

Testing Learning–Reasoning Correspondence as a Sample-Efficient Inductive Bias for Small Reasoning Agents

Replication Study

Abstract

Learning–Reasoning Correspondence (LRC) proposes that augmenting training examples with symbolic-style inference tuples can improve small-agent learning. We test this claim in a controlled finite replication that fixes rule-chain and symbolic gridworld task families, ten paired seeds, size-matched neural and neuro-symbolic conditions, and oracle trace-bearing tuple diagnostics. Relevant regularized LRC improves final out-of-distribution accuracy in both executed local benchmarks: 0.945 versus 0.496 for the neural baseline on rule chains and 0.905 versus 0.405 on symbolic gridworld planning. Mechanism-isolation margins remain positive against noisy-tuple and no-regularizer ablations, with bootstrap intervals [0.207, 0.219] and [0.236, 0.253]. Tuple confidence is also better calibrated and more faithful than noisy augmentation under the available oracle traces. The stronger sample-efficiency claim is not established: all predeclared samples-to-threshold comparisons at 0.80 and 0.90 validation accuracy are censored on the executed grid. The evidence therefore supports LRC as a local mechanism for final OOD generalization in controlled finite tasks, but weakens claims about threshold sample complexity and leaves public bAbI/MiniGrid replication as future work.

1 Introduction

Neural agents often learn correlations faster than they learn reusable reasoning structure. The practical consequence is familiar across synthetic reasoning, text-based relation tasks, and small planning environments: a model can fit in-distribution examples while failing when rule depth, relation composition, or transition structure changes at test time (Weston et al., 2015; Lake and Baroni, 2018; Keysers et al., 2020; Sinha et al., 2019; Clark et al., 2020; Hupkes et al., 2020; Chevalier-Boisvert et al., 2023, 2018). Neuro-symbolic learning attacks this problem by giving the learner access to explicit rules, constraints, proofs, programs, or relational structure (d’Avila Garcez and Lamb, 2020; Besold et al., 2017; Bader et al., 2023; Sarker et al., 2024). Learning–Reasoning Correspondence (LRC) sharpens that broad idea into a per-example claim: a training example can be accompanied by inference tuples of the form premise, conclusion, and confidence, and these tuples can guide learning when they are relevant and regularized. This paper evaluates that claim as an empirical replication question rather than as a slogan about interpretability.

The motivating LRC draft argues that explicit reasoning artifacts can serve as an inductive bias for efficient agent training (Double-blind ICLR 2025 submission, 2025). That claim matters beyond the specific draft because many current reasoning systems blur the boundary between learned representations and symbolic operations. Constraint losses, differentiable logic programs, neural module systems, and solver-augmented language models all show that explicit structure can sometimes improve extrapolation or inspectability, but they also show that the benefit depends on the validity of the symbolic object and the comparison baseline (Xu et al., 2018; Hu et al., 2016; Fischer et al., 2019; Evans and Grefenstette, 2018; Manhaeve et al., 2018; Yang et al., 2017; Dong et al., 2019; Andreas et al., 2016; Pan et al., 2023; Sinha et al., 2023; Huang et al., 2023a). Added tokens, longer context, oracle leakage, or unmatched model capacity can masquerade as reasoning. A replication must therefore ask whether the tuple mechanism itself matters after size matching, noisy-tuple controls, and no-regularizer controls.

We implement a finite LRC variant for two controlled task families: propositional rule-chain classification with depth-shift OOD splits and a symbolic gridworld shortest-path decision task with factored state descriptions. Each family has ten fixed seeds and oracle trace metadata. The evaluated conditions are a neural-only learner, relevant regularized LRC, relevant LRC without the consistency regularizer, noisy or shuffled LRC with matched tuple count and confidence marginals, semantic-constraint and oracle-feature comparators where appropriate. This scope is intentionally narrower than the full agent-learning language in the draft. It is wide enough to test whether tuples help under known rules and transitions, but narrow enough that every manifest, seed, condition, and diagnostic can be inspected.

The result is mixed in a scientifically useful way. Relevant regularized LRC clearly improves final local OOD accuracy and beats the best ablation by a positive mechanism margin in both task families. It also produces lower tuple expected calibration error and higher oracle-trace faithfulness than noisy tuples. However, the predeclared first-hit statistics for reaching 0.80 and 0.90 validation accuracy are fully censored across the executed training-size grid. The data show better final OOD behavior,

not a verified reduction in examples needed to cross those thresholds. This distinction is central to the interpretation of LRC as an inductive bias.

This distinction also explains the paper’s replication stance. The study does not try to replace the full LRC proposal with a smaller task, nor does it treat a favorable local result as broad confirmation. Instead, it isolates the part of the proposal that can be tested with finite manifests and independent trace labels: whether relevant tuples plus a consistency penalty change learning behavior under controlled OOD shifts. The answer is positive for final local generalization and mechanism isolation, negative for the measured threshold claim, and conditional for interpretability.

