Acc.
Base Qwen3-32B	0.545	0.224	0.445	0.246	0.841
Task-only Qwen3-32B	0.692	0.506	0.445	0.770	0.863
USSM	**0.712**	**0.536**	**0.517**	**0.792**	**0.877**

(b) Task-specific references

Task	Method	Metric	Score ↑
Correctness prediction	AKT	Accuracy	0.672
Written-response prediction	DP-MTL-style option tracing	Judge Acc.	0.464
Misconception diagnosis	BGE-small retrieval	SA	0.322
Future-response prediction	PFA + conditioned response	Joint Acc.	0.760
Error persistence	Logistic persistence	Accuracy	0.860

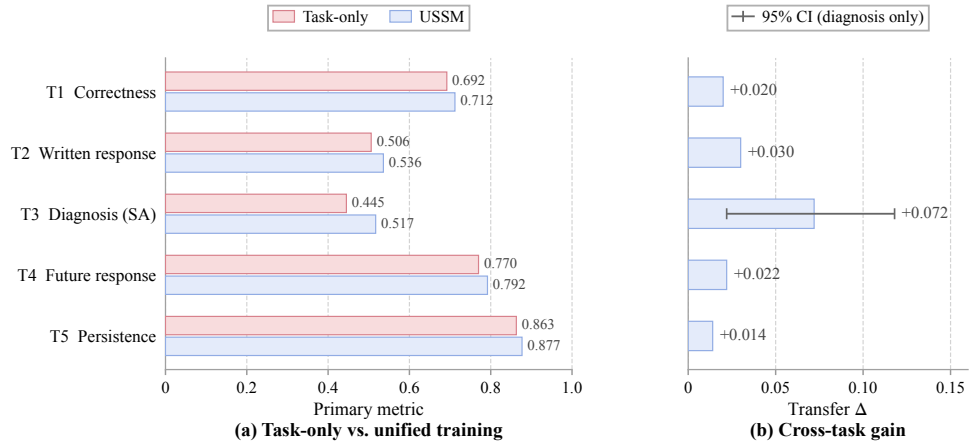
4.3 Semantic Frontier

The semantic-frontier analysis evaluates outputs that conventional knowledge-tracing and cognitive-diagnosis models do not naturally provide, including written responses, open misconception descriptions, and concrete future errors. We therefore compare USSM with teachers and a prompted language model on real longitudinal prefixes from FoundationalASSIST. Given only $H_{1:t}$, each rater predicts a misconception and a likely future error. Education experts additionally inspect later student behavior to adjudicate a reference misconception M^*, while later-error prediction is evaluated against the hidden suffix $H_{t+1:t+k}$. We use 200 prefixes and 20 teachers.

Agreement with M^* measures diagnosis quality, while later-error prediction measures whether the predicted error actually recurs. USSM achieves the highest score on both criteria in Table 2, outperforming both teachers and the prompted language model.

Table 2: Human evaluation on the same $H_{1:t}$ ($n = 200$ prefixes). Expert diagnosis measures agreement with M^*, while later-error prediction measures whether the named error appears in $H_{t+1:t+k}$. Holm-corrected McNemar p-values compare each baseline with USSM.

Rater	Expert diagnosis ↑	p vs. USSM	Later-error prediction ↑	p vs. USSM
Teacher	0.587	0.035	0.447	0.024
Prompted LM	0.513	< 0.001	0.393	< 0.001
USSM	**0.647**	–	**0.527**	–



(a) Task-only vs. unified training **(b) Cross-task gain**

Figure 4: Cross-task transfer under unified training. (**a**) Primary metric of each target task for the task-only model and USSM, with matched target-task exposure; misconception diagnosis uses SA. (**b**) Δ = USSM minus task-only. The misconception-diagnosis bar shows a question-clustered 95% confidence interval. Positive Δ indicates beneficial transfer from the other tasks.

Because all three raters are evaluated on the same 200 prefixes, the significance comparisons are paired rather than independent. We use McNemar's test on the per-prefix binary outcomes and apply Holm correction across the four baseline-versus-USSM comparisons formed by two baselines and two evaluation criteria. The resulting p-values test whether the discordant outcomes between each baseline and USSM are symmetric under the null hypothesis of equal paired accuracy. The teacher comparisons remain significant after correction for both expert diagnosis ($p = 0.035$) and later-error prediction ($p = 0.024$), while both prompted-LM comparisons have $p < 0.001$. Thus, the observed gaps are not explained by the aggregate rates alone, but are also supported by the paired pattern of wins and losses across the same student prefixes.

