ADAPTING, FAST AND SLOW: ON FEW-SHOT TRANSPORTABILITY OF COMPOSITIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Rules learned from source sequences can be composed to learn new computations from few target samples. We study the conditions and sample requirements for this form of compositional generalization in finite sequential causal models without unobserved confounding. Causal structure and knowledge of shared mechanisms identify local conditional distributions that can be reused across domains. We give a sufficient condition under which their estimates, called gates, can be composed by Circuit-TR to estimate the target concept—the output conditional given the inputs—without target samples. When the target composition and its source correspondences are unknown, we prove that Circuit-AD competes with this oracle by minimizing empirical log loss with a circuit-size penalty. The excess-risk bound separates source estimation error from composition selection; the selection cost decreases with target sample size and adapts to the comparator's size without requiring it in advance. Different compositions can compute the same noiseless function yet incur different risks when their gates are noisy. Neural-AD learns compositions of frozen source gates, allowing us to examine these distinctions experimentally. The results show how the available source operations affect adaptation and how process supervision reduces the samples needed to learn a composition.

1 INTRODUCTION

A learner can use abundant source data to acquire reusable prediction rules, then compose them to learn a new concept from fewer target samples. This is a form of *compositional generalization*: familiar components are combined in unfamiliar ways (Wiedemer et al., 2023; Schug et al., 2024). Understanding when this reuse is possible, and how much target data it requires, is a central question for learning across domains.

Sequence learning offers a setting in which such rules can be acquired without additional task labels. An autoregressive model learns to predict each entry from its preceding context, using the observed sequence itself to supply the training targets (Bengio et al., 2003). Some sequences also record an ordered computation, such as the successive values in the execution trace of a computer program. Predicting these entries is particularly interesting for transfer: the mechanisms that generate them may reappear in another domain, at different positions and with different arguments. Learning from execution traces has accordingly been studied as a route to reusable computations (Reed & de Freitas, 2016; Xhonneux et al., 2021).

For example, arithmetic traces can supply rules for maximum, minimum, and subtraction. Composing these operations can compute GCD even if GCD never appears in the source data. Library learning and modular meta-learning exploit related forms of reuse (Ellis et al., 2021; Alet et al., 2018). When operations are noisy, their prediction rules are conditional distributions. We call an estimate of such a local conditional a *gate*, its ordered composition a *circuit*, and the circuit's output conditional given its inputs a *concept*. Learning a target concept therefore requires both accurate gates and a useful composition.

Causal *transportability* asks whether available distributions and knowledge of shared and changing mechanisms determine a target quantity (Pearl & Bareinboim, 2011; Correa & Bareinboim, 2019). Here causal knowledge can map a target operation to a conditional distribution estimable from source

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.

observations, even when its parents and position change. This turns mechanism correspondences into a procedure for learning the target concept from source data.

Such correspondences may be unavailable, and shared mechanisms can be obscured by a different composition. Some target operations may also lack direct source matches, while a longer composition still approximates the desired concept. Target observations then provide evidence for choosing among compositions of the available source rules. This is a supervised domain-adaptation problem. Seminal analyses relate source and target learning through distributional discrepancies and shared hypothesis classes (Blitzer et al., 2007; Ben-David et al., 2010; Mansour et al., 2009); our setting makes the compositional structure explicit. Adaptation selects a composition of source-fitted gates, whose usefulness is measured by its risk on the target concept. Appendix I situates this question within the causal, adaptation, and compositional-learning literatures.

We study this problem in finite sequences generated in causal order. Domain knowledge specifies the source and target structure and which mechanisms they share; it supplies an oracle benchmark for learning when the target structure and its source correspondences are unknown. *Outcome supervision* supplies inputs and final outputs; *process supervision* additionally supplies intermediate values. Both concern the same concept but provide different evidence about its composition. Our contributions are:

1. **Transport from causal knowledge.** Under no unobserved confounding, we give a sufficient condition identifying the target concept from source conditionals. Circuit-TR composes their estimates, with excess risk bounded by accumulated gate-estimation error under source coverage (Section 3).
2. **Few-shot learning of compositions.** We prove that Circuit-AD competes with the supplied Circuit-TR predictor using target input/output pairs. The bound separates source estimation from the cost of selecting a composition, and adapts to its size without knowing it in advance (Section 4).
3. **Neural-AD and experiments.** Neural-AD learns reusable gates through next-token training and adapts their composition. Paired experiments examine how process supervision and the available source operations affect this learning; a noise analysis explains why different composition lengths can yield different approximation risks (Sections 5 and 6; Appendix F).

2 Sequential Computations and Learning

We model a sequence as the successive values of an ordered computation. Its causal mechanisms are represented by a Structural Causal Model (SCM) (Pearl, 2009, Definition 7.1.1). Capital letters denote variables, lowercase letters their values, and bold letters collections. We write $[T] = \{1, \dots, T\}$ and $V_{a:b} = (V_a, \dots, V_b)$.

Definition 2.1 (Structural causal model)**.** An SCM is a tuple $\mathcal{M} = \langle \mathbf{V}, \mathbf{U}, \mathcal{F}, P_{\mathbf{U}} \rangle$. Each endogenous variable $V_i \in \mathbf{V}$ has a mechanism $f_i \in \mathcal{F}$ and is assigned by $V_i := f_i(\mathbf{Pa}_i, \mathbf{U}_i)$, where $\mathbf{Pa}_i$ is an ordered list of endogenous parents and $\mathbf{U}_i \subseteq \mathbf{U}$ consists of exogenous variables. The distribution $P_{\mathbf{U}}(\mathbf{u})$ specifies the exogenous variables; the structural equations induce the distribution P of $\mathbf{V}$.

The causal diagram has an edge $V_j \to V_i$ for each parent. The sequence follows a known causal order, so parents precede their children; their argument order records how a mechanism uses them. Unless stated otherwise, the SCM is *Markovian*: independent exogenous variables U_i enter the respective equations. With this absence of unobserved confounding,

$$P(\mathbf{v}) = \prod_i P(v_i \mid \mathbf{pa}_i), \qquad P(v_i \mid \mathbf{pa}_i) = \int \mathbb{1}\{f_i(\mathbf{pa}_i, u_i) = v_i\}\, dP_{U_i}(u_i). \tag{1}$$

A *circuit* composes these local conditionals according to fixed ordered parent lists, shared across inputs. A *gate* estimates one local conditional. Under Markovianity, that conditional is determined by the structural function and its noise distribution, which justifies reuse when both are shared. Appendix D explains why shared mechanisms can induce different parent-conditionals under unobserved confounding.

We observe N independent complete source traces D from $\mathcal{M}$, with distribution P and diagram $\mathcal{G}$, and n independent target samples $D^\star$ from $\mathcal{M}^\star$, with distribution $P^\star$ and diagram $\mathcal{G}^\star$. Sampling is

independent across domains, and typically $N \gg n$. Source and target sequences have lengths T and $T^\star$; all positions take values in a finite vocabulary $\mathcal{V}$. The target input is the prefix $\mathbf{X} = V^\star_{1:M}$ and its output is $Y = V^\star_{T^\star}$. Their conditional table $P^\star(y \mid \mathbf{x})$ is the target *concept*. Outcome supervision observes $(\mathbf{X}, Y)$; process supervision additionally observes the intermediate values. Both concern the same concept. Appendix E.2 treats multiple sources.

All fitted gates have probabilities at least $\epsilon > 0$, bounding their log loss. For the transport results, true distributions are strictly positive and ϵ is chosen below every local conditional probability. The source coverage conditions needed for finite-sample estimation are specified with the full results in Appendix B.

Learning is evaluated by log loss. A predictor $\mu(y \mid \mathbf{x})$ estimates the target concept and has *risk* and *excess risk*

$$\mathcal{R}(\mu) := \mathbb{E}_{P^\star}[-\log \mu(Y \mid \mathbf{X})], \qquad \mathcal{E}(\mu) := \mathcal{R}(\mu) - \mathcal{R}(\mu_*), \tag{2}$$

where $\mu_*(y \mid \mathbf{x}) = P^\star(y \mid \mathbf{x})$. Thus $\mathcal{E}(\mu) = \mathbb{E}_{\mathbf{X}} D_{\mathrm{KL}}(P^\star(\cdot \mid \mathbf{X}) \,\|\, \mu(\cdot \mid \mathbf{X}))$ and is zero exactly when the target conditional is matched almost surely. We use natural logarithms, so risk is measured in nats. Main-text rates suppress constants depending on vocabulary, parent arity, probability bounds, source coverage, and confidence; $\widetilde{\mathcal{O}}$ also suppresses logarithmic factors.

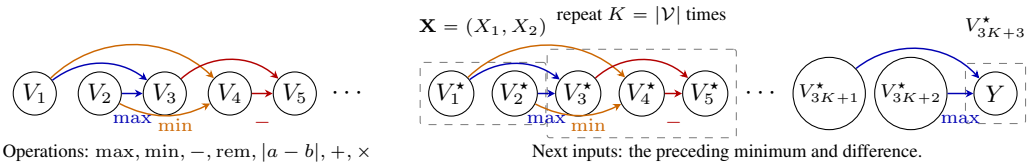
3 Circuit Transportability

We first ask what source data determine when domain knowledge supplies the target composition and its source correspondences. This separates the accuracy of the learned gates from the later task of discovering how to compose them.

3.1 A source library for GCD

Example 3.1 (Greatest common divisor)**.** The source in Figure 1 applies seven arithmetic operations, with several reused at later positions. Its complete traces supply observations for fitting their conditional distributions. The target repeats maximum, minimum, and subtraction a fixed number of times, then returns the maximum of the final pair. Without noise, $|\mathcal{V}|$ iterations suffice to compute GCD for every input pair in $\mathcal{V} = \{0, \ldots, 9\}$ (Equation (19)).

Both domains use this vocabulary; addition, subtraction, and multiplication are modulo ten, and $a \bmod 0 = a$. Each non-input operation independently returns a uniform value with probability 0.01, so identical operations have identical conditional tables. Given the target causal diagram, ordered parent lists, and corresponding source positions, we can fit its gates from source traces and marginalize the intermediate values to estimate its concept. The inputs and their distribution may change while these local conditionals remain shared.



(a) Source circuit (14 positions).

(b) GCD from maximum, minimum, and subtraction.

Figure 1: A source computation and a GCD composition of its operations. Circles are circuit positions; edge labels identify operations. Ellipses omit positions and their incident edges. The complete circuits and noise model are given in Appendix B.1.

3.2 Identification and risk

Domain knowledge is encoded by the source and target diagrams, including their ordered parent lists, and a discrepancy oracle specifying mechanism correspondences.

Definition 3.2 (Discrepancy oracle)**.** The discrepancy oracle $\Delta(i, i')$ is zero when the supplied information certifies that target position i and source position i' have the same arity, structural

function of the ordered arguments, and exogenous noise distribution. A value of one imposes no equality requirement.

The matched positions may have different parents and different indices. Under Markovianity, Equation (1) then implies equality of their conditional distributions after aligning the ordered arguments.

Definition 3.3 (Circuit transportability)**.** The query $P^\star(v^\star_{T^\star} \mid v^\star_{1:M})$ is circuit-transportable from P given $\Delta, \mathcal{G}, \mathcal{G}^\star$ if it has the same value for every source–target SCM pair whose source model entails P and that satisfies those diagrams and mechanism-sharing constraints.

A sufficient condition is that every target position after the observed prefix has a source match. Positivity identifies its conditional on all required arguments. This is a local sufficient criterion for transportability; other queries can be identified without matching every intermediate conditional (Appendix I.1). Circuit-TR fits the matched source conditionals and returns

$$\mu_{\mathrm{TR}}(y \mid \mathbf{x}) = \sum_{v^\star_{M+1:T^\star-1}} \prod_{i=M+1}^{T^\star} \mu^i_{\mathrm{TR}}(v^\star_i \mid \mathbf{pa}^\star_i), \qquad v^\star_{1:M} = \mathbf{x}, \quad v^\star_{T^\star} = y. \tag{3}$$

Here μ^i_{TR} estimates the conditional at target position i from its aligned source matches. Empirical conditional frequencies with a probability floor or neural predictors can supply these gates. The intermediate variables are summed out, so target traces are unnecessary.

Theorem 3.4 (Circuit-TR, informal)**.** *Suppose every target position after the prefix has a certified source match. If each transported gate has conditional KL error at most a uniformly over its arguments, then*

$$\mathcal{E}(\mu_{\mathrm{TR}}) \leqslant (T^\star - M)a. \tag{4}$$

For empirical source estimates, $a = \widetilde{\mathcal{O}}(1/N)$ with high probability when the required source parent configurations have probabilities bounded below. With the composition and matches supplied, this gives zero-shot target prediction.

The conditional KL chain rule adds local errors along the computation, and marginalization cannot increase KL divergence (Dasgupta, 1997). Source coverage matters because the gates must be accurate on the arguments reached by the target, which can differ in frequency from source arguments. Theorem B.3 gives the estimation and coverage dependence.

Circuit-TR is the oracle comparator for adaptation. When the supplied knowledge justifies several representations of a concept, each supplies a comparator; fitted gates can give them different risks. We now remove the supplied composition and matches and ask how many target samples suffice to compete with this predictor.

4 Symbolic Circuit Adaptation

Without the supplied graphs and mechanism matches, the source gates define a family of possible target computations. Circuit-AD uses target input/output pairs to select among them, aiming for risk comparable to Circuit-TR.

4.1 Selecting a composition from input/output pairs

Let $P_1, \ldots, P_d$ be source conditional distributions with at most c ordered arguments, and let $\hat{P}_1, \ldots, \hat{P}_d$ be gates fitted to them using only source data. A gate may be an empirical or neural estimate for a source position and proposed parent list.

For every size $\ell \geqslant M + 1$, let $\mathcal{H}_\ell$ contain the circuits formed by choosing a gate and a compatible ordered predecessor list at each position after the M inputs. Assume this family is nonempty. The predictor for a circuit $C \in \mathcal{H}_\ell$ is

$$\hat{\mu}_C(y \mid \mathbf{x}) = \sum_{v_{M+1:\ell-1}} \prod_{i=M+1}^{\ell} \hat{P}_{a_i}(v_i \mid v_{\pi_i}), \qquad v_{1:M} = \mathbf{x}, \quad v_\ell = y, \tag{5}$$

where $a_i \in [d]$ is the gate choice and π_i its ordered list of parent positions. The gates remain fixed. When source parents are unknown, the library can include estimates for every source position and predecessor list of arity at most c; Section 5 gives a neural construction.

For the n target input/output pairs, let $\widehat{\mathcal{R}}_{\mathrm{te}}(\mu) = -n^{-1} \sum_{(\mathbf{x},y)\in D^\star} \log \mu(y \mid \mathbf{x})$ be empirical risk. Circuit-AD selects

$$(\hat{\ell}, \hat{C}) \in \operatorname*{arg\,min}_{\ell \geqslant M+1,\ C\in\mathcal{H}_\ell} \left\{ \widehat{\mathcal{R}}_{\mathrm{te}}(\hat{\mu}_C) + \gamma_\ell \right\}, \qquad \hat{\mu} = \hat{\mu}_{\hat{C}}. \tag{6}$$

The penalty controls selection over all finite sizes by accounting for the number of compositions at each size, as in complexity-penalized estimation (Barron & Cover, 1991). For fixed maximum arity, its order is

$$\gamma_\ell = \mathcal{O}\left(\sqrt{\frac{(\ell - M)\log(2d\ell)}{n}} \right). \tag{7}$$

The exact coefficient is given in Equation (35). All n target samples are available for selection because the library was fitted using source data alone.

Theorem 4.1 (Circuit-AD oracle comparison, informal)**.** *Conditional on the source-fitted gates, with high probability over the n target pairs, the selected predictor satisfies, simultaneously for every $C \in \mathcal{H}_\ell$,*

$$\mathcal{E}(\hat{\mu}) \leqslant \mathcal{E}(\hat{\mu}_C) + 2\gamma_\ell. \tag{8}$$

In particular, if the supplied Circuit-TR composition has size L and uses this gate library, the theorem gives

$$\mathcal{R}(\hat{\mu}) \leqslant \mathcal{R}(\mu_{\mathrm{TR}}) + 2\gamma_L. \tag{9}$$

This bounds the additional population risk of learning the composition from outcomes. The proof controls empirical-to-population deviations simultaneously across candidates. A selected composition may also improve on Circuit-TR. Equation (37) and Appendix E.3 give the full result and its description-length interpretation.

Combining this inequality with the Circuit-TR estimation bound yields the dependence on source knowledge and circuit size.

Corollary 4.2 (Source estimation and adaptation, informal)**.** *Suppose the target query has a Circuit-TR representation of size L, including the inputs, using the source conditional distributions. If the gates used in this representation have conditional KL error at most a uniformly over their arguments, then, with high probability, Circuit-AD satisfies*

$$\mathcal{E}(\hat{\mu}) \leqslant (L - M)a + \mathcal{O}\left(\sqrt{\frac{(L - M)\log(2dL)}{n}} \right). \tag{10}$$

The first term bounds source estimation error; the second is the cost of selecting a composition and its size. A short representation limits both by using fewer gate applications and fewer structural choices. The learner need not know its size or recover the comparator circuit. These are sufficient target-sample guarantees; Theorem B.7 gives the full statement and proof. Approximate representations enter the more general comparison through their excess risk in Equation (8).

Why circuit size matters. A source supplying maximum, minimum, and subtraction permits a GCD computation with $\mathcal{O}(|\mathcal{V}|)$ positions. Remainder permits a construction with $\mathcal{O}(\log |\mathcal{V}|)$ positions, and the recurrence in Equation (20) uses only remainder gates followed by an addition. Repeated modular addition can implement subtraction, at greater length. Thus restricting the source library can increase the size of a representation while preserving its noiseless input/output function.

With comparable noise per gate, additional operations also offer more opportunities for errors. Circuits that compute the same noiseless function can therefore induce different concepts. Against one fixed target concept, length affects both the approximation risk of a composition and the statistical cost of selecting it. Appendices E.4 and F illustrate this distinction with GCD constructions and an exact noisy-chain calculation. The oracle inequality compares these risks and sizes together.

The candidate family grows rapidly with circuit size. Exact enumeration gives a statistical benchmark for selecting a composition; the next section introduces Neural-AD for learning a composition of the source gates.

5 NEURAL CIRCUIT ADAPTATION

Neural-AD separates a trainable generator of circuit structure from a shared gate predictor (Figure 2). Source next-token training learns both. Target adaptation freezes the gates and changes their composition. The available target observations determine how these structural choices can be evaluated.

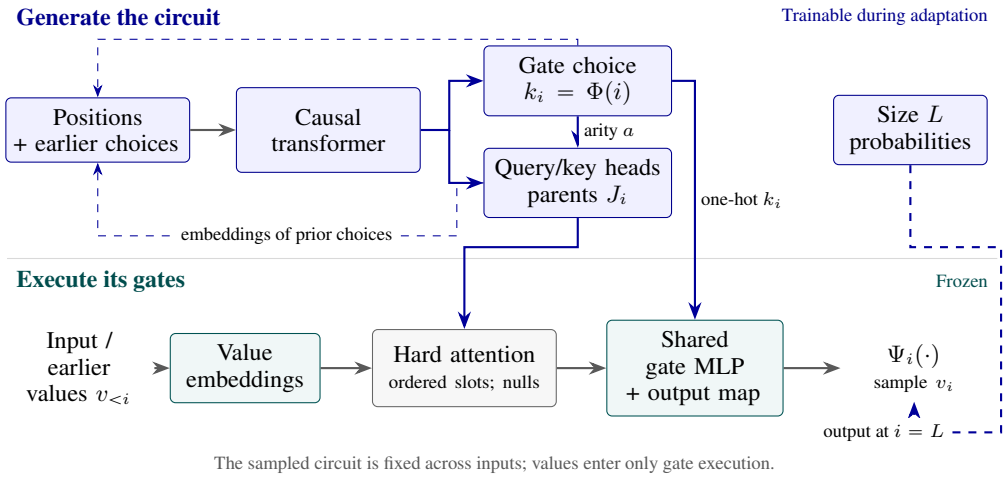


Figure 2: Neural-AD architecture at a generated position i. The upper path samples the gate, its ordered predecessors, and the circuit size. Hard attention routes the selected value embeddings into the frozen gate predictor below. Here $\Psi_i(\cdot) = \Psi(\cdot \mid v_{J_i}; k_i)$. Source training fits the generator and gate predictor without the earlier-choice inputs shown for target adaptation. Target adaptation changes the blue structural parameters. With process supervision, the observed trace also fixes the size.

5.1 REUSABLE GATES THROUGH HARD ATTENTION

The generator is a causal transformer over positions and earlier structural choices. It emits a gate indicator and query/key scores for c hard attention heads, which select ordered predecessor positions. A gate of arity $a \leqslant c$ uses the first a heads with distinct predecessors; unused heads receive a fixed null input. These choices are independent of execution values, so each sampled circuit has gate choices and ordered parent lists that remain fixed across inputs.

Let A collect the parents and Φ the gate indicators. Hard attention routes the chosen value embeddings into ordered slots. Their concatenation and a one-hot gate indicator enter a shared MLP and output map, giving

$$\Psi_\theta(v_i \mid v_{\mathbf{Pa}_i(A)}; \Phi(i)). \tag{11}$$

No position information enters the gate predictor, so a gate represents the same conditional table wherever it is applied. Its value embeddings, MLP, and output map are frozen during adaptation. Appendix H specifies the architecture.

Source fitting minimizes expected next-token log loss, with a penalty encouraging shared gates and small parent sets. After selecting a source representation, we refit its gates and freeze them. Their conditional errors on source-covered arguments control the risk of a transported composition (Lemma C.2); Appendix C.1 gives the objective and estimation guarantees.

5.2 ADAPTATION WITH PROCESS SUPERVISION

With complete target traces, each gate and parent choice can be evaluated directly. For a candidate C, let

$$\mu^\star_{C,i}(v_i \mid v_{1:i-1}) = \Psi(v_i \mid v_{\mathbf{Pa}_i(C)}; \Phi_C(i)) \tag{12}$$

use the frozen source gate predictor Ψ. The circuit's empirical trace loss is

$$\widehat{\mathcal{L}}^{\star}(C) = -\frac{1}{n(T^{\star} - M)} \sum_{v \in D^{\star}} \sum_{i=M+1}^{T^{\star}} \log \mu^{\star}_{C,i}(v_i \mid v_{1:i-1}). \tag{13}$$

The observed arguments make this objective separable: minimize over gates and ordered parents at each position, then fit the generator by maximum likelihood to those choices. The trace fixes the circuit size. The conditional KL chain rule bounds final-answer excess risk by the sum of the positions' population excess log losses (Proposition C.4). When target traces must also supply new gates, the extension in Appendix A permits their estimation.

5.3 Adaptation with outcome supervision

When only input/output pairs are observed, intermediate values must be marginalized. Let $\pi_{\theta^{\star}}(C)$ be the generator's probability of a circuit, including its size and all gate and parent choices. Each circuit defines the fitted predictor $\hat{\mu}_C$ of Equation (5). Prediction averages their conditional distributions:

$$\mu_{\theta^{\star}}(y \mid \mathbf{x}) = \sum_{C} \pi_{\theta^{\star}}(C) \hat{\mu}_C(y \mid \mathbf{x}). \tag{14}$$

We evaluate the generator using the penalized empirical risk

$$\widehat{F}(\theta^{\star}) = \widehat{\mathcal{R}}_{\mathrm{te}}(\mu_{\theta^{\star}}) + \mathbb{E}_{C \sim \pi_{\theta^{\star}}} \gamma_{|C|}, \tag{15}$$

with the same log loss and size penalty as Circuit-AD (Equations (6) and (35)). Concentrating on a single circuit recovers that circuit's selection criterion.

Rejection-sampling fine-tuning. Starting from the pretrained generator, we propose circuits, rank them by penalized empirical risk on the observed target pairs, and update the generator by likelihood fitting to retained compositions. New proposals sample the generator or modify earlier candidates. This rejection-sampling fine-tuning (RFT) follows expert iteration and self-training (Anthony et al., 2017; Singh et al., 2024). Checkpoints use Equation (15), estimated by sampling circuits and executions; Appendix C.4 gives the procedure.

Comparison with Circuit-AD. Finite search need not minimize this objective exactly. Its guarantee also accounts for the empirical Jensen gap $\widehat{D}(\pi) = \mathbb{E}_{C \sim \pi} \widehat{\mathcal{R}}_{\mathrm{te}}(\hat{\mu}_C) - \widehat{\mathcal{R}}_{\mathrm{te}}(\mu_{\pi}) \geqslant 0$. It measures the reduction in empirical log loss from averaging predictions, and is zero for a single circuit.

Theorem 5.1 (Neural-AD; informal)**.** *Suppose the fitted generator's penalized empirical risk is at most the minimum penalized empirical risk over the candidate circuits plus $\xi \geqslant 0$. Conditional on the fitted source gates, with high probability over the n target pairs, every candidate C of size L satisfies*

$$\mathcal{E}(\mu_{\hat{\theta}^{\star}}) \leqslant \mathcal{E}(\hat{\mu}_C) + 2\gamma_L + \xi + \widehat{D}(\pi_{\hat{\theta}^{\star}}). \tag{16}$$

Thus Neural-AD retains the Circuit-TR comparison while recording the effects of finite optimization and averaging over circuits. For a point mass, $\widehat{D} = 0$ and only the empirical optimization residual remains. Theorem C.5 gives the full statement; the experiments compare this search with exact Circuit-AD where enumeration is feasible.

6 Experiments

We test whether learning a composition can approach the risk of Circuit-TR, and how process supervision changes the target sample requirements. Circuit-AD and Neural-AD reuse gates fitted on source traces. ERM-target fits a plain transformer on target samples; ERM-src+tgt first pretrains on the same source traces. We report excess risk in nats, evaluating Neural-AD through Equation (14). All target-sample axes are logarithmic; detailed protocols appear in Appendix G.

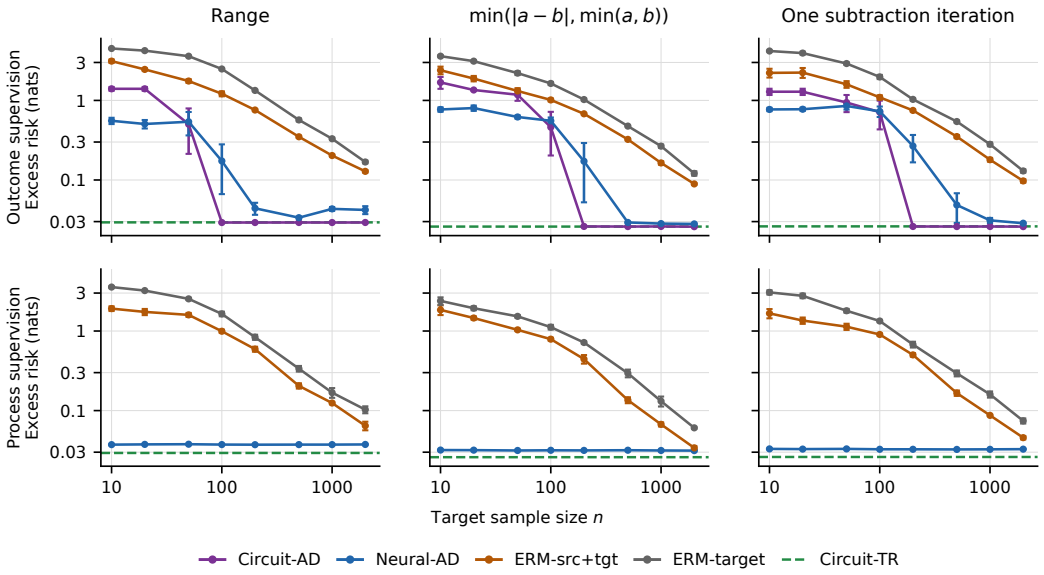


Figure 3: Learning three noisy computations under outcome supervision (top) and process supervision (bottom). Circuit-TR is supplied the target composition; Circuit-AD selects by exact enumeration. Points average five target datasets; bars show one standard error, conditional on one source fit. Both axes are logarithmic. Complete settings and results appear in Appendix G.4.1.

Learning to compete with Circuit-TR. We first use three- and four-gate computations over ten values, composed from maximum, minimum, and subtraction. Source and target operations independently return a uniform value with probability 0.1. The source supplies $20{,}000$ complete traces. This setting permits exact Circuit-AD enumeration and a direct comparison with Neural-AD using the same gates, candidate family, and penalized log loss (Appendix G.4.1).

In Figure 3, outcome-supervised Neural-AD at $n = 500$ attains excess risks 0.034, 0.029, and 0.049, compared with 0.029, 0.026, and 0.026 for Circuit-TR. Pretrained ERM remains at 0.324–0.351. Neural-AD's decoded circuits are within 0.01 nats of Circuit-AD's minimum penalized empirical risk in every seed at $n \geqslant 500$. Process supervision reaches excess risk 0.032–0.037 with only ten traces, since the observed intermediate values permit local gate and parent selection. The longer-circuit comparison in Appendix G.4.1 also shows where finite outcome search fails to find an available low-risk composition.

With 30% of input pairs withheld from target fitting, Neural-AD has mean excess risk 0.028–0.061 at $n = 500$, compared with 0.589–1.137 for pretrained ERM (Appendix G.6.1). This tests reuse on new full input combinations whose required local arguments are covered by the source.

Composing gates on real measurement windows. We reuse the seven-operation synthetic source library of Figure 1 to compute aggregates of electricity measurements (Trindade, 2015). Each input contains four hourly readings quantized to 100 levels; the target concepts are the noiseless range and largest adjacent absolute change. Adaptation and evaluation use different clients and years, with repeated input tuples excluded across splits (Appendix G.6.2).