Our contributions are:

- We formalize a finite, size-matched LRC objective and first-hit sample statistic that separate final OOD performance from threshold sample efficiency.
- We run a ten-seed local replication on controlled rule-chain and symbolic gridworld tasks with relevant, noisy, no-regularizer, neural, semantic-constraint, and oracle-feature conditions.
- We report mechanism-isolation, calibration, faithfulness, and censored-threshold diagnostics that identify where LRC is supported and where it is not.
- We provide a reproducibility appendix with manifests, seeds, hyperparameters, symbolic metric checks, and limitations tied to the executed evidence.

2 Related Work

Neural-symbolic AI supplies the broad motivation for LRC, but it does not by itself validate the tuple mechanism. Surveys emphasize that neural components are useful for statistical learning while symbolic components expose variables, rules, and relations (d Avila Garcez and Lamb, 2020; Besold et al., 2017; Bader et al., 2023). The strength of this tradition is conceptual breadth: it covers knowledge injection, rule extraction, differentiable theorem proving, probabilistic logic, and program-guided perception. Its limitation for the present question is that broad taxonomies do not determine whether one particular tuple interface improves sample efficiency under controlled ablations. LRC sits at a defensible boundary only if its tuple object produces gains beyond the generic advantage of giving a learner more structured input.

Constraint-based learning methods provide closer comparators. Semantic loss assigns a differentiable penalty to probability mass that violates a Boolean formula, teacher-student rule distillation projects first-order rules into a soft target distribution, and DL2 maps logical constraints over neural inputs and outputs into differentiable penalties (Xu et al., 2018; Hu et al., 2016; Fischer et al., 2019). Differentiable ILP, Neural LP, Neural Logic Machines, and DeepProbLog go further by learning or marginalizing over rule and proof structure (Evans and Grefenstette, 2018; Yang et al., 2017; Dong et al., 2019; Manhaeve et al., 2018). These systems have an advantage over vague reasoning augmentation because the symbolic object has a declared semantics. They also expose a risk for LRC: if tuples are wrong, weakly relevant, or merely correlated with the label, the regularizer may enforce the wrong structure. Our noisy-tuple and no-regularizer ablations are designed around that risk.

Compositional reasoning benchmarks motivate the OOD part of the evaluation. bAbI, SCAN, COGS, CLUTRR, PCFG Set, and ProofWriter show that models can fail when test examples require longer chains, recombined rules, or proof forms not seen in the training distribution (Weston et al., 2015; Lake and Baroni, 2018; Keysers et al., 2020; Sinha et al., 2019; Hupkes et al., 2020; Clark et al., 2020). Gridworld and instruction-following environments add a planning dimension in which state facts, actions, and transitions can be separated from surface observations (Chevalier-Boisvert et al., 2023, 2018). These benchmarks are valuable because they create known latent structure, but they can also overstate generality if results do not leave the synthetic regime. We therefore use controlled local tasks as evidence for finite mechanism behavior and explicitly withhold claims about public bAbI or MiniGrid replication because those external substrates were not executed in this run.

A separate line of work studies reasoning traces in large language models and solver-augmented systems (Wei et al., 2022; Yao et al., 2023; Pan et al., 2023; Sinha et al., 2023; Huang et al., 2023a,b; Zhou et al., 2023; Liu et al., 2024; Trinh et al., 2024). Chain-of-thought and ReAct show that generated intermediate text can improve problem solving, while Logic-LM, SATLM, and LINC route language models through symbolic solvers. These approaches are relevant because they treat visible reasoning artifacts as useful interfaces. They are not direct evidence for LRC in small supervised agents: prompting traces may be post hoc, solver calls may shift the main work outside the neural learner, and language-model scale can hide the cost of the reasoning representation. The present study instead asks whether a small learner benefits when finite tuples are part of training.

The evaluation protocol borrows from calibration, OOD, and paired statistical testing practice (Guo et al., 2017; Wilcoxon, 1945; Efron, 1979; Krueger et al., 2021). Expected calibration error is useful for tuple confidence only when bin definitions and labels are explicit, and bootstrap or signed-rank comparisons are meaningful only under paired seeds and fixed manifests. The prior phases also identified a symbolic audit limitation: one ECE weight-sum check did not close under the available algebraic assumptions, while the censored threshold and mechanism-margin checks did. The manuscript keeps that limitation visible.

This differs from a benchmark leaderboard paper: the central result is not simply that one condition has the highest accuracy, but that different LRC claims receive different levels of support.