4.4 Cross-Task Transfer

Figure 4 examines whether supervision from the other student-modeling tasks benefits a given target capability. Each comparison uses the same Qwen3-32B backbone and matches exposure to the target-task supervision, so the difference reflects the effect of unified training rather than additional examples from the target task.

The strongest positive transfer occurs on Eedi-Mining misconception diagnosis, where performance increases from 0.445 to 0.517. The gain is 0.072 with a question-clustered 95% confidence interval of $[+0.022, +0.118]$. All five tasks show positive transfer on the primary metrics in Figure 4, with the largest gain on misconception diagnosis. The remaining tasks improve by $+0.020$ for correctness prediction, $+0.030$ for written-response prediction, $+0.022$ for future-response prediction, and $+0.014$ for error persistence. Shared supervision therefore strengthens the learner representation across tasks, with the strongest benefit on open misconception diagnosis.

Table 3: Causal state-use checks. The current question stays; only the learner state is swapped. D is the donor-state rate minus the named control. Candidate-supported selection and no-candidate free generation are different metrics.

What is tested	Donor	Control	D
1. Can a probe read the rule? Candidates present.			
Cross-template probe	0.933	–	–
Transplant vs. random student	0.411	0.168	0.243
Transplant vs. noise	0.411	0.189	0.222
Same-rule / no-op	0.042 / 0.000	–	–
2. Can the state write the rule without options? New numbers.			
vs. a different trained rule (original)	$4/96$	$3/96$	0.010
vs. a different trained rule (A)	$56/96$	$6/96$	0.521
vs. a different trained rule (B)	$56/96$	$8/96$	0.500
Three new items in a row (A)	$14/32$	$0/32$	–
3. Can it induce a rule never seen in training?			
Held-out-rule triples	$4/8$	$8/8$ trained	–

4.5 Causal State Use

This analysis asks whether $Z(H)$ is used at generation time, or only readable by a probe. The test is simple: keep student B's current question q^B visible, replace $Z(H^B)$ with a donor state $Z(H^A)$, and ask whether the output moves toward A's error. Table 3 reports three increasingly strict versions of this intervention. Extra seed-level counts are in Appendix C. This is not in-context learning. In ICL the generator attends to the raw history while answering. Here the late-fusion mask forbids $H \to o$: the history is used only to form $Z(H)$, and at intervention time the raw history is removed. Swapping $Z(H^A)$ into B's query therefore changes a compressed learner state, not the demonstration prefix. If this intervention were ICL, dropping H or replacing only Z would not be a well-defined test.

Block 1 shows that the rule is both readable and causal when answer candidates are listed. A cross-template probe recovers known rules from $Z(H)$ at 0.933. Transplanting a donor state raises the donor error from 0.168 (a real random student) to 0.411 ($D = 0.243$, 95% CI $[0.126, 0.358]$). Noise, same-rule, and no-op controls stay smaller, so the flip is not a generic perturbation.

Block 2 removes the candidates, so the model must generate the error on new numbers rather than pick an exposed option. The original checkpoint barely moves ($4/96$ vs. $3/96$). After no-candidate training (checkpoint A), donor states hit the trained rule on $56/96$ items, against $6/96$ for a state tied to a different rule; checkpoint B, which adds a state-swap loss, does not help further. The same donor state can also keep the rule across three consecutive new items ($14/32$ vs. $0/32$). Figure 5 shows one such case: an ignore-denominator state writes 154, 234, and 45 on three unseen products.

Block 3 is the boundary. Held-out rules never seen in training reach only $4/8$ successful triples, while trained-rule controls remain at $8/8$. The state can carry and reuse procedures it has learned; it does not reliably induce unseen ones from a donor history alone.

5 Conclusion

USSM provides a unified framework for modeling learner behavior from semantic interaction histories. It compresses past interactions into an explicit layerwise learner-state trajectory that is reused across five student-modeling tasks. Experiments on multiple educational datasets show that a single USSM remains competitive with task-specific specialists and benefits from positive cross-task transfer. Human evaluation further shows that its semantic predictions align with expert judgments and later learner errors. Controlled state transplantation demonstrates that the learned state is not only readable but can directly steer generation toward learner-specific error patterns. Together, these results support explicit and reusable semantic learner states as a promising interface for unified student modeling.

