HIERARCHICAL CAUSAL DISTILLATION: SELECTING SUFFICIENT CAUSAL SUPERVISION

Anonymous authors
Paper under double-blind review

ABSTRACT

Distilling a large language model (LLM) requires choosing which aspects of its computation a smaller student should preserve. Responses to interventions on intermediate variables and computational routes make this choice testable. We propose Hierarchical Causal Distillation (HCD), based on the principle that a distilled model should preserve a compact causal structure sufficient for the required teacher responses. HCD asks which supervision objectives enable a student to retain these responses under intervention. HCD organizes supervision around behavior, high-level reasoning programs, their circuit realizations, and internal states. It trains students with different combinations of objectives and evaluates them against common task-specific behavioral and interventional requirements. Cross-model correspondences relate semantic variables and circuit roles, allowing teachers and students with different architectures to preserve the same required responses. We formalize the choice of causal granularity using Unified Causal Minimum Description Length (UCMDL), which selects the shortest adequate joint description of a teacher mechanism and cross-model alignment within a fixed candidate family. All candidates face the same intervention requirements. A finite-family confidence bound defines adequacy on independent selection data before description-length comparison. We report results on two controlled transformers with known programs and pretrained Qwen3.5 students on seven generated arithmetic tasks. In the controlled study, HCD selects program-and-circuit supervision (BPC) on both tasks. The evaluated BPC students achieve lower ordinary and intervention response errors than behavioral KD. The independently trained frequency-sorting student also achieves low state-intervention error without a separate state objective in student training. In the pretrained comparison, BPC improves base-answer accuracy by 30.0% relative to behavioral KD and achieves the lowest pooled mean teacher–student total variation on ordinary, program-intervention, and circuit-intervention queries. These results support program and circuit supervision for preserving the evaluated teacher responses during compression.

1 INTRODUCTION

Distillation requires deciding what a smaller student should preserve from its teacher. Output objectives transfer predictions (Hinton et al., 2015); representation and intervention objectives constrain the computation that produces them (Sun et al., 2019; Wu et al., 2022; Wadhwa et al., 2025). Ordinary agreement leaves open how the student will respond when an intermediate quantity, computational route, or internal state changes. Yet different internal realizations can preserve the same required responses. This motivates a central question: *which causal constraints are sufficient for faithful distillation?*

Consider a teacher that computes $z = a + b$ and then $y = zc$ (Figure 1). Matching ordinary answers constrains the final product. Replacing z tests its downstream use; blocking a route carrying z tests the role of that route. Substituting a broader internal state probes an additional correspondence. The supervision needed to preserve these responses is an empirical question: program and circuit objectives may already preserve a required state response, while another task may need a separate state objective.

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.

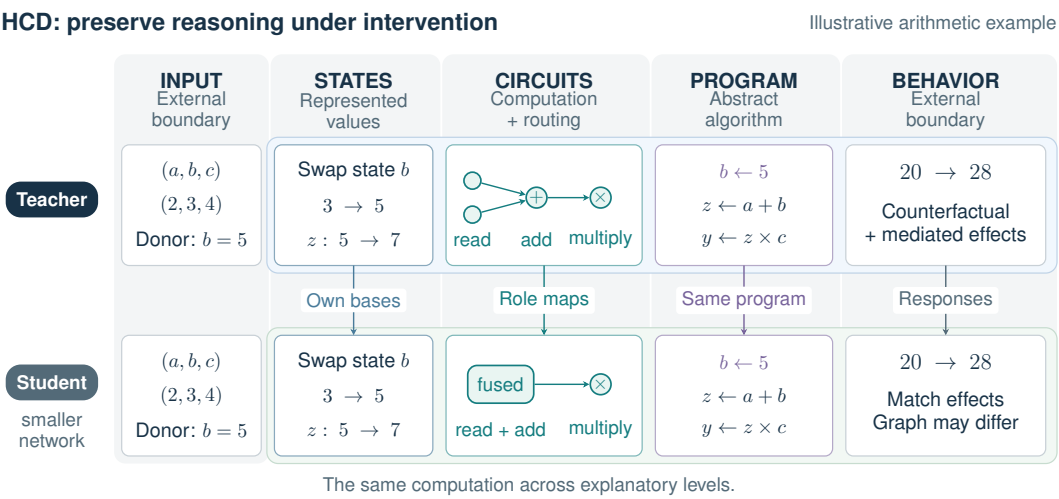


Figure 1: HCD's explanatory levels for $z = a + b$, $y = zc$. On input $(2, 3, 4)$, replacing the represented value $b = 3$ with donor value 5 changes $z : 5 \rightarrow 7$ and $y : 20 \rightarrow 28$. Each model uses its own donor states and bases; circuit operations may be fused. Vertical arrows denote cross-model correspondences. The columns show the representation, circuit, program, and response associated with one semantic intervention. Evaluation compares counterfactual outputs and required route-mediated effects.

We propose *Hierarchical Causal Distillation* (HCD) to select sufficient causal supervision. Candidates are trained independently with combinations of behavior (B), program (P), circuit (C), and state (S) objectives, then evaluated against common task-specific requirements. The objectives determine what information training supplies; the requirements determine what responses the student must preserve. Among adequate realizations, Unified Causal Minimum Description Length (UCMDL) selects a compact joint description of the teacher's relevant computation and its correspondence to the student. Adequacy is a prerequisite for description-length comparison.

Our controlled experiments use transformers compiled from known programs with Tracr (Lindner et al., 2023). HCD selects BPC on both tasks, and the evaluated students achieve lower ordinary and intervention response errors than behavioral KD. The independently trained frequency-sorting student also achieves low error on the required state-intervention family. In the pretrained objective comparison, Qwen3.5 students trained with BPC achieve 54.2% pooled accuracy on seven generated arithmetic tasks, versus 45.8% for BP and 41.7% for behavioral KD.

Novelty and contributions. Our novelty lies in formulating causal supervision as a selection problem: which combination of objectives yields a compact teacher–student correspondence that preserves the required intervention responses? Our contributions are threefold. *First*, we introduce HCD's partial order of behavioral, program, circuit, and state supervision. It separates the objectives used to train a student from the common causal requirements used to evaluate it, making supervision granularity an explicit choice. *Second*, we formulate UCMDL to select the shortest adequate joint description of a teacher mechanism and its correspondence to the student. A finite-family confidence bound defines adequacy for the selected trained realization under the specified intervention distributions. *Third*, we evaluate the supervision objectives on two controlled transformers and pretrained students on seven generated arithmetic tasks. The controlled BPC students achieve lower ordinary and intervention errors than behavioral KD, including low frequency-sorting state-intervention error under the shared evaluation-alignment procedure. On the generated arithmetic tasks, pretrained BPC students achieve the highest pooled base-answer accuracy and lowest teacher–student total variation (TV) distance among B, BP, and BPC. These findings connect the choice of supervision objectives to the response families preserved by the evaluated students.