With 100 target samples, Neural-AD attains excess risk 0.067 on adjacent change, close to Circuit-TR's 0.057, while pretrained ERM remains at 6.633 (Figure 4). Range is harder to learn from final answers: Neural-AD reaches 0.741, compared with 0.072 under process supervision and 0.071 for Circuit-TR. Thus reusable gates can yield accurate computations on new input distributions, while the difficulty of selecting their composition depends on the target observations.

GCD library interventions show how alternative compositions compensate for a missing remainder operation (Appendix G.2). Source-size and fixed-target noise studies separately examine gate-estimation error and accumulated noise (Appendices G.5.1 and G.5.2).

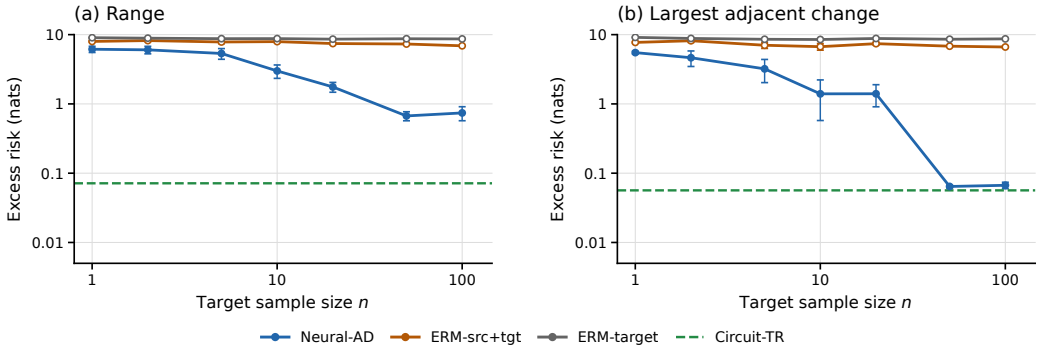


Figure 4: Outcome-supervised adaptation on electricity windows. Neural-AD averages five target datasets; each ERM baseline averages three for range and two for adjacent change. Bars show one standard error. Source gates and evaluation windows are shared across fits; Circuit-TR uses no target samples. Both axes are logarithmic. Process comparisons appear in Appendix G.6.2.

7 Conclusions

Compositional generalization depends on both what can be learned from the source and what must be learned about the target computation. Circuit transportability makes the first question precise: causal structure and shared mechanisms determine a composition of conditionals estimable from source traces. Our oracle comparison answers the second. Target input/output samples suffice for Circuit-AD to compete with that supplied composition, with an additional risk bound determined by the gate library, circuit size, and target sample size. The learner need not recover the supplied circuit or know its size.

The experiments with Neural-AD illustrate how this distinction shapes adaptation. Intermediate observations make local composition choices accessible with fewer target samples; final answers require searching over complete compositions. Restricting the source library can require different, longer computations, whose accumulated noise changes their risk even when they compute the same noiseless function. Thus the statistical goal is a low-risk composition of the learned gates. Efficiently finding such compositions in larger families remains an important computational question beyond the oracle guarantee.

AI use statement

In this work, we used ChatGPT, Codex, Claude, and Cursor for the following purposes:

- Ideation, writing, choice of notation, and structuring the paper.
- Literature overview and references.
- Drafting and checking mathematical arguments, including the conditional-estimation and selection proofs.
- Developing the codebase and preparing figures.
- Interpreting experimental results and preparing the reproducibility and implementation descriptions in Appendix G.1.

The authors are responsible for the final text, claims, proofs, code, and results.

Reproducibility statement

The assumptions and proofs for the transportability and adaptation results are given in Appendices B and C. Appendix H specifies the neural architecture. Appendix G describes the source and target computations, fitting and evaluation procedures, sample sizes, and replication; Appendix G.6.2 details preprocessing and data splits for the electricity study.

REFERENCES

Emmanuel Abbe, Samy Bengio, Aryo Lotfi, and Kevin Rizk. Generalization on the unseen, logic reasoning and degree curriculum. *Journal of Machine Learning Research*, 25(331):1–58, 2024a. URL https://www.jmlr.org/papers/v25/24-0220.html.

Emmanuel Abbe, Samy Bengio, Aryo Lotfi, Colin Sandon, and Omid Saremi. How far can transformers reason? the globality barrier and inductive scratchpad. In *Advances in Neural Information Processing Systems*, volume 37, pp. 27850–27895, 2024b. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/3107e4bdb658c79053d7ef59cbc804dd-Abstract-Conference.html.

Pieter Abbeel, Daphne Koller, and Andrew Y. Ng. Learning factor graphs in polynomial time and sample complexity. *Journal of Machine Learning Research*, 7(64):1743–1788, 2006. URL https://www.jmlr.org/papers/v7/abbeel06a.html.

Ferran Alet, Tomas Lozano-Perez, and Leslie P. Kaelbling. Modular meta-learning. In *Proceedings of the 2nd Conference on Robot Learning*, volume 87 of *Proceedings of Machine Learning Research*, pp. 856–868, 2018. URL https://proceedings.mlr.press/v87/alet18a.html.

Eric Allender, Joshua A. Grochow, Dieter van Melkebeek, Cristopher Moore, and Andrew Morgan. Minimum circuit size, graph isomorphism, and related problems. *SIAM Journal on Computing*, 47 (4):1339–1372, 2018. doi: 10.1137/17M1157970.

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Learning to compose neural networks for question answering. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1545–1554, 2016a. doi: 10.18653/v1/N16-1181. URL https://aclanthology.org/N16-1181/.

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Neural module networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016b.

Jacob Andreas, Dan Klein, and Sergey Levine. Modular multitask reinforcement learning with policy sketches. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 166–175, 2017. URL https://proceedings.mlr.press/v70/andreas17a.html.

Cem Anil, Yuhuai Wu, Anders Andreassen, Aitor Lewkowycz, Vedant Misra, Vinay Ramasesh, Ambrose Slone, Guy Gur-Ari, Ethan Dyer, and Behnam Neyshabur. Exploring length generalization in large language models. *arXiv preprint arXiv:2207.04901*, 2022.

Thomas Anthony, Zheng Tian, and David Barber. Thinking fast and slow with deep learning and tree search. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/d8e1344e27a5b08cdfd5d027d9b8d6de-Paper.pdf.

Elias Bareinboim and Judea Pearl. Transportability from multiple environments with limited experiments: Completeness results. *Advances in neural information processing systems*, 27, 2014.

Elias Bareinboim, Sanghack Lee, Vasant Honavar, and Judea Pearl. Transportability from multiple environments with limited experiments. *Advances in Neural Information Processing Systems*, 26, 2013.

Andrew R. Barron and Thomas M. Cover. Minimum complexity density estimation. *IEEE Transactions on Information Theory*, 37(4):1034–1054, 1991. doi: 10.1109/18.86996. URL https://isl.stanford.edu/~cover/papers/barron_cover_91.pdf.

S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan. A theory of learning from different domains. *Machine Learning*, 79(1-2):151–175, 2010. doi: 10.1007/s10994-009-5187-8.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155, 2003. URL https://www.jmlr.org/papers/v3/bengio03a.html.

Yoshua Bengio, Tristan Deleu, Nasim Rahaman, Nan Rosemary Ke, Sébastien Lachapelle, Olexa Bilaniuk, Anirudh Goyal, and Christopher Pal. A meta-transfer objective for learning to disentangle causal mechanisms. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=ryxWIgBFPS.

John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman. Learning bounds for domain adaptation. In J. Platt, D. Koller, Y. Singer, and S. Roweis (eds.), *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2007. URL https://proceedings.neurips.cc/paper_files/paper/2007/file/42e77b63637ab381e8be5f8318cc28a2-Paper.pdf.

Matthew Bowers, Theo X. Olausson, Lionel Wong, Gabriel Grand, Joshua B. Tenenbaum, Kevin Ellis, and Armando Solar-Lezama. Top-down synthesis for library learning. *Proceedings of the ACM on Programming Languages*, 7(POPL):1182–1213, 2023. doi: 10.1145/3571234. URL https://arxiv.org/abs/2211.16605.

Jack Brady, Julius von Kügelgen, Sebastien Lachapelle, Simon Buchholz, Thomas Kipf, and Wieland Brendel. Interaction asymmetry: A general principle for learning composable abstractions. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/735c847a07bf6dd4486ca1ace242a88c-Abstract-Conference.html.

Michael B. Chang, Abhishek Gupta, Sergey Levine, and Thomas L. Griffiths. Automatically composing representation transformations as a means for generalization. In *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1807.04640.

Henry Conklin, Bailin Wang, Kenny Smith, and Ivan Titov. Meta-learning to compositionally generalize. In *Proceedings of ACL-IJCNLP*, 2021.

J. Correa and E. Bareinboim. General transportability of soft interventions: Completeness results. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 10902–10912, Vancouver, Canada, Jun 2020. Curran Associates, Inc.

Juan D Correa and Elias Bareinboim. From statistical transportability to estimating the effect of stochastic interventions. In *IJCAI*, pp. 1661–1667, 2019.

Andrew Cropper. Playgol: Learning programs through play. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pp. 6074–6080, 2019. doi: 10.24963/ijcai.2019/841. URL https://www.ijcai.org/proceedings/2019/841.

Sanjoy Dasgupta. The sample complexity of learning fixed-structure Bayesian networks. *Machine Learning*, 29(2–3):165–180, 1997. URL https://cseweb.ucsd.edu/~dasgupta/papers/bayes-sample.pdf.

Eyal Dechter, Jon Malmaud, Ryan P. Adams, and Joshua B. Tenenbaum. Bootstrap learning via modular concept discovery. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*, pp. 1302–1309, 2013. URL https://www.ijcai.org/Abstract/13/196.

Kefan Dong and Tengyu Ma. First steps toward understanding the extrapolation of nonlinear models to unseen domains. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=7wrq3vHcMM.

Kevin Ellis, Lucas Morales, Mathias Sablé-Meyer, Armando Solar-Lezama, and Josh Tenenbaum. Learning libraries of subroutines for neurally–guided bayesian program induction. In *Advances in Neural Information Processing Systems*, volume 31, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/7aa685b3b1dc1d6780bf36f7340078c9-Abstract.html.

Kevin Ellis, Catherine Wong, Maxwell Nye, Mathias Sablé-Meyer, Luc Cary, Lucas Morales, Luke Hewitt, Armando Solar-Lezama, and Joshua B. Tenenbaum. DreamCoder: bootstrapping inductive program synthesis with wake-sleep library learning. *Proceedings of the 42nd ACM SIGPLAN International Conference on Programming Language Design and Implementation (PLDI)*, 2021.

William S. Evans and Leonard J. Schulman. Signal propagation and noisy circuits. *IEEE Transactions on Information Theory*, 45(7):2367–2373, 1999. doi: 10.1109/18.796377. URL https://users.cms.caltech.edu/~schulman/Papers/evansS99.pdf.

Gabriel Grand, Lionel Wong, Maddy Bowers, Theo X. Olausson, Muxin Liu, Joshua B. Tenenbaum, and Jacob Andreas. LILO: Learning interpretable libraries by compressing and documenting code. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2310.19791.

Steve Hanneke and Samory Kpotufe. On the value of target data in transfer learning. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://papers.nips.cc/paper_files/paper/2019/file/b91f4f4d36fa98a94ac5584af95594a0-Paper.pdf.

Steve Hanneke, Samory Kpotufe, and Yasaman Mahdaviyeh. Limits of model selection under transfer learning. In Gergely Neu and Lorenzo Rosasco (eds.), *Proceedings of Thirty Sixth Conference on Learning Theory*, volume 195 of *Proceedings of Machine Learning Research*, pp. 5781–5812. PMLR, 12–15 Jul 2023. URL https://proceedings.mlr.press/v195/hanneke23c.html.

Dieuwke Hupkes, Verna Dankers, Mathijs Mul, and Elia Bruni. Compositionality decomposed: how do neural networks generalise? *Journal of Artificial Intelligence Research*, 67:757–795, 2020.

Kasra Jalaldoust and Elias Bareinboim. Transportable representations for domain generalization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(11):12790–12800, Mar. 2024. doi: 10.1609/aaai.v38i11.29175. URL https://ojs.aaai.org/index.php/AAAI/article/view/29175.

Kasra Jalaldoust, Alexis Bellot, and Elias Bareinboim. Partial transportability for domain generalization. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=2V5LTfhcfd.

Daniel Keysers, Nathanael Schärli, Nathan Scales, Hylke Buisman, Daniel Furrer, Sergii Kashubin, Nikola Momchev, Danila Sinopalnikov, Lukasz Stafiniak, Tibor Tihon, Dmitry Tsarkov, Xiao Wang, Marc van Zee, and Olivier Bousquet. Measuring compositional generalization: A comprehensive method on realistic data. In *International Conference on Learning Representations*, 2020. URL https://arxiv.org/abs/1912.09713.

Najoung Kim and Tal Linzen. COGS: A compositional generalization challenge based on semantic interpretation. In *Proceedings of EMNLP*, 2020.

Lingjing Kong, Xin Liu, Guangyi Chen, Martin Q. Ma, Xiangchen Song, Yuekai Sun, Mikhail Yurochkin, Taylor W. Killian, Ruslan Salakhutdinov, Kun Zhang, Eric P. Xing, and Zhengzhong Liu. From reasoning traces to reusable modules: Understanding compositional generalization in language model reasoning. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026a. URL https://arxiv.org/abs/2606.18089.

Lingjing Kong, Shaoan Xie, Yang Jiao, Yetian Chen, Yanhui Guo, Simone Shao, Yan Gao, Guangyi Chen, and Kun Zhang. Learning by analogy: A causal framework for compositional generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 36615–36626, 2026b. URL https://openaccess.thecvf.com/content/CVPR2026/html/Kong_Learning_by_Analogy_A_Causal_Framework_for_Compositional_Generalization_CVPR_2026_paper.html.

Sébastien Lachapelle, Divyat Mahajan, Ioannis Mitliagkas, and Simon Lacoste-Julien. Additive decoders for latent variables identification and cartesian-product extrapolation. In *Advances in Neural Information Processing Systems*, volume 36, pp. 25112–25150, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/4ef594af0d9a519db8fb292452c461fa-Abstract-Conference.html.

Brenden M. Lake. Compositional generalization through meta sequence-to-sequence learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Brenden M. Lake and Marco Baroni. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2018.

Brenden M. Lake and Marco Baroni. Human-like systematic generalization through a meta-learning neural network. *Nature*, 623:115–121, 2023. doi: 10.1038/s41586-023-06668-3. URL https://www.nature.com/articles/s41586-023-06668-3.

Sanghack Lee, Juan Correa, and Elias Bareinboim. General transportability—synthesizing observations and experiments from heterogeneous domains. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(06):10210–10217, 2020. doi: 10.1609/aaai.v34i06.6582. URL https://ojs.aaai.org/index.php/AAAI/article/view/6582.

Leonid A. Levin. Randomness conservation inequalities; information and independence in mathematical theories. *Information and Control*, 61(1):15–37, 1984. URL https://www.cs.bu.edu/fac/lnd/pdf/inf.pdf.

Samuel Lippl and Kimberly Stachenfeld. When does compositional structure yield compositional generalization? a kernel theory. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/572a6f16ec44f794fb3e0f8a310acbc6-Abstract-Conference.html.

Sara Magliacane, Thijs Van Ommen, Tom Claassen, Stephan Bongers, Philip Versteeg, and Joris M Mooij. Domain adaptation by using causal inference to predict invariant conditional distributions. *Advances in neural information processing systems*, 31, 2018.

Divyat Mahajan, Mohammad Pezeshki, Charles Arnal, Ioannis Mitliagkas, Kartik Ahuja, and Pascal Vincent. Compositional risk minimization. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 42503–42553, 2025. URL https://proceedings.mlr.press/v267/mahajan25a.html.

Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation: Learning bounds and algorithms. In *22nd Conference on Learning Theory (COLT)*, 2009. URL https://www.cs.mcgill.ca/~colt2009/papers/003.pdf.

Jorge A. Mendez and Eric Eaton. Lifelong learning of compositional structures. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2007.07732.

Giambattista Parascandolo, Niki Kilbertus, Mateo Rojas-Carulla, and Bernhard Schölkopf. Learning independent causal mechanisms. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2018.

Judea Pearl. *Causality*. Cambridge university press, 2009.

Judea Pearl and Elias Bareinboim. Transportability of causal and statistical relations: A formal approach. In *Twenty-fifth AAAI conference on artificial intelligence*, 2011.

Jonas Peters, Peter Bühlmann, and Nicolai Meinshausen. Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(5):947–1012, 2016.

Qwen Team. Qwen2.5: A party of foundation models, September 2024. URL https://qwenlm.github.io/blog/qwen2.5/.

Scott Reed and Nando de Freitas. Neural programmer-interpreters. In *International Conference on Learning Representations*, 2016. URL https://arxiv.org/abs/1511.06279.

Mateo Rojas-Carulla, Bernhard Schölkopf, Richard Turner, and Jonas Peters. Invariant models for causal transfer learning. *The Journal of Machine Learning Research*, 19(1):1309–1342, 2018.

Dominik Rothenhäusler, Nicolai Meinshausen, Peter Bühlmann, and Jonas Peters. Anchor regression: Heterogeneous data meet causality. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 83(2):215–246, 2021.

Laura Ruis, Jacob Andreas, Marco Baroni, Diane Bouchacourt, and Brenden M. Lake. A benchmark for systematic generalization in grounded language understanding. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. Toward causal representation learning. *Proceedings of the IEEE*, 109(5):612–634, 2021.

Simon Schug, Seijin Kobayashi, Yassir Akram, Maciej Wołczyk, Alexandra Proca, Johannes von Oswald, Razvan Pascanu, João Sacramento, and Angelika Steger. Discovering modular solutions that generalize compositionally. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2312.15001.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, Cambridge, 2014.

Avi Singh, John D. Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Xavier Garcia, Peter J. Liu, James Harrison, Jaehoon Lee, Kelvin Xu, Aaron Parisi, Abhishek Kumar, Alex Alemi, Alex Rizkowsky, Azade Nova, Ben Adlam, Bernd Bohnet, Gamaleldin Elsayed, Hanie Sedghi, Igor Mordatch, Isabelle Simpson, Izzeddin Gur, Jasper Snoek, Jeffrey Pennington, Jiri Hron, Kathleen Kenealy, Kevin Swersky, Kshiteej Mahajan, Laura Culp, Lechao Xiao, Maxwell L. Bileschi, Noah Constant, Roman Novak, Rosanne Liu, Tris Warkentin, Yundi Qian, Yamini Bansal, Ethan Dyer, Behnam Neyshabur, Jascha Sohl-Dickstein, and Noah Fiedel. Beyond human data: Scaling self-training for problem-solving with language models. *Transactions on Machine Learning Research*, 2024. URL https://arxiv.org/abs/2312.06585.

Adarsh Subbaswamy, Peter Schulam, and Suchi Saria. Preventing failures due to dataset shift: Learning predictive models that transport. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 3118–3127. PMLR, 2019.

Artur Trindade. ElectricityLoadDiagrams20112014. UCI Machine Learning Repository, 2015. URL https://archive.ics.uci.edu/dataset/321/electricityloaddiagrams20112014.

Lazar Valkov, Dipak Chaudhari, Akash Srivastava, Charles Sutton, and Swarat Chaudhuri. HOUDINI: Lifelong learning as program synthesis. In *Advances in Neural Information Processing Systems*, volume 31, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/edc27f139c3b4e4bb29d1cdbc45663f9-Abstract.html.

Petar Veličković, Adrià Puigdomènech Badia, David Budden, Razvan Pascanu, Andrea Banino, Misha Dashevskiy, Raia Hadsell, and Charles Blundell. The CLRS algorithmic reasoning benchmark. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, 2022.

Heribert Vollmer. The complexity of deciding if a Boolean function can be computed by circuits over a restricted basis. *Theory Comput. Syst.*, 44(1):82–90, 2009. URL https://doi.org/10.1007/s00224-007-9030-9.

Thaddäus Wiedemer, Prasanna Mayilvahanan, Matthias Bethge, and Wieland Brendel. Compositional generalization from first principles. In *Advances in Neural Information Processing Systems*, volume 36, pp. 6941–6960, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/15f6a10899f557ce53fe39939af6f930-Abstract-Conference.html.

Thaddäus Wiedemer, Jack Brady, Alexander Panfilov, Attila Juhos, Matthias Bethge, and Wieland Brendel. Provable compositional generalization for object-centric learning. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2310.05327.

Louis-Pascal Xhonneux, Andreea-Ioana Deac, Petar Veličković, and Jian Tang. How to transfer algorithmic reasoning knowledge to learn new algorithms? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Keyulu Xu, Jingling Li, Mozhi Zhang, Simon S. Du, Ken-ichi Kawarabayashi, and Stefanie Jegelka. What can neural networks reason about? In *International Conference on Learning Representations*, 2020. URL https://arxiv.org/abs/1905.13211.

Yujun Yan, Kevin Swersky, Danai Koutra, Parthasarathy Ranganathan, and Milad Hashemi. Neural execution engines: Learning to execute subroutines. In *Advances in Neural Information Processing Systems*, volume 33, pp. 17298–17308, 2020. URL https://papers.nips.cc/paper/2020/hash/c8b9abffb45bf79a630fb613dcd23449-Abstract.html.

Hattie Zhou, Arwen Bradley, Etai Littwin, Noam Razin, Omid Saremi, Josh Susskind, Samy Bengio, and Preetum Nakkiran. What algorithms can transformers learn? a study in length generalization. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/45ed1a72597594c097152ef9cc187762-Paper-Conference.pdf.

Supplementary Material

A Circuit-TR with process supervision **18**
A.1 Circuit-TR with target observations . . . 18
A.2 Selecting a circuit for observed traces . . . 19

B Full statements and proofs **19**
B.1 The GCD construction . . . 20
B.2 Positivity and conditional estimation . . . 21
B.3 Composition and proof of Theorem 3.4 . . . 22
B.4 Pooling matched target positions . . . 23
B.5 Finite selection and proof of Theorem A.1 . . . 24
B.6 Proof of Corollary B.6 . . . 25
B.7 Composition selection and proof of Corollary 4.2 . . . 25

C Neural-AD: statements and proofs **27**
C.1 Source training and gate-estimation error . . . 27
C.2 Process supervision . . . 30
C.3 Outcome supervision and the Circuit-AD comparison . . . 31
C.4 Outcome-supervised circuit search . . . 32
C.5 Circuit-AD with a bounded verifier loss . . . 33

D Challenges due to unobserved confounders **34**

E Estimation, circuit size, and adaptation rates **35**
E.1 Matched and unmatched positions . . . 35
E.2 Multiple source domains . . . 36
E.3 Selection without a known circuit size . . . 36
E.4 Circuit size and the arithmetic examples . . . 36

F Concepts, composition length, and noise **38**
F.1 Restricting gates can change the required computation . . . 38
F.2 An exact example of length-dependent signal loss . . . 38
F.3 Gate estimation and the preference for shorter compositions . . . 39
F.4 Identification and computation length . . . 40

G Experimental settings and additional results **41**
G.1 Observation and evaluation protocol . . . 41
G.2 Learning the GCD concept from different source libraries . . . 42
G.3 What process supervision reveals . . . 45
G.4 Exact circuit selection . . . 46

G.5 Source estimation and composition noise 53
G.6 Reuse on new input combinations 54
G.7 Gates from pretrained language models 56
G.8 Numerical records and replication 58

H Detailed model architecture **59**

I Related work **61**
I.1 Causal and statistical transportability 61
I.2 Causal invariance and robust transfer 62
I.3 Independent mechanisms and causal representations 62
I.4 Learned libraries and modular adaptation 63
I.5 Compositional generalization, identification, and support 63
I.6 Execution traces and algorithmic learning 64
I.7 Statistical complexity and noisy computation 65
I.8 Learning to search for compositions 65

J Limitations **65**



(a) Source sequence (b) Target circuit

Figure 5: Argument alignment in Circuit-TR. The highlighted source and target positions share the subtraction mechanism and its noise distribution. The source position V_4 has ordered parents $\langle V_1, V_2 \rangle$ (blue), while the target position Y has ordered parents $\langle X_3, X_2 \rangle$ (orange). Single edges are the first parent and double edges are the second parent.

A CIRCUIT-TR WITH PROCESS SUPERVISION

The main analysis compares Circuit-AD with Circuit-TR, which is supplied with the target composition and source matches. Complete target traces allow Circuit-TR to fit gates at unmatched positions and to pool target observations with matched source observations. This appendix develops that extension and a Circuit-AD construction that competes with its oracle comparator when the supplied matches are represented by a partition. Circuit-AD constructs candidate gates from source data and a target fitting split, then selects a composition on an independent selection split, separating estimation and selection costs.

A.1 CIRCUIT-TR WITH TARGET OBSERVATIONS

Set $D^\star = \varnothing$ when every required target position has a source match and only source fitting is needed. The algorithm uses the discrepancy oracle of Definition 3.2.

Algorithm 1 Circuit-TR

Require: $D, D^\star; \Delta, \mathcal{G}, \mathcal{G}^\star; T, T^\star, M$
1: **for** $i = M+1, \ldots, T^\star$ **do**
2: $\quad \mathcal{J}_i \leftarrow \{i' \in [T] : \Delta(i, i') = 0\}$
3: $\quad D_i^{\mathrm{TR}} \leftarrow D^\star[Y : V_i^\star, \mathbf{R} : \mathbf{Pa}_i^\star]$
4: $\quad$ **for** $i' \in \mathcal{J}_i$ **do**
5: $\quad\quad D_i^{\mathrm{TR}} \leftarrow D_i^{\mathrm{TR}} \cup D[Y : V_{i'}, \mathbf{R} : \mathbf{Pa}_{i'}]$
6: $\quad$ **end for**
7: $\quad \mu^i_{\mathrm{TR}}(y \mid \mathbf{r}) \leftarrow \hat{P}(y \mid \mathbf{r}; D_i^{\mathrm{TR}})$
8: **end for**
9: **Return** $\mu_{\mathrm{TR}}(y \mid \mathbf{x}) \leftarrow \sum_{v^\star_{M+1:T^\star-1}} \prod_{i=M+1}^{T^\star} \mu^i_{\mathrm{TR}}(v_i^\star \mid \mathbf{pa}_i^\star)$

In D_i^{TR}, the output column at each contributing position is relabeled Y, and its ordered parent columns are relabeled by the common argument list $\mathbf{R}$. Thus $D[Y : V_{i'}, \mathbf{R} : \mathbf{Pa}_{i'}]$ copies the values of source position $V_{i'}$ and its ordered parents into those fitting columns; the target records are aligned in the same way. A union of datasets retains repeated records. Each gate is fitted with the conditional-frequency estimator in Equation (21).

Circuit-TR (Algorithm 1) pools these aligned records and then composes the fitted gates. Figure 5 illustrates how the ordered parents can differ across domains. A gate at a position with a source match can be fitted from source observations; otherwise it requires target observations. The excess-risk bound separates these contributions.

For the set $\mathcal{I}$ of target positions after the input prefix with a source match, Theorem B.3 gives

$$\mathcal{E}(\mu_{\mathrm{TR}}) = \widetilde{\mathcal{O}}\left(\frac{|\mathcal{I}|}{N} + \frac{T^\star - M - |\mathcal{I}|}{n}\right). \tag{17}$$

The second term is omitted when all required target positions have source matches, allowing $n = 0$. Fitting a gate at an unmatched position requires observations of its output and parents.

A.2 Selecting a circuit for observed traces

Keep the observed source and target positions and their causal order fixed. Each proposed pair of diagrams and mechanism-sharing partition yields a candidate predictor by the Circuit-TR construction. Circuit-AD constructs these candidates from source data and a target fitting split, then selects by empirical risk on an independent target selection split. Write $n = n_{\mathrm{tr}} + n_{\mathrm{te}}$ for their sizes; both splits belong to the target adaptation budget. The diagrams and partitions specify how each candidate is fitted; selection assesses the resulting predictor's risk.

Algorithm 2 Circuit-AD with process supervision

Require: $D, D^\star; T, T^\star, M$
1: $D^\star_{\mathrm{tr}}, D^\star_{\mathrm{te}} \leftarrow \mathrm{partition}(D^\star)$; $\mathcal{H} \leftarrow \varnothing$
2: **for** every pair of diagrams $(\mathcal{G}, \mathcal{G}^\star)$ with ordered predecessor parent lists **do**
3: **for** every partition of the source and target positions whose classes contain only positions of the same parent arity **do**
4: Set $\Delta(i, i') = 0$ within a class and 1 otherwise
5: $\mathcal{H} \leftarrow \mathcal{H} \cup \{\text{Circuit-TR}(D, D^\star_{\mathrm{tr}}; \Delta, \mathcal{G}, \mathcal{G}^\star; M)\}$
6: **end for**
7: **end for**
8: **Return** $\mu_{\mathrm{AD}} \in \arg\min_{\mu \in \mathcal{H}} \sum_{v \in D^\star_{\mathrm{te}}} -\log \mu(v_{T^\star} \mid v_{1:M})$

Theorem A.1 (Circuit-AD)**.** *Suppose the source and target sequences have comparable lengths of order T, and fixed fractions of the n target samples are used for fitting and selection. Let μ_{TR} be the oracle comparator, supplied with the correct diagrams and a partition into classes of equal mechanisms and noise distributions, with parent arguments aligned. It uses the same fitting data and rule as Circuit-AD, and its discrepancy oracle is zero exactly within each class. Then, with high probability over the target selection samples,*

$$\mathcal{E}(\mu_{\mathrm{AD}}) \leqslant \mathcal{E}(\mu_{\mathrm{TR}}) + \mathcal{O}\left(\sqrt{\frac{T^2 \log T}{n}}\right). \tag{18}$$

The additional term bounds the risk cost of selecting among the candidate predictors. Source estimation and any required target estimation remain in $\mathcal{E}(\mu_{\mathrm{TR}})$. The comparator need not minimize risk among these fitted predictors; Circuit-AD may select one with lower risk. If the search restricts each position to a fixed number of parents and still includes the oracle diagrams, the selection cost improves to $\mathcal{O}(\sqrt{T \log T / n})$. Both rates follow from a bound on the number of candidates; the search itself can be computationally expensive. This candidate family also fits gates and varies source diagrams and sharing partitions, so its count differs from that of the frozen-library selector in Section 4.1. Theorem B.5 gives the result for arbitrary source and target lengths and split sizes.