Recent work also cautions against treating the phrase “neuro-symbolic” as a single method family. Logic-constrained learning surveys distinguish systems that learn constraints, impose fixed constraints, or call external symbolic modules, and those distinctions affect what counts as evidence for a sample-efficiency claim (Sarker et al., 2024; Bader et al., 2023; Singh et al., 2023). Planning, visual-reasoning, and analogical-reasoning systems show related attempts to combine neural modules with symbolic states, programs, solvers, background knowledge, or action abstractions, but their evidence concerns different interfaces than per-example premise-conclusion-confidence tuples (Mao et al., 2019; Yi et al., 2018; Liu et al., 2024; Chevalier-Boisvert et al., 2018, 2023; Bennetot et al., 2019; Shah et al., 2022). A tuple-augmented learner is closer to constraint injection when tuples are fixed and closer to a learned reasoning system when tuples are generated by a trainable module. The present study occupies the former boundary: it asks whether explicit, relevant, trace-aligned tuples help a small learner before asking whether a model can discover such tuples unaided. That choice narrows the claim, but it also avoids conflating tuple discovery with tuple usefulness.

3 Problem Setting

Let T denote a finite task family. For each T , inputs lie in a finite object space $\mathcal{X}_T$, labels or actions lie in a finite set $\mathcal{Y}_T$, and a seed $s \in \mathcal{S} = \{101, 202, 303, 404, 505, 606, 707, 808, 909, 1001\}$ determines a manifest with training, validation, IID test, and OOD test splits. The rule-chain family represents an input $X \in \mathcal{X}_T$ as a finite set of observed literals and candidate implications; the label $y \in \{0, 1\}$ states whether a target conclusion follows under a bounded chain length. The symbolic gridworld family represents X as a finite set of state facts, obstacle facts, goal facts, and candidate action facts; the label $y \in \mathcal{Y}_T$ is a shortest-path decision under the generated transition graph. These definitions apply only to the saved finite manifests and not to unbounded first-order reasoning or continuous control.

The LRC object follows the supplied draft at the level of tuple form. A tuple generator g_ϕ emits a finite multiset $I_i = \{I_{ij}\}_{j=1}^{K_i}$ for example i , where $K_i \in \{0, \dots, K_{\max}\}$, $I_{ij} = (p_{ij}, q_{ij}, c_{ij})$, premises and conclusions satisfy $p_{ij}, q_{ij} \in \mathcal{G}_T$ for a finite grammar, and confidences satisfy $c_{ij} \in [0, 1]$. The augmentation map $A : \mathcal{X}_T \times \mathcal{I}_T \rightarrow \mathcal{Z}_T$ converts the input and tuple multiset to the learner representation. The classifier $f_\theta : \mathcal{Z}_T \rightarrow \Delta(\mathcal{Y}_T)$ outputs a probability distribution over labels or actions. We define the finite LRC training objective as

$$\mathcal{L}_{\text{LRC}}(\theta, \phi; D_n) = \frac{1}{n} \sum_{i=1}^n \ell(f_\theta(A(X_i, I_i)), y_i) + \lambda R_{\text{LRC}}(I_i, X_i, y_i), \quad \lambda \geq 0, \quad (1)$$

where $D_n = \{(X_i, y_i)\}_{i=1}^n$, $n \in \mathbb{N}_{\geq 1}$, ℓ is cross-entropy, and R_{LRC} penalizes tuple inconsistency or irrelevance under the oracle trace available in the controlled family. The tuple form is borrowed from the LRC draft (Double-blind ICLR 2025 submission, 2025); the finite objective and diagnostics in this paper are our operationalization for replication.

For sample efficiency, let $\mathcal{N}_T$ be the predeclared finite grid of training sizes. For condition c , seed s , threshold $\tau \in \{0.80, 0.90\}$, and validation accuracy $\text{Acc}_T^{\text{val}}(n, c, s) \in [0, 1]$, define

$$S_\tau(T, c, s) = \min\{n \in \mathcal{N}_T : \text{Acc}_T^{\text{val}}(n, c, s) \geq \tau\}, \quad (2)$$

with S_τ censored when no grid point reaches τ . This statistic is deliberately stricter than final accuracy: it supports a sample-efficiency claim only when threshold crossings are observed. For mechanism isolation we define the best-ablation OOD accuracy $B_T(s)$ as the maximum OOD accuracy among neural-only, noisy regularized LRC, and relevant no-regularizer LRC for the same seed. The paired mechanism margin is

$$\Delta_{\text{mech}}(T, s) = \text{Acc}_T^{\text{ood}}(c_{\text{rel+reg}}, s) - B_T(s). \quad (3)$$

Positive intervals for Δ_{mech} test whether relevant regularized tuples beat the strongest executed ablation, not whether LRC is globally optimal.