2 Related Work

Reasoning-oriented distillation. Teacher rationales can supervise smaller models (Hsieh et al., 2023), while recent methods emphasize key reasoning steps (Feng et al., 2024a), use executable programs to check generated solutions (Zhu et al., 2024), or augment training with counterfactual inputs and diverse rationales (Feng et al., 2024b). These works enrich the examples or traces supplied to the student. HCD asks which responses to interventions *inside* the teacher and student must be preserved, then compares objective sets against that common requirement. Input-level counterfactual augmentation and internal causal substitution address different questions.

Causal distillation. Knowledge distillation transfers output distributions and can also match intermediate representations (Hinton et al., 2015; Sun et al., 2019). Interchange Intervention Training encourages neural computations to follow an abstract causal model (Geiger et al., 2022). DIITO applies interchange interventions to aligned hidden states and trains the student to match the teacher's intervened outputs (Wu et al., 2022). Circuit Distillation aligns representations of functionally corresponding circuit components (Wadhwa et al., 2025). These approaches specify an internal correspondence as part of the objective. HCD compares supervision configurations under common intervention requirements and selects an adequate realized correspondence by description length.

Causal abstraction and circuit testing. Causal abstraction relates computations through their responses to interventions (Rubenstein et al., 2017; Geiger et al., 2025). Distributed Alignment Search learns correspondences between abstract variables and neural subspaces (Geiger et al., 2024), while circuit discovery and testing examine whether selected routes preserve behavior and how localized they are (Conmy et al., 2023; Shi et al., 2024). Unrestricted alignment maps can make an abstraction uninformative (Sutter et al., 2025). HCD specifies a correspondence family before selection and includes its description in the objective. It seeks a compact cross-model relation that satisfies the task's intervention requirements.

Description length and controlled models. Minimum description length compares models through concise descriptions (Rissanen, 1978; Grünwald, 2007), including descriptions of causal structure (Wendong et al., 2025). UCMDL describes a teacher's relevant computation jointly with the correspondence that relates it to a student. Tracr compiles RASP programs into transformers with known computational structure (Weiss et al., 2021; Lindner et al., 2023). This setting supports direct evaluation of how supervision granularity affects the transfer of specified causal dependencies.

3 Hierarchical Causal Distillation

HCD takes a teacher, a student architecture, and a set of required behavioral and intervention responses. It trains candidate supervision configurations independently, evaluates their causal adequacy, and selects a compact adequate realization. The framework separates three decisions: which responses define success, which objectives train the student, and which adequate correspondence has the shortest description.

3.1 A Hierarchy of Causal Supervision

A *program* specifies semantic intermediate variables and their updates. A *circuit* comprises neural operations and routes that realize those updates. A *state* is an internal representation at a specified model site. Table 1 summarizes the corresponding supervision objectives. Behavioral supervision is included throughout.

Table 1: Information supplied by each supervision objective. In the controlled study, P also supervises semantic intermediate values. The S objective adds a distinct family of state-intervention response targets.

Objective	What the student matches	Example for $z = a + b$, $y = zc$
Behavior (B)	Ordinary teacher outputs and task-specific answer constraints	Predict y on the original input.
Program (P)	Responses under semantic intermediate substitutions	Replace z and recompute the answer.
Circuit (C)	Responses to interventions on computational routes	Block a route carrying z and compare its effect.
State (S)	Responses to additional internal-state substitutions	Substitute a specified internal state and match the response.

The supervision hierarchy is

$$B \longrightarrow BP \longrightarrow \left\{ \begin{array}{c} BPC \\ BPS \end{array} \right\} \longrightarrow BPCS. \tag{1}$$

Each arrow adds a supervision objective. This is a partial order: BPC and BPS extend BP with different constraints. Candidates are trained independently and assessed against common requirements.

A task uses branches with meaningful intervention families. Behavioral KD (B) is eligible whenever it meets all requirements. Program supervision already uses semantic internal correspondences; S adds response targets from a distinct state-intervention family, supporting the BPS and $BPCS$ branches.

3.2 Common Behavioral and Interventional Requirements

Each intervention specifies an abstract change and a model-specific correspondence that realizes it inside the teacher or student. The teacher correspondence defines the reference response. Student correspondences may use different sites or coordinates within a common permitted family. This allows models with different internal dimensions to express the same required causal relationship.

Let q_m^j denote model m's answer distribution after intervention j, with $j = 0$ denoting ordinary execution. The query and model-specific correspondence are implicit in this notation. For normalized Jensen–Shannon divergence $J(p, q) = D_{\mathrm{JS}}(p, q)/\ln 2 \in [0, 1]$, the controlled-study response errors are

$$D_B(T, S) = \mathbb{E}_x J(q_T^0, q_S^0), \qquad D_k(T, S) = \mathbb{E}_{\xi \sim \mu_k} J(q_T^j, q_S^j), \quad k \in \{P, C, S\}. \tag{2}$$

Here $\xi = (x_b, x_d, j)$ is an intervention query and μ_k its sampling distribution. Lower error indicates closer agreement. A task fixes its required criteria $\mathcal{K}$ and tolerances τ_k before comparison. We write $D_k(c)$ when making the dependence on a realized student and its correspondence explicit.

In the controlled study, program adequacy also checks agreement with the executable program and the accuracy of decoded semantic intermediate values. These criteria apply to every candidate, including B. A semantic readout tests whether a representation encodes the required value; D_S tests the answer produced after a state substitution. Program supervision can therefore contain semantic-state targets while a distinct state-intervention objective remains optional. Appendix A defines both errors and their aggregation.

Thus a BPC student can be required to satisfy a state-intervention criterion. This evaluates its responses on an additional intervention family under the common correspondence-fitting procedure. The pretrained comparison reports total variation on ordinary, program-intervention, and circuit-intervention queries, as defined in Section 4.

3.3 Supervision Objectives

Program supervision. Program interventions replace a semantic intermediate variable and recompute its descendants. The teacher correspondence is validated against the executable program. In

the controlled study, the student learns from the resulting program answer and required semantic intermediate values. In the language-model study, P adds teacher answer-distribution matching on program-intervention queries.