AI USE STATEMENT

Generative AI tools were used to assist with language editing and presentation of the manuscript. All research questions, methodological decisions, experimental design, implementation, analysis, and scientific claims were developed and verified by the authors. The authors take full responsibility for the content of the paper.

ETHICS STATEMENT

This work studies student modeling using existing educational datasets and does not recruit new student participants. The underlying data may involve minors and are therefore treated as sensitive educational records. We use the datasets only under their original access conditions and do not attempt to recover or infer student identities. The learner state and predicted misconceptions are model representations derived from limited interaction histories and should not be interpreted as permanent labels of individual students. In particular, such predictions should not be used on their own for grading, placement, tracking, or access to educational resources. Any deployment of student-modeling systems based on these representations should retain appropriate privacy safeguards and meaningful human oversight.

REPRODUCIBILITY STATEMENT

We provide the task definitions, model formulation, training objective, data splits, evaluation protocols, and baseline specifications in the main paper and appendix. The appendix further reports dataset-level statistics, task-specific evaluation settings, specialist baselines, and detailed intervention protocols. We will release the preprocessing code, task construction pipeline, training configurations, prompt and serialization templates, evaluation scripts, and split seeds required to reproduce the reported experiments. Datasets with restricted access, including FoundationalASSIST, must be obtained under their original access terms.

REFERENCES

Ghodai Abdelrahman, Sherif Abdelfattah, Qing Wang, and Yu Lin. DBE-KT22: A knowledge tracing dataset based on online student evaluation. *arXiv preprint arXiv:2208.12651*, 2022.

Suyeong An, Junghoon Kim, Minsam Kim, and Juneyoung Park. No task left behind: Multi-task learning of knowledge tracing and option tracing for better student assessment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022.

Prarthana Bhattacharyya, Joshua Mitton, Ralph Abboud, and Simon Woodhead. Faster, cheaper, more accurate: Specialised knowledge tracing models outperform llms. *arXiv preprint arXiv:2603.02830*, 2026.

Marcel Binz, Elif Akata, Matthias Bethge, Franziska Brändle, Fred Callaway, Julian Coda-Forno, Peter Dayan, Can Demircan, Maria K. Eckstein, Noémi Éltető, Thomas L. Griffiths, Susanne Haridi, Akshay K. Jagadish, Ji-An Li, Alexander Kipnis, Sreejan Kumar, Tobias Ludwig, Marvin Mathony, Marcelo Mattar, Alireza Modirshanechi, Surabhi S. Nath, Joshua C. Peterson, Milena Rmus, Evan M. Russek, Tankred Saanum, Johannes A. Schubert, Luca M. Schulze Buschoff, Nishad Singhi, Xin Sui, Mirko Thalmann, Fabian J. Theis, Vuong Truong, Vishaal Udandarao, Konstantinos Voudouris, Robert Wilson, Kristin Witte, Shuchen Wu, Dirk U. Wulff, Huadong Xiong, and Eric Schulz. A foundation model to predict and capture human cognition. *Nature*, 644:1002–1009, 2025.

John Seely Brown and Richard R. Burton. Diagnostic models for procedural bugs in basic mathematical skills. *Cognitive Science*, 2(2):155–192, 1978.

Xinghe Chen, Naiming Liu, and Shashank Sonkar. MalruleLib: Large-scale executable misconception reasoning with step traces for modeling student thinking in mathematics. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2026.

Zhendong Chu, Jian Xie, Shen Wang, Zichao Wang, and Qingsong Wen. UniEDU: Toward unified and efficient large multimodal models for educational tasks. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, 2025.

Albert T. Corbett and John R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, 4(4):253–278, 1994.

Aritra Ghosh, Neil Heffernan, and Andrew S. Lan. Context-aware attentive knowledge tracing. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2020.

Danial Hooshyar, Yeongwook Yang, Gustav Sir, Tommi Karkkainen, Raija Hamalainen, Mutlu Cukurova, and Roger Azevedo. Problems with large language models for learner modelling: Why llms alone fall short for responsible tutoring in k–12 education. *arXiv preprint arXiv:2512.23036*, 2025.