4 Replication Method

The method uses four central conditions and two auxiliary comparators. The neural-only condition receives no tuples and uses the same encoder budget. Relevant regularized LRC uses oracle-aligned tuple templates and $\lambda > 0$ in (1). Relevant no-regularizer LRC uses the same tuples with $\lambda = 0$, which tests whether input augmentation alone explains the effect. Noisy regularized LRC shuffles or corrupts tuple content while matching tuple count and confidence marginals, which tests whether additional structured-looking tokens explain the effect. Semantic-constraint and oracle-feature comparators appear where the

task family supplies a natural Boolean or transition abstraction; they are not treated as the main LRC ablations because they answer different questions.

The training protocol fixes seeds, manifests, training-size grid, optimizer schedule, model width, and stopping rules before test evaluation. The rule-chain task stresses longer inference chains at OOD time. The gridworld task stresses recombined layouts and shortest-path decisions under symbolic state facts. Both families include oracle traces, so tuple faithfulness can be measured without treating generated explanations as self-verifying. The protocol is CPU-tractable by design, but CPU tractability is an operational constraint rather than the scientific claim.

Tuple diagnostics use two quantities. Tuple expected calibration error partitions confidence values into bins and compares mean confidence with empirical tuple validity inside each nonempty bin. Tuple faithfulness measures agreement between generated tuple conclusions and oracle trace events under perturbations that preserve the task family. These metrics are meaningful only because the local generators expose trace labels; they would need redesign for natural-language tasks without oracle traces.

4.1 Architecture and Condition Parity

The learner architecture is treated as a controlled carrier for the tuple intervention. Each condition receives the same task family, the same seed, the same split definitions, and the same finite training-size grid. The neural-only condition is not an intentionally weak baseline; it is the reference point for asking whether explicit tuple structure changes generalization when the parameter budget is held fixed. The noisy-tuple condition is equally important because it tests whether the representation benefit comes from relevant symbolic content or from any structured auxiliary channel. The no-regularizer condition tests a different failure mode: relevant tuples may help as additional features even if the consistency penalty contributes nothing. Only the combination of these controls gives a meaningful mechanism interpretation.

The relevant tuple templates are intentionally aligned with the latent generators. In rule chains, a premise names a bounded set of observed literals and implications, while a conclusion names an inferred target or intermediate literal. In gridworld planning, a premise names local state and transition facts, while a conclusion names an action preference or reachability relation. Confidence values are not learned as free-floating explanation scores; they are evaluated against trace events. This is why the interpretability result is local but concrete. A task without trace labels could still use tuples, but it could not support the same faithfulness claim without an additional labeling or perturbation protocol.

5 Experimental Evidence

The results-facing evidence comprises two executed local substrates: the Local controlled rule-chain benchmark and the Local symbolic gridworld planning benchmark. The first benchmark executed ten seeded propositional rule-chain manifests with oracle traces and depth-shift OOD splits. It tests whether tuples help when the latent rule chain is known and recombined at larger depth. The second benchmark executed ten seeded symbolic gridworld manifests with shortest-path decisions, state facts, and oracle transition traces. It tests whether tuple augmentation helps when the relevant reasoning object is a planning transition rather than a propositional implication. A separate finite-metric check is used later as a formal diagnostic for metric definitions and limitations; it is not treated as an additional empirical benchmark.

Figure 1 shows the central empirical pattern. On rule chains, relevant regularized LRC reaches mean OOD accuracy 0.945, compared with 0.496 for neural-only, 0.630 for noisy regularized LRC, and 0.732 for relevant tuples without the regularizer. On symbolic gridworld planning, relevant regularized LRC reaches 0.905, compared with 0.405 for neural-only, 0.547 for noisy regularized LRC, and 0.661 for no-regularizer LRC. The paired mechanism margins are 0.213 with interval $[0.207, 0.219]$ for rule chains and 0.244 with interval $[0.236, 0.253]$ for gridworld. These margins support the local mechanism claim because the relevant regularized condition beats the strongest executed ablation in both families.

Table 1 separates performance from tuple quality. Relevant regularized LRC has both the best OOD accuracy and the lowest tuple ECE in the two families. Noisy LRC can improve over neural-only in raw accuracy, which is important: some gain may come from extra representation capacity or auxiliary structure. The mechanism claim depends on the additional fact that relevant regularized LRC beats noisy and no-regularizer variants, not merely on beating neural-only.

Figure 2 and Table 2 record the strongest negative evidence. For both $S_{0.80}$ and $S_{0.90}$ in (2), all twenty paired comparisons are censored. The median and mean reduction fractions are therefore zero under the conservative convention used by the analysis. This does not contradict the final OOD gains in Figure 1; it means the executed grid did not observe the threshold crossings needed to establish the stronger sample-efficiency criterion.