In particular, once the Circuit-TR comparator has excess risk at most $\eta/2$, a target sample size of order $\widetilde{\mathcal{O}}(T^2/\eta^2)$ suffices for Circuit-AD to achieve excess risk at most η (Corollary B.6). When all required target positions have source matches, sufficiently many source samples allow Circuit-TR to construct an accurate candidate; Circuit-AD then uses target samples to select a candidate with comparable risk. Otherwise, the fitting split also fits gates at unmatched positions.

The same selection argument applies when candidates pool matching positions within the target as well as the source. The Circuit-TR comparator must use the same pooling rule; Appendix B.4 gives this extension.

B Full statements and proofs

All probability statements below use $0 < \delta < 1$. Risk and excess risk refer throughout to the log loss in Equation (2). The proofs first bound the excess risk of a supplied Circuit-TR composition, then bound the additional risk of selecting a composition from target samples.

B.1 THE GCD CONSTRUCTION

To specify the probabilistic example, let $\mathcal{V} = \{0, \ldots, |\mathcal{V}| - 1\}$ with V_1, V_2 independent and uniform in the source, and X_1, X_2 independent and uniform in the target. Every operation returns its deterministic value with probability $1 - p_{\text{noise}}$ and an independent uniform value otherwise, with independent noise at each position. Addition, multiplication, and subtraction are reduced modulo $|\mathcal{V}|$; remainder uses $a \bmod 0 = a$.

The source in Figure 1a is specified below. Each operation gives the deterministic value before its independent replacement noise. It supplies maximum, minimum, subtraction, remainder, absolute difference, addition, and multiplication. Further computations may use any of the fitted gates, with new parent arguments.

Position	Operation	Position	Operation
V_3	$\max(V_1, V_2)$	V_9	$V_1 V_2$
V_4	$\min(V_1, V_2)$	V_{10}	$\min(V_5, V_6)$
V_5	$V_3 - V_4$	V_{11}	$V_8 - V_9$
V_6	$V_3 \bmod V_4$	V_{12}	$\max(V_{10}, V_{11})$
V_7	$\lvert V_1 - V_2 \rvert$	V_{13}	$V_9 \bmod V_8$
V_8	$V_1 + V_2$	V_{14}	$V_7 + V_{10}$

The target in Figure 1b has inputs $V_1^\star = X_1, V_2^\star = X_2$ and repeats noisy versions of

$$\begin{aligned} V_{3k}^\star &\leftarrow \max(V_{3k-2}^\star, V_{3k-1}^\star), \\ V_{3k+1}^\star &\leftarrow \min(V_{3k-2}^\star, V_{3k-1}^\star), \\ V_{3k+2}^\star &\leftarrow V_{3k}^\star - V_{3k+1}^\star \pmod{|\mathcal{V}|}, \qquad k = 1, \ldots, K. \end{aligned} \tag{19}$$

A final noisy maximum of $V_{3K+1}^\star, V_{3K+2}^\star$ gives Y, at position $T^\star = 3K + 3$. Without noise, $K = |\mathcal{V}|$ suffices to compute GCD, taking $\gcd(0, 0) = 0$. With noise, marginalizing the intermediate variables gives this circuit's concept $P^\star(Y \mid X_1, X_2)$.

In the noiseless construction, one iteration replaces (a, b) by $(\min(a, b), |a - b|)$. This preserves the GCD. While both entries are positive, their sum decreases by at least one per iteration. For unequal positive inputs, the first iteration reduces the sum to $\max(a, b) \leqslant |\mathcal{V}| - 1$; for $a = b$, it already produces $(a, 0)$. Thus $K = |\mathcal{V}|$ is sufficient for all inputs, and further iterations preserve the final GCD. Modular subtraction agrees with ordinary subtraction on the ordered maximum and minimum. Positive replacement noise makes every local conditional positive; at positive noise the query is the distribution of this specified circuit's final output.

Replacing the third line of Equation (19) by $V_{3k}^\star \bmod V_{3k+1}^\star$ gives a remainder-based circuit with the same three-operation pattern. The paired GCD experiment uses this construction with $K = 4$ over ten values, giving 15 positions (Appendix G.3). It needs only logarithmically many iterations to compute GCD without noise. A shorter implementation dispenses with maximum and minimum at every iteration. For its noiseless computation, fix K for the vocabulary and unroll

$$r_0 = X_1, \quad r_1 = X_2, \quad r_{j+1} = r_{j-1} \bmod r_j \ (j = 1, \ldots, K), \qquad Y = (r_K + r_{K+1}) \bmod |\mathcal{V}|. \tag{20}$$

This uses K remainder operations and one addition, hence $K + 3$ positions including the inputs. Without replacement noise, once a remainder is zero, the pair alternates between $(g, 0)$ and $(0, g)$, where $g = \gcd(X_1, X_2)$; its sum is always g. After at most one initial step the operands are ordered, and within each two nonterminal steps their maximum decreases by a factor of at least two. Thus $K = 2\lceil \log_2 |\mathcal{V}| \rceil + 2$ suffices. Direct enumeration verifies that $K = 5$ suffices at $|\mathcal{V}| = 10$ and $K = 10$ at $|\mathcal{V}| = 100$, giving eight and thirteen positions respectively.

Each noisy construction defines its own concept. To compare their induced conditionals with one common target concept, Appendix E.4 fixes the output-noised GCD conditional and bounds each construction's approximation risk. Appendix F examines why a restricted gate library can retain a longer noiseless representation while changing the conditional distributions attainable at positive noise.

B.2 POSITIVITY AND CONDITIONAL ESTIMATION

For an evaluation distribution Q with conditional $q(y \mid x)$, write

$$\mathcal{R}_Q(\mu) = \mathbb{E}_Q[-\log \mu(Y \mid X)], \qquad \mathcal{E}_Q(\mu) = \mathcal{R}_Q(\mu) - \mathcal{R}_Q(q).$$

We omit the subscript when $Q = P^\star$, as in Equation (2). The optimal risk is $\mathcal{R}_Q(q) = H_Q(Y \mid X)$.

The explicit estimation bounds use two positive lower bounds. True conditional probabilities are at least ϵ, and each parent configuration at a position contributing data has probability at least ρ under that position's data distribution. For finite variables, strict positivity supplies both bounds. Keeping them separate records the different costs of predicting rare outputs and observing rare parent configurations. The bounds hold uniformly over parent arguments, so they also control gate errors when target compositions visit those arguments with different frequencies. No lower bound on the probability of a complete sequence is needed. For a position without parents, take $\rho = 1$.

Let $\bar{P}(\cdot \mid x; D)$ be empirical conditional frequencies, or the uniform distribution if x is absent from D. For $0 < \epsilon < 1/|\mathcal{Y}|$, define

$$\hat{P}(\cdot \mid x; D) = \operatorname*{arg\,min}_{q \in \mathcal{P}_\epsilon(\mathcal{Y})} \|q - \bar{P}(\cdot \mid x; D)\|_2^2, \tag{21}$$

where $\mathcal{P}_\epsilon(\mathcal{Y})$ is the set of probability vectors with $q(y) \geqslant \epsilon$ for every $y \in \mathcal{Y}$. The adjustment preserves normalization and makes log loss bounded. The next lemma permits pooling several positions from the same sequence without treating those positions as independent observations.

Lemma B.1 (Conditional estimation)**.** *Let the data consist of independent blocks $b = 1, \ldots, B_0$, with $r_b \geqslant 1$ records (X_{bj}, Y_{bj}) in block b. Records within a block may be dependent. Suppose every record has the same conditional distribution $p(y \mid x) \geqslant \epsilon$, and $P(X_{bj} = x) \geqslant \rho > 0$ for every $x \in \mathcal{X}$. Set*

$$m = \frac{\sum_b r_b}{\max_b r_b}, \qquad h = \log \frac{4|\mathcal{X}||\mathcal{Y}|}{\delta}. \tag{22}$$

For any evaluation distribution Q with conditional distribution p, with probability at least $1 - \delta$,

$$\mathcal{E}_Q(\hat{P}) \leqslant \min\left\{\log \frac{1}{\epsilon}, \frac{32|\mathcal{Y}|h}{\epsilon\rho m}\right\}. \tag{23}$$

The evaluation covariate distribution need not equal any training covariate distribution.

Proof. Fix x, y, let $r = \max_b r_b$, and define the counts

$$C_b = \sum_{j=1}^{r_b} \mathbb{1}\{X_{bj} = x\}, \qquad C_{by} = \sum_{j=1}^{r_b} \mathbb{1}\{X_{bj} = x, Y_{bj} = y\}, \qquad Z_b = C_{by} - p(y \mid x) C_b.$$

Writing $\mu_x = \sum_b \mathbb{E}C_b$, we have $\mu_x \geqslant \rho \sum_b r_b$, $\mathbb{E}Z_b = 0$, and

$$\sum_b \operatorname{Var}(C_b) \leqslant r\mu_x, \qquad \sum_b \operatorname{Var}(Z_b) \leqslant r\mu_x.$$

For the second inequality, Cauchy–Schwarz gives $Z_b^2 \leqslant r_b \sum_j \mathbb{1}\{X_{bj} = x\}(\mathbb{1}\{Y_{bj} = y\} - p(y \mid x))^2$; taking expectations uses only the common conditional distribution. Both centered block counts and Z_b have absolute value at most r. Bernstein's inequality and a union bound imply, simultaneously for all x, y, with probability at least $1 - \delta$,

$$\sum_b C_b \geqslant \mu_x - \sqrt{2r\mu_x h} - \tfrac{2}{3} r h,$$

$$\left|\sum_b Z_b\right| \leqslant \sqrt{2r\mu_x h} + \tfrac{2}{3} r h.$$

Indeed, there are at most $|\mathcal{X}| + 2|\mathcal{X}||\mathcal{Y}|$ one-sided events, each of probability at most e^{-h}.

If $\rho m \geqslant 32h$, then $\mu_x \geqslant 32rh$, the denominator $\sum_b C_b$ is at least $\mu_x/2$, and the empirical conditional frequency $\bar{P}(y \mid x)$ satisfies

$$|\bar{P}(y \mid x) - p(y \mid x)|^2 \leqslant \frac{32rh}{\mu_x} \leqslant \frac{32h}{\rho m}.$$

Since $p(\cdot \mid x) \in \mathcal{P}_\epsilon(\mathcal{Y})$, nonexpansiveness of Euclidean projection gives $\|\hat{P}(\cdot \mid x) - p(\cdot \mid x)\|_2 \leqslant \|\bar{P}(\cdot \mid x) - p(\cdot \mid x)\|_2$. Using $\log t \leqslant t - 1$,

$$D_{\mathrm{KL}}\big(p(\cdot \mid x) \,\|\, \hat{P}(\cdot \mid x)\big) \leqslant \sum_y \frac{(p(y \mid x) - \hat{P}(y \mid x))^2}{\hat{P}(y \mid x)} \leqslant \frac{32|\mathcal{Y}|h}{\epsilon \rho m}.$$

Average under the evaluation covariate distribution. Excess risk is exactly this expected KL, since

$$\mathcal{E}_Q(\hat{P}) = \mathbb{E}_{Q(X)} D_{\mathrm{KL}}\big(p(\cdot \mid X) \,\|\, \hat{P}(\cdot \mid X)\big). \tag{24}$$

For every realization of the fitting data, $\mathcal{E}_Q(\hat{P}) \leqslant \log(1/\epsilon)$. If $\rho m < 32h$, the other term in Equation (23) exceeds $|\mathcal{Y}|/\epsilon \geqslant \log(1/\epsilon)$, so the deterministic bound proves the remaining case. □

Lemma B.2 (Causal pooling of matched positions)**.** *Suppose there are N independent source blocks containing $J \geqslant 1$ matched records each and n independent target blocks containing one matched record each. All records have conditional distribution $p(y \mid x) \geqslant \epsilon$ and $P(X = x) \geqslant \rho$. Assume that records within each block can be ordered by the causal order so that, for every x, y, $\mathbb{1}\{X = x\}$ is measurable before the corresponding outcome and*

$$\mathbb{E}\big[\mathbb{1}\{X = x\}\big(\mathbb{1}\{Y = y\} - p(y \mid x)\big) \,\big|\, \textit{preceding variables}\big] = 0.$$

If $\rho(N + n/J) \geqslant 32h$ for $h = \log(4|\mathcal{X}||\mathcal{Y}|/\delta)$, the estimator formed from all $JN + n$ records satisfies, for any evaluation distribution Q with conditional p, with probability at least $1 - \delta$,

$$\mathcal{E}_Q(\hat{P}) \leqslant \min\left\{\log\frac{1}{\epsilon}, \frac{32|\mathcal{Y}|h}{\epsilon\rho(JN + n)}\right\}.$$

Proof. Fix x, y and let C be the total number of records with $X = x$, and let $\mu = \mathbb{E}C \geqslant \rho(JN + n)$. The block counts are independent, have maximum block size J, and have variance at most their block size times their mean. Bernstein's inequality gives, simultaneously for all x, $C \in [\mu/2, 2\mu]$ whenever $\mu \geqslant 32Jh$.

Order the residuals within each source block by the causal order and concatenate the independent blocks. The displayed conditional-mean assumption makes $\mathbb{1}\{X = x\}(\mathbb{1}\{Y = y\} - p(y \mid x))$ a martingale difference with absolute value at most one and predictable variance at most $\mathbb{1}\{X = x\}$. On the count event, Freedman's inequality therefore gives, simultaneously for all x, y,

$$\left|\sum \mathbb{1}\{X = x\}(\mathbb{1}\{Y = y\} - p(y \mid x))\right| \leqslant \sqrt{4\mu h} + \frac{2}{3}h.$$

Since $\mu \geqslant 32h$, division by $C \geqslant \mu/2$ and elementary algebra give $|\bar{P}(y \mid x) - p(y \mid x)|^2 \leqslant 32h/\mu \leqslant 32h/(\rho(JN + n))$. Projection nonexpansiveness and the same KL bound as in Lemma B.1 finish the proof. If the displayed rate exceeds the deterministic upper bound $\log(1/\epsilon)$, the latter gives the minimum. □

B.3 Composition and proof of Theorem 3.4

Theorem B.3 (Circuit-TR excess-risk bound)**.** *Let $\mathcal{I} = \{i \in \{M + 1, \ldots, T^\star\} : \mathcal{J}_i \neq \varnothing\}$ and $c = \max_{i>M} |\mathbf{Pa}_i^\star|$. Assume the conditional-probability bound ϵ and parent-configuration bound ρ at the target positions and their matched source positions. With $h = \log(4(T^\star - M)|\mathcal{V}|^{c+1}/\delta)$, with probability at least $1 - \delta$,*

$$\mathcal{E}(\mu_{\mathrm{TR}}) \leqslant \frac{32|\mathcal{V}|h}{\epsilon\rho}\left(\frac{|\mathcal{I}|}{N} + \frac{T^\star - M - |\mathcal{I}|}{n}\right). \tag{25}$$

If every position after the prefix has a source match, the second term is absent and $n = 0$ is allowed. Otherwise $n \geqslant 1$ is required. The excess risk is also at most $\log(1/\epsilon)$.

Let $X = V_{1:M}^\star$, $Z = V_{M+1:T^\star-1}^\star$, $Y = V_{T^\star}^\star$, and let $p_i(\cdot \mid \mathbf{pa}_i^\star)$ and $q_i(\cdot \mid \mathbf{pa}_i^\star)$ denote the true and estimated local conditionals. The known causal order and absence of confounding give

$$P^\star(z, y \mid x) = \prod_{i=M+1}^{T^\star} p_i(v_i \mid \mathbf{pa}_i^\star), \qquad Q(z, y \mid x) = \prod_{i=M+1}^{T^\star} q_i(v_i \mid \mathbf{pa}_i^\star).$$

The log-sum inequality, followed by the chain rule for KL, yields

$$\begin{aligned}\mathcal{E}(\mu_{\mathrm{TR}}) &= \mathbb{E}_X D_{\mathrm{KL}}\big(P^\star(Y \mid X) \,\|\, Q(Y \mid X)\big) \\ &\leqslant \mathbb{E}_X D_{\mathrm{KL}}\big(P^\star(Z, Y \mid X) \,\|\, Q(Z, Y \mid X)\big) \\ &= \sum_{i=M+1}^{T^\star} \mathbb{E}_{P^\star} D_{\mathrm{KL}}\big(p_i(\cdot \mid \mathbf{Pa}_i^\star) \,\|\, q_i(\cdot \mid \mathbf{Pa}_i^\star)\big). \end{aligned} \tag{26}$$

Thus the query excess risk is at most the sum of local excess risks, evaluated under the target distribution of their parent arguments.

If $J_i = |\mathcal{J}_i| > 0$, each source sequence contributes a block of J_i matched records and each target sequence contributes one record. By Equation (1) and the certified sharing of mechanisms and noise distributions, all records share p_i after parent alignment. Lemma B.1 applies with

$$m_i = \frac{J_i N + n}{J_i} = N + \frac{n}{J_i}. \tag{27}$$

For $J_i = 0$, it applies with $m_i = n$. Union-bound over the $T^\star - M$ positions after the prefix, taking $h = \log(4(T^\star - M)|\mathcal{V}|^{c+1}/\delta)$. Equations (23) and (26) give

$$\mathcal{E}(\mu_{\mathrm{TR}}) \leqslant \frac{32|\mathcal{V}|h}{\epsilon\rho}\left(\sum_{i\in\mathcal{I}} \frac{1}{N + n/J_i} + \frac{T^\star - M - |\mathcal{I}|}{n}\right). \tag{28}$$

Replacing $N + n/J_i$ by N gives Equation (25). When every required target position has a source match, Equation (26) directly gives Equation (4) with any uniform local KL bound a. The empirical estimate supplies $a = \widetilde{\mathcal{O}}(1/N)$. The refined bound allows dependence within a sequence. When there are no unmatched positions, omit the last term, including when $n = 0$. If an unmatched position has no target data, use the deterministic bound $\mathcal{E}(\mu_{\mathrm{TR}}) \leqslant \log(1/\epsilon)$ instead.

Under the independent-local-noise SCM assumptions, the matched residuals satisfy the conditional-mean hypothesis of Lemma B.2 after ordering positions causally. If $\rho(N + n/J_i) \geqslant 32h$ for every matched position, that lemma replaces the matched-position term in Equation (28) by $1/(J_i N + n)$:

$$\mathcal{E}(\mu_{\mathrm{TR}}) \leqslant \frac{32|\mathcal{V}|h}{\epsilon\rho}\left(\sum_{i\in\mathcal{I}} \frac{1}{J_i N + n} + \frac{T^\star - M - |\mathcal{I}|}{n}\right).$$

This is the $J_i N + n$ pooling gain. The sequence-level coverage condition protects against missing parent configurations: without it, repeated positions in a block can repeat an unobserved parent configuration and cannot supply independent coverage.

Finally, the deterministic query-loss bound follows because, for every prefix and label,

$$Q(y \mid x) = \sum_z q_{T^\star}(y \mid \mathbf{pa}_{T^\star}^\star) \prod_{i=M+1}^{T^\star - 1} q_i(v_i \mid \mathbf{pa}_i^\star) \geqslant \epsilon. \tag{29}$$

The product before the final position sums to one. This argument also applies to candidate diagrams and to conditionals that depend on the full predecessor prefix.

B.4 Pooling matched target positions

The candidate partition in process-supervised Circuit-AD (Algorithm 2) can also prescribe pooling target positions with the same conditional distribution after parent alignment. Given a correct partition, let J_i be the number of matched source positions and $J_i^\star \geqslant 1$ the number of matched target positions, including i itself. The alternative estimator pools all their aligned records. For $n \geqslant 1$, independent source sequences contribute blocks of size J_i when $J_i > 0$, and target sequences contribute blocks of size $J_i^\star$. Hence the conditional-estimation lemma applies with

$$m_i = \frac{J_i N + J_i^\star n}{\max\{J_i, J_i^\star\}}.$$

With the same floor and coverage assumptions at every pooled position, and $h = \log(4(T^\star - M)|\mathcal{V}|^{c+1}/\delta)$, the Circuit-TR predictor $\mu_{\mathrm{TR}}^{\mathrm{pool}}$ satisfies, with probability at least $1-\delta$,

$$\mathcal{E}(\mu_{\mathrm{TR}}^{\mathrm{pool}}) \leqslant \frac{32|\mathcal{V}|h}{\epsilon\rho} \sum_{i=M+1}^{T^\star} \frac{1}{m_i}.$$

The proof is the same block-estimation and KL-composition argument as above. Under causal ordering and $\rho m_i \geqslant 32h$, the causal-residual proof gives denominator $J_i N + J_i^\star n$ instead. Pooling more positions does not justify treating their parent configurations as independently observed.

An exhaustive Circuit-AD search using this fitting rule has the same structural candidate count. Its true-partition candidate is $\mu_{\mathrm{TR}}^{\mathrm{pool}}$ fitted on $D, D_{\mathrm{tr}}^\star$. The finite-selection lemma below therefore gives, with probability at least $1-\delta$,

$$\mathcal{E}(\mu_{\mathrm{AD}}^{\mathrm{pool}}) \leqslant \mathcal{E}(\mu_{\mathrm{TR}}^{\mathrm{pool}}) + 2\log(1/\epsilon)\sqrt{\frac{\log(2H_0/\delta)}{2n_{\mathrm{te}}}}.$$

The Circuit-TR comparator must use the same pooling rule as the candidates. It may differ from Algorithm 1, which pools only position i from each target sequence.

B.5 Finite selection and proof of Theorem A.1

Lemma B.4 (Selection on held-out target data)**.** *Let $\mathcal{H}$ be a nonempty finite set of predictors constructed independently of $D_{\mathrm{te}}^\star$, and assume $\mu(y \mid x) \geqslant \epsilon$ for every $\mu \in \mathcal{H}$. If $\hat{\mu}$ minimizes empirical risk on $n_{\mathrm{te}} \geqslant 1$ independent target samples, then, with probability at least $1-\delta$,*

$$\mathcal{E}(\hat{\mu}) \leqslant \min_{\mu\in\mathcal{H}} \mathcal{E}(\mu) + 2\log(1/\epsilon)\sqrt{\frac{\log(2|\mathcal{H}|/\delta)}{2n_{\mathrm{te}}}}. \tag{30}$$

Proof. Condition on the data used to construct $\mathcal{H}$. Every candidate loss lies in $[0, \log(1/\epsilon)]$. Hoeffding's inequality and a union bound control the absolute difference between empirical and population risk for every candidate by $\log(1/\epsilon)\sqrt{\log(2|\mathcal{H}|/\delta)/(2n_{\mathrm{te}})}$. Apply this deviation bound to the empirical minimizer and a population minimizer, and use empirical optimality. Integrating over the fitting data gives the unconditional statement. This is the usual finite-class selection argument (Shalev-Shwartz & Ben-David, 2014), with its loss bound explicit. □

Theorem B.5 (Circuit-AD oracle inequality)**.** *Consider the process-supervised Circuit-AD construction in Algorithm 2. Let $n_{\mathrm{te}} = |D_{\mathrm{te}}^\star| \geqslant 1$, and let H_0 be an upper bound on its number of structural candidates. Let μ_{TR} be the Circuit-TR predictor using the supplied correct diagrams and a partition into classes of equal mechanisms and noise distributions, with parent arguments aligned. It uses the same fitting data $D, D_{\mathrm{tr}}^\star$ and fitting rule as Circuit-AD. With probability at least $1-\delta$,*

$$\mathcal{E}(\mu_{\mathrm{AD}}) \leqslant \mathcal{E}(\mu_{\mathrm{TR}}) + 2\log(1/\epsilon)\sqrt{\frac{\log(2H_0/\delta)}{2n_{\mathrm{te}}}}. \tag{31}$$

In Circuit-AD, the supplied correct diagrams and mechanism partition are among the candidates. Class members have equal arity, and their ordered parent lists align the arguments. Their Circuit-TR predictor uses the same fitting data and rule as every other candidate. Equation (29) bounds all candidate query losses, so Lemma B.4 proves the additive oracle inequality. For this partition-based candidate family, the comparator's discrepancy oracle is zero exactly between positions in the same class; arbitrary partial sets of pairwise certifications need not produce one of these fitted candidates.

For completeness, a partition of $T + T^\star$ positions can be encoded by assigning each position one of $T + T^\star$ labels, giving at most $(T + T^\star)^{T+T^\star}$ partitions. For a position with $i-1$ predecessors, the number of ordered parent lists is

$$\sum_{k=0}^{i-1} \frac{(i-1)!}{(i-1-k)!} \leqslant e(i-1)!.$$

Consequently, the graph count for s positions is bounded by $((2s)!)^s$, and

$$|\mathcal{H}| \leqslant H_0 := (T+T^\star)^{T+T^\star}((2T)!)^T((2T^\star)!)^{T^\star}. \tag{32}$$

In particular,

$$\log H_0 = \mathcal{O}\big((T+T^\star)\log(T+T^\star) + T^2\log(2T) + (T^\star)^2\log(2T^\star)\big). \tag{33}$$

This count is an upper bound; incompatible arities can be excluded and identical candidate predictors can be merged.

If all candidate parent lists have length at most c, replace the graph count by

$$G_c(s) = \prod_{i=1}^{s}\sum_{k=0}^{\min(c,i-1)}\frac{(i-1)!}{(i-1-k)!}.$$

The resulting log count is $\mathcal{O}((T+T^\star)\log(T+T^\star) + cT\log(2T) + cT^\star\log(2T^\star))$. The restricted class must contain the true ordered parent lists. For comparable source and target lengths and n_{te} a fixed fraction of n, these counts give the rates in Theorem A.1.

B.6 Proof of Corollary B.6

Corollary B.6 (Sufficient target sample size)**.** *Fix the observed positions, their causal order, and the candidate family. Suppose the oracle candidate has excess risk at most $\eta/2$ with probability at least $1-\delta/2$. If*

$$n_{\mathrm{te}} \geqslant \frac{8\log^2(1/\epsilon)\log(4H_0/\delta)}{\eta^2}, \tag{34}$$

then Circuit-AD has excess risk at most η with probability at least $1-\delta$.

Use confidence $\delta/2$ for the Circuit-TR estimation event and $\delta/2$ for Lemma B.4. If the oracle candidate's excess risk is at most $\eta/2$ and $n_{\mathrm{te}} \geqslant 8\log^2(1/\epsilon)\log(4H_0/\delta)/\eta^2$, the selection term is at most $\eta/2$. Adding the two terms proves the result with probability at least $1-\delta$. When all target positions after the prefix have source matches, Theorem 3.4 supplies the estimation event using sufficiently large N, even with no target fitting records.

B.7 Composition selection and proof of Corollary 4.2

Theorem B.7 (Adaptation and circuit size)**.** *Let $n_{\mathrm{te}} \geqslant 1$ and $B = \log(1/\epsilon)$. Construct the source estimates independently of the target selection pairs, with every fitted gate having probability floor ϵ, and hold them fixed during selection. Fix a Circuit-TR representation C_* composed of $P_1, \ldots, P_d$ whose query equals $P^\star(y \mid \mathbf{x})$, and let L be its size, including the M input positions. Suppose that, with probability at least $1-\delta/2$, each fitted gate used by C_* has conditional KL error at most a relative to its true source conditional, uniformly over parent configurations. Define*

$$\gamma_\ell = B\sqrt{\frac{(\ell-M)\log\big(2d(c+1)\ell^c\big) + \log(4/\delta)}{2n_{\mathrm{te}}}}. \tag{35}$$

Let $\widehat{\mathcal{R}}_{\mathrm{te}}(\mu)$ denote empirical risk on the target input/output pairs. Select $\hat{\mu}$ by minimizing $\widehat{\mathcal{R}}_{\mathrm{te}}(\hat{\mu}_C) + \gamma_\ell$ over all $\ell \geqslant M+1$ and $C \in \mathcal{H}_\ell$. Then, with probability at least $1-\delta$,

$$\mathcal{E}(\hat{\mu}) \leqslant (L-M)a + 2\gamma_L. \tag{36}$$

For a candidate of size ℓ, write $k = \ell - M$. At position i, the number of ordered predecessor lists of length at most c is

$$A_c(i) = \sum_{j=0}^{\min(c,i-1)}\frac{(i-1)!}{(i-1-j)!} \leqslant (c+1)\ell^c.$$

Consequently, $|\mathcal{H}_\ell| \leqslant [d(c+1)\ell^c]^k$. Every candidate query has probability at least ϵ: after fixing the input prefix, the final conditional has that floor and the preceding candidate conditionals sum to one.

Condition on the source data. For each ℓ, apply Hoeffding's inequality to the query losses and use confidence weight $(\delta/2)2^{-k}$. The union bound over $\mathcal{H}_\ell$ and then over $k \geqslant 1$ gives, with probability at least $1 - \delta/2$ over the target selection pairs,

$$|\mathcal{R}(\hat{\mu}_C) - \widehat{\mathcal{R}}_{\mathrm{te}}(\hat{\mu}_C)| \leqslant \gamma_\ell \qquad \text{for every } \ell \geqslant M+1,\ C \in \mathcal{H}_\ell.$$

Indeed, the probability contribution at size ℓ is at most $\frac{\delta}{2}2^{-k}$ because γ_ℓ^2 contains $[\log(4/\delta) + k\log(2d(c+1)\ell^c)]B^2/(2n_{\mathrm{te}})$. The penalties diverge with ℓ, while empirical losses are in $[0, B]$, so only finitely many sizes can minimize the penalized criterion.