Each model substitutes a donor representation from its own computation at the site assigned to the semantic variable. For base and donor activations h_m^b, h_m^d and an orthonormal correspondence basis $U_{m,v}$ for variable v, the substitution is

$$\widetilde{h}_m^b = h_m^b + U_{m,v}U_{m,v}^{\mathsf{T}}(h_m^d - h_m^b), \qquad U_{m,v}^{\mathsf{T}}U_{m,v} = I. \tag{3}$$

The intervention replaces the selected coordinates while preserving their orthogonal complement. The model then recomputes downstream activations to produce the intervened answer. A semantic readout assesses the value encoded at the assigned site; the model's own computation determines its effect on the answer. This construction permits different teacher and student dimensions while preserving a common semantic intervention. Appendix A specifies joint substitutions and the complete losses.

Circuit supervision. Circuit supervision matches responses when computational routes associated with the program are changed. In the controlled study, it also matches how blocking those routes alters the effect of a program intervention. For model m, intervention j, and a block B_R on roles R, this effect is

$$M_{m,R} = (q_m^j - q_m^0) - (q_m^{j,B_R} - q_m^{0,B_R}). \tag{4}$$

The first difference measures the intervention's effect with routes intact; the second measures its effect with those routes blocked. Their contrast measures the intervention-by-block interaction under the specified reference policy. Appendix A gives the controlled-study loss; the language-model study matches teacher distributions on its circuit-intervention queries.

State supervision. State supervision matches responses under an additional internal-state substitution family. For frequency sorting, this family includes joint substitutions beyond the single-variable program training distribution. It supplies the alternative BPS branch in Eq. (1).

For a supervision configuration h, the training objective is

$$\mathcal{L}_h = \mathcal{L}_B + \sum_{\ell \in h \setminus \{B\}} \lambda_\ell \mathcal{L}_\ell + \lambda_{\text{map}} \Omega(\Phi_S). \tag{5}$$

The behavioral term $\mathcal{L}_B$ combines ordinary output distillation with task-specific answer constraints. The correspondence Φ_S relates the student to the required semantic variables and circuit roles; Ω regularizes its learned readouts when used. Appendix A specifies the controlled-study objectives and intervention operators. Appendix D gives the language-model query construction and metrics. Evaluation correspondences are estimated within the same permitted family for all methods, using examples separate from final evaluation.

3.4 Selecting a Compact Adequate Realization

A realization $c = (h, S, a, \Phi_S)$ combines a supervision configuration h, a trained student S, a teacher causal description $a = (P, \mathcal{C}_T, \Phi_T)$, and a student correspondence Φ_S. The teacher description contains the semantic program P, its retained circuit $\mathcal{C}_T$, and the map Φ_T relating program variables and circuit roles to the teacher. UCMDL measures the joint description length

$$L_{\text{UC}}(c) = \underbrace{L(P, \mathcal{C}_T, \Phi_T \mid T)}_{\text{teacher description}} + \underbrace{L(\Phi_S \mid a, S)}_{\text{student correspondence}}. \tag{6}$$

With model weights and task requirements as side information, the code measures causal-description and correspondence complexity. Appendix B.1 describes the coding scheme.

Let $\mathcal{F}$ be the evaluated realizations for a task and student architecture, using validated teacher descriptions. HCD first forms the adequate set

$$\mathcal{F}_{\text{adq}} = \left\{ c \in \mathcal{F} : \overline{D}_k(c) \le \tau_k \text{ for every } k \in \mathcal{K} \right\}, \tag{7}$$

where $\overline{D}_k$ is an upper confidence bound on a required error. The intervention families, permitted correspondences, and tolerances are shared across candidates. For a nonempty adequate set, HCD selects

$$c^* \in \arg\min_{c \in \mathcal{F}_{\text{adq}}} L_{\text{UC}}(c). \tag{8}$$

When the adequate set is empty, the procedure reports that no candidate meets the specified requirements. Appendix B gives the finite-family confidence bound. Selection and final evaluation use disjoint data.

Finite-family adequacy guarantee. Condition on teacher validation, student training, and alignment fitting, and fix the finite candidate family before examining selection data. For $M = |\mathcal{F}|$ realizations, $K = |\mathcal{K}|$ bounded criteria, and n_k independent observations for criterion k, use

$$\overline{D}_k(c) = \min\left\{1, \widehat{D}_k(c) + \sqrt{\frac{\ln(MK/\delta)}{2n_k}}\right\}. \tag{9}$$

Hoeffding's inequality and a union bound give simultaneous coverage of all candidate–criterion pairs with probability at least $1 - \delta$. Thus selecting the shortest candidate whose upper bounds meet every tolerance retains the adequacy guarantee under the specified intervention distributions. The guarantee concerns the selected trained realization. Independently trained students are evaluated separately. Appendix B gives the derivation and sampling conditions.

Shared correspondence estimation. Evaluation correspondences are fitted on separate alignment data with student weights held fixed. All configurations, including behavioral KD, use the same permitted sites, ranks, and readout family. Maps are held fixed during selection.

Additional supervision can yield a simpler correspondence, and one configuration can yield correspondences of different complexity. HCD selects the shortest adequate realization within the evaluated family.

4 Experimental Setup

The controlled study tests adequacy-constrained selection. The pretrained study compares B, BP, and BPC objectives using ordinary and intervention queries.

4.1 Controlled Transformers

We use two RASP programs compiled by Tracr (Weiss et al., 2021; Lindner et al., 2023): count-less-frequency and frequency sorting. Both operate on length-six sequences over four symbols, with program-defined semantic variables and intervention sites. Students preserve teacher depth with narrower residual streams: 12 versus 29 dimensions for count-less-frequency and 32 versus 82 for sorting. Each task uses 12,000 training queries and 2,048 queries for each of alignment, development, selection, and testing. Complete base sequences are partitioned before donor pairing, and sampled queries may reuse sequences within a partition. Appendix C provides the model, data, and optimization details.

Table 2 lists the candidate configurations and common response requirements. Both tasks also require executable-program agreement and semantic-state readout accuracy. Student tolerances are 0.05 for response errors and 0.10 for semantic-state readout errors. HCD applies these thresholds to selection-set upper bounds and uses UCMDL to compare adequate realizations. The selected configuration is then evaluated in an independent training run on separate test examples.

Table 2: Controlled-study requirements and candidate supervision configurations. Every configuration for a task is assessed against the same required response families.