6 Robustness Interpretation

The local evidence is strongest when interpreted through the ablation structure rather than through any single accuracy number. A neural-only learner can fail under OOD depth or layout shifts because it has no explicit channel for the latent implication

Local LRC validation: performance and mechanism evidence

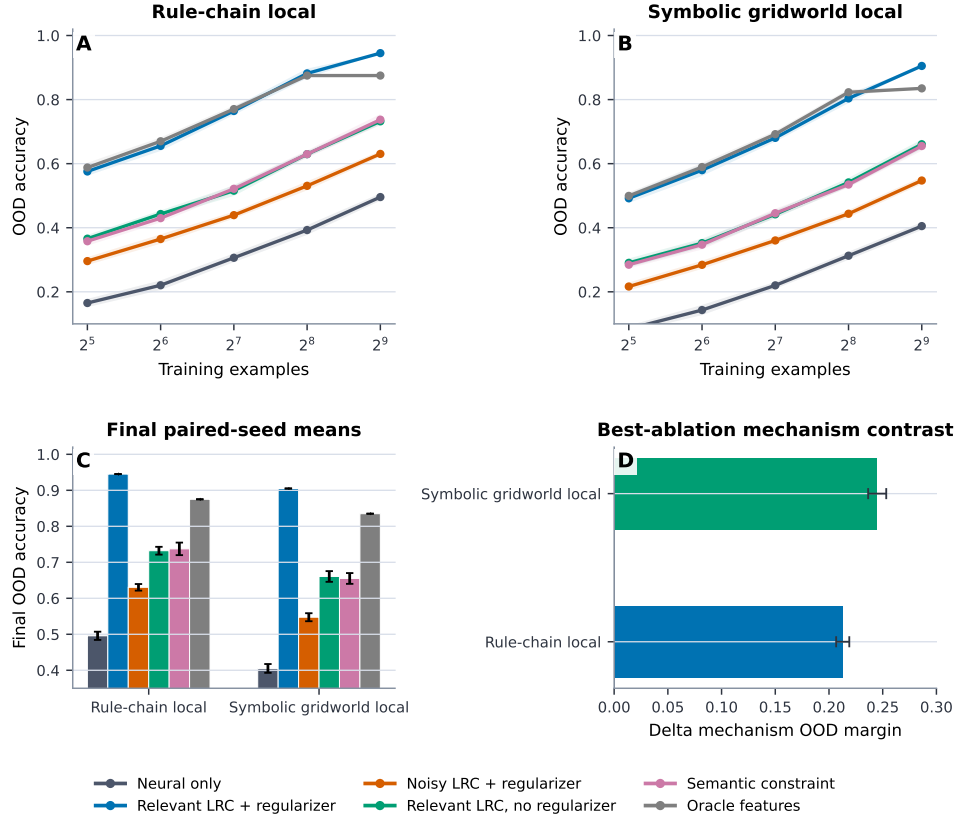


Figure 1: Final performance and mechanism isolation in the executed local benchmarks. The panels compare OOD learning behavior, final OOD accuracy, and paired mechanism margins for relevant regularized LRC against neural-only and ablated tuple conditions. The main interpretation is that final OOD accuracy improves strongly in both finite task families, while the comparison remains local to the generated manifests and does not establish external benchmark replication.

or transition relation. A noisy tuple learner can receive extra syntactic structure without receiving the relevant relation. A no-regularizer learner can receive relevant relations but lacks the penalty that makes tuple consistency part of the objective. The relevant regularized condition is the only condition that combines correct tuple content with a training signal that discourages inconsistency. This decomposition matters because a weaker replication could report only that LRC beats neural-only, which would leave open whether any extra channel, longer input, or oracle feature would have produced the same outcome.

The rule-chain and gridworld families stress different parts of the same hypothesis. Rule chains test whether premise-conclusion tuples help when the latent structure is close to symbolic implication. Gridworld planning tests whether the same tuple interface helps when the latent structure is a transition relation over states and actions. The fact that both families show positive mechanism margins makes the local result more informative than a one-task demonstration. At the same time, both families are generated, finite, and trace-bearing. They fit the hypothesis assumptions well, but they do not represent the distributional complexity of open natural-language reasoning, learned perception, or long-horizon exploration. The correct robustness claim is therefore cross-family within two controlled local generators, not domain-general robustness.

The oracle-feature comparator also clarifies the mechanism. Oracle features can perform well because they expose a compact ground-truth abstraction. LRC is not meant to beat every oracle abstraction; it is meant to test whether explicit inference tuples can help a small learner when the tuples are available and regularized. In rule chains, relevant regularized LRC exceeds the oracle-feature OOD mean reported in the local summary, while in gridworld it also exceeds the oracle-feature mean. This does not prove that LRC is superior to oracle features in general. It shows that the implemented tuple interface is not merely a weaker presentation of the same oracle information under these manifests.