On the simultaneous deviation event, penalized empirical optimality gives, for every fitted circuit predictor $\hat{\mu}_C$ with $C \in \mathcal{H}_\ell$,

$$\begin{aligned}\mathcal{R}(\hat{\mu}) &\leqslant \widehat{\mathcal{R}}_{\mathrm{te}}(\hat{\mu}) + \gamma_{\hat{\ell}} \\ &\leqslant \widehat{\mathcal{R}}_{\mathrm{te}}(\hat{\mu}_C) + \gamma_\ell \\ &\leqslant \mathcal{R}(\hat{\mu}_C) + 2\gamma_\ell.\end{aligned}$$

This compares population risks without a size penalty on the selected predictor's risk. The penalty permits simultaneous comparison across all finite circuit sizes; it does not require the comparator size to be known. Subtracting the target Bayes risk gives, simultaneously for every fitted circuit predictor,

$$\mathcal{E}(\hat{\mu}) \leqslant \mathcal{E}(\hat{\mu}_C) + 2\gamma_\ell, \qquad C \in \mathcal{H}_\ell. \tag{37}$$

This proves Equation (8) independently of the source estimation assumption. Source estimation error and any failure to represent the target query are already included in the fitted comparator's excess risk.

Apply the comparison to the Circuit-TR representation C_* of size L. Its fitted predictor $\hat{\mu}_{C_*}$ uses the same gates as the selector. The representation supplies a joint conditional distribution of its intermediate variables and output given the inputs, whose output marginal is $P^\star(y \mid \mathbf{x})$; these intermediates need not coincide with those of the target SCM. The KL chain rule and marginalization therefore give

$$\mathcal{E}(\hat{\mu}_{C_*}) \leqslant \sum_{i=M+1}^{L} \mathbb{E} D_{\mathrm{KL}}\big(P_{a_i}(\cdot \mid \mathbf{pa}_i) \,\|\, \hat{P}_{a_i}(\cdot \mid \mathbf{pa}_i)\big) \leqslant (L-M)a$$

on the source estimation event, with the expectation taken under this representation and the target input distribution. Apply the preceding population-risk comparison to C_* and subtract the target Bayes risk. The source and target events each fail with probability at most $\delta/2$, proving Equation (36). One may choose C_* to have the smallest size among representations satisfying the stated gate-error assumption; the selector does not need to know this size.

To obtain the source estimation event, apply Lemma B.1 with confidence $\delta/(2d)$ to each required source conditional. The proof controls every parent configuration simultaneously. If each estimate has effective block count at least m, as defined in that lemma, and coverage at least ρ, take

$$a = \min\left\{B, \frac{32|\mathcal{V}|\log(8d|\mathcal{V}|^{c+1}/\delta)}{\epsilon\rho m}\right\}.$$

For one source position and a specified predecessor list per estimate, $m = N$. Different estimates may use the same source samples; a union bound does not require their independence. Only the conditionals used by the comparator circuit need satisfy the source estimation assumptions. Other candidates still have bounded query loss and can be included in selection. For fixed c, ϵ, and confidence, $2\gamma_L = \mathcal{O}(\sqrt{(L-M)\log(2dL)/n_{\mathrm{te}}})$. Taking $n_{\mathrm{te}} = n$ gives Equation (10).

Separating approximation and source estimation. The fitted-comparator inequality Equation (37) applies even when no composition of the true source conditionals represents the target query exactly. The next proposition separates approximation and source estimation for a fixed source composition. Its cross term accounts for the fact that KL divergence does not satisfy a triangle inequality.

Proposition B.8 (Approximation and source estimation)**.** *Fix a circuit C of size ℓ formed from true source conditional distributions, with query μ_C, and write $A_C = \mathcal{E}(\mu_C)$. Suppose its fitted gates*

have conditional KL error at most a uniformly over their arguments, on an event of probability at least $1 - \delta/2$. Let $b_C = (\ell - M)a$, and assume both the true conditional distributions and fitted gates have probability floor ϵ. With probability at least $1 - \delta$, the size-penalized selector satisfies

$$\mathcal{E}(\hat{\mu}) \leqslant A_C + b_C + \frac{\sqrt{A_C b_C}}{\epsilon} + 2\gamma_\ell. \tag{38}$$

Proof. For probability vectors p, q, r with $q, r \geqslant \epsilon$, the identity

$$D_{\mathrm{KL}}(p\|r) = D_{\mathrm{KL}}(p\|q) + D_{\mathrm{KL}}(q\|r) + \sum_y (p_y - q_y)\log(q_y/r_y)$$

and the $1/\epsilon$-Lipschitz property of the logarithm give

$$D_{\mathrm{KL}}(p\|r) \leqslant D_{\mathrm{KL}}(p\|q) + D_{\mathrm{KL}}(q\|r) + \frac{\sqrt{D_{\mathrm{KL}}(p\|q) D_{\mathrm{KL}}(q\|r)}}{\epsilon}.$$

Here Cauchy–Schwarz and Pinsker's inequality imply $\|p - q\|_2^2 \leqslant \frac{1}{2}\|p - q\|_1^2 \leqslant D_{\mathrm{KL}}(p\|q)$, and likewise for q, r. Apply the bound with the true target conditional, μ_C, and $\hat{\mu}_C$, and average over the target inputs. The final-gate floor gives $\mu_C, \hat{\mu}_C \geqslant \epsilon$ by the argument in Equation (29). Another application of Cauchy–Schwarz controls the cross term. KL composition, evaluated under the circuit C and the target input distribution, bounds the expected divergence from μ_C to $\hat{\mu}_C$ by b_C. Combine this with Equation (37) and the two events of failure probability at most $\delta/2$ each. □

C NEURAL-AD: STATEMENTS AND PROOFS

Source next-token fitting controls the gates' conditional KL error. Conditional on these learned gates, target adaptation selects their composition. The results below relate these two stages to the query risk.

C.1 SOURCE TRAINING AND GATE-ESTIMATION ERROR

On complete source traces, a source circuit representation $S = (A, \Phi)$ has probability $\pi_\theta(S)$, and the empirical next-token loss, averaged over these circuit choices, is

$$\widehat{\mathcal{L}}(\theta) = -\frac{1}{NT} \sum_{v \in D} \sum_{i=1}^{T} \mathbb{E}_{S \sim \pi_\theta} \log \Psi_\theta(v_i \mid v_{\mathbf{Pa}_i(S)}; \Phi(i)). \tag{39}$$

Source structural choices are independent across positions, allowing exact evaluation of this expectation by local enumeration. A penalty on the number of distinct gate indicators used and the number of parent edges encourages sharing and small parent sets. After selecting one source circuit representation, we refit its gates and retain the indicators used by that representation. Under source coverage, their conditional KL errors control the excess risk of a transported composition by the conditional KL chain rule (Lemma C.2). The analysis uses these gate-estimation errors without requiring recovery of the source graph.

For a hard source representation $S = (A, \Phi)$, write $q_{\theta,S,i}(\cdot \mid v_{1:i-1}) = \Psi_\theta(\cdot \mid v_{\mathbf{Pa}_i(S)}; \Phi(i))$ for its prediction at position i. The corresponding population next-token loss and its difference from the Bayes next-token loss satisfy

$$\mathcal{L}(\theta) = \frac{1}{T} \sum_{i=1}^{T} \mathbb{E}_{S \sim \pi_\theta} \mathbb{E}_P[-\log q_{\theta,S,i}(V_i \mid V_{1:i-1})], \tag{40}$$

$$\mathcal{L}(\theta) - \frac{H_P(V_{1:T})}{T} = \frac{1}{T} \sum_{i=1}^{T} \mathbb{E}_{S \sim \pi_\theta} \mathbb{E}_P D_{\mathrm{KL}}(P(\cdot \mid V_{1:i-1}) \,\|\, q_{\theta,S,i}(\cdot \mid V_{1:i-1})). \tag{41}$$

For fixed parameters, the generator does not condition its circuit choices on sequence values. The sum includes every source position, with $V_{1:0}$ the empty context. A fixed hard representation is included by taking π_θ to be a point mass.

Finite source samples. Let Θ be a fixed model class that includes the permitted gate predictors and distributions on hard source representations. For each θ, the per-trace loss is the average log loss in Equation (40), including its expectation over S. Let $\mathcal{N}_N$ be the size of a finite uniform cover of these loss functions at radius $1/N$, with centers in the class. Such a cover exists because the trace space is finite and the floored log losses are bounded. The independent observations are the N source traces.

Proposition C.1 (Source prediction from empirical fitting)**.** *Assume all gate probabilities in Θ are at least ϵ. Let $\Omega(\theta) = \mathbb{E}_{S\sim\pi_\theta}[d(S) + \sum_i |\mathbf{Pa}_i(S)|]$ and $\lambda \geqslant 0$. With probability at least $1-\delta$ over the source traces, every fit satisfying*

$$\widehat{\mathcal{L}}(\hat{\theta}) + \lambda\Omega(\hat{\theta}) \leqslant \inf_{\theta\in\Theta}\{\widehat{\mathcal{L}}(\theta) + \lambda\Omega(\theta)\} + \zeta$$

obeys

$$\mathcal{L}(\hat{\theta}) + \lambda\Omega(\hat{\theta}) \leqslant \inf_{\theta\in\Theta}\{\mathcal{L}(\theta) + \lambda\Omega(\theta)\} + 2u_N + \zeta, \tag{42}$$

where

$$u_N = \log(1/\epsilon)\sqrt{\frac{\log(2\mathcal{N}_N/\delta)}{2N}} + \frac{2}{N}.$$

If a selected hard representation and its refitted gates, denoted $\tilde{\theta} \in \Theta$, satisfy

$$\widehat{\mathcal{L}}(\tilde{\theta}) + \lambda\Omega(\tilde{\theta}) \leqslant \widehat{\mathcal{L}}(\hat{\theta}) + \lambda\Omega(\hat{\theta}) + \eta,$$

then Equation (42) *holds for $\tilde{\theta}$ with $\zeta + \eta$ in place of ζ.*

Proof. A trace's expected average log loss is in $[0, \log(1/\epsilon)]$. Hoeffding's inequality and a union bound control empirical minus population loss at every cover center by the square-root term in u_N. Approximating any represented loss changes each of its empirical and population means by at most $1/N$. Thus $|\widehat{\mathcal{L}}(\theta) - \mathcal{L}(\theta)| \leqslant u_N$ simultaneously over Θ. On this event, for every comparator θ,

$$\begin{aligned}\mathcal{L}(\hat{\theta}) + \lambda\Omega(\hat{\theta}) &\leqslant \widehat{\mathcal{L}}(\hat{\theta}) + \lambda\Omega(\hat{\theta}) + u_N \\ &\leqslant \widehat{\mathcal{L}}(\theta) + \lambda\Omega(\theta) + u_N + \zeta \\ &\leqslant \mathcal{L}(\theta) + \lambda\Omega(\theta) + 2u_N + \zeta.\end{aligned}$$

The penalty is a function of the represented choices and has no sampling deviation. Taking the infimum proves the first conclusion. The same argument, preceded by the asserted empirical comparison for $\tilde{\theta}$, adds η. Subtracting $H_P(V_{1:T})/T$ yields the corresponding excess-loss comparison. □

Both fitting errors are measured on the empirical sample used in the comparison. The uniform event permits data-dependent initialization, checkpoint selection, and selection of a hard representation within the fixed class. Their quality is recorded by the achieved errors ζ and η. In particular, the retained gates are assessed after refitting. The next lemma converts their source conditional KL errors into an excess-risk bound for a target composition.

Lemma C.2 (Error of reusing a learned gate)**.** *Let source position j have conditional distribution $P_j(\cdot \mid a)$ on its ordered arguments, and let the gate q_j estimate that distribution. Suppose every argument configuration to be used in the target has source probability at least $\rho_j > 0$. Put*

$$e_j = \mathbb{E}_P D_{\mathrm{KL}}\big(P_j(\cdot \mid \mathbf{Pa}_j) \,\|\, q_j(\cdot \mid \mathbf{Pa}_j)\big).$$

At any matched target position, the conditional KL error of using q_j is at most e_j/ρ_j, under any target distribution on these arguments. If a target circuit uses source gates j_i, its final-query excess log risk satisfies

$$\mathcal{E}(\hat{\mu}_C) \leqslant \sum_{i=M+1}^{L} \frac{e_{j_i}}{\rho_{j_i}}. \tag{43}$$

Here the true circuit and the fitted circuit use the same composition, and each target conditional distribution agrees with its matched source distribution after argument alignment.

Proof. For each covered argument tuple a, nonnegativity of KL gives

$$P(\mathbf{Pa}_j = a) D_{\mathrm{KL}}\big(P_j(\cdot \mid a) \,\|\, q_j(\cdot \mid a)\big) \leqslant e_j.$$

Divide by the source argument probability and average over any target argument distribution. The chain rule for conditional KL then bounds the conditional KL divergence between the true and fitted trace distributions by the sum of their local divergences. Marginalizing the intermediate values cannot increase KL, giving Equation (43). □

This lemma assumes the matching needed by Circuit-TR for the comparator circuit; it does not assume that target adaptation is given that matching. Small average source KL error need not control error on arguments rare under the source. The coverage bound converts that average error into the reuse bound above.

A consequence of sparse population fitting. The two counts in the source objective favor sharing equal conditional tables and omitting unused arguments. The following population characterization makes this precise for a class with arbitrary floored conditional tables. It concerns parent sets and conditional tables of the observed distribution. Prediction and reuse require the gate errors above, independently of this characterization.

Proposition C.3 (Sparse source representation)**.** *Suppose the ordered Markovian source has full support, its true conditional tables have the model's probability floor, and every true parent has an effect on its child's conditional distribution for some setting of the other parents. Suppose each position has at most c parents. Consider hard representations with distinct ordered parents, hard gate indicators, a fixed active-slot pattern for each arity, and arbitrary floored conditional tables. The representation permits enough indicators of each arity, and argument permutations are allowed.*

For sufficiently small $\lambda > 0$, every global minimizer of

$$\mathcal{L}(\theta) + \lambda \left(d(S) + \sum_{i=1}^{T} |\mathbf{Pa}_i(S)| \right) \tag{44}$$

selects the true parent sets and assigns the same gate indicator exactly to positions whose conditional tables agree after an argument permutation. The conditional tables are correct in their selected argument order. Indicator labels and symmetric argument orders need not be unique.

Proof. There are finitely many hard parent and indicator-partition choices. For each such choice, the floored conditional tables form a compact set, and source cross-entropy is continuous on it. A choice unable to realize the true next-token conditionals therefore has a strictly positive minimum excess risk. Let $g > 0$ be the minimum of these excesses over the finitely many nonrealizing choices, or $g = +\infty$ if there are none. The structural penalty is at most $(c+1)T$. Since the true representation is permitted, choosing $0 < \lambda < g/((c+1)T)$ excludes all nonrealizing structures from a global minimum. Within a realizing structure, the penalty is fixed and a population minimizer must achieve the Bayes risk. It remains to characterize minimum-penalty exact representations.

First, each selected parent set contains the true parent set. Otherwise vary an omitted essential parent while fixing all other predecessors. Full support permits both predecessor assignments. The true conditional changes for a suitable choice of the other true parents, whereas the selected-argument predictor remains unchanged, contradicting exact fit.

Next consider one occupied indicator, used at s positions with a shared table of arity a. Full support implies that its table is specified on the entire argument product at every such position. Consequently its set of essential argument slots is the same wherever it is used. If it has $k \geqslant 1$ inessential slots, delete these slots at all s positions and retain the order of the others. This keeps every conditional distribution unchanged, removes ks edges, and does not increase the number of indicators. The representation permits the reduced arity. Thus a minimum-penalty exact representation cannot contain such slots. Combined with containment of the true parents, its selected parent sets are exactly the true ones.

With these parent sets fixed, full support determines each conditional table at every argument tuple. Two tables unequal under every allowed argument permutation cannot share an indicator. Conversely, tables equal under such a permutation can use one indicator by aligning their ordered parents to a

common representative. If they used different indicators, merging them would strictly reduce the penalty without changing the predictions or edges. Hence the minimum-penalty representation has precisely the asserted sharing, up to relabeling and argument symmetries. □

The fixed active-slot convention ensures that a shared indicator has the same argument space wherever it is used. Without it, different null patterns could distinguish positions even under a common indicator. For an expected objective over hard representations, any global minimum in a class containing all point masses can assign positive probability only to globally minimizing hard representations: its objective is an average of their objective values. The population conclusion therefore also applies to representations in the support of such a minimum.

C.2 Process supervision

Condition on the fitted source gates, and let $\mathcal{H}_{T^\star}$ contain all permitted circuits of the observed target size. Write $\mathcal{L}^\star(C)$ for the population counterpart of Equation (13) and

$$\mathcal{L}^\star(\pi) = \mathbb{E}_{C\sim\pi}\mathcal{L}^\star(C), \qquad \widehat{\mathcal{L}}^\star(\pi) = \mathbb{E}_{C\sim\pi}\widehat{\mathcal{L}}^\star(C).$$

The distribution π can correlate choices at different positions. Each circuit has fixed gate choices and ordered parent lists, selected without observing its execution values.

Proposition C.4 (Process-supervised fitting)**.** *Let $h^\star = H_{P^\star}(V^\star_{M+1:T^\star} \mid \mathbf{X})/(T^\star - M)$. For any distribution π on $\mathcal{H}_{T^\star}$, its final-query predictor satisfies*

$$\mathcal{E}(\mu_\pi) \leqslant (T^\star - M)\big[\mathcal{L}^\star(\pi) - h^\star\big]. \tag{45}$$

Suppose the gates have probability floor ϵ and the target sample contains n independent traces. With probability at least $1-\delta$, every fitted distribution satisfying

$$\widehat{\mathcal{L}}^\star(\hat\pi) \leqslant \min_{C\in\mathcal{H}_{T^\star}} \widehat{\mathcal{L}}^\star(C) + \eta$$

obeys

$$\mathcal{E}(\mu_{\hat\pi}) \leqslant (T^\star - M)\left[\min_{C\in\mathcal{H}_{T^\star}} \mathcal{L}^\star(C) - h^\star + 2u^\star_n + \eta\right],$$

$$u^\star_n = \log(1/\epsilon)\sqrt{\frac{\log(2|\mathcal{H}_{T^\star}|/\delta)}{2n}}.$$

Proof. For a fixed circuit, the conditional chain rule gives

$$\begin{aligned}(T^\star - M)\big[\mathcal{L}^\star(C) - h^\star\big] &= \sum_{i=M+1}^{T^\star} \mathbb{E}_{P^\star} D_{\mathrm{KL}}\big(P^\star(\cdot \mid V^\star_{1:i-1}) \,\big\|\, \mu^\star_{C,i}(\cdot \mid V^\star_{1:i-1})\big)\\ &\geqslant \mathcal{E}(\hat\mu_C),\end{aligned}$$

where the inequality marginalizes the intermediate values. Convexity of KL in its second argument yields $\mathcal{E}(\mu_\pi) \leqslant \mathbb{E}_{C\sim\pi}\mathcal{E}(\hat\mu_C)$, proving Equation (45). Hoeffding's inequality and a union bound over $\mathcal{H}_{T^\star}$ give $|\widehat{\mathcal{L}}^\star(C) - \mathcal{L}^\star(C)| \leqslant u^\star_n$ simultaneously for all candidates. Averaging this event over any fitted π preserves the deviation bound. The assumed empirical fitting inequality therefore gives $\mathcal{L}^\star(\hat\pi) \leqslant \min_C \mathcal{L}^\star(C) + 2u^\star_n + \eta$. Combine this with Equation (45). □

Selecting and fitting a circuit. For each position, evaluate every legal gate and parent list on the observed trace. Since the legal circuit family is the product of these local choices and its empirical trace loss is their sum, choosing a local minimizer at every position gives a global minimizing circuit $\hat C$. Maximizing $\log \pi_{\theta^\star}(\hat C)$ then fits the generator to this selection. The log probability conditions each choice on the preceding choices of $\hat C$ as in Equation (66). If the fitted generator assigns probability at least $1-\beta$ to $\hat C$, its empirical trace loss is within $\beta\log(1/\epsilon)$ of the discrete minimum. This follows by bounding the loss of all other circuits by $\log(1/\epsilon)$.

The generator's expected trace loss is evaluated over complete circuits. In particular, an autoregressive generator requires averaging over the preceding choices when computing the expected local log loss at a later position. Its fitted distribution need not be a point mass, even when the selected circuit minimizes the discrete loss exactly.

C.3 Outcome supervision and the Circuit-AD comparison

Condition on the source sample and its fitted gates. Let $\mathcal{H} = \bigcup_{L \geqslant M+1} \mathcal{H}_L$ be the candidate family of Theorem B.7, possibly restricted to a maximum size. Target observations are independent of source fitting. Every fitted gate has probability floor ϵ, so every final-query predictor has the same floor. Use $n_{\mathrm{te}} = n$ and the log-loss penalty γ_L in Equation (35).

For a distribution π on $\mathcal{H}$, define

$$\begin{aligned}
\mu_\pi(y \mid \mathbf{x}) &= \sum_C \pi(C)\hat{\mu}_C(y \mid \mathbf{x}), \\
\widehat{F}(\pi) &= \widehat{\mathcal{R}}_{\mathrm{te}}(\mu_\pi) + \mathbb{E}_\pi \gamma_{|C|}, \\
\widehat{D}(\pi) &= \mathbb{E}_\pi \widehat{\mathcal{R}}_{\mathrm{te}}(\hat{\mu}_C) - \widehat{\mathcal{R}}_{\mathrm{te}}(\mu_\pi).
\end{aligned}$$

The last quantity is nonnegative by convexity of the negative logarithm. It is an empirical Jensen gap, measuring the improvement obtained by averaging conditional predictions. For an unbounded size family, restrict attention to distributions with finite expected penalty.

Theorem C.5 (Neural-AD and Circuit-AD)**.** *With probability at least $1 - \delta$ over the target pairs, the following holds simultaneously for every fitted distribution $\hat{\pi}$, including a distribution chosen using those pairs. If*

$$\widehat{F}(\hat{\pi}) \leqslant \inf_{L, C \in \mathcal{H}_L} \{\widehat{\mathcal{R}}_{\mathrm{te}}(\hat{\mu}_C) + \gamma_L\} + \xi, \qquad \xi \geqslant 0, \tag{46}$$

then

$$\mathcal{E}(\mu_{\hat{\pi}}) \leqslant \inf_{L, C \in \mathcal{H}_L} \{\mathcal{E}(\hat{\mu}_C) + 2\gamma_L\} + \xi + \widehat{D}(\hat{\pi}). \tag{47}$$

If a family Π contains the point mass at every candidate circuit, then

$$\inf_{\pi \in \Pi} \widehat{F}(\pi) \leqslant \inf_{L, C \in \mathcal{H}_L} \{\widehat{\mathcal{R}}_{\mathrm{te}}(\hat{\mu}_C) + \gamma_L\}. \tag{48}$$

Consequently, any fit within ξ of the left-hand infimum satisfies Equation (46)*.*

Proof. The counting and concentration argument of Theorem B.7, conditional on the source data, gives the event

$$|\mathcal{R}(\hat{\mu}_C) - \widehat{\mathcal{R}}_{\mathrm{te}}(\hat{\mu}_C)| \leqslant \gamma_{|C|} \qquad (C \in \mathcal{H})$$

with probability at least $1 - \delta/2$, and hence at least $1 - \delta$. On this event, convexity of log loss gives

$$\begin{aligned}
\mathcal{R}(\mu_{\hat{\pi}}) &\leqslant \mathbb{E}_{\hat{\pi}} \mathcal{R}(\hat{\mu}_C) \\
&\leqslant \mathbb{E}_{\hat{\pi}}[\widehat{\mathcal{R}}_{\mathrm{te}}(\hat{\mu}_C) + \gamma_{|C|}] \\
&= \widehat{F}(\hat{\pi}) + \widehat{D}(\hat{\pi}) \\
&\leqslant \inf_{L, C \in \mathcal{H}_L} \{\widehat{\mathcal{R}}_{\mathrm{te}}(\hat{\mu}_C) + \gamma_L\} + \xi + \widehat{D}(\hat{\pi}) \\
&\leqslant \inf_{L, C \in \mathcal{H}_L} \{\mathcal{R}(\hat{\mu}_C) + 2\gamma_L\} + \xi + \widehat{D}(\hat{\pi}).
\end{aligned}$$

Subtract the target Bayes risk. This event controls every circuit simultaneously, so it also controls averages with target-data-dependent weights. A point mass at C satisfies $\widehat{F}(\delta_C) = \widehat{\mathcal{R}}_{\mathrm{te}}(\hat{\mu}_C) + \gamma_{|C|}$, proving Equation (48). □

For a finite horizon, it suffices that every candidate point mass belongs to the closure of Π: the probability floor makes the empirical risk continuous, and the size penalty is bounded on that family. Appendix H gives a sufficient parameterization. The oracle comparison itself does not need this representability assumption; it needs only the achieved comparison in Equation (46).

Interpreting the two residuals. The residual ξ bounds the excess of the attained penalized empirical risk over the Circuit-AD empirical optimum. It includes errors from search, likelihood fitting, and finite-execution scoring. The theorem is conditional on this attained value; it does not give a computation budget that guarantees it. The empirical Jensen gap can be estimated from sampled circuits' conditional probabilities on the target sample. It vanishes for a point mass; more generally, if $\pi(C_0) \geqslant 1 - \beta$, then

$$0 \leqslant \widehat{D}(\pi) \leqslant \beta\,[\log(1/\epsilon) + \epsilon^{-1}].$$

Indeed, the average component empirical risk differs from the empirical risk of C_0 by at most $\beta \log(1/\epsilon)$, and the probabilities of μ_π and $\hat{\mu}_{C_0}$ differ by at most β at each observed answer. The ϵ^{-1}-Lipschitz bound for log loss supplies the other term. This sufficient bound explains how concentration on one circuit recovers the symbolic comparison.

Averaged prediction can have lower empirical risk than every individual circuit. For example, equal-weight Bernoulli predictors with success probabilities $1/4$ and $3/4$ average to $1/2$; on a balanced binary sample this strictly improves log loss, and equal circuit sizes leave the penalty unchanged. Thus Equation (48) is an inequality. A separately decoded single circuit has its own Circuit-AD guarantee with its own empirical optimization residual. Its objective gap does not measure ξ or $\widehat{D}$ for averaged prediction.

C.4 Outcome-supervised circuit search

Rejection-sampling fine-tuning maintains a structural generator and an archive of candidate circuits. The gate predictor is fixed. The score used to rank an individual circuit is

$$s(C) = -\widehat{\mathcal{R}}_{\mathrm{te}}(\hat{\mu}_C) - \gamma_{|C|}.$$

Larger scores correspond to lower penalized empirical risk. Each round has four steps.

1. **Propose.** Sample structures from the current generator and form local changes to retained or pretrained structures. A local change redraws a gate and its parents or adds a final gate. The longer-composition study also repeats a preceding block with shifted parents and removes positions that are not ancestors of the output. Proposals always use the retained gates, arities, and legal ordered parent lists.
2. **Evaluate and retain.** Execute proposed circuits on the observed target inputs. For each pair $(\mathbf{x}, y)$, average the final gate's probability of y across executions, then take its negative logarithm. The resulting estimate of empirical risk, together with the size penalty, ranks circuits. The archive records distinct circuits, averages their score estimates across visits, and re-evaluates leading candidates with fresh executions.
3. **Fit.** For the leading retained circuits $C_0, \ldots, C_{m-1}$, in decreasing score order, take maximum-likelihood updates minimizing

$$-\sum_{j=0}^{m-1} \frac{2^{-j}}{\sum_{h=0}^{m-1} 2^{-h}} \log \pi_{\theta^\star}(C_j). \tag{49}$$

   The log probability includes size and every gate and parent choice up to the output. Target values enter evaluation; likelihood fitting receives the selected structures.
4. **Select a checkpoint.** Sample fresh circuits and executions from the fitted generator. Average their answer probabilities before taking the log loss, and add the expected penalty, evaluated from the size probabilities. Retain the checkpoint with lowest estimated value of Equation (15).

The longer-composition study additionally proposes changes to archived circuits with high mean probability of the observed answers, penalized by the bounded-loss size penalty. This proposal guide uses the given training pairs. Log loss ranks archive candidates and selects checkpoints; the generator is fitted to the selected circuits by the weighted likelihood objective above. Appendices G.1 and G.4.1 gives the proposal proportions, execution counts, fitting budgets, and stopping rules of the reported studies.

Monte Carlo approximation. For a fixed circuit, averaging the final gate's distribution over sampled intermediate values is an unbiased estimate of its answer distribution. Taking the negative logarithm introduces an upward bias in estimated log loss, in addition to sampling variance. Averaging independently estimated log losses across archive visits reduces variance but does not remove this finite-execution bias. Increasing executions per scored pair reduces both sources of error. The same issue applies when estimating the averaged predictor's risk at checkpoint selection. The theorem concerns the exact conditional distributions of the fitted circuits and their empirical risk on the full target sample; numerical estimates of objective gaps require control of these approximation errors.

C.5 Circuit-AD with a bounded verifier loss

This subsection considers a bounded-verifier variant of the selection criterion.

Condition on the source sample and its fitted gates. For each allowed size $L \geqslant M+1$, let $\mathcal{H}_L$ be the finite set of candidate circuits built from these gates, using the same arities and ordered predecessor choices as Circuit-AD. The candidate predictor is $\hat{\mu}_C$. The target observations $(\mathbf{X}_t, Y_t)$, $t = 1, \ldots, n$, are independent with distribution $P^\star$ and independent of the source fitting data.

For a fixed verifier $r : \mathcal{V}^M \times \mathcal{V} \times \mathcal{V} \to [0,1]$, define

$$\ell_r(C; \mathbf{x}, y) = \mathbb{E}_{\hat{Y} \sim \hat{\mu}_C(\cdot|\mathbf{x})}[1 - r(\mathbf{x}, \hat{Y}, y)], \tag{50}$$

$$\mathcal{R}_r(\hat{\mu}_C) = \mathbb{E}_{P^\star} \ell_r(C; \mathbf{X}, Y), \qquad \widehat{\mathcal{R}}_r(\hat{\mu}_C) = \frac{1}{n} \sum_{t=1}^{n} \ell_r(C; \mathbf{X}_t, Y_t). \tag{51}$$

The expectation over the circuit's generated answer is part of the loss. In particular, exact-answer reward measures sampled-answer accuracy, not the accuracy of the most likely answer. Monte Carlo rollouts estimate this loss without observing target intermediate values.