Task	Required response families	Supervision configurations
Count-less-frequency	B, P, C	B, BP, BPC
Frequency sorting	B, P, C, S	B, BP, BPC, BPS, $BPCS$

4.2 Mathematical Reasoning with Pretrained Language Models

We evaluate on generated four-choice arithmetic questions with explicit intermediates, inspired by calculation structures in GSM8K (Cobbe et al., 2021) and SVAMP (Patel et al., 2021). We compare Qwen3.5-2B students supervised by a Qwen3.5-4B teacher (Qwen Team, 2026) under B, BP, and BPC on the seven programs in Table 3.

Each family computes an intermediate value $z = f(a, b)$ and a final answer $y = g(z, c)$ (Table 3). A base question is paired with a donor question. The program counterfactual inserts the donor's intermediate value into the base question's final step: $y_{\mathrm{cf}} = g_{\mathrm{base}}(z_{\mathrm{donor}}, c_{\mathrm{base}})$. The choices include the base, donor, and counterfactual answers; retained pairs have distinct values for all three. This defines a target for the changed intermediate's effect.

Table 3: Generated arithmetic task families. Each question computes an intermediate value z before producing the final answer y.

Task family	Intermediate z	Final answer y
Sum–multiply	$a + b$	zc
Product–subtract	ab	$z - c$
Difference–multiply	$a - b$	zc
Exact division	$a + b$	z/c
Maximum–add	$\max(a, b)$	$z + c$
Conditional adjustment	$a + b$	$z + c$ if $z \geq 20$; $z - c$ otherwise
Remainder	$a + b$	$z \bmod c$

For example, a sum–multiply base with $(a, b, c) = (3, 4, 5)$ has $z = 7$ and answer 35. A donor with $(8, 4, 3)$ has $z = 12$ and answer 36. Substituting $z = 12$ into the base's final step gives the counterfactual answer 60.

Identifying causal correspondences. The solution program supplies semantic intermediates. Candidate states are tested for encoding an intermediate and transferring its effect under substitution; circuit candidates are tested for their role in this effect. Correspondences are validated on separate examples and define the distillation interventions.

Distillation protocol. The students share initial weights, training examples, and an optimization schedule. Each task uses separate data partitions for student training, validation, and testing. Discovery and calibration data are used to identify and fit interventions. The loss matches the teacher's answer-choice distribution and encourages probability mass on valid choices. KD uses ordinary base and donor queries; BP adds program-intervention queries; BPC adds both program- and circuit-intervention queries. Appendix D describes the query construction and metric definitions.

Metrics. Accuracy measures the fraction of generated base questions answered correctly. Teacher–student agreement is measured by total variation distance, $\mathrm{TV}(p, q) = \frac{1}{2}\sum_i |p_i - q_i|$, between their answer distributions on the same query. Mean TV is reported separately for ordinary (B), program-intervention (P), and circuit-intervention (C) queries. Accuracy evaluates base-answer correctness; TV evaluates agreement with the teacher's answer-choice distribution.

5 Experimental Results

We first report controlled-study response errors, then compare pretrained students on generated arithmetic tasks.

5.1 Controlled Transformers

Table 4 compares independently trained students at the selected BPC configuration with behavioral KD on separate test examples. Each response error is an average normalized Jensen–Shannon divergence between a deterministic teacher answer and the student's answer distribution. State-intervention error D_S is diagnostic for count-less-frequency and required for frequency sorting.

Table 4: Controlled-transformer response errors. Lower is better; values lie in $[0, 1]$. HCD selects BPC on both tasks. The state response is a required criterion for frequency sorting.

Task	Method	Supervision	D_B	D_P	D_C	D_S
Count-less-frequency	HCD	BPC	**0.0050**	**0.0138**	**0.0090**	**0.0123**
	KD	B	0.0125	0.7072	0.9052	0.7144
Frequency sorting	HCD	BPC	**0.0004**	**0.0092**	**0.0003**	**0.0077**
	KD	B	0.0567	0.5469	0.6692	0.4851

5.1.1 Interventions distinguish students with similar ordinary outputs

On count-less-frequency, behavioral KD attains ordinary error $D_B = 0.0125$, alongside program and circuit errors of 0.7072 and 0.9052. HCD achieves low ordinary error (0.0050) and reduces the intervention errors to 0.0138 and 0.0090. The intervention queries distinguish these students' responses to changes in semantic intermediates and computational routes.

Frequency sorting shows the same pattern across response families. HCD reduces ordinary error from 0.0567 to 0.0004, program error from 0.5469 to 0.0092, and circuit error from 0.6692 to 0.0003. Both ordinary and intervention agreement improve in the reported results.

5.1.2 Program-and-circuit supervision yields low state-intervention error

The selection procedure chooses BPC on both tasks. Frequency sorting includes BPS and $BPCS$ as alternatives and requires a distinct state-intervention response. A separately trained student at the selected BPC configuration achieves $D_S = 0.0077$ without an S objective in student-weight training, compared with 0.4851 for behavioral KD.

The state family includes joint substitutions beyond the single-variable program training distribution. Evaluation correspondences are fitted with student weights held fixed, using program targets for program comparisons and teacher targets for circuit and state comparisons. Under this shared procedure, the BPC student achieves low response error on the additional state query family.

P supplies semantic intermediate targets, and C supplies route-response targets. The student attains the reported state-intervention agreement with residual width 32, compared with the teacher's 82.

5.2 Pretrained Models on Generated Arithmetic Tasks

Table 5: Pooled test results for Qwen3.5-2B students distilled from a Qwen3.5-4B teacher on seven generated arithmetic tasks. Accuracy is measured on held-out base questions; higher accuracy and lower teacher–student TV are better.

Supervision	Accuracy (%) ↑	Mean TV B ↓	Mean TV P ↓	Mean TV C ↓
KD (B)	41.7	0.248	0.246	0.249
BP	45.8	0.193	0.192	0.181
BPC	**54.2**	**0.166**	**0.165**	**0.151**

On the pooled test set, accuracy increases from 41.7% for behavioral KD to 45.8% for BP and 54.2% for BPC (Table 5). Calculated from the unrounded accuracies, the BPC result is 12.5 percentage points above KD and 8.3 points above BP across the seven generated tasks.

Across the pooled test examples, BPC has the lowest mean TV in all three query families, followed by BP and KD. The same ordering holds for base-answer accuracy, so BPC improves both answer correctness and teacher agreement in this comparison.

6 DISCUSSION

6.1 IMPLICATIONS FOR CAUSAL SUPERVISION

State responses with program and circuit supervision. The frequency-sorting result shows that student-weight training with BPC can yield low error on the required state-intervention family under the common evaluation alignment (Table 4). P supervises semantic values, and C constrains downstream route effects. These signals may support the observed state responses.