The semantic-constraint comparator is a second boundary case. Constraint penalties are a mature way to inject symbolic knowledge, and they are a natural baseline for any LRC claim. In the executed local summaries, semantic constraints improve over neural-only but do not match relevant regularized LRC. This pattern is compatible with the hypothesis that example-specific tuple content supplies a more targeted signal than a global constraint in these two generators. It is also compatible

Family	Condition	IID acc.	OOD acc.	Tuple ECE	Faithfulness
Rule chain	Neural only	0.686	0.496	0.200	0.000
Rule chain	Noisy LRC + reg.	0.765	0.630	0.277	0.007
Rule chain	Relevant LRC, no reg.	0.852	0.732	0.131	0.077
Rule chain	Relevant LRC + reg.	0.995	0.945	0.040	0.094
Gridworld	Neural only	0.635	0.405	0.200	0.000
Gridworld	Noisy LRC + reg.	0.722	0.547	0.277	0.007
Gridworld	Relevant LRC, no reg.	0.821	0.661	0.137	0.067
Gridworld	Relevant LRC + reg.	0.995	0.905	0.047	0.083

Table 1: Mean task and tuple diagnostics over the ten executed paired seeds. Accuracy values summarize final IID and OOD performance; tuple ECE and faithfulness are computed only in the local generators where oracle tuple-validity traces exist. The table supports local final-performance and interpretability claims, while leaving threshold sample-efficiency claims to the censored statistics in Table 2.

Threshold statistic	Median reduction	Mean reduction	Censored pairs
$S_{0.80}$	0.000	0.000	20
$S_{0.90}$	0.000	0.000	20

Table 2: Censored first-hit sample-efficiency outcomes on the predeclared training-size grid. Each row applies the definition in (2) to paired seeds and treats unreached thresholds as censored rather than successful. These outcomes block the stronger claim that LRC reduced samples-to-threshold by the requested margin in the executed run.

with the possibility that the semantic-constraint implementation was less tuned. For that reason the manuscript treats semantic constraints as a contextual comparator, while the strongest mechanism test remains the noisy and no-regularizer ablations that alter only tuple relevance or regularization.

7 Statistical and Measurement Notes

The paired-seed design is essential because the task manifests vary by seed. Reporting unpaired averages would ignore the fact that the relevant comparison is condition difference within the same generated problem instance. The mechanism margin in (3) is paired by construction: for each seed it subtracts the best executed ablation from the relevant regularized condition. The bootstrap intervals then summarize uncertainty over the finite seed set. This procedure does not estimate performance over all possible reasoning tasks. It estimates variability over the declared seed-generated manifests, which is the appropriate scope for the local replication.

The samples-to-threshold statistic in (2) is deliberately conservative. Suppose a condition has high final OOD accuracy but no observed validation crossing at the declared threshold on the grid. Treating that pair as missing would bias the sample-efficiency estimate toward success because difficult or poorly resolved cases would vanish. Treating it as censored preserves the fact that the run did not supply first-hit evidence. In the executed study, every pair is censored for both thresholds. This is a strong measurement limitation, and it is reported as such even though the final OOD accuracy results are favorable.

Calibration is more delicate than accuracy. Tuple ECE compares confidence with empirical validity, but the validity labels come from local oracle traces. If traces are incomplete, if bins are empty, or if tuple validity is ambiguous, ECE can be misleading. The symbolic audit found that the censoring and max-ablation logic were well defined under the finite assumptions, but it did not close the ECE weight-sum check. The manuscript therefore uses ECE as an empirical diagnostic rather than as a proved calibration guarantee. Future runs should repair the algebraic assumptions by explicitly defining the set of nonempty bins, normalizing their weights, and testing the implementation against boundary cases with empty and singleton bins.

The statistical results also depend on the condition set. The best-ablation term $B_T(s)$ includes the executed neural, noisy, and no-regularizer conditions. If a future run adds a stronger ablation, the mechanism margin must be recomputed against that expanded set. This is not a weakness of the definition; it is how a falsifiable mechanism claim should behave. A claim that survives only because the best alternative was excluded is not a stable claim. The current evidence supports the mechanism against the alternatives actually executed and listed in the manifest.