Choose weights $w_L > 0$ with $\sum_L w_L \leqslant 1$, and upper bounds $H_L \geqslant |\mathcal{H}_L|$ for the nonempty candidate families. Set

$$\gamma_L^{(r)} = \sqrt{\frac{\log(2H_L/(\delta w_L))}{2n}}. \tag{52}$$

One choice is $w_L = 6/(\pi^2 (L-M)^2)$ and $H_L = [dL^c]^{L-M}$. This bound counts an indicator and at most c ordered predecessors at each generated position; the arity and distinctness restrictions can only reduce the count. With this choice the penalty increases with circuit size and is of order $\sqrt{((L-M)\log(2dL^c) + \log(1/\delta))/n}$, up to the logarithm of the size weight. All statements can be restricted to a fixed maximum size.

Theorem C.6 (Circuit-AD with verifier loss)**.** *With probability at least $1-\delta$ over the target sample,*

$$\left|\mathcal{R}_r(\hat{\mu}_C) - \widehat{\mathcal{R}}_r(\hat{\mu}_C)\right| \leqslant \gamma_L^{(r)} \quad \textit{for every } C \in \mathcal{H}_L \textit{ and every allowed } L. \tag{53}$$

Consequently, if Circuit-AD selects $\hat{C}$ to minimize $\widehat{\mathcal{R}}_r(\hat{\mu}_C) + \gamma_{|C|}^{(r)}$, then

$$\mathcal{R}_r(\hat{\mu}_{\hat{C}}) \leqslant \inf_{L, C \in \mathcal{H}_L} \{\mathcal{R}_r(\hat{\mu}_C) + 2\gamma_L^{(r)}\}. \tag{54}$$

An empirical optimization error $\xi \geqslant 0$ adds ξ to the right side.

Proof. For a fixed $C \in \mathcal{H}_L$, the n losses are independent and belong to $[0,1]$. Hoeffding's inequality bounds failure of its deviation inequality by $2\exp(-2n(\gamma_L^{(r)})^2) = \delta w_L / H_L$. A union bound over the circuits of every size gives total failure probability at most $\delta \sum_L w_L \leqslant \delta$. On the uniform event, for any comparator $C \in \mathcal{H}_L$,

$$\begin{aligned}
\mathcal{R}_r(\hat{\mu}_{\hat{C}}) &\leqslant \widehat{\mathcal{R}}_r(\hat{\mu}_{\hat{C}}) + \gamma_{|\hat{C}|}^{(r)} \\
&\leqslant \widehat{\mathcal{R}}_r(\hat{\mu}_C) + \gamma_L^{(r)} + \xi \\
&\leqslant \mathcal{R}_r(\hat{\mu}_C) + 2\gamma_L^{(r)} + \xi.
\end{aligned}$$

Take the infimum over comparators. □

The comparator can be any supplied Circuit-TR construction using the fitted gates. Its risk accounts for the quality of the available gates and their match to the target computation.

D CHALLENGES DUE TO UNOBSERVED CONFOUNDERS

Markovianity justifies the local reuse of conditional distributions in Circuit-TR. When the exogenous noises are independent and each enters only one structural equation, the noise at a position is independent of its parents. Consequently, the conditional distribution given those parents is determined by that position's mechanism and noise distribution. Sharing them across domains preserves the conditional distribution after aligning the ordered parent arguments, even when the parent positions and their distribution change.

This assumption concerns the underlying circuit, including its intermediate variables. Under outcome supervision, those variables are absent from the target dataset and are marginalized when predicting the final output. Their absence does not introduce a shared exogenous cause into the circuit. Unobserved confounding is different: the same exogenous variable can influence a position and one of its parents. Conditioning on the parent then changes the distribution of the noise entering that position, and this change can depend on the domain.

Example D.1 (An unchanged mechanism with a changing conditional distribution)**.** Consider the source sequence (V_1, V_2) in Figure 6. Let

$$U_{12} \sim \mathrm{Bern}(1/2), \qquad U_1 \sim \mathrm{Bern}(q), \qquad U_2 \sim \mathrm{Bern}(0.1)$$

be mutually independent, and define

$$V_1 \leftarrow U_1 \oplus U_{12}, \qquad V_2 \leftarrow (V_1 \vee U_{12}) \oplus U_2, \tag{55}$$

where $\oplus$ and $\vee$ denote exclusive or and logical or. The source has $q = 0.1$. Each target has variables $(V_1^\star, V_2^\star)$ governed by the same equations with exogenous variables $U_1^\star, U_2^\star, U_{12}^\star$, and query variables $X = V_1^\star$ and $Y = V_2^\star$. Consider two possible targets, one with $q = 0.2$ and the other with $q = 0.8$. Only the distribution of U_1 changes across domains. In particular, the structural equation at the second position and the joint distribution of its two exogenous arguments are shared across domains. Using X and Y to index the target positions, both source–target pairs therefore have the same graph and sharing information: $\Delta(Y, 2) = 0$, while $\Delta(X, 1) = 1$ permits the input mechanism to differ.

The first position is uniform in the source and both targets. Conditional on $X = 0$, however, $U_1^\star = U_{12}^\star$ and in a target with parameter q,

$$P^\star(U_{12}^\star = 1 \mid X = 0) = q.$$

At $X = 0$ the outcome is $U_{12}^\star \oplus U_2^\star$, whereas at $X = 1$ it is $1 \oplus U_2^\star$. It follows that

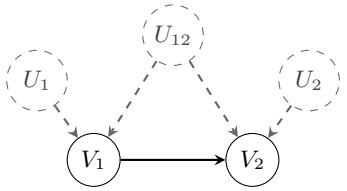
$$P^\star(Y = 1 \mid X = 0) = 0.1 + 0.8q, \qquad P^\star(Y = 1 \mid X = 1) = 0.9. \tag{56}$$

Thus the source conditional $P(V_2 = 1 \mid V_1 = 0)$ is 0.18, while the two possible target conditionals at $X = 0$ are 0.26 and 0.74.

The source distribution, target input marginal, graphs, and sharing information are identical in the two cases, but the target query differs. Hence this information does not determine $P^\star(y \mid x)$, even when both available distributions are known exactly. Every observable joint probability is at least 0.05; the obstruction persists under strict positivity. Labeled target observations can distinguish these targets; the example concerns what the supplied source and structural information identifies without those observations.

Transportability beyond local matching. Circuit transportability asks whether the available distributions and structural constraints determine the target query. The local matching condition used by Circuit-TR is sufficient in the Markovian setting; it is not a complete characterization of this broader question. Example D.1 shows that a shared mechanism and noise distribution need not yield a shared observational conditional under confounding. Statistical transportability provides methods that account for shared exogenous causes and partially observed target domains (Correa & Bareinboim, 2019). Such methods can require factors involving several positions, rather than the individual parent-conditionals composed by Circuit-TR.

Conversely, an unmatched position need not prevent transport of the final query. In a Markovian circuit, an intermediate position with no directed path to the output disappears when the intermediate variables are marginalized. The discrepancy oracle records which local equalities are supplied; failure to supply an equality does not by itself establish that the query cannot be transported.



Domain	q	Probability $V_1 = 1$	Probability $V_2 = 1 \mid V_1 = 0$
Source	0.1	0.5	0.18
Target 1	0.2	0.5	0.26
Target 2	0.8	0.5	0.74

Figure 6: Unobserved confounding in Example D.1. The source graph is shown; each target has the same graph with $X = V_1^\star$ and $Y = V_2^\star$. Dashed nodes and arrows show exogenous variables and their effects. The target rows report the corresponding probabilities under $P^\star$ for $V_1^\star, V_2^\star$. Across domains, only the distribution of U_1 changes. The mechanism at the second position and the joint distribution of its exogenous variables are shared, yet its conditional distribution changes. The two targets have the same input marginal and are equally compatible with the supplied source information.

Partial information about the target query. Even when the source does not determine the entire query, it may constrain the possible target distributions. Partial transportability studies the resulting bounds (Jalaldoust et al., 2024). This possibility does not require confounding. Consider a Markovian chain $X \to Z \to Y$ whose transported conditional distribution satisfies

$$P^\star(Z = 1 \mid X = 0) = 0.1, \qquad P^\star(Z = 1 \mid X = 1) = 0.9,$$

while the target mechanism for Y is unrestricted. Writing $a = P^\star(Y = 1 \mid Z = 0)$ and $b = P^\star(Y = 1 \mid Z = 1)$ gives

$$\big(P^\star(Y = 1 \mid X = 0), P^\star(Y = 1 \mid X = 1)\big) = (0.9a + 0.1b,\ 0.1a + 0.9b).$$

As a, b vary, these pairs occupy a proper subset of the possible binary conditionals. In particular, their difference has magnitude at most 0.8, a constraint that also holds when all conditional probabilities are strictly positive. The transported conditional distribution therefore supplies information about the final query even without determining it. With process supervision, the remaining conditional probabilities can be estimated from the observed target pairs (Z, Y). Under outcome supervision those pairs are unavailable, and any further estimation must use the observed (X, Y) pairs through the composed query.

E Estimation, circuit size, and adaptation rates

E.1 Matched and unmatched positions

Fix the supplied circuit representation and source matches. Complete target traces allow gates at unmatched positions in this representation to be fitted locally, as in Appendix A.1. The bound below describes this estimator. If all required positions have source matches, the target dataset may be empty and the unmatched-position term is omitted. Alternative representations of the same concept are compared separately by the composition oracle inequality.

Let $\mathcal{I}$ contain the positions after the observed prefix with a certified source match, put $J_i = |\mathcal{J}_i|$, and let $c = \max_{i>M} |\mathbf{Pa}_i^\star|$. The proof of Theorem B.3 gives the refined bound

$$\mathcal{E}(\mu_{\mathrm{TR}}) \leqslant \frac{32|\mathcal{V}|h}{\epsilon\rho}\left(\sum_{i\in\mathcal{I}} \frac{1}{N + n/J_i} + \frac{T^\star - M - |\mathcal{I}|}{n}\right), \tag{57}$$

where $h = \log(4(T^\star - M)|\mathcal{V}|^{c+1}/\delta)$. The effective count accounts for dependence among records from the same sequence using independent sequence blocks. Under the causal residual condition in Lemma B.2 and the sequence-level coverage condition $\rho(N + n/J_i) \geqslant 32h$, the matched-position term can instead be written $1/(J_i N + n)$, recovering the pooling gain. Without that condition, more source matches may improve estimation in particular distributions, but the generic bound does not multiply the independent sample count by J_i.

Source estimation depends on the size and coverage of the local conditional tables. With approximately uniform c-parent configurations, ρ is of order $|\mathcal{V}|^{-c}$. The output floor also depends on vocabulary size: for a uniform-replacement noise model, the smallest conditional probability is $p_{\mathrm{noise}}/|\mathcal{V}|$. These dependencies determine the source sample size needed to supply an accurate candidate before target selection.

E.2 MULTIPLE SOURCE DOMAINS

As an extension of the single-source setting, suppose source domain $s \in [S]$ supplies $N_s \geqslant 1$ independent sequences of length T_s, independently across domains and of the target observations. Let J_{is} count source positions in domain s matched to target position i. Circuit-TR pools their aligned records with the target records exactly as in Algorithm 1. Put

$$r_i = \max\{1, J_{i1}, \ldots, J_{iS}\}, \qquad m_i = \frac{n + \sum_{s=1}^{S} J_{is} N_s}{r_i}.$$

When the pooled data are nonempty, Lemma B.1 applies with effective block count m_i: each independent sequence contributing matched records is a block, and r_i bounds its size. Under the same local probability and coverage assumptions, with probability at least $1 - \delta$,

$$\mathcal{E}(\mu_{\mathrm{TR}}) \leqslant \frac{32|\mathcal{V}|h}{\epsilon\rho} \sum_{i=M+1}^{T^\star} \frac{1}{m_i},$$

with $h = \log(4(T^\star - M)|\mathcal{V}|^{c+1}/\delta)$. An unmatched position has $m_i = n$. If $\rho m_i \geqslant 32h$, the causal-residual argument replaces the matched-position denominator by $n + \sum_s J_{is} N_s$. Thus observations with matching conditional distributions can be reused both across positions and across independent samples from different domains.

The process-supervised Circuit-AD construction in Algorithm 2 similarly enumerates a graph for each domain and a partition of all $\sum_s T_s + T^\star$ positions. The finite-selection proof is unchanged, with the candidate count adjusted to those graphs and positions. The circuit-size bound in Theorem B.7 also permits source estimates from any of these domains, provided they are constructed independently of target selection observations. Its source-estimation term requires the same accuracy and coverage conditions for the gates used by the comparator.

E.3 SELECTION WITHOUT A KNOWN CIRCUIT SIZE

The penalty in Equation (35) permits comparison with every finite circuit formed from the fitted gate library without knowing the comparator's size. Its description-length interpretation follows directly from the counting proof. Assign weight $w_\ell = 2^{-(\ell - M)}$ to size $\ell \geqslant M + 1$ and equal weight within each nonempty $\mathcal{H}_\ell$. The resulting circuit weights $\nu(C) = w_\ell / |\mathcal{H}_\ell|$ sum to at most one, and

$$-\log \nu(C) = \log |\mathcal{H}_\ell| + (\ell - M) \log 2 \leqslant (\ell - M) \log\bigl(2d(c+1)\ell^c\bigr).$$

Thus the penalty accounts for describing both the size and the choices within that size. The union bound allocates confidence according to these weights, yielding the risk comparison in Equation (37). This is a statistical form of universality relative to the available gates: the guarantee applies to every finite comparator, including the supplied Circuit-TR composition.

Complexity-penalized likelihood estimation balances approximation and description length (Barron & Cover, 1991). In library learning, Cropper (2019) compares sample bounds when additional reusable predicates shorten a logical representation. Here, the description specifies gate indices and ordered parent lists, and bounded-log-loss concentration yields the square-root penalty in Equation (35). There is also an analogy with universal search, where a description determines the resources assigned to a candidate. Levin's Kt complexity additionally includes the logarithm of execution time (Levin, 1984, Section 1.3). The present penalty allocates statistical confidence and gives a risk guarantee; it does not account for execution time or establish computational optimality of the search.

E.4 CIRCUIT SIZE AND THE ARITHMETIC EXAMPLES

Circuit-TR uses a supplied composition. After gates have been fitted from source data, the size-penalized selector in Equation (6) compares their compositions using target input/output pairs, marginalizing the intermediate variables. Its oracle inequality Equation (37) bounds the selected predictor's excess risk by that of every such circuit plus its size-dependent selection cost.

Example E.1 (GCD with subtraction and remainder)**.** Consider a fixed target that returns $\gcd(x_1, x_2)$ with probability $1 - p_{\mathrm{noise}}$ and an independent uniform value otherwise. Take independent uniform

inputs and $0 < p_{\text{noise}} \leqslant 1/2$. The source supplies operations with this same uniform-replacement noise.

Using $\max, \min, -$, the construction in Equation (19) has $\ell_- = 3|\mathcal{V}| + 3$ positions. If the source also supplies remainder, replacing subtraction by remainder with $a \bmod 0 := a$ gives a circuit of size $\ell_{\text{mod}} = \mathcal{O}(\log |\mathcal{V}|)$. The remainder chain in Equation (20) further reduces the operation count within the same logarithmic order, using a final noisy sum. Each construction has one final noisy operation. Conditional on no replacement at the preceding operations, its output has exactly the target conditional distribution $P^\star(\cdot \mid x_1, x_2)$. Consequently the approximation error of each circuit is at most $(\ell - 3)\log(1/(1 - p_{\text{noise}}))$.

To isolate composition from source estimation, first take the source conditional distributions as known. Equation (8) gives the following sufficient bounds for the same target:

$$\begin{aligned} \text{subtraction:} \quad & \widetilde{\mathcal{O}}\left(p_{\text{noise}}|\mathcal{V}| + \sqrt{\frac{|\mathcal{V}|}{n}}\right), \\ \text{remainder:} \quad & \widetilde{\mathcal{O}}\left(p_{\text{noise}} \log |\mathcal{V}| + \sqrt{\frac{\log |\mathcal{V}|}{n}}\right). \end{aligned} \tag{58}$$

The remainder constructions improve the orders of both the approximation and selection terms. The displayed sizes describe explicit representations over their respective gate libraries. The argument below proves the bounds and includes finite source-estimation error. The comparison is useful when the noise accumulated over the circuit is small; Appendix F gives an exact illustration of the loss of signal in longer representations.

The subtraction update preserves the GCD of the operand pair and, while the operands are positive and unequal, decreases their maximum. Equal operands produce a zero in the next step. Thus $K_- = |\mathcal{V}|$ iterations suffice on $\{0, \ldots, |\mathcal{V}| - 1\}$. For remainder, writing the larger operand as a and the smaller as b, the next pair is $(b, a \bmod b)$. Within two nonterminal steps its maximum is at most half the previous maximum. Hence $K_{\text{mod}} = 2\lceil \log_2 |\mathcal{V}| \rceil + 1$ iterations suffice for $|\mathcal{V}| \geqslant 2$. With $a \bmod 0 = a$, both recurrences preserve the final maximum after a zero is reached. Three operations per iteration and the final maximum give $\ell_- = 3K_- + 3$ and $\ell_{\text{mod}} = 3K_{\text{mod}} + 3$.

For the common-target comparison, fix

$$P^\star(y \mid x_1, x_2) = (1 - p_{\text{noise}})\mathbb{1}\{y = \gcd(x_1, x_2)\} + \frac{p_{\text{noise}}}{|\mathcal{V}|},$$

with $\gcd(0, 0) = 0$ and uniform independent inputs. Every source operation independently replaces its deterministic answer by a uniform value with probability p_{noise}. For any of these noiselessly correct constructions C of size ℓ, let F be the event of no replacement in the $\ell - 3$ operations preceding the final operation. Its probability is $(1 - p_{\text{noise}})^{\ell-3}$, independently of the inputs. On F the final maximum or sum receives the noiseless terminal operands, and its output conditional is $P^\star(\cdot \mid x_1, x_2)$. Therefore, for some conditional distribution Q_C,

$$\mu_C = (1 - p_{\text{noise}})^{\ell-3} P^\star + \left[1 - (1 - p_{\text{noise}})^{\ell-3}\right] Q_C.$$

The pointwise lower bound on μ_C yields

$$A_C = \mathcal{E}(\mu_C) \leqslant (\ell - 3)\log \frac{1}{1 - p_{\text{noise}}}. \tag{59}$$

For known source conditional distributions, Equation (37) consequently gives

$$\mathcal{E}(\hat{\mu}) \leqslant (\ell - 3)\log \frac{1}{1 - p_{\text{noise}}} + 2\gamma_\ell.$$

With two parents per operation, holding the number of available source conditional distributions and the confidence level fixed, the selection contributions for subtraction and remainder are, respectively,

$$\mathcal{O}\left(\log(1/\epsilon)\sqrt{\frac{|\mathcal{V}| \log(2|\mathcal{V}|)}{n}}\right) \quad \text{and} \quad \mathcal{O}\left(\log(1/\epsilon)\sqrt{\frac{\log(2|\mathcal{V}|)\log\log(4|\mathcal{V}|)}{n}}\right).$$

Taking $\epsilon = p_{\text{noise}}/|\mathcal{V}|$ for the known conditional distributions and using $-\log(1-p_{\text{noise}}) \leqslant 2p_{\text{noise}}$ for $p_{\text{noise}} \leqslant 1/2$ proves Equation (58). The logarithms hidden by $\widetilde{\mathcal{O}}$ include $\log(|\mathcal{V}|/p_{\text{noise}})$.

For learned source gates, let $b_C = (\ell - 2)a$ and take ϵ no larger than their output floor and $p_{\text{noise}}/|\mathcal{V}|$. Proposition B.8 adds $b_C + \sqrt{A_C b_C}/\epsilon$ to the approximation and selection terms. These estimation terms vanish as the uniform gate-error bound $a \to 0$, at fixed vocabulary and noise. All terms concern the same target distribution specified above.

Expressibility and minimum size. Minimum circuit size asks how few operations compute a specified function over a basis; clone membership asks whether the basis can express that function (Allender et al., 2018; Vollmer, 2009). For probabilistic circuits the analogous objects are conditional distributions, and approximation is measured here by excess risk. A finite basis yields a countable set of finite compositions, while the oracle inequality also allows comparison with compositions that approximate a query. The statistical bound controls the additional risk from selecting among these representations. The computational cost of carrying out that search is a further consideration.

F Concepts, composition length, and noise

A target concept is a conditional table $P^\star(y \mid \mathbf{x})$. A circuit represents that table after its intermediate variables are marginalized. Several circuits can represent the same concept; when their required conditionals are identified from the source, each supplies a valid Circuit-TR representation. They need not generate the same intermediate traces. Process supervision supplies complete execution traces; outcome supervision supplies only their input/output pairs. With noisy mechanisms, equality of the noiseless input/output functions alone does not imply equality of the conditional tables.

F.1 Restricting gates can change the required computation

Removing a gate from the source library can increase the required composition length while preserving the noiseless input/output function. For example, on $\{0, \ldots, q-1\}$, modular addition implements subtraction: starting at a and adding b exactly $q-1$ times yields $(a-b) \bmod q$. Hence a library containing maximum, minimum, and modular addition still implements the noiseless subtraction-based GCD construction in Appendix E.4. The additional operations matter: each introduces its own noise and, with fitted gates, its own estimation error. A restriction can consequently preserve noiseless expressibility while making every available construction incur greater excess risk for the target concept.

This distinction is relevant to the strictly positive distributions assumed in our risk bounds. The output conditionals of different noisy GCD circuits can approximate one common concept without being identical. Their discrepancy depends on the noise level and the chosen compositions. The common-target comparison in Appendix E.4 quantifies this approximation. The following finite example isolates the effect exactly.

F.2 An exact example of length-dependent signal loss

For this example, take the source conditional distributions as known and use them as exact gates. Consider chains of unary gates on an alphabet of size $q \geqslant 2$. The noiseless map g of each gate is a permutation, and its conditional distribution is

$$K_g(y \mid x) = (1-p)\mathbb{1}\{y = g(x)\} + \frac{p}{q}, \qquad 0 < p < 1. \tag{60}$$

Replacement events and replacement values are independent across gate applications. All inputs have positive probability. Here length m counts gates along the chain, so the circuit size including its input is $m+1$.

Proposition F.1 (Exact concepts in homogeneous noisy chains)**.** *Let C be a chain of $m \geqslant 1$ gates with composed permutation f_C, and put $r = 1-p$. Its concept is*

$$\mu_C(y \mid x) = r^m \mathbb{1}\{y = f_C(x)\} + \frac{1-r^m}{q}. \tag{61}$$

If the target is the concept of a chain C_* of length m_*, then $\mu_C = P^\star$ if and only if $f_C = f_{C_*}$ and $m = m_*$. Among chains with $f_C = f_{C_*}$ and $m \geqslant m_*$, excess risk is strictly increasing in m.

Proof. Conditional on no replacement, the output is $f_C(x)$. After a replacement, the current value is uniform; every subsequent permutation and uniform replacement preserves that distribution. This proves Equation (61). Its unique most probable output is $f_C(x)$, with probability

$$a_m = \frac{1 + (q-1)r^m}{q}.$$

Equality of the conditional tables forces their most probable outputs and their strictly decreasing probabilities a_m to agree. For equal composed permutations, the excess risk is independent of the input distribution and equals

$$\mathcal{E}(\mu_C) = a_{m_*} \log \frac{a_{m_*}}{a_m} + (1 - a_{m_*}) \log \frac{1 - a_{m_*}}{1 - a_m}. \tag{62}$$

The derivative of this expression with respect to a_m is $(a_m - a_{m_*})/[a_m(1 - a_m)]$. It is negative when $m > m_*$, while a_m decreases with m. □

The coefficient r^m measures the retained dependence on the noiseless computation. For small p, it is approximately $\exp(-mp)$: retaining substantial dependence on the noiseless computation requires controlling the product of length and noise. In this model, if C_* is the unique shortest representation of its noiseless map over the full gate library, and restricting the source library removes that representation, no chain of the remaining gates exactly represents its positive-noise concept. Longer chains may still approximate it. The conclusion uses homogeneous replacement noise and permutation chains; circuit size in a general graph also counts gates outside a single path, and additional computation can reduce noise through redundancy. Signal loss and the resources required for reliable noisy computation are studied by Evans & Schulman (1999); the proposition here compares the exact conditional tables of different representations.

Example F.2 (A missing gate with a longer substitute)**.** On $\{0, 1, 2\}$, let $g_1(x) = x + 1 \bmod 3$ and $g_2(x) = x + 2 \bmod 3$. The target circuit applies g_2 once with noise p, whereas the source supplies only g_1 with the same noise. Two applications of g_1 compute g_2 without noise. More generally, lengths $m = 2, 5, 8, \ldots$ have the same noiseless map. At positive noise their excess risks against the fixed one-gate target are given by Equation (62) with $m_* = 1$.

There is a positive excess-risk lower bound even if a predictor averages the output conditionals of arbitrarily many source chains. Every source chain assigns probability at most a_2 to $g_2(x)$: a chain with the correct noiseless map has length at least two; a chain with a different map assigns this value probability below $1/3$. An average satisfies the same bound. Grouping the output events into $\{g_2(x)\}$ and its complement therefore gives

$$\inf_{\mu} \mathcal{E}(\mu) = a_1 \log \frac{a_1}{a_2} + (1 - a_1) \log \frac{1 - a_1}{1 - a_2} > 0, \tag{63}$$

where the infimum ranges over the conditional predictors induced by source chains and their averages. The two-gate chain attains the bound. This gap tends to zero as $p \downarrow 0$. Thus a restriction can exclude exact representation of a positive-noise concept by source-gate compositions while retaining arbitrarily accurate approximation as the gates approach determinism.

F.3 Gate estimation and the preference for shorter compositions

Source estimation adds to the effect of intrinsic gate noise. If each fitted gate has conditional KL error at most a uniformly over its arguments, the chain rule and marginalization give

$$\mathbb{E}_{\mathbf{x}} D_{\mathrm{KL}}\big(\mu_C(\cdot \mid \mathbf{x}) \,\|\, \hat{\mu}_C(\cdot \mid \mathbf{x})\big) \leqslant (\ell - M)a. \tag{64}$$

This bound applies to general circuits of size ℓ, not only chains. It compares a specified composition of true source conditionals with the same composition of fitted gates. For a composition that only approximates the target concept, Proposition B.8 combines its approximation risk with this estimation contribution and the cost of target selection.

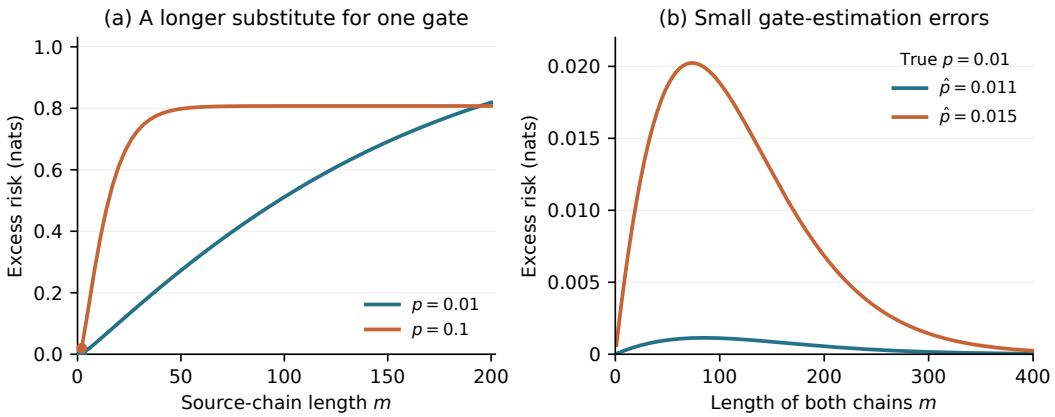


Figure 7: Exact calculations on three symbols; no training or Monte Carlo estimation is used. **Left:** source chains g_1^m, $m = 2, 5, 8, \ldots$, compared with the fixed one-gate g_2 target in Example F.2. Each curve has the indicated common gate noise p in both source and target. Longer chains preserve the noiseless map while changing its conditional table. **Right:** gate-estimation error alone. For each length m, the target is the true m-gate chain with $p = 0.01$, and the predictor uses the same chain with a perturbed replacement probability $\hat{p}$. Here the reference concept changes with m to isolate estimation error. Small local errors can produce much larger excess risk after composition; eventual convergence of both chains to the uniform conditional makes this discrepancy decrease again.

The right panel of Figure 7 evaluates the chain comparison exactly. With true replacement probability 0.01 and fitted probability 0.011, the single-gate excess risk is approximately 3.15×10^{-5} nats; at length 85 the excess risk is approximately 1.14×10^{-3} nats, about 36 times as large. This example shows how accurate source gates can yield appreciable errors when repeatedly composed. At larger lengths, convergence of both conditional distributions to the uniform distribution reduces their discrepancy again.

When candidate circuits implement the same noiseless map through gates of comparable reliability, shorter compositions can preserve more signal and incur less source estimation error. Proposition F.1 gives a setting where this preference is exact. Across the full candidate family, the empirical risk term in Equation (6) evaluates each composition's output conditional on the observed input/output pairs, while the size penalty controls the statistical cost of selecting it. A shortest representation of the wrong map can have high risk. The objective therefore balances empirical risk and circuit size. The expected size penalty in Neural-AD serves the same comparison for its distribution over compositions.