Jules King, L. Burleigh, Simon Woodhead, Panagiota Kon, Perpetual Baffour, Scott Crossley, Walter Reade, and Maggie Demkin. Eedi – mining misconceptions in mathematics. Kaggle, 2024.

Unggi Lee, Joo Young Kim, Ran Ju, Minyoung Jung, and Jeyeon Eo. A training-free large reasoning model-based knowledge tracing framework for unified prediction and prescription. *arXiv preprint arXiv:2601.01708*, 2026.

Runze Li, Siyu Wu, Jun Wang, and Wei Zhang. CIKT: A collaborative and iterative knowledge tracing framework with large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, 2025.

Zitao Liu, Qiongqiong Liu, Jiahao Chen, Shuyan Huang, Jiliang Tang, and Weiqi Luo. pykt: A python library to benchmark deep learning based knowledge tracing models. In *Advances in Neural Information Processing Systems*, 2022.

Zitao Liu, Qiongqiong Liu, Jiahao Chen, Shuyan Huang, and Weiqi Luo. simplekt: A simple but tough-to-beat baseline for knowledge tracing. In *The Eleventh International Conference on Learning Representations*, 2023a.

Zitao Liu, Qiongqiong Liu, Teng Guo, Jiahao Chen, Shuyan Huang, Xiangyu Zhao, Jiliang Tang, Weiqi Luo, and Jian Weng. XES3G5M: A knowledge tracing benchmark dataset with auxiliary information. In *Advances in Neural Information Processing Systems*, 2023b.

Frederic M. Lord. *Applications of Item Response Theory to Practical Testing Problems*. Lawrence Erlbaum Associates, Hillsdale, NJ, 1980.

Philip I. Pavlik, Hao Cen, and Kenneth R. Koedinger. Performance factors analysis—a new alternative to knowledge tracing. In *Proceedings of the 14th International Conference on Artificial Intelligence in Education*, pp. 531–538, 2009.

Chris Piech, Jonathan Bassen, Jonathan Huang, Surya Ganguli, Mehran Sahami, Leonidas J. Guibas, and Jascha Sohl-Dickstein. Deep knowledge tracing. In *Advances in Neural Information Processing Systems*, 2015.

Georg Rasch. *Probabilistic Models for Some Intelligence and Attainment Tests*. Danish Institute for Educational Research, Copenhagen, 1960.

Shashank Sonkar, Xinghe Chen, Naiming Liu, Richard G. Baraniuk, and Mrinmaya Sachan. Llm-based cognitive models of students with misconceptions. *arXiv preprint arXiv:2410.12294*, 2024.

Kurt VanLehn. *Mind Bugs: The Origins of Procedural Misconceptions*. MIT Press, Cambridge, MA, 1990.

Fei Wang, Qi Liu, Enhong Chen, Zhenya Huang, Yuying Chen, Yu Yin, Zai Huang, and Shijin Wang. Neural cognitive diagnosis for intelligent education systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

Zichao Wang, Angus Lamb, Evgeny Saveliev, Pashmina Cameron, Yordan Zaykov, Jose Miguel Hernandez-Lobato, Richard E. Turner, Richard G. Baraniuk, Craig Barton, Simon Peyton Jones, Simon Woodhead, and Cheng Zhang. Results and insights from diagnostic questions: The neurips 2020 education challenge. In *Proceedings of the NeurIPS 2020 Competition and Demonstration Track*, 2021.

Eamon Worden, Cristina Heffernan, Neil Heffernan, and Shashank Sonkar. Foundationalassist: Dataset for foundational knowledge tracing & pedagogical grounding of large language models. In *Artificial Intelligence in Education*, 2026.

Le Wu, Xiangzhi Chen, Fei Liu, Junsong Xie, Chenao Xia, Zhengtao Tan, Mi Tian, Jinglong Li, Kun Zhang, Defu Lian, Richang Hong, and Meng Wang. Edustudio: Towards a unified library for student cognitive modeling. *Frontiers of Computer Science*, 19:198342, 2025.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. C-Pack: Packed resources for general Chinese embeddings. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2024.

A Extended Evaluation

This appendix provides the evaluation details that complement the primary results in the main paper. We first specify the semantic scoring protocol, then report the higher-precision task-level results, auxiliary metrics, and the multi-seed robustness summary. Keeping these results together avoids fragmenting the evaluation across many small task-specific tables.