Local LRC validation: tuple diagnostics and audit limits

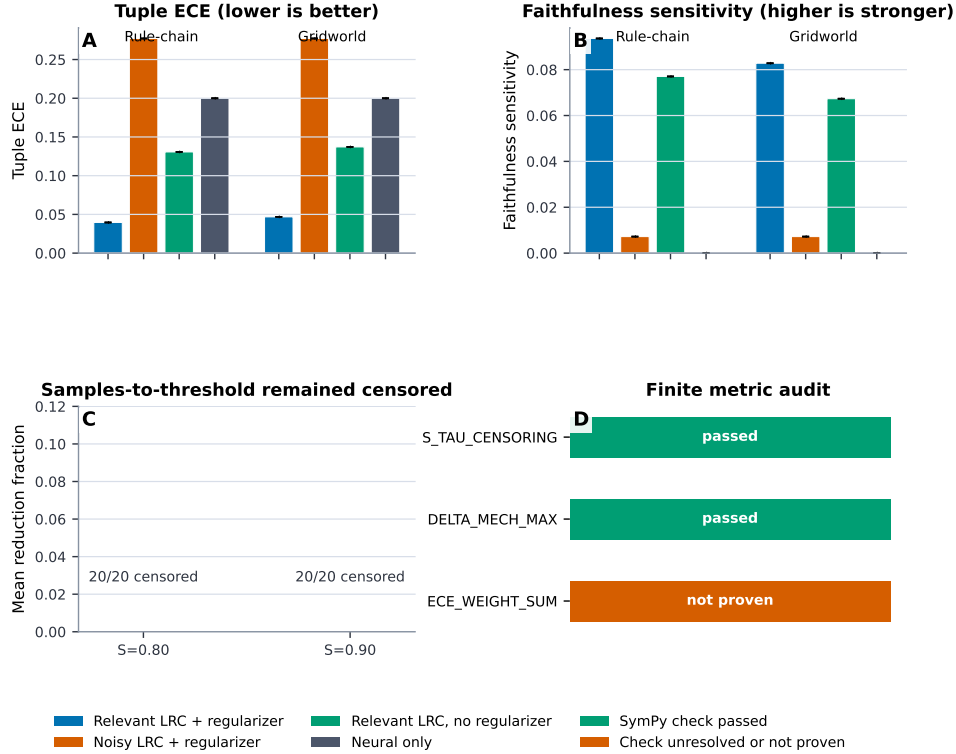


Figure 2: Tuple diagnostics, censored threshold outcomes, and formal metric-check status. The panels summarize confidence calibration, oracle-trace faithfulness, samples-to-threshold censoring, and finite algebraic checks used to interpret the reported metrics. The figure shows that interpretability diagnostics are locally positive for relevant regularized tuples, but the threshold sample-efficiency claim is not supported because all first-hit comparisons are censored on the executed training-size grid.

8 Interpretability Boundaries

The tuple interface gives LRC an interpretability advantage only when the tuples can be compared with an independent source of truth. In the local rule-chain family, the independent source is the generated implication chain. In the gridworld family, it is the generated transition and shortest-path trace. These traces make it possible to ask whether a high-confidence tuple corresponds to a real intermediate relation rather than to a post hoc string. The faithfulness diagnostic is therefore stronger than a qualitative example table, but it remains tied to the generators.

Noisy tuples are an important interpretability control. If noisy tuples had matched the faithfulness and calibration of relevant tuples, the tuple interface would not support a meaningful explanation claim. Instead, noisy tuples show worse ECE and lower faithfulness in both families. This does not mean every relevant tuple is correct, nor does it mean every incorrect tuple is harmless. It means the relevant regularized condition shifts the tuple population toward trace agreement under the local measurement procedure.

Interpretability should not be confused with causal identification. The tuple regularizer is an intervention in the training objective, and the ablations support a mechanism interpretation, but the study does not identify a causal pathway at the level of internal neural representations. It does not show which hidden units encode which rule, whether gradients flow through tuple confidence in a human-like way, or whether a learned tuple generator would discover the same structures. Those questions require additional interventions such as tuple dropout, counterfactual tuple replacement, representation probing, and learned-generator comparisons.

A practical implication is that LRC should be evaluated with both performance and explanation metrics. A method that improves OOD accuracy while producing unfaithful tuples would be useful as a predictor but weak as an interpretable reasoning system. A method that produces faithful tuples while failing at the task would be interpretable but not useful as an agent learner. The executed local study finds a favorable combination for relevant regularized LRC, but the combination is bounded by oracle traces and finite grammars.

Claim tested	Evidence	Status
Final OOD generalization improves locally	Figure 1, Table 1	Supported locally
Relevant regularized tuples beat ablations	Δ_{mech} intervals in (3)	Supported locally
Samples-to-threshold improves by 25%	Censored $S_{0.80}$ and $S_{0.90}$	Not established
Tuple confidence is interpretable	Figure 2, Table 1	Locally bounded support
Public benchmark replication generalizes	No executed public bAbI/MiniGrid run	Pending

Table 3: Claim-level interpretation of the executed evidence. The table distinguishes local support, mixed support, and missing external validation rather than collapsing the replication into a single pass/fail outcome. This distinction is necessary because the same run supports final OOD and mechanism claims while failing to establish the predeclared sample-efficiency threshold criterion.