Common intervention requirements. All candidates share base and donor distributions, intervention targets, block references, permitted correspondences, and error tolerances. Correspondences are fitted on separate alignment data and held fixed during selection. Adequacy is defined by expected error under these intervention distributions, giving the description-length comparison a common set of response requirements.

Compact causal correspondences. Adequacy determines eligibility, and the joint code compares eligible teacher–student correspondences. Additional supervision can yield a simpler map: for example, a student trained with more objectives could admit a lower-rank correspondence that needs fewer bits to describe. A common program grammar, map family, and coefficient precision define the comparison. With model weights as side information, UCMDL measures the complexity of the causal description and correspondence.

Agreement across different representation widths. The controlled students achieve low intervention response errors with narrower residual streams than their teachers. Fitted correspondences link semantic variables and circuit roles across the two models. These results illustrate how causal agreement can be evaluated across different representation widths through the specified sites, subspaces, and intervention distributions.

Program and circuit queries for pretrained students. The pretrained comparison shows higher pooled base-answer accuracy and closer teacher agreement as program and circuit queries are added to behavioral KD. BPC achieves the strongest result among the three evaluated objective sets on both measures. This supports using program and circuit queries as additional supervision for the generated arithmetic tasks.

6.2 LIMITATIONS AND FUTURE WORK

Our evaluation covers two controlled tasks with known programs and seven generated arithmetic tasks with pretrained models. The pretrained study compares objective sets with different training query sets and reports pooled base-answer accuracy and teacher–student TV. Future work will extend the full selection procedure to pretrained models, broaden the task and model coverage, and evaluate matched query budgets and executable counterfactual accuracy.

7 CONCLUSION

HCD makes the choice of causal supervision explicit by evaluating objective sets against common response requirements and comparing the description lengths of adequate realizations. In the controlled transformers, the procedure selected BPC for both tasks. The evaluated BPC students achieved lower ordinary and intervention response errors than behavioral KD. The separately trained frequency-sorting student also achieved low state-intervention error under the common evaluation alignment, with BPC objectives for student-weight training.

On the generated arithmetic tasks, pretrained BPC students reached 54.2% pooled base-answer accuracy and the lowest teacher–student TV among the compared objectives. Together, these results support program and circuit supervision for improving the evaluated students' answer accuracy and agreement with teacher intervention responses. HCD organizes these objectives around task-specific causal requirements and a compact cross-model description.

AI USE STATEMENT

AI assistants supported literature search, code implementation, interpretation of results, drafting, and language revision. The authors are responsible for the content of the submitted manuscript.

ETHICS STATEMENT

Causal distillation transfers specified computational dependencies from a teacher to a student. These dependencies can include errors and undesirable behavior. Applications should therefore evaluate task correctness, harmful responses, and relevant safety outcomes alongside teacher agreement.

REPRODUCIBILITY STATEMENT

Section 3 defines the supervision hierarchy, causal requirements, selection rule, and finite-family bound. Appendix A specifies intervention operators and training objectives; Appendix B gives the description-length code and the bound's derivation. Appendix C reports the controlled-study protocol, and Appendix D defines the generated-task construction and evaluation metrics.

REFERENCES

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 arXiv:2110.14168*, 2021. URL https://arxiv.org/abs/2110.14168.

Arthur Conmy, Augustine Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. Towards automated circuit discovery for mechanistic interpretability. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://arxiv.org/abs/2304.14997.

Kaituo Feng, Changsheng Li, Xiaolu Zhang, Jun Zhou, Ye Yuan, and Guoren Wang. Keypoint-based progressive chain-of-thought distillation for LLMs. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 13241–13255. PMLR, 2024a. URL https://proceedings.mlr.press/v235/feng24e.html.

Tao Feng, Yicheng Li, Chenglin Li, Hao Chen, Fei Yu, and Yin Zhang. Teaching small language models reasoning through counterfactual distillation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 5831–5842, 2024b. doi: 10.18653/v1/2024.emnlp-main.333. URL https://aclanthology.org/2024.emnlp-main.333/.

Atticus Geiger, Zhengxuan Wu, Hanson Lu, Josh Rozner, Elisa Kreiss, Thomas Icard, Noah Goodman, and Christopher Potts. Inducing causal structure for interpretable neural networks. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pp. 7324–7338. PMLR, 2022. URL https://proceedings.mlr.press/v162/geiger22a.html.

Atticus Geiger, Zhengxuan Wu, Christopher Potts, Thomas Icard, and Noah Goodman. Finding alignments between interpretable causal variables and distributed neural representations. In *Proceedings of the Third Conference on Causal Learning and Reasoning*, volume 236, pp. 160–187. PMLR, 2024. URL https://proceedings.mlr.press/v236/geiger24a.html.

Atticus Geiger, Duligur Ibeling, Amir Zur, Maheep Chaudhary, Sonakshi Chauhan, Jing Huang, Aryaman Arora, Zhengxuan Wu, Noah Goodman, Christopher Potts, and Thomas Icard. Causal abstraction: A theoretical foundation for mechanistic interpretability. *Journal of Machine Learning Research*, 26(83):1–64, 2025. URL https://jmlr.org/beta/papers/v26/23-0058.html.

Peter D. Grünwald. *The Minimum Description Length Principle*. MIT Press, 2007.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015. URL https://arxiv.org/abs/1503.02531.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963. doi: 10.1080/01621459.1963.10500830.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 8003–8017, 2023. doi: 10.18653/v1/2023.findings-acl.507. URL https://aclanthology.org/2023.findings-acl.507/.

David Lindner, János Kramár, Sebastian Farquhar, Matthew Rahtz, Thomas McGrath, and Vladimir Mikulik. Tracr: Compiled transformers as a laboratory for interpretability. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://arxiv.org/abs/2301.05062.

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. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.naacl-main.168. URL https://aclanthology.org/2021.naacl-main.168/.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

Jorma Rissanen. Modeling by shortest data description. *Automatica*, 14(5):465–471, 1978. doi: 10.1016/0005-1098(78)90005-5.

Paul K. Rubenstein, Sebastian Weichwald, Stephan Bongers, Joris M. Mooij, Dominik Janzing, Moritz Grosse-Wentrup, and Bernhard Schölkopf. Causal consistency of structural equation models. In *Proceedings of the 33rd Conference on Uncertainty in Artificial Intelligence*, 2017. URL https://arxiv.org/abs/1707.00819.