A.1 LLM-as-Judge Evaluation

We use a frozen *LLM-as-judge* protocol with **gpt6-astra** whenever the target requires semantic equivalence rather than exact string match. Throughout the paper, **SA** denotes semantic accuracy under this protocol.

Written response. The judge receives the question, the reference response, and the model prediction, and returns yes/no on whether the prediction matches the reference answer, including equivalent multiple-choice options and mathematically equivalent free-form expressions. Accuracy is the yes rate on the pooled $3{,}693$-example written-response test set.

Misconception diagnosis. On the held-out Eedi-Mining fold ($n = 894$), the judge receives the question, the student's incorrect response, the predicted misconception text, and the expert reference misconception, and returns yes/no on whether the prediction names the same underlying error; the resulting yes rate is **SA**. Generative base and task-only outputs each correspond to $398/894$ SA hits; unified outputs correspond to $462/894$. For BGE-small retrieval, we take the top-ranked taxonomy misconception description for each query and score it with the same judge, yielding $288/894$ ($\mathrm{SA}_{\mathrm{BGE}}$=0.322).

Future response. Correctness on the future item uses the gold label. The open response is judged with the same protocol as written-response prediction. A prediction counts as correct only when both future correctness and the judged response match the target.

A.2 Task-Level Results

Table 4 reports the primary metric for each of the five student-modeling tasks. The untuned base model is substantially weaker than trained models on correctness, written-response, and future-response prediction, showing that the gains are not explained by the pretrained language model alone. Unified training improves over the corresponding task-only checkpoint on all five primary metrics in the reported results, with the largest absolute gain on misconception diagnosis.

Table 4: Primary task-level results with additional precision. Correctness prediction, written-response prediction, future-response prediction, and error persistence use accuracy. Misconception diagnosis uses SA (gpt6-astra) on the held-out Eedi-Mining fold ($n = 894$) for all reported models, including BGE retrieval.

Model	Correct. Acc.	Written Acc.	Diagnosis SA	Future Acc.	Persist. Acc.
Base Qwen3-32B	0.544580	0.223666	0.445000	0.246304	0.841189
Task-only	0.691652	0.505551	0.445190	0.769783	0.862705
Unified	**0.711652**	**0.535551**	**0.516779**	**0.791783**	**0.876705**

Table 5: Auxiliary task-specific metrics. Written-response prediction and error persistence have no additional secondary metric in the current evaluation and are therefore represented by their primary scores in Table 4.

Correctness prediction

Method	Accuracy	Macro-F1	AUROC
Base Qwen3-32B	0.544580	0.474429	0.633796
AKT	0.672000	0.671627	0.742101
Task-only	0.691652	**0.690917**	**0.763184**
USSM	**0.711652**	0.687526	0.760078

Misconception diagnosis (SA on the held-out Eedi-Mining fold, $n = 894$)

Method	Semantic accuracy (SA)
BGE-small retrieval	0.322000
Base Qwen3-32B	0.445000
Task-only	0.445190
USSM	**0.516779**

Future-response prediction

Method	Joint accuracy	Correctness accuracy	Response accuracy
Base Qwen3-32B	0.246304	0.379565	0.264783
PFA + conditioned response	0.760000	0.797174	0.760870
Task-only	0.769783	**0.802341**	**0.773816**
USSM	**0.791783**	0.812565	0.796913

Specialist references. The task-specific references used in the main comparison are AKT for correctness prediction (0.672), DP-MTL-style option tracing for written responses (0.464), BGE-small retrieval with judged taxonomy descriptions for Eedi misconception diagnosis (0.322 SA), PFA with conditioned response for future-response prediction (0.760), and logistic persistence for error persistence (0.860). For correctness prediction, the additional specialist scores are DKT at 0.669, IRT at 0.640, PFA at 0.620, and BKT at 0.541.

Table 5 consolidates the available secondary metrics that were previously split across several small tables. These metrics provide a more detailed view of model behavior without replacing the primary metrics used in the main paper. For correctness prediction, the unified model improves the primary accuracy while the task-only checkpoint remains slightly higher on Macro-F1 and AUROC in the detailed evaluation. For future-response prediction, the decomposition shows that the joint score depends on both future correctness and response generation rather than on either component alone.