With process supervision, observed intermediate values allow local fitting and selection. The final conditional predictor is obtained by composing the fitted gates and marginalizing the intermediate variables. Under outcome supervision, candidate circuits are evaluated through log loss on the observed input/output pairs. Comparing the excess risks of the true supplied composition, its source-fitted version, and the selected predictor helps assess approximation, the effect of source estimation, and the remaining adaptation gap on longer noisy computations.

F.4 Identification and computation length

Classical transportability asks whether a target quantity is determined by available distributions and structural assumptions. Its formulas can combine several identified quantities through products, marginalization, and recursive reductions (Pearl & Bareinboim, 2011; Bareinboim & Pearl, 2014). Circuit representations make an additional resource explicit: the number of applications of reusable mechanisms needed to realize or approximate a target concept, and the error incurred by executing their fitted versions. A conditional at a target position with no directly certified source match may still be represented or approximated by a composition of source conditionals. Exact identification through a composition requires a structurally justified equality with the target query. Computation alone cannot distinguish two models that agree on all available information but disagree on that query.

Description length provides a complementary reason to prefer economical representations. The circuit weights in Appendix E.3 allocate statistical confidence across every finite size. That section develops the connection with Levin's universal search and Kt complexity. Here longer descriptions enlarge the selection cost, while noisy execution can add an approximation cost.

G Experimental settings and additional results

We first specify the shared learning and evaluation protocol, then give the source-library and supervision comparisons from Section 6. Further studies examine exact circuit selection, the contributions of source estimation and gate noise to composition risk, and reuse on new input combinations and real measurement windows. A final study composes gates extracted from pretrained language models.

G.1 Observation and evaluation protocol

In the synthetic studies, source data contain complete execution traces. Under outcome supervision, target fitting receives only the input prefix and final output. Under process supervision, it additionally receives the intermediate values. Within each paired study, both modes learn the same target concept, the conditional table of the final output given the inputs. Target fitting and checkpoint selection use the n supplied samples. Generating more executions on an observed target input does not provide new target labels.

We report excess risk $\mathcal{E}$ from Equation (2) in nats, averaging the conditional KL divergence under the target input distribution. For noiseless computations, the target conditional entropy is zero, so risk equals excess risk. Where reported as a complementary diagnostic, answer error is one minus the accuracy of the most probable output against the noiseless computation.

Uncertainty is computed across independently sampled datasets, conditional on the source fit when it is shared. Intermediate positions and rollouts are not independent data replicates. The symbolic selection study in Appendix G.4.2 uses the stratified uncertainty estimate specified there; the language-model study evaluates fixed extracted gates without independently trained model replicates.

Shared source gates and target fitting. The GCD and electricity studies use the source circuits described in Appendix G.2. Each source is fitted once per vocabulary, using $N = 10^6$ traces for $|\mathcal{V}| = 10$ and $N = 3 \times 10^6$ for $|\mathcal{V}| = 100$, with 10% held out. Source inputs are uniform; later values follow the source circuit, with independent uniform-replacement probability 0.01 at each non-input position. The gate predictor has two hidden layers of width 128 or 256 and value embeddings of dimension 32 or 64, respectively. Its probabilities have floor $\epsilon = 0.01/|\mathcal{V}|$. The structural generator has one causal transformer layer, width 16, and four attention heads. Source fitting minimizes expected next-token log loss with coefficient 0.01 on occupied gate indicators and parent edges. Batches contain 512 traces. The learning rate of the gate predictor is 0.003 for ten values and 0.001 for 100 values; the structural-generator learning rate is 0.005. The respective limits are 30,000 and 80,000 updates, with held-out-objective stopping. The selected source assignment is refitted, and the gate predictor is frozen during target adaptation. The repeated source operations may retain different fitted indicators; all retained indicators are available to Neural-AD.

Outcome-supervised Neural-AD uses rejection-sampling fine-tuning as described in Appendix C. Each round proposes 4,096 circuits, scores them by empirical log loss plus the size penalty, and fits the generator to the best 16. The archive's 128 leading candidates are re-evaluated on fresh executions. Circuit scores use four executions per observed pair, averaging the final gate's output probability before taking its negative logarithm, and at most 500 of the supplied target pairs per round. Generator fitting uses 20 Adam updates with learning rate 0.05 and rank weights proportional to 2^{-r} for $r = 0, \ldots, 15$. Checkpoints use 64 sampled circuits and executions per pair to estimate the log loss of the mixture in Equation (14), plus its expected size penalty. Fitting ends after 80 rounds or 30 rounds without improvement. These Monte Carlo estimates introduce integration error; additional executions do not increase the target sample size.

The proposal distribution combines samples from the generator with local changes to retained circuits. Once the archive is populated, 25% of proposals modify its best circuits under the penalized log loss, and 20% modify circuits ranked by a proposal guide based on observed-answer probability and circuit size. In the two-input GCD studies, a further 25% modify the pretrained source composition.

The electricity computations have four inputs and omit this last proposal source. Forty percent of generator proposals receive one or two local changes. Changes can replace a gate or ordered-parent choice, add a position, or repeat a block; positions that do not affect the output are removed before scoring. The final score and checkpoint criterion remain penalized log loss.

The GCD and electricity studies use the penalty γ_ℓ of Equation (35) with regularization constant $B = 1$ and $\delta = 0.05$. The uniform-deviation guarantee for the floored log loss uses $B = \log(1/\epsilon)$; the exhaustive comparison in Appendix G.4.1 uses that constant.

Under process supervision, Neural-AD evaluates every legal gate and ordered-parent choice at each observed position, selects the empirical trace-log-loss minimizer, and fits its generator to that circuit. The output position is fixed by the trace length. ERM-target and ERM-src+tgt use a plain one-layer transformer of width 64 with four soft-attention heads and an MLP of the same width as the gate predictor. ERM-target fits all parameters; ERM-src+tgt pretrains on the same source traces, then freezes its MLP and output mapping while adapting attention, embeddings, and normalization. Both maximize the marginal likelihood of the observed answer over sampled intermediate tokens under outcome supervision, and minimize next-token log loss on the complete trace under process supervision. They use no size penalty. Their outcome fits use eight executions per pair, batches of at most 32 samples, and at most 2,000 Adam updates with learning rate 0.001.

Neural-AD denotes its averaged conditional from Equation (14). The selected-circuit diagnostic separately evaluates its greedily decoded circuit, without averaging over the generator. Circuit-TR uses the supplied composition and source-to-target gate assignments, with no target fitting. Its excess risk reflects gate estimation error and any difference between the supplied composition's conditional and the target concept. Exact Circuit-AD is reported only where its candidate family can be enumerated.

G.2 Learning the GCD concept from different source libraries

Source interventions. The full source contains maximum, minimum, subtraction, remainder, absolute difference, sum, and product, with reuse across its 14 positions (Figure 1). The no-remainder source has 12 positions. The restricted source retains maximum, minimum, sum, and product across ten positions. The restricted source circuits change the parent lists of later positions as specified in Table 1. The ten-value source fits retain 13, ten, and nine indicators, respectively, including an unconditional gate. The 100-value fits retain 11, ten, and eight indicators, respectively; electricity uses the full library. The operation names describe the source computation; the retained predictors are fitted conditional distributions.

	No-remainder source		Restricted source	
Position	Operation	Parents	Operation	Parents
3	Maximum	$(1, 2)$	Maximum	$(1, 2)$
4	Minimum	$(1, 2)$	Minimum	$(1, 2)$
5	Subtraction	$(3, 4)$	Addition	$(1, 2)$
6	Absolute difference	$(1, 2)$	Multiplication	$(1, 2)$
7	Addition	$(1, 2)$	Minimum	$(5, 6)$
8	Multiplication	$(1, 2)$	Maximum	$(3, 7)$
9	Minimum	$(5, 6)$	Addition	$(4, 8)$
10	Subtraction	$(7, 8)$	Multiplication	$(7, 9)$
11	Maximum	$(9, 10)$	—	—
12	Addition	$(6, 9)$	—	—

Table 1: Source interventions. Positions 1 and 2 are independent uniform inputs. Parent lists give ordered position indices. Each operation is independently replaced by a uniform value with probability 0.01; addition, subtraction, and multiplication are modulo $|\mathcal{V}|$.

One concept, several possible compositions. Inputs are independent and uniform on $\{0, \ldots, |\mathcal{V}| - 1\}$. At $|\mathcal{V}| = 10$, the target concept is

$$P^\star(y \mid a, b) = 0.99\, \mathbb{1}\{y = \gcd(a, b)\} + 0.01/10.$$

Only input/output pairs are supplied for adaptation in this study. The search cap is 27 positions, including the two inputs.

With the full library, Circuit-TR is supplied a fixed remainder chain followed by the sum of its last two values: five remainder operations, giving eight positions. The convention $a \bmod 0 = a$ makes a completed noiseless chain alternate between GCD and zero; the final sum gives GCD regardless of this alternation. The chain length is fixed in advance and suffices for every input pair. With no remainder, the supplied subtraction construction has 27 positions and also computes GCD on every noiseless input. Each supplied composition provides one reference for the risk comparison. No reference circuit is supplied for the max/min/sum/product library.

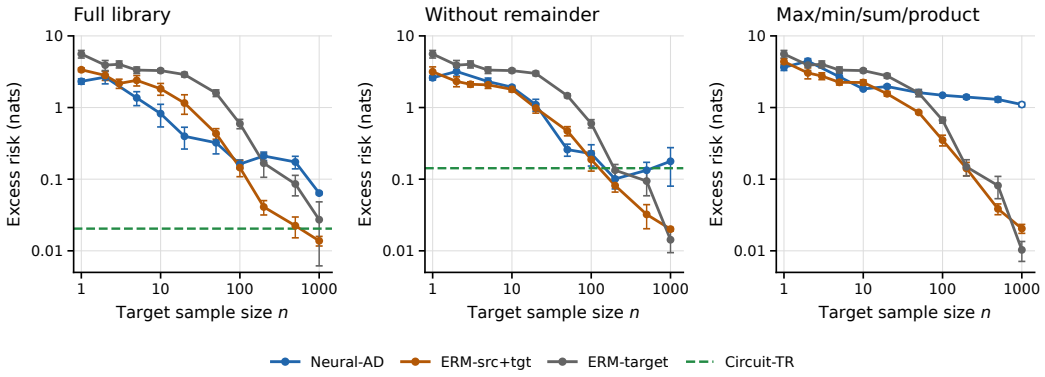


Figure 8: Outcome-supervised GCD concept learning over ten values under the three source libraries. Neural-AD evaluates the mixture conditional. Both ERM baselines maximize marginal answer likelihood. Error bars are one standard error over target datasets, conditional on a single source fit; open markers denote the four-dataset max/min/sum/product condition at $n = 1000$; other points have five datasets. Circuit-TR evaluates the supplied remainder or subtraction composition without target observations.

Method	$n = 20$	$n = 100$	$n = 1000$
Full library			
Neural-AD	0.397 ± 0.134 (5)	0.163 ± 0.023 (5)	0.064 ± 0.003 (5)
ERM-src+tgt	1.156 ± 0.351 (5)	0.145 ± 0.037 (5)	0.014 ± 0.002 (5)
ERM-target	2.888 ± 0.211 (5)	0.595 ± 0.089 (5)	0.027 ± 0.021 (5)
Without remainder			
Neural-AD	1.096 ± 0.206 (5)	0.227 ± 0.075 (5)	0.178 ± 0.098 (5)
ERM-src+tgt	0.984 ± 0.148 (5)	0.188 ± 0.059 (5)	0.020 ± 0.001 (5)
ERM-target	2.993 ± 0.210 (5)	0.598 ± 0.079 (5)	0.014 ± 0.005 (5)
Max/min/sum/product			
Neural-AD	1.964 ± 0.105 (5)	1.484 ± 0.030 (5)	1.101 ± 0.074 (4)
ERM-src+tgt	1.557 ± 0.114 (5)	0.351 ± 0.060 (5)	0.020 ± 0.003 (5)
ERM-target	2.768 ± 0.201 (5)	0.667 ± 0.062 (5)	0.010 ± 0.003 (5)

Table 2: GCD concept excess risk in nats. Means and standard errors are over target-data seeds; the seed count accompanies each estimate. Neural-AD is the mixture conditional.

Removing remainder leaves useful alternative computations. In the ten-value study, Neural-AD reaches excess risk 0.259 at $n = 50$ and 0.101 at $n = 200$, compared with 0.142 for the supplied

subtraction composition. Decoded circuits use absolute difference and can be substantially shorter than that reference: at $n = 50$ their total sizes are ten to 15 positions. Thus final answers can support a different composition from the one used to construct Circuit-TR. With the full library, the advantage over the ERM baselines is largest at small target sample sizes; pretrained ERM reaches a similar risk around $n = 100$ and lower risk at larger budgets.

The max/min/sum/product library retains higher risk over the tested budgets. This intervention restricts efficient computations available to the search, but does not establish that no longer composition can compute GCD. In the noiseless modular arithmetic, subtraction can be synthesized by repeated addition, since $a - b = a + (|\mathcal{V}| - 1)b \pmod{|\mathcal{V}|}$. The distinction matters with positive gate noise: a longer noiselessly valid construction may induce a less accurate conditional table. Appendices F and G.5.2 examine this interaction between representation, length, and noise.

The same intervention over 100 values further separates the libraries. Here the target concept is noiseless GCD, inputs are uniform over all 10,000 pairs, and Neural-AD searches up to 30 positions. At $n = 50$, its excess risk is 0.136 ± 0.031 with the full library, 1.591 ± 0.035 without remainder, and 3.003 ± 0.026 with only max/min/sum/product. Each estimate averages three target datasets, with one standard error conditional on the corresponding source fit.

What the generator has learned. A decoded full-library circuit at $n = 100$ first computes $u = \max(a, b)$ and $v = \min(a, b)$, then $r_1 = u \bmod v$, $r_2 = v \bmod r_1$, $r_3 = r_1 \bmod r_2$, and outputs $|r_2 - r_3|$. Its ideal operations give GCD on all 100 input pairs. This check concerns the noiseless computation; replacing its operations by the fitted noisy gates determines its risk against the target concept. Across the five seeds, four decoded circuits at $n = 100$ and all five at $n = 500$ compute GCD exactly in this ideal interpretation. Other generator mass still contributes to the reported Neural-AD conditional. For example, at $n = 100$ its mean excess risk is 0.163, whereas the decoded circuits have mean excess risk 0.052.

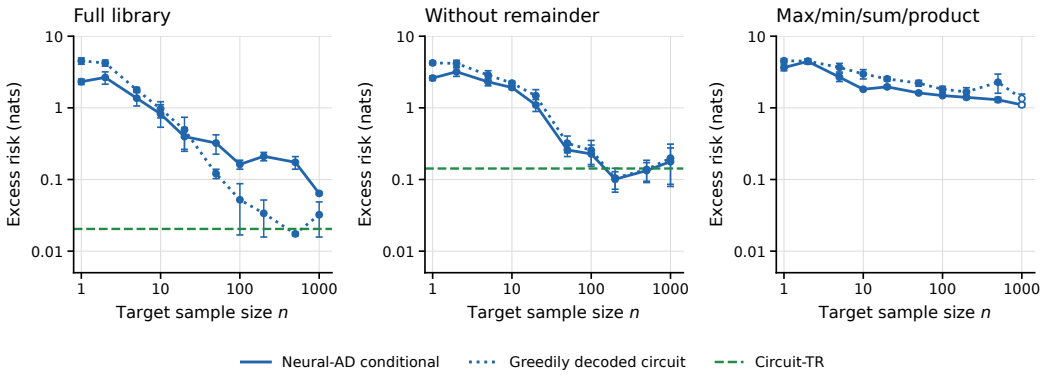


Figure 9: Neural-AD's mixture risk and the risk of its greedily decoded circuit on the same ten-value GCD concept. These are distinct output conditionals. Lower decoded-circuit risk indicates that generator mass on other circuits affects the mixture; decoding need not improve risk at every sample size.

The RFT baseline comparison in Figure 10 examines the contribution of the optimizer without Neural-AD's gate-selection architecture. It uses the same observed answers and reports the same conditional risk.

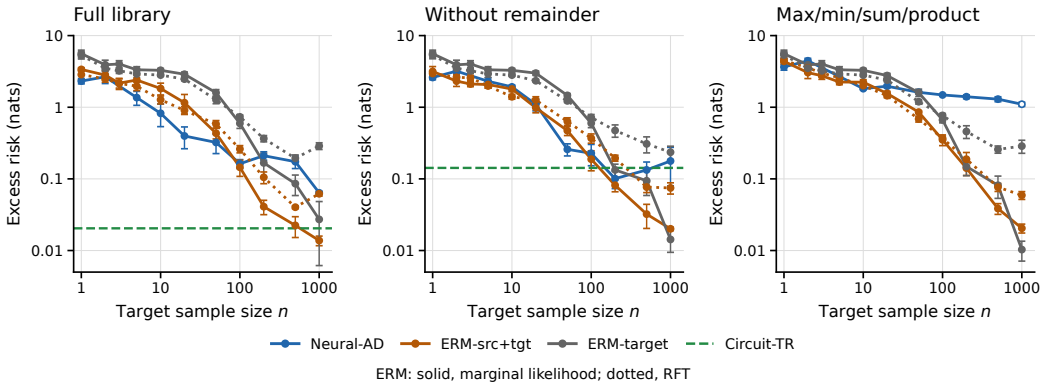


Figure 10: ERM optimizer comparison on the ten-value GCD concept. The default ERM baselines maximize marginal answer likelihood; the RFT variants fine-tune the same plain-transformer architectures by rejection sampling. Each variant is aggregated separately over target-data seeds, conditional on the same source fit for ERM-src+tgt.

G.3 What process supervision reveals

This paired study exposes the same target samples with and without intermediate values. This holds the target concept and sample budget fixed while changing the evidence available for selecting the computation. It uses the full source circuit. Source replacement noise is 0.01; the target noise is specified separately below, so the supplied fitted composition need not reproduce the target conditional exactly.

A complete GCD circuit. The target is the maximum/minimum/remainder construction in Equation (19), followed by a final maximum, with $K = 4$ for ten values. This fixed circuit has 15 positions and computes GCD on every noiseless input. Each target gate instead uses independent replacement probability 0.001. A complete noisy execution supplies the process condition; its inputs and final output supply the outcome condition. Both are evaluated against the conditional table of this same noisy circuit. Circuit-TR receives that circuit and the corresponding source gate assignments. Unlike the answer-replacement concept in Appendix G.2, this target includes noise at intermediate positions.

At $|\mathcal{V}| = 10$ and $n = 100$, process-supervised Neural-AD has mean excess risk 0.049, compared with 0.210 using final answers alone and 0.059 for Circuit-TR. By $n = 1000$, the process and outcome means are 0.056 and 0.058, respectively, across all five seeds. Intermediate values allow each local choice to be evaluated directly. With only final answers, those choices must be assessed through the composed conditional. The process result therefore supplies evidence that useful gates are available, while the remaining outcome gap also depends on finite search and regularization.

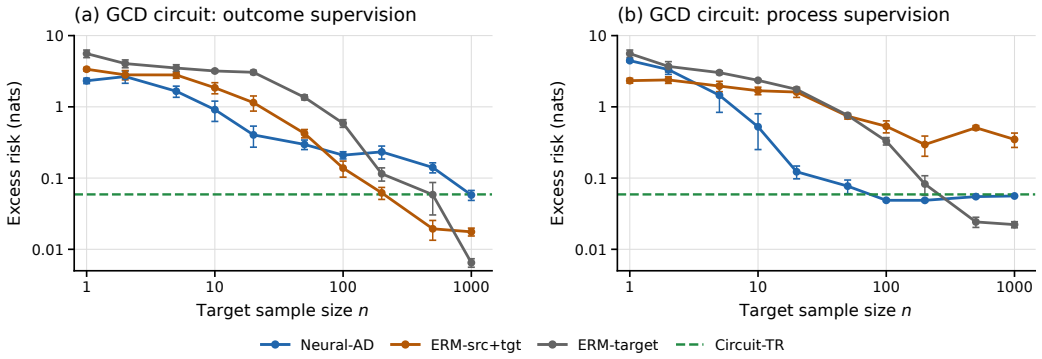


Figure 11: A complete GCD circuit over ten values under outcome and process supervision. Both panels use the same target executions. Points average five target datasets; error bars give one standard error, conditional on a shared source fit.

G.4 Exact circuit selection

G.4.1 Neural search compared with exact circuit selection

Source and targets. This study makes the symbolic selection problem small enough to solve exactly, so that the circuit found by neural search can be compared with its empirical optimum.

All positions take values in $\{0, \ldots, 9\}$. This study uses a five-position source, distinct from the richer source in Figure 1, with independent uniform inputs V_1, V_2 and positions $V_3 = \max(V_1, V_2)$, $V_4 = \min(V_1, V_2)$, and $V_5 = V_1 - V_2 \pmod{10}$. At every non-input position, independently, a uniform draw replaces the operation's output with probability 0.1. One source dataset of $20{,}000$ sequences supplies the fitted gates shared by all target fits. The target computations are

Noiseless computation	$T^\star$	Maximum candidate size
$\max(a,b) - \min(a,b)$	5	6
$\min(\max(a,b) - \min(a,b), \min(a,b))$	6	6
Equation (19), $K = 1$, followed by maximum	6	6
Equation (19), $K = 2$, followed by maximum	9	9

Target operations have the same replacement probability as the source. The target concept is the conditional distribution of the final output given the two inputs, including noise in the intermediate positions. The two truncated subtraction computations are evaluated against their own conditionals; they do not compute mathematical GCD on every input. Subtraction receives the maximum and minimum in the target, whereas its source parents are the original inputs. Process fitting receives complete target traces; outcome fitting sees only their input/output pairs. Both are evaluated against the same conditional table.

Architecture and source fitting. The structural generator has one causal transformer layer, width 16, and four ordinary attention heads. It parameterizes two ordered hard parent selections and eight initial gate indicators, with arities $(0, 0, 1, 1, 2, 2, 2, 2)$. The shared gate predictor uses 32-dimensional value embeddings and an MLP with two hidden layers of width 128. Its output probabilities have floor $\epsilon = 0.01/|\mathcal{V}|$; it receives no position embedding. The source fit retains an unconditional gate and gates for maximum, minimum, and modular subtraction.

Source fitting minimizes the exact expectation of next-token cross-entropy over legal gate and parent selections, with coefficient 0.01 on the expected number of occupied indicators and parent edges. Ten percent of source traces are held out. The generator is initialized using position-specific indicators and parent pairs of low empirical conditional entropy, calculated on the source training traces. The first 30% of at most 3,000 updates fit the gate predictor with this generator fixed; subsequent updates

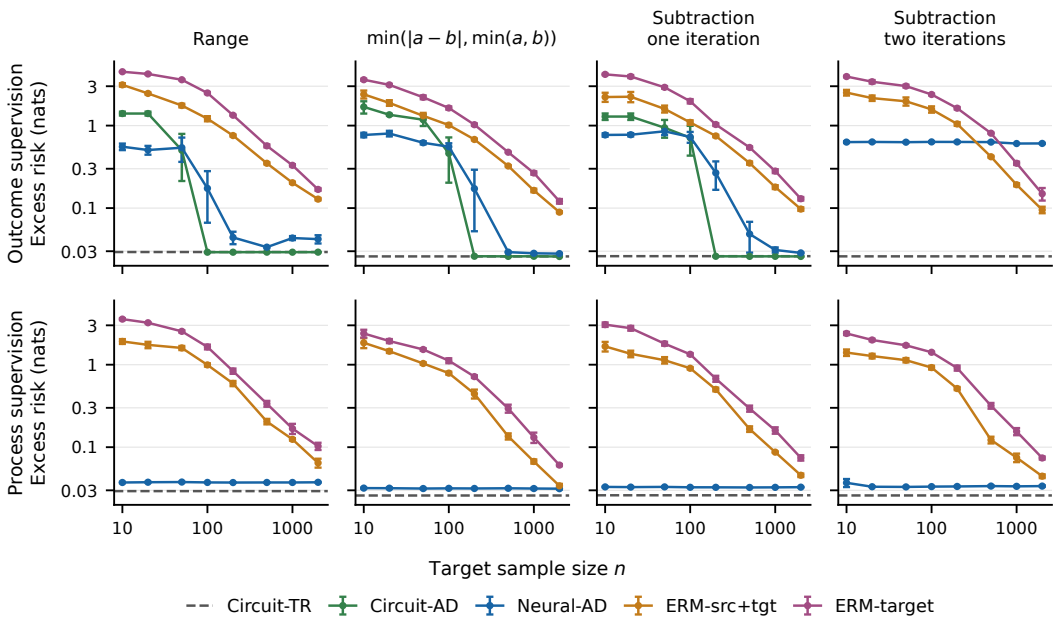


Figure 12: Short-circuit comparison with exact Circuit-AD. Excess risk is plotted against target sample size n on logarithmic axes, with paired outcome and process conditions. Points average five target datasets conditional on one source fit; bars show one standard error. For the two-iteration subtraction target, outcome ERM has four seeds through $n = 1000$ and two at $n = 2000$; Circuit-AD is not enumerated. Neural-AD averages 400 sampled circuits and executions per input. All 100 input pairs are evaluated against each task's fixed noisy conditional.

train both, with the penalty increasing linearly over the next 30%. Adam learning rates are 0.003 for the gate predictor and 0.02 for the generator, with batches of 256. The held-out objective is checked every 50 updates with patience 12. Held-out penalized loss selects a hard source assignment from the argmax and 64 sampled assignments. The gate predictor is refitted for 1,000 updates under that assignment, then frozen. Its recorded parameter hash is unchanged by every reported target fit.

Outcome fitting. Neural-AD initializes the structural generator from its source fit and uses the RFT procedure described in Appendix C. Each round proposes 4,096 circuits. Once the archive is populated, half come from the current generator, one quarter from local changes to its 16 best retained circuits, and one quarter from local changes to the pretrained composition. Forty percent of generator samples also receive one or two local changes. A change redraws a used position's gate and ordered parents, or grows the circuit by one position. This study uses neither the answer-probability proposal guide nor block-repetition proposals.

Each proposed circuit is scored by its empirical log loss plus its size penalty of Equation (35), with $B = \log(1/\epsilon)$, $\delta = 0.05$, $d = 4$, and $c = 2$. The probability of an observed answer is estimated from four executions, averaging the final gate's output probability; its negative logarithm supplies the loss. Scoring uses all pairs up to $n = 500$ and a random subset of 500 at larger budgets. The finite-execution log-loss estimate can be biased. An archive stores running mean scores and re-evaluates its 128 best candidates on fresh executions each round. The generator fits the top 16 by maximum likelihood, with normalized weights proportional to 2^{-r} for rank $r = 0, \ldots, 15$, using 20 Adam updates at learning rate 0.05. Checkpoints minimize the estimated log loss of the averaged predictor plus its expected size penalty, using 64 sampled circuits and executions per pair. Fitting stops after at most 80 rounds or 30 rounds without improvement. All scoring and checkpoint choices use only the supplied target pairs.

Process fitting. With intermediate values observed, each position's empirical log loss is computed for every legal gate and ordered-parent choice. The per-position minimizers give an exact hard minimizer of the empirical trace loss over these choices. The target generator is initialized afresh and

fitted by maximum likelihood to that circuit for at most 300 updates, stopping early if it assigns more than 99.9% mass to the selected structure and retaining the highest-mass checkpoint. The observed trace length fixes the output position. Thus the paired experiment compares local fitting from traces with finite RFT search from final answers, in addition to the difference in available supervision.

Comparison methods. Circuit-TR uses the supplied target composition and source gate assignments, without target data, and evaluates its conditional table exactly. Circuit-AD enumerates every circuit over the same four fitted gates with sizes $M + 1, \ldots, 6$, evaluating all 305,242 conditional tables exactly. It minimizes empirical log loss plus the same size penalty; ties favor smaller circuits. We do not enumerate the nine-position candidate family.

The ERM baselines use a plain one-layer transformer of width 64 with four soft-attention heads, residual connections, layer normalization, a width-128 MLP, and the same probability floor. ERM-target trains all parameters from target observations. ERM-src+tgt first trains on the same source traces, then freezes its MLP and output mapping while adapting attention, embeddings, and normalization. Neither baseline has gate indicators or a size penalty. Outcome fitting maximizes the marginal log likelihood of the observed answer over sampled intermediate tokens, using eight executions per pair and self-normalized importance weights. Process fitting minimizes next-token cross-entropy on the observed traces. Outcome fitting uses batches of at most 32 samples and process fitting at most 64; both use Adam learning rate 0.001 and at most 2,000 updates, with empirical-loss stopping. Under process supervision, the baselines receive every intermediate value supplied to Neural-AD.

Method	Source information	Target circuit	Target samples
Circuit-TR	Fitted gates	Supplied	None
Circuit-AD	Fitted gates	Unknown	Input/output
Neural-AD	Fitted gates and pretrained generator	Unknown	Input/output
ERM-target	None	Unknown	Input/output
ERM-src+tgt	Source training sequences	Unknown	Input/output

Table 3: Information under outcome supervision. Circuit-TR also receives the source-to-target gate assignments. Circuit-AD and Neural-AD use the same fitted gates and eligible sizes. In the paired process condition, Neural-AD and both ERM baselines instead receive complete traces.

Evaluation and replication. The evaluator computes the target conditional on all 100 uniformly weighted input pairs. Excess risk is the average KL divergence from this conditional to the fitted conditional. Neural-AD's output conditional averages 400 sampled circuits and their executions per input, integrating the final gate's output probabilities. Circuit-AD and Circuit-TR are evaluated exactly. ERM output conditionals marginalize sampled intermediate tokens. A separate stored error metric compares the argmax answer with the noiseless computation.

Budgets are $n \in \{10, 20, 50, 100, 200, 500, 1000, 2000\}$; samples are nested within each target-data seed. Except for the stated two-iteration ERM comparisons, each plotted point averages five independent target datasets, with standard error equal to the sample standard deviation divided by the square root of the number of seeds. The exceptions are the two-iteration subtraction target's outcome ERM rows, with four seeds through $n = 1000$ and two at $n = 2000$. All uncertainty is conditional on one shared source fit. Circuit-TR's repeated reference records are identical and count as one fitted comparator.