Claudia Shi, Nicolas Beltran-Velez, Achille Nazaret, Carolina Zheng, Adrià Garriga-Alonso, Andrew Jesson, Maggie Makar, and David M. Blei. Hypothesis testing the circuit hypothesis in LLMs. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-2997. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/abccb8a90b30d45b948360ba41f5a20f-Abstract-Conference.html.

Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. Patient knowledge distillation for BERT model compression. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pp. 4323–4332, 2019. doi: 10.18653/v1/D19-1441. URL https://aclanthology.org/D19-1441/.

Denis Sutter, Julian Minder, Thomas Hofmann, and Tiago Pimentel. The non-linear representation dilemma: Is causal abstraction enough for mechanistic interpretability? In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-5000. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/dbb98528c9870377f3f0d133aae6050b-Abstract-Conference.html.

Somin Wadhwa, Silvio Amir, and Byron C. Wallace. Circuit distillation, 2025. URL https://arxiv.org/abs/2509.25002.

Gail Weiss, Yoav Goldberg, and Eran Yahav. Thinking like transformers. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pp. 11080–11090. PMLR, 2021. URL https://proceedings.mlr.press/v139/weiss21a.html.

Liang Wendong, Simon Buchholz, and Bernhard Schölkopf. Algorithmic causal structure emerging through compression. In *Proceedings of the Fourth Conference on Causal Learning and Reasoning*, volume 275 of *Proceedings of Machine Learning Research*, pp. 201–242. PMLR, 2025. URL https://proceedings.mlr.press/v275/wendong25a.html.

Zhengxuan Wu, Atticus Geiger, Joshua Rozner, Elisa Kreiss, Hanson Lu, Thomas Icard, Christopher Potts, and Noah Goodman. Causal distillation for language models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4288–4295, 2022. doi: 10.18653/v1/2022.naacl-main.318. URL https://aclanthology.org/2022.naacl-main.318/.

Xuekai Zhu, Biqing Qi, Kaiyan Zhang, Xinwei Long, Zhouhan Lin, and Bowen Zhou. PaD: Program-aided distillation can teach small models reasoning better than chain-of-thought fine-tuning. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2571–2597, 2024. doi: 10.18653/v1/2024.naacl-long.142. URL https://aclanthology.org/2024.naacl-long.142/.

APPENDIX: SUPPLEMENTARY MATERIAL

A INTERVENTIONS AND TRAINING OBJECTIVES

A.1 PROGRAM AND STATE SUBSTITUTIONS

An intervention query $\xi = (x_b, x_d, j)$ specifies a base input, a donor when needed, and the intervened variables or routes. A program intervention replaces a variable's update with its donor value and recomputes descendants. Simultaneous substitutions use values from the same unmodified donor computation. An undefined intervened answer is assigned to the additional answer class $\perp$.

Equation (3) defines the substitution of donor coordinates into a base activation for model m and variable v. Teacher and student each use donor activations from their own computation. Both the base and donor computations contribute to student optimization. Independently substituted variables at the same site use orthogonal subspaces; overlapping subspaces require a specified substitution order. These conventions determine the joint intervention when several variables are changed together.

The alignment Φ_m specifies sites, subspaces, circuit-role assignments, and semantic readouts. A readout $\alpha_{m,v}$ decodes a variable from its assigned coordinates for supervision and evaluation. The model's own downstream computation produces the answer. Intervention sites are internal representations preceding the answer logits.

A.2 BEHAVIOR AND PROGRAM LOSSES IN THE CONTROLLED STUDY

For task labels y and program responses q_P^j in the compiled-transformer study, the losses are

$$\mathcal{L}_B = \mathbb{E}_{(x,y)} \left[-\log q_S(y \mid x) + \lambda_{\mathrm{KD}} D_{\mathrm{KL}}(q_T^0 \| q_S^0) \right], \tag{10}$$

$$\mathcal{L}_P = \mathbb{E}_{\mu_P} \left[D_{\mathrm{KL}}(q_P^j \| q_S^j) + \frac{\eta}{|\mathcal{V}_\star|} \sum_{v \in \mathcal{V}_\star} \ell_v\big(\alpha_{S,v}(Z_S^j), z_{P,v}^j\big) \right]. \tag{11}$$

Z_S^j denotes the student's intervened intermediate representations, $z_{P,v}^j$ is the program's semantic target, and $\mathcal{V}_\star$ is the set of required variables. Categorical targets use cross entropy; a numerical target with range $[a_v, b_v]$ uses squared error normalized by $(b_v - a_v)^2$. The program answer is deterministic in the controlled tasks, so its KL term is cross entropy. All training divergences use natural logarithms, and sequence losses average over non-BOS positions. When the ordinary teacher answer equals the task label, the two terms in $\mathcal{L}_B$ reduce to a rescaled cross-entropy loss.

A.3 CIRCUIT AND STATE LOSSES IN THE CONTROLLED STUDY

A role r maps to a set of model messages $E_{m,r}$. A block B_R replaces the messages in $\bigcup_{r \in R} E_{m,r}$ with the prescribed reference values. Singleton and joint blocks use one shared assignment. Each model's reference values remain constant across the four response conditions of a circuit query.

For a program intervention j and a role set R, the contrast $M_{m,R}$ is defined in Eq. (4). The circuit loss is

$$\mathcal{L}_C = \mathbb{E}_{\xi,R} \left[\frac{1}{4} \sum_{\substack{u \in \{0,j\} \\ b \in \{\varnothing, B_R\}}} D_{\mathrm{KL}}(q_T^{u,b} \| q_S^{u,b}) + \gamma \left\| M_{T,R} - M_{S,R} \right\|_2^2 \right]. \tag{12}$$

The contrast compares how a block changes the program intervention's effect. Its penalty places additional optimization weight on the interaction computed from the four matched responses. State supervision uses the additional state query family:

$$\mathcal{L}_S = \mathbb{E}_{\mu_S} D_{\mathrm{KL}}(q_T^j \| q_S^j). \tag{13}$$

A distinct state objective requires a distinct query family or distinct teacher response targets. When its queries and targets coincide with the program targets, adding $\mathcal{L}_S$ reweights the same response constraint.

The alignment regularizer in Eq. (5) is $\Omega(\Phi_S) = \sum_{v \in \mathcal{V}_\star} (\|W_{S,v}\|_F^2 + \|b_{S,v}\|_2^2)$ for affine semantic readouts. The intervention bases remain orthonormal during optimization through QR orthonormalization.