A.3 Multi-Seed Robustness

The three-seed summary is reported separately from the single-run high-precision values above because its purpose is to quantify training variation rather than provide another task leaderboard. Across all five tasks, the mean ordering remains consistent with the point-estimate comparison, and the standard deviations are small relative to the observed task-only versus unified gaps.

Table 6: Three-seed robustness summary for Qwen3-32B task-only and unified training. Entries are mean ± standard deviation.

Model	Correct. Acc.	Written Acc.	Diagnosis SA	Future Acc.	Persist. Acc.
Task-only	0.6917±0.0036	0.5056±0.0059	0.4450±0.0083	0.7698±0.0042	0.8627±0.0025
Unified	0.7117±0.0032	0.5356±0.0054	0.5168±0.0075	0.7918±0.0037	0.8767±0.0021

Table 7: Learner-state architecture ablation under matched multi-task supervision. All variants use the same Qwen3-32B backbone, data, target-task exposure, and optimization budget.

Model	Correct. Acc.	Written Acc.	Diagnosis SA	Future Acc.	Persist. Acc.
Multi-task SFT	0.7058	0.5274	0.4989	0.7836	0.8724
+ State token	0.7066	0.5288	0.5012	0.7849	0.8731
+ State bottleneck	0.7098	0.5330	0.5114	0.7896	0.8755
USSM	**0.7117**	**0.5356**	**0.5168**	**0.7918**	**0.8767**

A.4 Learner-State Architecture Ablation

We first separate the contribution of ordinary multi-task training from the learner-state bottleneck. All variants use the same Qwen3-32B initialization, training examples, task-sampling ratio, target-token objective, and optimization budget. *Multi-task SFT* directly exposes the semantic history to the generated output. + *State token* inserts `<STUDENT_STATE>` but keeps standard causal attention, so the output can still attend to the raw history. + *State bottleneck* activates the late-fusion mask and blocks the direct $H \to o$ path, forcing history-dependent information through the learner state. The full USSM further uses the counterfactual recombination described in Section 3.

These results isolate two effects. First, ordinary multi-task SFT already accounts for most of the raw predictive gain, which is expected because all tasks share educational semantics and the same backbone. Merely adding a state token changes performance only marginally, suggesting that an unused special token is not itself sufficient. Activating the late-fusion bottleneck produces a more consistent improvement, with the largest change on misconception diagnosis, where semantic information from the learner history is most important. Counterfactual recombination provides a smaller additional gain on top of the bottleneck. This pattern supports the interpretation that the main role of the proposed architecture is not simply to increase multi-task capacity, but to make the learner state a functionally sufficient interface for history-dependent prediction.

We next test whether the full layerwise trajectory is necessary once the bottleneck is active. The prediction query and current item are kept fixed, and only the representation injected at the state position is changed. *Last layer only* restores $z^{(L)}$ but no earlier states. *Last 16 layers* restores only the top quarter of the trajectory. *Every 8th layer* uses a sparse set of states across depth. *Layer-shuffled* retains all state vectors but permutes their layer assignments, controlling for the total number of injected vectors while destroying their depth correspondence.

The trajectory ablation tests a stronger claim than task accuracy alone. A final-layer state preserves part of the donor effect but substantially reduces directed transfer relative to the full trajectory. Restoring progressively broader coverage across depth recovers more of the effect. Importantly, the layer-shuffled control remains well below the full trajectory despite injecting the same collection of state vectors. This pattern indicates that the benefit is not explained only by storing more hidden-state vectors: the correspondence between a state representation and the transformer layer at which it is restored also matters. Together with the architecture ablation above, this distinguishes the proposed learner-state mechanism from ordinary multi-task SFT while showing that the layerwise trajectory is most important for reusable and causally controllable state, even when conventional prediction scores are relatively close.

Table 8: Layerwise trajectory ablation for state transplantation. The donor-directed rate and random-student control use the candidate-supported setting, while the final column reports donor-rule hits under no-candidate free generation.

State representation	Donor	Random	D_{transfer}	No-candidate donor hits
Last layer only	0.286	0.169	0.117	$25/96$
Last 16 layers	0.349	0.170	0.179	$39/96$
Every 8th layer	0.366	0.169	0.197	$44/96$
Layer-shuffled	0.302	0.171	0.131	$31/96$
Full trajectory	**0.411**	0.168	**0.243**	$56/96$

Table 9: Task coverage and post-filtering split sizes. A source appears only for tasks that its released fields can support.