What the paired comparison establishes. With ten traces, Neural-AD already reaches excess risk within 0.011 nats of Circuit-TR on all four tasks; its argmax answer agrees with each task's noiseless computation in every seed at every reported budget. The ERM baselines benefit from process supervision but need larger samples to approach this risk. Under outcome supervision, the three smaller tasks reach a comparable level with hundreds of target samples. At $n = 2000$, Neural-AD's excess risks on these tasks are 0.028–0.042, compared with 0.089–0.128 for ERM-src+tgt and 0.121–0.168 for ERM-target. These comparisons concern the target conditional, including its noise, rather than only its most likely answer.

Method	$n = 10$	$n = 100$	$n = 500$	$n = 2000$
Range				
Circuit-TR		0.029 (no target data)		
Circuit-AD	1.400 ± 0.076	0.029 ± 0.000	0.029 ± 0.000	0.029 ± 0.000
Neural-AD	0.555 ± 0.050	0.173 ± 0.107	0.034 ± 0.001	0.042 ± 0.005
ERM-src+tgt	3.118 ± 0.149	1.213 ± 0.086	0.349 ± 0.011	0.128 ± 0.004
ERM-target	4.496 ± 0.047	2.485 ± 0.068	0.569 ± 0.024	0.168 ± 0.006
$\min(\lvert a - b \rvert, \min(a, b))$				
Circuit-TR		0.026 (no target data)		
Circuit-AD	1.684 ± 0.286	0.461 ± 0.258	0.026 ± 0.000	0.026 ± 0.000
Neural-AD	0.769 ± 0.049	0.555 ± 0.056	0.029 ± 0.000	0.028 ± 0.000
ERM-src+tgt	2.393 ± 0.265	1.014 ± 0.041	0.324 ± 0.008	0.089 ± 0.002
ERM-target	3.600 ± 0.101	1.635 ± 0.074	0.476 ± 0.006	0.121 ± 0.008
Subtraction, one iteration				
Circuit-TR		0.026 (no target data)		
Circuit-AD	1.287 ± 0.116	0.713 ± 0.280	0.026 ± 0.000	0.026 ± 0.000
Neural-AD	0.770 ± 0.037	0.729 ± 0.109	0.049 ± 0.020	0.029 ± 0.000
ERM-src+tgt	2.219 ± 0.281	1.093 ± 0.066	0.351 ± 0.016	0.097 ± 0.005
ERM-target	4.179 ± 0.086	1.974 ± 0.119	0.543 ± 0.016	0.130 ± 0.006
Subtraction, two iterations				
Circuit-TR		0.026 (no target data)		
Circuit-AD	–	–	–	–
Neural-AD	0.631 ± 0.014	0.634 ± 0.010	0.631 ± 0.010	0.608 ± 0.009
ERM-src+tgt	$2.511 \pm 0.198^{(4)}$	$1.572 \pm 0.138^{(4)}$	$0.419 \pm 0.011^{(4)}$	$0.095 \pm 0.009^{(2)}$
ERM-target	$3.931 \pm 0.078^{(4)}$	$2.383 \pm 0.058^{(4)}$	$0.808 \pm 0.014^{(4)}$	$0.149 \pm 0.026^{(2)}$

Table 4: Outcome-supervised excess risk in nats, mean $\pm$ one standard error over five target datasets. Superscripts give the seed count where fewer are available. The dash denotes a candidate family that is not enumerated.

Method	$n = 10$	$n = 100$	$n = 500$	$n = 2000$
Range				
Circuit-TR		0.029 (no target data)		
Neural-AD	0.037 ± 0.001	0.037 ± 0.000	0.037 ± 0.000	0.038 ± 0.000
ERM-src+tgt	1.914 ± 0.129	0.994 ± 0.045	0.205 ± 0.015	0.064 ± 0.008
ERM-target	3.568 ± 0.097	1.642 ± 0.109	0.337 ± 0.027	0.103 ± 0.010
$\min(\lvert a-b\rvert, \min(a,b))$				
Circuit-TR		0.026 (no target data)		
Neural-AD	0.032 ± 0.000	0.032 ± 0.000	0.032 ± 0.000	0.031 ± 0.001
ERM-src+tgt	1.843 ± 0.257	0.790 ± 0.032	0.135 ± 0.012	0.034 ± 0.002
ERM-target	2.391 ± 0.259	1.120 ± 0.076	0.295 ± 0.033	0.061 ± 0.002
Subtraction, one iteration				
Circuit-TR		0.026 (no target data)		
Neural-AD	0.033 ± 0.000	0.033 ± 0.000	0.033 ± 0.000	0.033 ± 0.000
ERM-src+tgt	1.667 ± 0.222	0.905 ± 0.030	0.166 ± 0.013	0.046 ± 0.002
ERM-target	3.061 ± 0.189	1.336 ± 0.046	0.294 ± 0.025	0.074 ± 0.006
Subtraction, two iterations				
Circuit-TR		0.026 (no target data)		
Neural-AD	0.037 ± 0.004	0.033 ± 0.001	0.034 ± 0.001	0.034 ± 0.001
ERM-src+tgt	1.403 ± 0.126	0.922 ± 0.048	0.122 ± 0.011	0.045 ± 0.002
ERM-target	2.391 ± 0.104	1.414 ± 0.023	0.318 ± 0.025	0.074 ± 0.003

Table 5: Process-supervised excess risk in nats, mean $\pm$ one standard error over five target datasets. Neural-AD and both ERM baselines receive the same complete target traces. Circuit-TR is the supplied-composition reference.

Penalized selection and finite search. For the enumerable families, we evaluate the penalized empirical log loss of Neural-AD's greedily decoded circuit exactly and compare it with the Circuit-AD minimum on the same sample (Table 6). At $n \geqslant 500$, every decoded circuit is within 0.01 nats of this minimum. At very small budgets the methods can decode the same short circuit while their risks differ, because Neural-AD averages output conditionals over its circuit distribution. The decoded-circuit gap is therefore a search diagnostic, not the optimization residual of the averaged predictor in the neural theorem.

Target	Diagnostic	$n = 50$	$n = 100$	$n = 200$	$n = 500$
Range	Maximum gap	0.115	0.152	0.007	0.000
	Within 0.01 nats	3/5	3/5	5/5	5/5
$\min(\lvert a-b\rvert, \min(a,b))$	Maximum gap	0.302	2.374	0.279	0.000
	Within 0.01 nats	3/5	1/5	4/5	5/5
Subtraction, one iteration	Maximum gap	0.062	1.326	1.766	0.004
	Within 0.01 nats	3/5	3/5	2/5	5/5

Table 6: Penalized empirical log loss of Neural-AD's decoded circuit minus the exact Circuit-AD minimum on the same target sample. Maximum gap is in nats; the second row counts seeds within 0.01 nats. The conditionals in Figure 12 average sampled circuits and are a different object from this diagnostic.

For the nine-position target, outcome Neural-AD remains near 0.61–0.64 nats through $n = 2000$, whereas Circuit-TR is at 0.026 and process Neural-AD is near 0.033. Its decoded circuit has one gate in every seed through $n = 500$. At these budgets the decoded circuit's penalized objective is better than the supplied circuit's; at $n = 1000$ and 2000, the supplied circuit has the better objective but the search misses it. The supplied and process-supervised predictors establish that a low-risk composition is available. This example distinguishes a finite-sample preference for short circuits from a failure to find a better penalized composition.

G.4.2 Circuit selection from input/output pairs

Target	$n = 5$	$n = 20$	$n = 100$	Gates at $n = 20$
Swapped composition	0.715 ± 0.169	0.0486 ± 0.0005	0.0486 ± 0.0005	2
Repeated shift	0.743 ± 0.175	0.0484 ± 0.0005	0.0484 ± 0.0005	2
Irrelevant intermediate	0.0655 ± 0.0006	0.0655 ± 0.0006	0.0655 ± 0.0006	1

Table 7: Outcome-supervised Circuit-AD: excess risk $\mathcal{E}$ in nats, mean $\pm$ one standard error. Each entry averages 20 independent datasets for each of three fixed target input distributions. The last column counts non-input positions in the selected circuit; at $n = 20$ the count is the same in all 60 runs. Candidate output conditionals are computed exactly.

Tasks. The source has $T = 3$ positions. Its first variable is uniform on $\{0, \ldots, 99\}$, its second applies $s(v) = v + 1 \pmod{100}$ to the first, and its third applies $r(v) = 99 - v$ to the second. At each of positions two and three, independently, the operation is replaced by a uniform vocabulary draw with probability 0.1. Targets also have three positions and one observed input:

Target	Second position	Third position
Swapped composition	$r(V_1)$	$s(V_2)$
Repeated shift	$s(V_1)$	$s(V_2)$
Irrelevant intermediate	$r(V_1)$	$s(V_1)$
Shift then doubling	$s(V_1)$	$2V_2 \bmod 100$
Doubling then square	$2V_1 \bmod 100$	$V_2^2 \bmod 100$
Increased replacement noise	$r(V_1)$	$s(V_2)$

The increased-noise target uses replacement probability 0.25; the others use 0.1. It applies the same operations as the swapped-composition target, with different conditional distributions. The target input distribution is uniform or proportional to a^v, for $a = 0.6$ or $a = 2/7$.

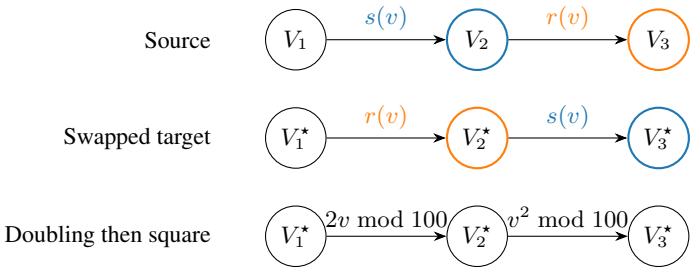


Figure 13: SCMs in the exact selection experiment. Matching colors denote shared conditional distributions. Source training observes every position; target selection observes only $V_1^\star$ and $V_3^\star$. Circuit output conditionals sum over generated intermediate values.

Fitting and selection. For every source position, we fit a gate for each earlier parent choice, including the empty parent list. This gives six gates: three unconditional distributions and three one-parent conditionals. Estimation uses $N = 10^5$ source traces and the probability-floor estimator with $\epsilon = 10^{-4}$. These gates are fixed during target selection. Neither the source parents nor the target parents or mechanism matches are supplied to Circuit-AD.

For each size $\ell \in \{2, 3, 4, 5\}$, including the input, we enumerate every legal composition of these gates. There are respectively 6, 54, 648, and 9,720 circuits. All n target input/output pairs are used to minimize empirical risk plus

$$\gamma_\ell = \log(1/\epsilon)\sqrt{\frac{(\ell - 1)\log(24\ell) + \log(80)}{2n}}.$$

No target intermediate value enters gate fitting or selection. The source-fit gates can include predictors with parent arguments other than the generating source parents; every candidate remains a composition of fitted source predictors.

Evaluation and replication. We evaluate the target conditional and each selected predictor's conditional table exactly, summing over all target input values with their true probabilities. Each target and each of the three input distributions has 20 independent source/target datasets. The mean gives equal weight to the three fixed distributions. Its standard error is obtained by adding the three within-distribution variances of the sample means and dividing by nine before taking the square root. Target budgets are $n \in \{5, 10, 20, 50, 100, 200\}$, with observations nested across budgets.

At $n \geqslant 20$, all runs select two gates for the swapped and repeated targets, and one for the irrelevant-intermediate target. The latter omits an intermediate position that does not affect the final output. Targets involving doubling or squaring retain larger risk across the reported budgets (Table 8). Target samples select a composition within the enumerated family while the fitted gates remain fixed.

Target	$n = 20$	$n = 100$	$n = 200$
Swapped composition	0.0486 ± 0.0005	0.0486 ± 0.0005	0.0486 ± 0.0005
Repeated shift	0.0484 ± 0.0005	0.0484 ± 0.0005	0.0484 ± 0.0005
Irrelevant intermediate	0.0655 ± 0.0006	0.0655 ± 0.0006	0.0655 ± 0.0006
Shift then doubling	2.789 ± 0.056	2.658 ± 0.013	2.574 ± 0.026
Doubling then square	2.670 ± 0.013	2.554 ± 0.022	2.532 ± 0.020
Increased replacement noise	0.657 ± 0.092	0.269 ± 0.002	0.265 ± 0.004

Table 8: Excess risk of outcome-supervised Circuit-AD over all six targets. Source sample size, candidate family, and uncertainty calculation are the same as in Table 7. The final three rows use doubling, squaring, or a higher replacement probability than the source.

G.5 Source estimation and composition noise

G.5.1 Source sample size

A separate Circuit-TR study uses ten target SCMs with ten positions, two inputs, 100 values, and at most two parents per position. A 15-position source supplies all required gates. With the target structure and source matches given and no target observations, increasing the source sample size from 10^3 to 10^6 reduces excess risk from 2.358 ± 0.273 to 0.151 ± 0.025. The estimates average three datasets within each task, then report one standard error across the ten tasks. Composed conditional tables and target distributions are evaluated exactly. This study isolates source estimation error in the supplied Circuit-TR comparator; it does not estimate the cost of selecting an unknown composition.

G.5.2 Approximating one fixed noisy-GCD target

To examine the approximation term in Equation (58), we fix the target concept to

$$P^\star(y \mid x_1, x_2) = (1-p)\mathbb{1}\{y = \gcd(x_1, x_2)\} + p/100,$$

with uniform inputs in $\{0, \ldots, 99\}$ and $p \in \{0.01, 0.1\}$. We compare remainder- and subtraction-based compositions of known conditional distributions, with replacement probability p at every operation. Their internal noise can produce different conditional tables; each is evaluated against the same $P^\star$ above. A composition with K iterations has size $\ell = 3K + 3$. We evaluate all 10,000 input pairs, approximating each composed output conditional by 1,000 intermediate trajectories per pair. Using known gates removes source estimation error from this comparison.

Within these two constructions, the first noiselessly correct remainder and subtraction circuits have $K = 9$ and $K = 98$, respectively. At $p = 0.01$, their excess risks are 0.154 and 1.728 nats. A shorter subtraction circuit with $K = 23$ has excess risk 0.703 despite being noiselessly incorrect on some inputs: additional noisy operations can cost more than they gain in arithmetic correctness. Using gates fitted from 10^5 source samples instead raises the excess risks of the 30- and 297-position constructions to 0.279 and 2.248 nats, respectively, estimated with 200 trajectories per input. This comparison adds source estimation error to the intrinsic noise of the known gates. These are approximation errors for the same target concept, the comparator term in Equation (8). The comparison illustrates how the available operations and composition length affect that term; it does not establish minimum circuit sizes or adaptation sample complexity. The bound $(\ell - 3)\log(1/(1-p))$ is shown only where the noiseless computation is correct, as required by Appendix E.4.

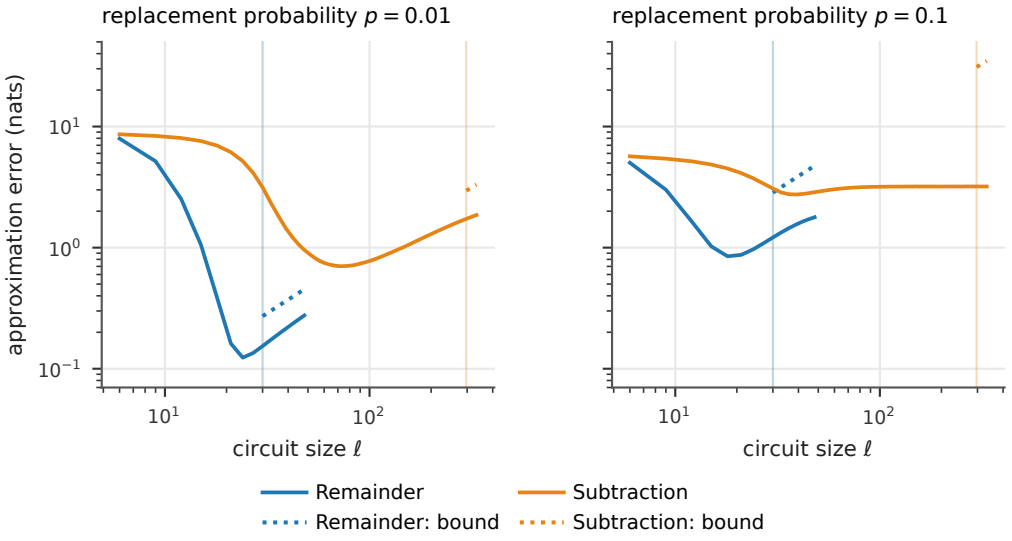


Figure 14: Approximation error against one fixed noisy-GCD concept over 100 values. Solid curves show Monte Carlo estimates of excess risk for compositions of known conditional distributions; dotted curves show the bound where the corresponding noiseless computation is correct. Vertical lines mark the first such size within each construction.

G.6 Reuse on new input combinations

G.6.1 New target input tuples with source-covered arguments

We repeat the three enumerable tasks of Appendix G.4.1, withholding a random 30% of the 100 input pairs from target adaptation in each seed. Source fitting still uses its original traces. The target conditional and all methods are evaluated only on the withheld pairs. Thus no evaluated full input tuple is labeled during target fitting, while its local gate arguments may be observed in the source.

Computation	Circuit-AD	Neural-AD	ERM-src+tgt	ERM-target
Range	$.030 \pm .001$	$.039 \pm .005$	$1.137 \pm .120$	$2.414 \pm .453$
$\min(\lvert a-b\rvert, \min(a,b))$	$.025 \pm .001$	$.028 \pm .002$	$.695 \pm .037$	$1.512 \pm .048$
One subtraction iteration	$.026 \pm .001$	$.061 \pm .031$	$.589 \pm .041$	$1.330 \pm .372$

Table 9: Excess risk on held-out input pairs at $n = 500$, under outcome supervision. Entries are mean $\pm$ one standard error across three target-data and split seeds, conditional on one source fit. The final task is a one-iteration computation, not complete GCD.

The small risks of Circuit-AD and Neural-AD in Table 9 show reuse on new full input combinations. The relevant coverage is that of the local argument configurations needed by a useful composition, rather than direct exposure to every target input tuple.

G.6.2 Computing aggregates of electricity measurements

Inputs and concepts. We use ElectricityLoadDiagrams20112014 (Trindade, 2015), a collection of 15-minute electricity measurements from 370 clients. We form hourly means, then quantize each client's measurements into 100 levels using the 99.5th percentile of its positive readings over 2012–2013 as the upper reference value: the level is $\lfloor 99x/x_{\text{ref}} \rfloor$, clipped to $\{0, \ldots, 99\}$. Nineteen clients without development-period consumption are excluded. The quantization step is client-specific, with median 2.5 kW. An input contains four consecutive hourly levels, omitting windows with gaps or invalid quantized values. The two target concepts are deterministic: the range of the four levels and their largest adjacent absolute change.

The n adaptation samples are drawn from a pool of 4.3 million windows from 246 development clients in 2012–2013. Evaluation uses 105 other clients in 2014. Quantized input tuples occurring in the development pool are removed from the evaluation pool, leaving approximately 174,000 windows; a fixed sample of 20,000 is evaluated. For each seed, the first n windows of a random ordering of the development pool supply nested target samples. Risk is averaged over the evaluation windows. The split tests computation on a different client/year distribution and on unseen full input tuples.

Source gates and supplied compositions. The source is the synthetic full-library circuit in Figure 1, fitted on traces over 100 values. Its gates are frozen and reused on the argument distributions induced by real measurement windows. Circuit-TR computes range with a tree of maxima, a tree of minima, and a final absolute difference: seven gates and 11 positions. For largest adjacent change, it uses three absolute differences followed by two maxima: five gates and nine positions. These noiseless computations also define the intermediate values exposed under process supervision. Their supplied fitted-gate risks are 0.0715 and 0.0565 nats, respectively. Neural-AD searches circuits with at most 16 positions. Evaluation uses 200 executions per input; since the target answer is deterministic, excess risk equals log loss at that answer.

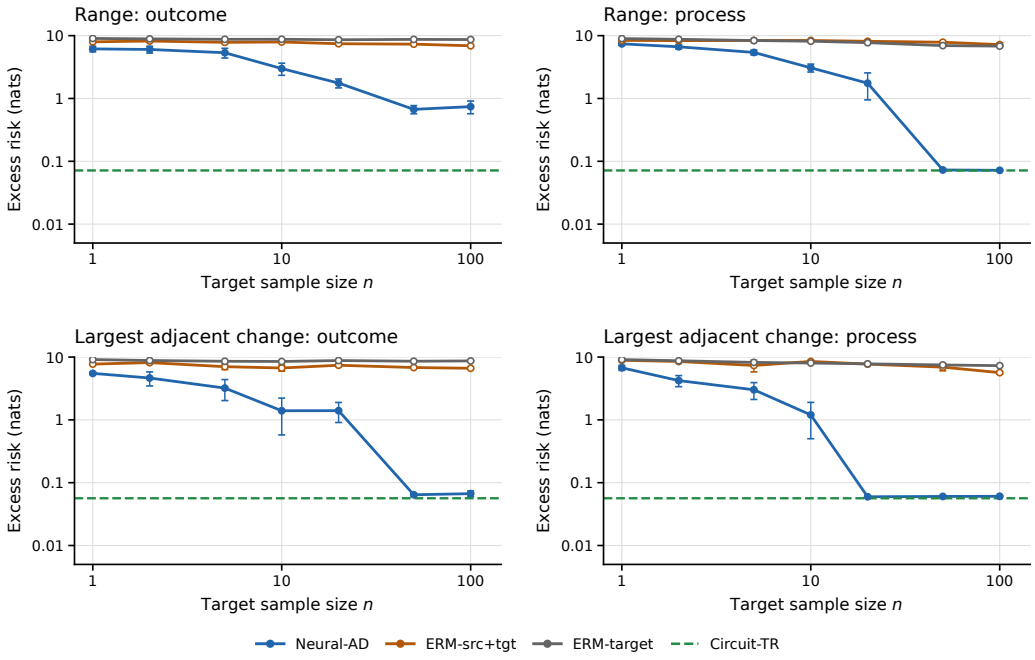


Figure 15: Electricity-window aggregates under outcome and process supervision. Source gates are fitted on synthetic traces; adaptation and evaluation use distinct client/year splits with repeated full input tuples excluded across splits. Error bars give one standard error across target-data seeds, conditional on the same source fit and evaluation windows. Neural-AD uses five target datasets at every budget. Each ERM baseline uses three for range and two for adjacent change.

At $n = 100$, Neural-AD attains mean excess risk 0.067 for largest adjacent change under outcome supervision and 0.061 under process supervision, compared with 0.057 for the supplied Circuit-TR composition. For range, process supervision reaches 0.072, close to Circuit-TR's 0.071, while outcome supervision remains at 0.741. These estimates average five target datasets. The two aggregates thus differ in the difficulty of selecting a composition from final answers, despite using the same fitted library and input windows.

Aggregate	Supervision	n	Circuit-TR	Neural-AD	ERM-src+tgt	ERM-target
Range	Outcome	20	0.071	1.759 ± 0.285 (5)	7.438 ± 0.231 (3)	8.604 ± 0.075 (3)
Range	Outcome	100	0.071	0.741 ± 0.169 (5)	6.901 ± 0.054 (3)	8.670 ± 0.103 (3)
Range	Process	20	0.071	1.751 ± 0.802 (5)	8.135 ± 0.240 (3)	7.734 ± 0.105 (3)
Range	Process	100	0.071	0.072 ± 0.001 (5)	7.198 ± 0.464 (3)	6.817 ± 0.104 (3)
Adjacent change	Outcome	20	0.057	1.401 ± 0.493 (5)	7.409 ± 0.374 (2)	8.799 ± 0.004 (2)
Adjacent change	Outcome	100	0.057	0.067 ± 0.007 (5)	6.633 ± 0.186 (2)	8.687 ± 0.002 (2)
Adjacent change	Process	20	0.057	0.060 ± 0.001 (5)	7.718 ± 0.128 (2)	7.798 ± 0.104 (2)
Adjacent change	Process	100	0.057	0.061 ± 0.001 (5)	5.653 ± 0.264 (2)	7.304 ± 0.042 (2)

Table 10: Excess risk in nats for electricity aggregates, reported as mean $\pm$ one standard error, with the number of target datasets in parentheses. Circuit-TR uses no target samples. The source fit and evaluation windows are shared across target datasets.

G.7 Gates from pretrained language models

		Source model: Qwen2.5-Instruct		
Target computation	$T^\star$	1.5B	3B	7B
Quadratic	6	70.9	78.7	93.2
Median	7	97.0	97.9	100.0
Symmetric sum	8	61.9	69.7	90.0
Range	8	99.1	99.3	100.0
Fibonacci	10	77.0	99.9	100.0
LCG	10	60.6	64.6	86.3
GCD, remainder	30	0.6	5.8	98.3

Table 11: Exact-answer accuracy (%) from Circuit-TR with fixed language-model gates and supplied target circuits. No target training samples are used. Evaluation covers all 10,000 input pairs for two-input tasks and a fixed sample of 10,000 input triples otherwise. Predicted output probabilities use 1,000 Monte Carlo trajectories per input. Each entry evaluates one fixed set of gates.

Source gates. We use the released Qwen2.5-Instruct 1.5B, 3B, and 7B checkpoints (Qwen Team, 2024), loaded in bfloat16 with their chat templates. Model identifiers are `Qwen/Qwen2.5-1.5B-Instruct`, `Qwen/Qwen2.5-3B-Instruct`, and `Qwen/Qwen2.5-7B-Instruct`. The 1.5B and 7B checkpoints use Apache 2.0; the 3B checkpoint uses the Qwen Research License. These models supply gates as conditional distributions, without parameter updates in this experiment.

For each of the nine arithmetic operations, we prompt the model on every argument configuration: 10,000 for each binary operation and 100 for each unary operation, totaling 60,300 configurations per model. Prompts state the operation and include three fixed worked examples. Three prompt formats are compared by mean operator accuracy, using the known single-operation answers rather than target-composition performance. All three models select the format with short worked reasoning. This calibration and the worked examples are source-side supervision; the zero-target-data claim concerns the subsequent compositions.

The model greedily generates up to 48 tokens of reasoning, ending in an answer tag. At that tag we calculate a distribution over zero-padded two-digit outputs: first-digit probabilities are normalized over digits, then multiplied by the corresponding normalized second-digit probabilities. The resulting distribution is projected to the floor $\epsilon = 10^{-4}$. These are conditional distributions given the generated reasoning, not a marginal over all possible language-model reasoning sequences. The cached tables define the gates used during composition.

If generation does not produce the answer tag, digits are scored immediately after the original prompt. This occurs in 3,378, 133, and eight of the 10,000 remainder configurations for 1.5B, 3B, and 7B, respectively, and in one multiplication configuration for 1.5B. These cases remain in the evaluated

tables. Table 12 reports each operator's argmax accuracy; the smaller models' remainder errors are substantial despite their high accuracy on most other operations.

Operation	1.5B	3B	7B
Addition	96.69	99.99	100.00
Subtraction	99.89	98.09	100.00
Multiplication	84.24	87.19	96.18
Minimum	99.87	99.90	100.00
Maximum	99.01	99.26	100.00
Remainder	56.46	88.69	96.17
Copy	100.00	100.00	100.00
Increment	100.00	100.00	100.00
Decrement	100.00	100.00	100.00

Table 12: Single-operation exact-answer accuracy (%) over every argument configuration in $\{0, \ldots, 99\}$, after the stated extraction procedure. Columns are Qwen2.5-Instruct model sizes. These fixed tables are used for every target computation.

Target computations. The available library contains nine operations: addition, subtraction, multiplication, minimum, maximum, remainder, copy, increment, and decrement. Each circuit in Table 13 uses a subset of these gates. Arithmetic is modulo 100 when needed, with $a \bmod 0 = a$. Each target circuit specifies its ordered parents and gate assignments. Inputs are uniform on $\{0, \ldots, 99\}$.

Target	Inputs M	Length $T^\star$	Noiseless computation
Quadratic	3	6	$(ax + b)x \bmod 100$, from (x, a, b)
Median	3	7	$\max\{\min(a, b), \min(\max(a, b), c)\}$
Symmetric sum	3	8	$(ab + bc + ac) \bmod 100$
Range	3	8	$\max(a, b, c) - \min(a, b, c)$
Fibonacci	2	10	Eight additions of the preceding two values
LCG	2	10	Four updates $x \leftarrow ax + 1$, from (x, a)
GCD, remainder	2	30	Nine maximum/minimum/remainder iterations, then final maximum

Table 13: Targets for pretrained-model composition. Length includes input and output positions. Recurrences use arithmetic modulo 100; LCG denotes a linear congruential recurrence. Every target operation is noiseless.

Target composition and evaluation. The target circuits in Table 11 specify the ordered parents and the operation used at each position. No target traces or input/output samples are used for fitting or selecting a composition. Inputs are uniform. Evaluation covers every two-input pair, and a fixed sample of 10,000 uniformly drawn triples for each three-input task; all model sizes use the same inputs. Predicted output probabilities average 1,000 intermediate trajectories per input. The target's exact answer has probability one, so its conditional entropy is zero and risk equals excess risk.

This setting examines the composition inequality with supplied approximate gates. Its noiseless target does not satisfy the positive-target assumptions of the finite-sample estimation theorems, which are not invoked here. Table 14 compares risk with the sum of local gate losses along the true target computation, the upper bound in the KL composition argument. Uniform single-operation accuracy can hide errors on frequently reached arguments. For example, the 3B maximum gate has accuracy 99.26% over uniform argument pairs, but only 17.19% on the argument pairs reached at the final maximum in the noiseless target computation, where one argument is always zero. What controls the composition bound is local risk on these target arguments, not average accuracy under uniform arguments. The extraction procedure also fixes a two-digit answer representation; its gate errors include formatting behavior and do not measure arithmetic ability independently of that representation.