A.4 Controlled-Study Response and Semantic-State Errors

All response errors use the normalized divergence

$$J(p, q) = \frac{D_{\mathrm{KL}}(p\|\bar{p}) + D_{\mathrm{KL}}(q\|\bar{p})}{2 \ln 2}, \qquad \bar{p} = \tfrac{1}{2}(p + q). \tag{14}$$

Per-position divergences are averaged within each sequence to give one bounded observation. For two point-mass answers, J is zero for agreement and one for disagreement. For a softmax student, J measures distributional discrepancy, including the probability assigned to the teacher's answer.

Circuit response error applies J to the designated circuit intervention queries, using corresponding blocked roles and the same reference policy in both models. Singleton and joint blocks are sampled under the specified intervention distribution. The four-condition profile in Eq. (12) specifies the training objective. Program response error D_P compares teacher and student. Program adequacy additionally compares the student with the executable program's response q_P^j and its semantic states. For categorical states, the bounded readout error d_v is disagreement with the program state. For numerical states,

$$d_v(\widehat{z}, z) = \min\left\{1, \frac{|\widehat{z} - z|}{b_v - a_v}\right\}. \tag{15}$$

The semantic-state criterion takes the maximum d_v over required variables for each query, then averages over queries. It evaluates decoded intermediate values; D_S evaluates the final answer distribution after a state intervention. These are separate criteria in $\mathcal{K}$. The student response tolerance of 0.05 applies to the required teacher–student response errors and executable-program response error; the 0.10 tolerance applies to the semantic-state readout criterion. Count-less-frequency uses D_S as a diagnostic, while frequency sorting includes it in the adequacy decision.

B Description Length and Independent Selection

B.1 The Joint Description

The description in Eq. (6) expands as

$$\begin{aligned} L_T(a) &= L_{\mathcal{G}}(P) + L_C(\mathcal{C}_T \mid P) + L_{\Phi,T}(\Phi_T \mid P, \mathcal{C}_T), \\ L_{\mathrm{UC}}(c) &= L_T(a) + L_{\Phi,S}(\Phi_S \mid a, S). \end{aligned} \tag{16}$$

The program code describes operations, dependencies, constants, and any additional intermediate variables. The circuit code records retained messages and their role assignments. Alignment codes record sites, ranks, subspaces, and readout parameters. Information determined by an earlier component is shared through conditioning. The model weights, task requirements, and common coding conventions are given when describing the correspondence.

The coding scheme uses a prefix integer code, indices within specified finite families, and quantized coefficients. All candidates share a program grammar, enumeration order, integer code, and quantization convention specified before selection. A circuit retaining k of N messages is described by k, an index of width $\lceil \log_2 \binom{N}{k} \rceil$, and the role assignments. Learned coefficients are represented at 8- or 16-bit precision in $[-8, 8]$, with the precision and matrix dimensions also recorded. A decoded basis is orthonormalized by a deterministic QR decomposition. Adequacy is assessed using the represented map so that the evaluated correspondence matches its description.

Joint coding can favor a more detailed teacher description when it admits a simpler student alignment. It also distinguishes students trained with the same objectives when their correspondences have different complexity. The supervision configuration describes the training objectives, while the code length describes the realized causal correspondence. Equation (8) minimizes this length over the evaluated adequate candidates.

B.2 Teacher Validation and Student Alignment

For the controlled tasks, the source program and its compiled computation identify the teacher's semantic variables and intervention sites. Validation checks ordinary outputs, intervened outputs, and intermediate values against the executable program. If a candidate retains only part of the teacher circuit, omitted messages are replaced by the prescribed references and the retained model must preserve the required responses. Alternative teacher alignments are compared with the same reference teacher interventions. Adequate descriptions realize every required operation.

For evaluation, each method uses the same data allocation and permitted alignment family. Correspondence estimation fits the sites, subspaces, and readouts that relate the student to the teacher description, with student weights held fixed. Program comparisons use program targets; circuit and direct state comparisons use teacher targets. All configurations, including behavioral KD, use this procedure before selection data are examined.

B.3 Adequacy on an Independent Sample

Condition on teacher validation, student training, alignment estimation, and any development choices. In Eq. (9), M counts the preselection realizations, K counts the required bounded criteria, and n_k counts independent observations from the specified law for criterion k. The confidence level is set by $\delta \in (0, 1)$. For each fixed candidate and criterion, Hoeffding's inequality bounds the probability that the expected error exceeds the empirical mean by ϵ by $\exp(-2n_k\epsilon^2)$ (Hoeffding, 1963). Substituting $\epsilon = \sqrt{\ln(MK/\delta)/(2n_k)}$ gives a probability at most $\delta/(MK)$. A union bound over the candidate–criterion pairs yields simultaneous coverage at least $1 - \delta$. On this event, every member of $\mathcal{F}_{\mathrm{adq}}$ satisfies all required expected-error tolerances, including the member chosen by description length.

Teacher criteria assessed on the same selection sample are included among the required bounds. The family size M counts every trained-model/correspondence realization whose selection errors are compared. The value of K counts individual bounded criteria, including program-answer and semantic-state checks.

Several interventions on one base can be averaged into one bounded observation; n_k then counts independent base draws. With a separate donor pool, independent base and donor-index draws define the sampling distribution conditional on that pool. The unit of observation follows the sampling design: grouped interventions sharing a base contribute one observation, while independently sampled queries can be treated separately. Exact enumeration gives the error over the specified finite query population. A bound conditional on a fixed base or donor pool certifies expected error under that pool's query distribution.

A candidate is adequate when every required upper bound meets its tolerance. The bound accounts jointly for empirical error, sample size, and the number of candidates and criteria. Its coverage applies to the evaluated realizations under their specified intervention distributions. An independently trained student at the selected supervision level receives a separate final evaluation; its performance is reported from independent test data.

C Controlled-Study Details

C.1 Programs, Teachers, and Students

Both tasks use the vocabulary {`a`, `b`, `c`, `d`} and six positions following a beginning-of-sequence (BOS) token. Count-less-frequency uses histogram counts to compare token frequencies. Frequency sorting orders tokens by frequency-derived keys with Tracr's stable tie rule. The designated program quantities include histogram counts and, for sorting, the downstream sorting computation. BOS and padding are excluded from semantic intervention targets.

Students use GELU, learned positional embeddings, and zero dropout. Their attention masks match the teacher tasks, and their initial weights are independent of the compiler weights. A single affine readout per semantic variable is shared across sequence positions. Intervention sites are residual outputs after attention or feed-forward operations at the selected program, circuit, and state locations. Alignment subspace ranks are $\{2, 4, 8\}$, subject to the available width.