Task	Source	N_{tr}	N_{va}	N_{te}	Protocol / target
Correctness prediction					
	DBE-KT22	3,515	221	225	Next-interaction first-response correctness; new-student/warm-item evaluation.
	Eedi Response	15,736	984	984	Held-out response conditioned on the same student's profile.
	FoundationalASSIST	30,009	1,863	1,882	Identifiable first-response correctness after support/privacy filtering.
	XES3G5M	52,009	2,888	7,222	Next-interaction correctness on official temporal test students.
	MalRule	3,000	300	600	Systematic-rule correctness with same- and cross-template tests.
Written response					
	DBE-KT22	3,472	218	223	Selected-option generation.
	Eedi Response	15,736	984	984	Profile-conditioned held-out option generation.
	FoundationalASSIST	29,995	1,864	1,886	Privacy-filtered structured mathematical first-response generation.
	MalRule	3,000	300	600	Executable systematic-error response generation.
Misconception diagnosis					
	Eedi-Mining	3,476	0	894	Open diagnosis against expert misconceptions; fixed question-grouped five-fold split, folds 0–3 for training and fold 4 for testing.
Future response					
	DBE-KT22	3,186	200	200	Correctness and selected response on a specified item at $\Delta = 3$.
	FoundationalASSIST	70,811	4,403	4,400	Correctness and open response at $\Delta \in \{1, 3, 5\}$.
Error persistence					
	DBE-KT22	114	15	12	Same question and same wrong option recurs within the next three same-question records.
	FoundationalASSIST	7,272	974	964	Exact normalized wrong response recurs within the next three same-question records.
	Total task instances	**241,331**	**15,214**	**21,076**	

B Dataset Construction and Coverage

The benchmark combines sources with different forms of supervision rather than forcing every dataset into every task. FoundationalASSIST and DBE-KT22 provide the broadest longitudinal coverage, while XES3G5M contributes large-scale correctness traces, Eedi-Mining supplies expert misconception labels, and MalRule provides controlled executable error rules for mechanism tests. This division is important because the pooled benchmark evaluates a shared learner representation across complementary sources rather than a single-dataset leaderboard.

Table 9 reports the post-filtering example counts for each task–source pair. Real longitudinal evaluations use student-disjoint splits where applicable, while MalRule uses controlled same-template and cross-template examples. The final one-epoch fit merges the original training and validation partitions into $256{,}545$ task instances; the frozen $21{,}076$-instance test set is not used for training.

Table 10 summarizes the source-level scope before the final context-length filter. The datasets differ substantially in temporality and available response fields. In particular, Eedi Response lacks timestamps in the public mirror and is therefore used for profile-conditioned prediction rather than next-step forecasting, while Eedi-Mining contains no longitudinal student sequence and is used only for response-level diagnosis. These distinctions explain why not every source contributes to every task.

Table 10: Source-level scope and split construction before the final context-length filter.

Source	Students	Questions / labels	History cap	Split and scope
DBE-KT22	1,193	212	6	954/119/120 train/validation/test students; temporal new-student, predominantly warm-item multiple-choice evaluation.
Eedi Response	4,918	445	6	3,934/492/492 users; user-disjoint deterministic split. The public mirror omits timestamps, so the protocol is profile-conditioned rather than temporal.
FoundationalASSIST	5,000	2,986	12	4,000/500/500 students; temporal student-disjoint split. Free-text targets are filtered to privacy-safe structured mathematics, and future items occur one, three, or five interactions ahead.
XES3G5M	18,066	4,137	6	13,008/1,445/3,613 students; official temporal test users. The released data record correctness but not selected responses.
Eedi-Mining	–	2,587 labels	–	4,370 question–wrong-answer diagnosis queries under a fixed question-grouped five-fold split; folds 0–3 provide 3,476 training examples and fold 4 provides 894 test examples.
MalRule	synthetic	21 rules	4	3,000/300/600 synthetic profiles over 132 surfaces; the test combines same-template and cross-template profiles across fractions, decimals, and subtraction.