The experiment evaluates these fixed gates and supplied compositions; it does not compare learning an unknown composition or directly fine-tuning the language model.

	1.5B		3B		7B	
Task	Risk	Local-loss sum	Risk	Local-loss sum	Risk	Local-loss sum
Quadratic	2.275	2.975	1.833	2.320	0.615	0.697
Median	0.339	0.464	0.164	0.207	0.030	0.040
Symmetric sum	2.870	4.357	2.543	3.467	0.881	1.077
Range	0.132	0.208	0.078	0.118	0.045	0.050
Fibonacci	0.812	0.830	0.086	0.088	0.079	0.080
LCG	2.923	6.050	2.720	4.932	1.116	1.558
GCD, remainder	8.854	114.373	8.316	45.522	0.247	1.374

Table 14: Risk and summed local log loss in nats for the noiseless targets. Local losses are averaged over arguments reached by the target circuit and summed over non-input positions; this sum bounds the exact composed predictor's risk. Reported predictor risks use 1,000 Monte Carlo trajectories per input. The fixed gates and evaluation inputs are those of Table 11.

G.8 Numerical records and replication

For the ten-value GCD studies, target budgets are $n \in \{1, 2, 5, 10, 20, 50, 100, 200, 500, 1000\}$; the GCD-concept ERM grids also include $n = 3$. The 100-value library comparison uses $n = 50$ and three target datasets. Electricity comparisons use $n \in \{1, 2, 5, 10, 20, 50, 100\}$. The number of target datasets for each reported estimate appears in Table 15. Samples are nested within each seed. GCD evaluation covers all input pairs and uses 400 sampled circuits and executions per input for Neural-AD; electricity uses the fixed 20,000-input evaluation specified above. Repeated Circuit-TR records represent one supplied fitted composition and are counted once.

Table 15: Available replication for the experimental studies. Each entry gives target sample sizes followed by the number of target-data seeds. The source fit is shared within each library and vocabulary condition.

Task / library	Method / supervision	Target sample sizes n: seeds
Electricity change, $\|\mathcal{V}\| = 100$, full	ERM-src+tgt / outcome+process; ERM-target / outcome+process	1, 2, 5, 10, 20, 50, 100: 2
Electricity change, $\|\mathcal{V}\| = 100$, full	Neural-AD / outcome+process	1, 2, 5, 10, 20, 50, 100: 5
GCD concept, $\|\mathcal{V}\| = 10$, full	ERM-src+tgt / outcome; ERM-target / outcome	1, 2, 3, 5, 10, 20, 50, 100, 200, 500, 1000: 5
GCD concept, $\|\mathcal{V}\| = 10$, full	Neural-AD / outcome	1, 2, 5, 10, 20, 50, 100, 200, 500, 1000: 5
GCD concept, $\|\mathcal{V}\| = 10$, full	ERM-src+tgt (RFT) / outcome; ERM-target (RFT) / outcome	1, 2, 3, 5, 10, 20, 50, 100, 200, 500, 1000: 5
GCD concept, $\|\mathcal{V}\| = 10$, max/min/sum/product	ERM-src+tgt / outcome; ERM-target / outcome	1, 2, 3, 5, 10, 20, 50, 100, 200, 500, 1000: 5
GCD concept, $\|\mathcal{V}\| = 10$, max/min/sum/product	Neural-AD / outcome	1, 2, 5, 10, 20, 50, 100, 200, 500: 5; 1000: 4
GCD concept, $\|\mathcal{V}\| = 10$, max/min/sum/product	ERM-src+tgt (RFT) / outcome; ERM-target (RFT) / outcome	1, 2, 3, 5, 10, 20, 50, 100, 200, 500, 1000: 5

Task / library	Method / supervision	Target sample sizes n: seeds
GCD concept, $\|\mathcal{V}\| = 10$, no remainder	ERM-src+tgt / outcome; ERM-target / outcome	1, 2, 3, 5, 10, 20, 50, 100, 200, 500, 1000: 5
GCD concept, $\|\mathcal{V}\| = 10$, no remainder	Neural-AD / outcome	1, 2, 5, 10, 20, 50, 100, 200, 500, 1000: 5
GCD concept, $\|\mathcal{V}\| = 10$, no remainder	ERM-src+tgt (RFT) / outcome; ERM-target (RFT) / outcome	1, 2, 3, 5, 10, 20, 50, 100, 200, 500, 1000: 5
GCD concept, $\|\mathcal{V}\| = 100$, full	Neural-AD / outcome	50: 3
GCD concept, $\|\mathcal{V}\| = 100$, max/min/sum/product	Neural-AD / outcome	50: 3
GCD concept, $\|\mathcal{V}\| = 100$, no remainder	Neural-AD / outcome	50: 3
GCD circuit, $\|\mathcal{V}\| = 10$, full	ERM-src+tgt / outcome+process; ERM-target / outcome+process; Neural-AD / outcome+process	1, 2, 5, 10, 20, 50, 100, 200, 500, 1000: 5
Electricity range, $\|\mathcal{V}\| = 100$, full	ERM-src+tgt / outcome+process; ERM-target / outcome+process	1, 2, 5, 10, 20, 50, 100: 3
Electricity range, $\|\mathcal{V}\| = 100$, full	Neural-AD / outcome+process	1, 2, 5, 10, 20, 50, 100: 5

The archived records identify each source fit, target concept, noise level, penalty scale, and evaluation setting. Marginal-likelihood and RFT-trained ERM variants are distinct estimators and are summarized separately. Standard errors across target seeds do not quantify uncertainty from retraining the source or from finite Monte Carlo integration. Numerical values for the decoded-circuit diagnostic are kept separate from the mixture risks reported for Neural-AD.

H Detailed model architecture

Neural-AD has a causal transformer that generates gate indicators and parent choices, and a shared predictor that applies the selected gate to its ordered argument values. Source training fits both parts from observed sequences. Target adaptation trains the generator while keeping the predictor fixed.

Generating structural choices. Position i has a learned token p_i, with starred parameters for the target. A one-layer causal transformer maps the position tokens to embeddings. At target position i, embeddings of the earlier gate and parent choices have been added to their position tokens: if position $j < i$ uses indicator k_j and ordered parents $J_j = (j_1, \ldots, j_a)$, its token is

$$u_j^\star = p_j^\star + b_\Phi^\star(k_j) + \sum_{h=1}^{c} b_A^\star(j_h), \qquad j_h = 0 \text{ for } h > a.$$

Here $b_\Phi^\star$ and $b_A^\star$ are learned embeddings, and 0 denotes the null argument. The implementation shares $b_A^\star$ across heads: swapping two parents leaves this history token unchanged. The head-specific selections and ordered argument concatenation still distinguish their roles when the gate executes. This encoding conditions later choices on the selected parent set and gate; the history token does not preserve the order of the parent list. The input positions and the current position have token $p_j^\star$, without an added choice. Running the causal transformer on $u_{1:i}^\star$ gives the current embeddings $e_{1:i}^\star$. A linear map of $e_i^\star$ gives gate-indicator logits. For argument slot h, linear query and key maps give predecessor scores

$$s_{h,i,j}^\star = \frac{\langle Q_h^\star e_i^\star, K_h^\star e_j^\star \rangle}{\sqrt{r}}, \qquad j < i.$$

Here r is the query and key dimension. The gate indicator is sampled first. Its fixed arity $a \leqslant c$ specifies how many heads select arguments; gates with $a > i - 1$ are masked. The first a heads then select distinct predecessors in order, using softmax probabilities with previously selected predecessors masked. The other heads receive a null argument. After these choices are sampled, their embeddings are added to $p_i^\star$ for the next generation step. Thus later choices can depend on earlier choices. The generator never receives the input values, intermediate execution values, or observed target answer.

The fixed active slots $1, \ldots, a$ and the arity of an indicator are preserved in every composition. All null arguments use the same embedding. These conventions make an indicator specify a gate with a fixed ordered argument space.

Gate predictor. A shared value embedding maps each $v \in \mathcal{V}$ and the null value $\varnothing$ to a vector. The predictor Ψ is an MLP applied to the concatenated embeddings of the c argument slots and the one-hot gate indicator. If its output logits are z_y, it returns

$$\Psi_\theta(y \mid v_J; k) = \epsilon + (1 - |\mathcal{V}|\epsilon)\frac{\exp z_y}{\sum_{y' \in \mathcal{V}} \exp z_{y'}}, \qquad 0 < \epsilon < |\mathcal{V}|^{-1}. \tag{65}$$

Unused slots are filled with $\varnothing$. No position embedding or prefix representation is supplied to this MLP. Consequently, the same indicator and ordered argument values give the same conditional distribution wherever the gate is placed.

The two attention operations have different roles. Soft causal attention within the generator processes position tokens and previously chosen structure. The resulting query/key scores specify hard selections of arguments for the gate predictor. A sampled composition therefore evaluates one gate at each generated position, using values at fixed predecessor positions.

Source choices and loss. During source training, the generator receives only the position tokens $p_{1:T}$, without feedback from sampled choices. Its embeddings are therefore deterministic given the parameters, and its hard choices at different positions are independent. Write $S = (A, \Phi)$ for a hard source representation and $\pi_\theta(S)$ for its probability. At position i, let $\alpha_{\theta,i}(k, J)$ be the probability of indicator k and ordered parent list J. It is the product of the indicator probability and the sequentially masked parent probabilities above. The next-token loss is

$$\widehat{\mathcal{L}}(\theta) = -\frac{1}{NT} \sum_{v \in D} \sum_{i=1}^{T} \sum_{(k,J)} \alpha_{\theta,i}(k, J) \log \Psi_\theta(v_i \mid v_J; k).$$

The inner sum ranges over legal choices, including zero-arity gates. This computes the expectation in Equation (39) exactly by enumerating the choices at each position, without enumerating complete source representations.

Let $p_{i,k} = \sum_J \alpha_{\theta,i}(k, J)$ and let $a(k)$ be the arity of indicator k. The expected structural penalty also has an exact expression:

$$\begin{aligned}\Omega(\theta) &= \mathbb{E}_{S \sim \pi_\theta}\left[d(S) + \sum_{i=1}^{T} |\mathbf{Pa}_i(S)|\right] \\ &= \sum_{k=1}^{d_{\max}} \left[1 - \prod_{i=1}^{T}(1 - p_{i,k})\right] + \sum_{i=1}^{T}\sum_{k=1}^{d_{\max}} p_{i,k} a(k).\end{aligned}$$

Here $d(S)$ counts occupied indicators and $d_{\max}$ is their available capacity. Differentiating $\widehat{\mathcal{L}}(\theta) + \lambda\Omega(\theta)$ updates the generator, its query/key and indicator projections, the value embeddings, and the gate MLP.

Source fitting and retained gates. The generator is initialized to prefer different indicators at the source positions and parent choices with low empirical conditional entropy. These preferences use source training traces. The gate predictor is fitted first with the initialized structural probabilities held fixed; joint training then updates both parts while increasing the penalty to its specified value. A held-out source sample selects a checkpoint and one hard representation from the greedily decoded representation, sampled alternatives, and candidates formed by replacing a positive-arity gate with a zero-arity gate. With that representation fixed, the gate predictor is refitted by next-token cross-entropy. Its occupied indicators, arities, and argument conventions define the d retained gates. Appendix C.1 states prediction guarantees in terms of the achieved source fitting error.

Target parameters and size. Outcome-supervised adaptation starts from the pretrained generator and its position tokens, query/key maps, and indicator projection. In outcome adaptation, the embeddings of prior gate and parent choices start at zero and are trainable. A categorical distribution over $L \in \{M+1, \ldots, L_{\max}\}$ specifies circuit size; with process supervision, this distribution is fixed at the observed size $T^\star$. Process-supervised fitting instead initializes fresh structural parameters and fits them to the circuit obtained by exact local selection. The target parameters $\theta^\star$ include the generator (including its feed-forward block), position tokens, structural projections, feedback embeddings, and, with outcome supervision, size logits. The value embeddings, null embedding, gate MLP, output map, and retained indicator set remain fixed.

For outcome supervision, sample L and then the choices at positions $M+1, \ldots, L$. The probability of the resulting circuit is

$$\pi_{\theta^\star}(C) = \Pr_{\theta^\star}(L) \prod_{i=M+1}^{L} \Pr_{\theta^\star}\Big(k_i, J_i \mid (k_j, J_j)_{j=M+1}^{i-1}\Big). \tag{66}$$

The input positions have no generated choices. Conditional on the sampled circuit, execution draws values successively from its frozen gates and returns the value at position L. Its $L - M$ gate applications include the output gate.

Representation of candidate circuits. The point-mass condition in Theorem C.5 holds in the closure of a sufficiently wide parameterization. To see this, set the feedback embeddings and the transformer's attention and feed-forward residual branches to zero. Its output is then the position token. With position and structural query/key dimensions at least $L_{\max}$, choose coordinate vectors for these tokens and keys. Each head's linear query map can assign an arbitrarily dominant score to the prescribed predecessor at each position. Sequential masking preserves the desired distinct parent choices. The indicator projection can likewise concentrate on a prescribed gate at every position, and the size distribution can concentrate at its prescribed value. This constructs a point mass at every legal circuit in the limit. It does not require representing every distribution over circuits. If some candidate point masses are unavailable, the inclusion comparison of optimal empirical values applies only to the representable candidates.

Prediction. The predictor in Equation (14) averages circuit output distributions under the generator. Sampling a circuit and its intermediate values, then averaging the final gate's output distribution, estimates this predictor. On small instances, summation over all intermediate values gives the exact conditional distribution of a fixed circuit. For log loss, averaging answer probabilities precedes taking their negative logarithm. Greedy structural choices define a single circuit whose performance can be evaluated separately from the averaged predictor.

I Related work

Our analysis connects three questions: which source conditionals identify a target concept under supplied structure, how their estimation errors affect a composed predictor, and how target samples permit selection when the composition is unknown. The following discussion relates these questions to causal transportability, modular learning, compositional generalization, and statistical model selection. The comparisons distinguish the observations available to the learner, the object being reused, and the guarantee sought.

I.1 Causal and statistical transportability

Transportability formalizes when available distributions, together with assumptions about shared and changing mechanisms, determine a quantity in another domain (Pearl & Bareinboim, 2011; Bareinboim et al., 2013). General transportability combines observational and experimental distributions from heterogeneous domains to identify conditional causal effects (Lee et al., 2020); soft-intervention transportability also permits stochastic and conditional interventions (Correa & Bareinboim, 2020). Statistical transportability concerns observational queries such as $P^\star(y \mid \mathbf{x})$ and allows unobserved confounding and incomplete target observations (Correa & Bareinboim, 2019). Its identification procedures decompose and combine available quantities, including marginalization over intermediate

variables. Circuit-TR follows this identification perspective for a sequential computation: supplied matches identify the required target conditionals from the source distribution. Their fitted gates are composed according to the target's ordered parent arguments.

The circuit setting makes the number and arrangement of gate applications explicit. A source mechanism can be reused at several target positions, and different circuits can represent the same concept. Under the stated Markovian assumptions, local matching supplies a sufficient transport condition; Appendix D discusses queries requiring more general identification arguments. The risk analysis then accounts for estimating gates from source traces and for selecting their composition from target samples. These are separate questions from identification with exactly known distributions.

Transportability also connects directly to representation learning. Jalaldoust & Bareinboim (2024) characterize when the target conditional score of a representation can be obtained from source distributions, and connect this property to observable source invariances under mechanistic-stability and faithfulness assumptions. Jalaldoust et al. (2024) bound target functionals when the source distributions and structural constraints leave them only partially determined, and use those bounds for robust learning. Our target concept is a conditional table represented by a circuit. The comparison concerns the risk of fitting and composing source gates and the target-sample cost of competing with the supplied Circuit-TR predictor. Partial transportability concerns the remaining range of possible target quantities under the available information.

I.2 Causal invariance and robust transfer

Different invariance approaches use different causal assumptions and pursue different guarantees. Invariant causal prediction uses stability across environments to infer causal predictors and construct confidence statements (Peters et al., 2016). Rojas-Carulla et al. (2018) assume a subset of covariates gives an invariant outcome conditional and establish an adversarial optimality result for domain generalization. Magliacane et al. (2018) use joint causal inference over context and system variables to infer separating sets supporting invariant conditionals, without requiring the graph or intervention targets in advance. These approaches relate changes across environments to features of a common system. Circuit-TR instead receives cross-position mechanism matches and an ordered target composition; Circuit-AD uses target samples to select a composition of the fitted gates.

Invariance need not mean equality of an observed outcome conditional. The surgery estimator of Subbaswamy et al. (2019) constructs identifiable interventional distributions that remain stable when specified mechanisms change. Anchor regression controls sensitivity to a class of shifts in linear structural models through a modified least-squares objective (Rothenhäusler et al., 2021). Their robustness criteria express which environmental changes a predictor must withstand. Here shared mechanisms justify reuse of their local conditional distributions, while the target concept can change with their composition. The adaptation guarantee controls risk under the target distribution supplying the samples, relative to a Circuit-TR comparator justified by the supplied structural information.

I.3 Independent mechanisms and causal representations

The independent causal mechanisms perspective treats a system as autonomous mechanisms whose organization supports reuse across environments (Schölkopf et al., 2021). Parascandolo et al. (2018) learn separate mechanisms by competition among modules that map transformed observations back to a reference distribution. Bengio et al. (2020) use adaptation speed after sparse mechanism changes as a signal for learning causal structure and representations. Their analysis explains why a correctly fitted causal conditional that remains unchanged has zero expected adaptation gradient. Under outcome supervision, circuit adaptation studies another use of modularity: source gates remain fixed while their assignments and ordered arguments define a new computation. Target compositions can differ substantially from the source computation, and the statistical comparison depends on the available library and comparator size.

Causal representation learning also asks how causal variables can be recovered from observations that do not expose them directly (Schölkopf et al., 2021). Our source execution traces expose the sequential variables used to fit conditional gates. Process supervision exposes the target intermediate values as well; outcome supervision leaves them unobserved and evaluates the composed output conditional by marginalization. This information structure permits a direct comparison between

learning local choices and learning a complete composition. It also separates the representation of a concept as a circuit from the discovery of causal variables in perceptual data. The present results quantify the value of reusable mechanisms when source traces expose the variables on which their fitted gates operate.

Recent causal accounts connect mechanism identification to compositional generalization. Kong et al. (2026b) study a latent causal hierarchy and give parent-support conditions under which observed combinations determine new ones, together with conditions for recovering the hierarchy. Kong et al. (2026a) model reasoning traces through latent skills and routing choices, relating their identification and recombination to locally witnessed interfaces. These works make local coverage central to composition. Our variables are observed in the source traces, and the risk analysis follows the estimation of their conditionals through a supplied circuit and the subsequent selection of a composition from target samples.

I.4 Learned libraries and modular adaptation

Learning reusable computations and searching their compositions have a substantial history. Exploration–compression learns abstractions from solved tasks (Dechter et al., 2013); EC^2 adds neural guidance of synthesis (Ellis et al., 2018); DreamCoder combines library growth, search, and training on generated tasks (Ellis et al., 2021). Stitch improves abstraction discovery through corpus-guided synthesis (Bowers et al., 2023), and LILO combines compression with language-model-guided synthesis and documentation (Grand et al., 2024). These systems learn the vocabulary in which later tasks can have shorter solutions. Our outcome-supervised analysis begins with a source-fitted gate library and studies the risk of its compositions. DreamCoder's likelihood-based formulation includes probabilistic computations. Here local conditionals are estimated from source traces, causal assumptions certify their reuse, and target samples select their arrangement.

Alet et al. (2018) provide a particularly close adaptation procedure: learn neural modules across tasks, freeze them, and select a composition by empirical loss on a new task. Their composition schemes allow repeated modules and varying structures. HOUDINI combines typed compositions of neural functions with symbolic search and parameter learning (Valkov et al., 2018). Mendez & Eaton (2021) explicitly separate fitting a structure over frozen components from subsequently modifying or expanding the library. Circuit-AD shares the fixed-library selection principle. Here complete source traces provide observations of the gates' arguments and outputs, and causal matching supplies a Circuit-TR comparator; the analysis separates its estimation error from the target-sample cost of learning a composition.

Neural Module Networks assemble shared modules using a question's linguistic structure (Andreas et al., 2016b); learning the assembly mechanism also permits training from answers without observed layouts (Andreas et al., 2016a). Their layout varies with the question. Compositional recursive learners choose representation transformations from their current state and can learn when to stop (Chang et al., 2019). Each candidate in our setting instead fixes its gates and ordered parents across all inputs. Policy-sketch reinforcement learning likewise distinguishes supplied compositions, which permit direct reuse, from compositions learned through target interaction (Andreas et al., 2017). These distinctions concern which structural information is available; the sketches' state-dependent subpolicies and reward objective differ from our conditional gates and risk.

I.5 Compositional generalization, identification, and support

SCAN, COGS, and gSCAN test recombinations of familiar constituents (Lake & Baroni, 2018; Kim & Linzen, 2020; Ruis et al., 2020). CFQ's distribution-based compositionality assessment constructs splits with similar distributions of individual rule applications and different distributions of their combinations (Keysers et al., 2020). Such coverage of constituent identities is different from coverage of a gate's ordered argument values. The latter controls source estimation error in our results. The distinction between systematic recombination and increased computation length is also central to the taxonomy of Hupkes et al. (2020). Meta-learning can encourage compositional behavior through suitable task distributions and objectives (Lake, 2019; Conklin et al., 2021; Lake & Baroni, 2023); it need not take the form of explicit symbolic circuit search.

Several theories identify conditions under which fitted components determine unseen combinations. Wiedemer et al. (2023) assume a shared composition function and establish extrapolation from exact fitting under component coverage, regularity, a Jacobian rank condition, and agreement at an initial configuration. Lachapelle et al. (2023) identify latent blocks using additive decoders and extend their combinations to Cartesian-product support. For object-centric learning, Wiedemer et al. (2024) distinguish decoder extrapolation from correct encoding of new combinations and impose compositional consistency to obtain both. Brady et al. (2025) extend this identification approach beyond additivity by restricting interactions between latent blocks. These results use structural and support assumptions to justify recombination. Our source traces expose execution variables directly. We estimate gates from these observations and compare their candidate compositions through the resulting output conditionals.

Schug et al. (2024) study identification and recombination of neural modules in a teacher–student setting. With matched model classes, full input support, suitable task variation, and connected compositional support, exact population fitting permits new combinations up to the modules' identification equivalence. Circuit-AD's comparison has a different target: low risk for the output conditional, without requiring recovery of one circuit representation. Lippl & Stachenfeld (2025) further show that compositional structure in a kernel constrains, but does not itself ensure, correct behavior on unseen combinations. Explicit component structure and successful recombination are separate properties.

Structured extrapolation also has quantitative risk analyses. Dong & Ma (2023) bound target squared error for additive functions using marginal coverage and connectivity of observed input combinations, even when joint supports differ. Compositional Risk Minimization models stochastic conditionals with additive attribute energies and trains by conditional log loss (Mahajan et al., 2025). Its extrapolation result uses an affine-hull condition, correct source conditionals, and the target attribute prior. These are close precedents for exploiting local structure despite global shift. Our composition instead applies conditional gates sequentially; source coverage concerns their arguments, and target samples select among their possible compositions.

Abbe et al. (2024a) study Boolean functions with withheld input regions and analyze a minimum-Fourier-degree bias for specified model classes. Their simplicity measure differs from circuit description length relative to a gate library: a high-degree function can have a short circuit. This comparison also separates two forms of novelty. Their unseen region belongs to one function's input domain; here source and target can use different computations while each reused gate has adequate local argument coverage. Circuit-AD's target samples and its population risk use the same target distribution.

I.6 Execution traces and algorithmic learning

Neural Programmer-Interpreters learn reusable procedures from execution traces that include calls, arguments, and termination information (Reed & de Freitas, 2016). CLRS supplies structured intermediate algorithm states as supervision (Veličković et al., 2022). Our process observations contain realized circuit values; they can leave gate identities and parent choices unknown. These values nevertheless permit evaluating local conditionals, which makes the statistical role of process supervision explicit.

The source-trace/target-outcome distinction also appears directly in algorithmic transfer: Xhonneux et al. (2021) pretrain graph processors using source traces and learn target algorithms from final outputs. Neural Execution Engines compose pretrained numerical subroutines within supplied algorithms (Yan et al., 2020). Their masks can depend on execution state, whereas our circuits have fixed parent lists. Together these works establish operational precedents for learning from traces and reusing subcomputations, with different routing and supervision conventions.

Xu et al. (2020) connect algorithmic alignment to sample complexity under local auxiliary supervision, sequential module training, and stability assumptions. Their analysis and our process results both make component errors and their propagation explicit. Our bounds concern conditional KL error after source fitting and the risk of the target output conditional. The globality analysis of Abbe et al. (2024b) instead concerns learning barriers for specified Transformer models; intermediate supervision and inductive scratchpads can change the learning difficulty. Length-generalization studies likewise investigate repeated computation and accumulated error (Anil et al., 2022; Zhou et al., 2024). Our

oracle comparison controls the selection cost using the number of possible gate assignments and ordered parent lists at each circuit size.

I.7 Statistical complexity and noisy computation

Learning local conditional distributions on a fixed graph has established sample-complexity analyses under KL loss (Dasgupta, 1997). Abbeel et al. (2006) give parameter- and structure-learning guarantees for bounded-degree factor graphs. These analyses relate local statistical complexity to joint distribution estimation. Circuit-TR uses this factorization across domains: source conditionals are reused at certified target positions, with source coverage controlling their estimation error on target arguments.

Playgol supplies a direct precedent for the connection between a learned library, representation length, and sufficient sample bounds (Cropper, 2019). In a realizable logical-learning setting, it counts hypotheses and compares bounds when added background predicates shorten a composition. Circuit-AD similarly pays for both gate choices and ordered parent lists, while allowing estimated stochastic gates and approximate representations. The comparison is between risk bounds for available representations; the shortest expression need not have the lowest risk when its gates have unequal errors.

Minimum-complexity estimation balances likelihood fit against description length (Barron & Cover, 1991). Our square-root penalty comes from bounded-loss concentration over circuit descriptions, conditional on the source-fitted gates. Levin's description-length allocation of search resources supplies a complementary computational connection (Levin, 1984); the confidence assigned to a composition in our proof does not bound its search time. Appendix E.3 develops this distinction.

Transfer-learning lower bounds address a related but different statistical question. Hanneke & Kpotufe (2019) characterize binary-classification rates using source and target samples, noise, and distributional relations. Hanneke et al. (2023) show that selecting a model without knowing its transfer distance can yield rates substantially slower than those attainable when this information is supplied. Our comparator is a particular predictor built from fitted gates. Its risk comparison is conditional on a fixed fitted library; the transfer-learning rates above instead quantify how distributional relations govern the statistical value of source labels.

Information loss through noisy gates is studied by Evans & Schulman (1999), including the role of depth and redundancy in reliable computation. Appendix F uses a restricted chain family to relate length to the conditional table being represented. Two circuits with identical noiseless outputs can define different concepts once their gate noise is included. Shorter chains can preserve more signal in that family; reliable computation with redundant error correction explains why length alone cannot rank every noisy circuit.

I.8 Learning to search for compositions

Expert iteration alternates search with neural fitting to improved decisions (Anthony et al., 2017). ReST$^{\mathrm{EM}}$ generates outputs, retains those passing correctness feedback, and fits the model to them (Singh et al., 2024). Neural-AD applies this search-and-fit principle to compositions of frozen gates. It scores candidate circuits by empirical risk plus the size penalty and fits a generator over their structural choices. The resulting predictor averages circuit output conditionals. Its comparison with Circuit-AD therefore includes the attained empirical optimization error and the empirical Jensen gap. The guarantee thus separates the statistical cost of selecting a composition from the quality of the search.

J Limitations

Causal and distributional assumptions. The analysis assumes finite shared vocabularies, a known causal order, and Markovian SCMs. Sequential indexing must agree with causal order; arbitrary sequence data need not satisfy this condition. Strict positivity ensures that every local argument configuration can be observed, while the fitted probability floor bounds log loss. Coverage can still be poor when vocabulary size or parent arity is large. The source-estimation corollary is informative

when the comparator's gates are accurately estimated on the arguments it uses. Appendix D shows why the local transport procedure can fail with unobserved confounding.

Available gates and candidate circuits. Outcome-supervised Circuit-AD selects compositions of frozen source gates. Its attainable risk depends on the gate library and on the allowed compositions. Restricting the library can require longer computations; with noisy gates, these may approximate the target concept less accurately (Appendix F). The oracle inequality compares the selected predictor with the supplied Circuit-TR construction, including its gate-estimation error. Process supervision additionally permits gates to be fitted from observed local inputs and outputs (Appendix A). For a fixed gate library, the circuit-size bound controls the target-sample cost of selecting a composition.

Neural parameterization and optimization. The source guarantees depend on the conditional KL errors of the retained gates and coverage of their arguments. The sparse population characterization assumes a sufficiently expressive class and global optimization. Outcome adaptation uses finite RFT search over compositions of the frozen gates. Its log-loss comparison with Circuit-AD includes an empirical optimization residual and, for the averaged predictor, the gap between component and marginal log losses (Theorem C.5). Monte Carlo scoring introduces bias and variance in ranking and checkpoint selection. The analysis gives no bound on the computation needed to attain a prescribed residual. A finite length limit restricts the available comparator family.

Empirical scope. The composition-selection experiments use controlled discrete computations and specified aggregates of electricity measurements. The pretrained-model experiments supply target circuits and cache single-operation predictions. The neural comparisons use a shared source fit, so their uncertainty concerns target datasets conditional on that fit. Process and outcome results compare learning with different target observations; performance on the long noisy targets additionally depends on the retained gates, the size penalty, and the numerical search. Broader vocabularies, independent source fits, and naturally occurring target computations remain important extensions.