Table 6: Controlled models. The student feed-forward width is twice its residual width. Teacher parameter counts are those specified for the compiled models.

Quantity	Count-less-frequency	Frequency sorting
Teacher transformer blocks	2	5
Teacher heads / residual width	1 / 29	1 / 82
Teacher parameters	4,287	109,356
Student transformer blocks	2	5
Student heads / residual width	2 / 12	4 / 32

C.2 Intervention Queries and Answers

The compiled program identifies where each required variable is computed. Substitution occurs after that computation and before the variable's downstream use, followed by recomputation of all descendants. Base and donor sequences have equal length under the primary query law. Whole-sequence and single-position substitutions are distinguished. Teacher validation compares the program, named intermediate computations, and compiled-transformer responses, including self-substitutions and identity blocks.

Count-less-frequency answers use the bounded histogram-count classes; sorting answers use the four token classes. Both answer sets include $\perp$. The teacher is decoded as a deterministic function. Numerical outputs are assigned to the nearest allowed answer within 10^{-4}; ambiguous or more distant values map to $\perp$. The student predicts a softmax distribution over the same answer classes. Reported response errors average over non-BOS positions and then over sampled queries.

Count-less-frequency requires B, P, C responses. Sorting additionally requires a distinct state-intervention family, including joint substitutions beyond the single-variable program training distribution. State-response measurements for count-less-frequency are reported as a separate diagnostic. Every method uses the same required families and tolerances for its task.

C.3 Data and Optimization

Complete base sequences are assigned to disjoint data partitions before intervention donors are paired. Each sequence belongs to one partition. The partition roles are teacher-description construction (10%), teacher validation (10%), training (40%), alignment estimation (10%), development (10%), selection (10%), and test (10%). Donors are drawn from separate pools within the corresponding partition. The reported sample counts refer to query examples, which may reuse sequences within a partition. The partition percentages describe assignment of base sequences; the query counts describe the subsequent sampling of base–donor interventions. For the confidence bound in Eq. (9), n_k follows the independent sampling unit described in Appendix B.

Table 7: Training and evaluation settings specified for the controlled study. The response and semantic-state tolerances apply to different error definitions.

Quantity	Setting
Optimizer	AdamW
Learning rates	3×10^{-4} for student weights; 10^{-3} for maps
Weight decay / gradient norm cap	10^{-2} / 1
Batch size	256 query examples
Training query examples	12,000
Alignment, development, selection, test	2,048 query examples per partition
Training update range	4,000–20,000 updates
Objective weights	$\lambda_{\mathrm{KD}} = \lambda_P = \lambda_C = \lambda_S = \eta = \gamma = 1$
Readout regularization	$\lambda_{\mathrm{map}} = 10^{-4}$
Map ranks / coefficient precision	$\{2, 4, 8\}$ / 8 or 16 bits
Teacher error tolerances	0.05 for each required bounded error
Student error tolerances	0.05 for responses; 0.10 for semantic-state readouts

For each task, HCD evaluates the supervision configurations in Table 2, including behavioral KD as the B configuration. Adequacy is assessed on independent selection data, and the shortest adequate realization determines the selected supervision level. Final performance is measured with an independently trained student at that level, using alignment data and test examples disjoint from selection. The selected supervision level is retained throughout final evaluation.

D Generated Mathematical-Reasoning Task Details

The generated four-choice tasks draw on calculation structures in GSM8K (Cobbe et al., 2021) and SVAMP (Patel et al., 2021), with explicit intermediates and program counterfactuals. A Qwen3.5-4B teacher (Qwen Team, 2026) supervises Qwen3.5-2B students using B, BP, or BPC. Table 5 reports base-answer accuracy and teacher agreement.

D.1 Base, Donor, and Counterfactual Answers

For each question, its task family defines an intermediate function f and a final-answer function g, as specified in Table 3. Let the base and donor intermediate values be $z_{\mathrm{b}} = f_{\mathrm{b}}(a_{\mathrm{b}}, b_{\mathrm{b}})$ and $z_{\mathrm{d}} = f_{\mathrm{d}}(a_{\mathrm{d}}, b_{\mathrm{d}})$. Their ordinary answers are $y_{\mathrm{b}} = g_{\mathrm{b}}(z_{\mathrm{b}}, c_{\mathrm{b}})$ and $y_{\mathrm{d}} = g_{\mathrm{d}}(z_{\mathrm{d}}, c_{\mathrm{d}})$. The program counterfactual keeps the base final step and substitutes the donor intermediate value:

$$y_{\mathrm{cf}} = g_{\mathrm{b}}(z_{\mathrm{d}}, c_{\mathrm{b}}). \tag{17}$$

The four answer choices contain y_{b}, y_{d}, and y_{cf}. A pair is retained only when these three answers are distinct.

D.2 Identifying Neural States and Circuits

The known solution program supplies the intermediate value for each question. Candidate state correspondences are tested for both encoding that value and transferring its effect when a representation from one example is substituted into another. Candidate circuit components are tested for whether they carry the transferred effect through to the answer. The resulting correspondences are validated on separate examples before student evaluation, using the known intermediate variables and their predicted effects on the answer.

The comparison focuses on program and circuit intervention families. State correspondences associated with semantic intermediate variables are used in program and circuit identification.

D.3 Distillation Protocol

The students start from identical Qwen3.5-2B weights and use shared data partitions and optimization settings. For each task, discovery data identify candidate interventions and calibration data are used to fit them. Separate partitions support student training, validation, and testing. The supervision configuration determines which queries from each sampled pair enter the loss.

For each query used in training, the objective matches the teacher's distribution over answer choices and encourages the student to place probability mass on valid answer choices. KD (B) uses ordinary base and donor queries. BP adds program-intervention queries, and BPC adds both program- and circuit-intervention queries. The known semantic intermediates define and validate those intervention queries.

D.4 Metric Interpretation

For teacher and student answer distributions p_x and q_x on the same query x, total variation distance is

$$\mathrm{TV}(p_x, q_x) = \frac{1}{2} \sum_i |p_x(i) - q_x(i)| \in [0, 1]. \tag{18}$$

Mean TV is computed separately for ordinary (B), program-intervention (P), and circuit-intervention (C) queries; zero indicates identical distributions and one disjoint distributions. Base-question accuracy measures answer correctness. Appendices A and C define the controlled-study normalized Jensen–Shannon errors and adequacy tolerances.