C Causal Analysis Details

The causal experiments separate three increasingly demanding questions. The candidate-supported intervention asks whether changing the learner state can redirect a prediction when possible answers are visible. The no-candidate setting removes this shortcut and requires the model to generate the donor-consistent error on new numerical instances. The held-out-rule setting then asks whether the same mechanism extends to rules that were not available during training. These settings therefore test state use, procedural reuse, and the current generalization boundary, respectively.

Under candidate support, the donor-state rate is 0.411, compared with 0.168 for a random-student state, giving $D_{\mathrm{transfer}} = 0.243$ with a 95% confidence interval of $[0.126, 0.358]$. In the no-candidate setting, the original checkpoint shows little directed transfer ($4/96$ versus $3/96$), whereas checkpoint A reaches $56/96$ donor-rule hits against $6/96$ wrong-rule hits. Checkpoint B adds a state-swap objective but remains at $56/96$ donor-rule hits with $8/96$ wrong-rule hits, indicating no additional gain in this evaluation. The three-item test further requires the same transplanted state to preserve the rule across consecutive queries, reaching $14/32$ successful sequences versus $0/32$ for the control state.

For held-out rules, we evaluate histories of length 3, 6, and 12 over $5{,}376$ decodes. New-rule three-item sequences reach $4/8$, whereas trained-rule controls reach $24/24$ item-level successes and $8/8$ successful three-item sequences. This contrast distinguishes reliable reuse of learned procedures from the more difficult problem of inducing a previously unseen procedure from history alone.

C.1 Qualitative Free-Generation Example

Figure 5 illustrates the no-candidate setting. The same transplanted state is reused across three new numerical instances. The generated responses follow the same ignore-denominator multiplication rule, showing that the effect is not tied to one exposed answer option or one specific set of numbers.

D Prompt and Output Formats

All five tasks use the same serialization $[H, \texttt{<STUDENT_STATE>}, \tau, q, o]$. The learner state is formed from H before τ appears, so causal attention keeps $Z(H)$ independent of the downstream task instruction. The instruction changes only the requested output, while the history encoder, learner-state trajectory, and generation interface remain shared. Table 11 summarizes the output field and the primary scoring rule for each task.

The shared interface is important for interpreting the unified model. Task identity changes the requested output, but it does not enter the learner state and does not introduce a separate encoder or task-specific representation. Consequently, the cross-task comparisons in the main paper evaluate different queries of the same learner-state mechanism rather than a collection of independently parameterized student models.

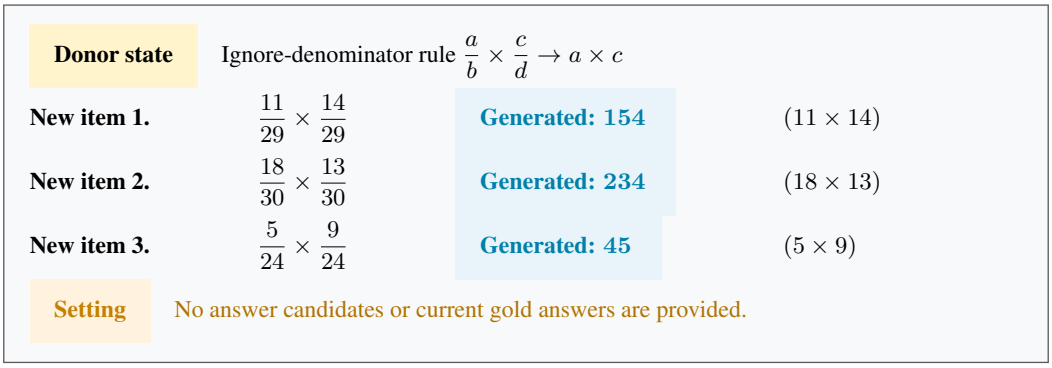


Figure 5: Free-generation examples under state transplantation. The same donor learner state reproduces the trained ignore-denominator rule on new numerical instances.

Table 11: Shared output interface for the five student-modeling tasks.

Task	Output field	Primary scoring
Correctness prediction	`<correctness>`	Exact accuracy
Written-response prediction	`<predicted_response>`	LLM-judge accuracy
Misconception diagnosis	`<misconception>`	Semantic accuracy
Future-response prediction	correctness + response at distance Δ	Joint correctness and response accuracy
Error persistence	`<persistence>`	Exact accuracy