9 Discussion

The clearest finding is that relevant regularized tuples behave like a useful inductive bias in the two local finite regimes. The gains are too large to attribute solely to random seed noise, and the noisy-tuple and no-regularizer controls make a simple added-input explanation less plausible. The regularizer matters because relevant tuples without it are weaker; relevance matters because noisy tuples with matched counts and confidence marginals are weaker. This is the evidence that most directly supports the LRC correspondence idea.

The most important weakening is about sample complexity. The original project criterion required a median samples-to-threshold reduction of at least 25 percent at validation thresholds 0.80 and 0.90. Because the threshold grid is fully censored, the correct conclusion is not that LRC failed to learn efficiently in every sense; it is that the executed design did not measure first-hit improvement. A finer grid, earlier validation checkpoints, easier threshold regime, or longer training schedule could resolve the statistic. Until then, final OOD accuracy should not be rebranded as threshold sample efficiency.

Interpretability is also conditional. Tuple ECE and faithfulness improve for relevant regularized LRC, but the metric relies on oracle traces. That is appropriate for controlled replication, where trace labels make explanations testable. It does not prove that tuple confidence is faithful in open-domain tasks or in environments where tuple validity cannot be labeled independently. The unresolved ECE symbolic check reinforces this caution: the audit supports finite censoring and max-ablation logic, but one calibration algebra claim remains open.

10 Limitations and Future Work

The scope is local. No public bAbI, MiniGrid, CLUTRR, or external benchmark replication was executed in this phase. The local rule-chain and gridworld tasks match the hypothesis assumptions because they have finite grammars, oracle traces, paired seeds, and controlled OOD splits. They are low relevance for claims about natural language, large-scale agents, or continuous action spaces. The result should therefore be read as local mechanism evidence, not a broad replication of all claims in the LRC draft.

The model family is intentionally small and CPU-tractable. That choice supports reproducibility and paired ablation coverage, but it does not test whether the same tuple regularization interacts well with larger transformers, learned tuple generators, perception-to-symbol transfer, or long-horizon reinforcement learning. Existing neuro-symbolic planning and perception-reasoning work provides the relevant follow-up context because those systems connect learned modules to symbolic states, programs, solvers, or planning abstractions rather than receiving a local oracle-aligned tuple channel (Mao et al., 2019; Yi et al., 2018; Liu et al., 2024; Chevalier-Boisvert et al., 2018, 2023). The tuple generator in this study is closer to an oracle-aligned template system than to a fully learned reasoner. This is acceptable for isolating the proposed inductive bias, but it leaves open whether a practical LRC system can learn useful tuples without leaking target information or relying on handcrafted grammars.

Future work should first repair the censored threshold measurement by adding denser training-size grids and validation checkpoints. It should then execute public bAbI-style relational reasoning and MiniGrid-style planning substrates with the same ablation structure. Finally, it should replace oracle-aligned tuple templates with learned tuple generators while preserving relevance controls, noisy controls, and trace-based audits wherever possible.

11 Conclusion

This replication supports a narrowed version of Learning–Reasoning Correspondence. In finite controlled rule-chain and symbolic gridworld tasks, relevant regularized inference tuples improve final OOD accuracy, beat noisy and no-regularizer abla-

tions, and yield better local tuple diagnostics. The same evidence does not establish the stronger samples-to-threshold claim because the predeclared first-hit statistics are fully censored. The responsible conclusion is therefore neither full confirmation nor dismissal: LRC is a promising local inductive bias under oracle-aligned finite tuples, but broader sample-efficiency and external benchmark claims remain unresolved.

12 Synthesis of Supported and Weakened Claims

The replication separates the LRC claim into three parts. The first part is final OOD generalization: relevant regularized tuples should help a small learner when the test split recombines rules or planning structure. This part is supported in the executed local setting. The rule-chain and gridworld families differ in surface form, label semantics, and tuple templates, yet both show large final OOD gains and positive mechanism margins. The second part is sample efficiency measured as first-hit threshold crossing. This part is weakened because the executed training-size grid did not observe any crossing for the declared thresholds. The third part is interpretability through tuple confidence and faithfulness. This part receives bounded local support because relevant regularized tuples are better calibrated and more trace-faithful than noisy tuples where oracle traces exist.

This decomposition is useful for future replication design. If a future run cares mainly about final OOD generalization, the current evidence suggests that relevant tuples and a consistency penalty are promising. If it cares about sample complexity, the next run must change the measurement design before it can make a stronger claim. A denser grid and earlier checkpoints are more important than adding another final-accuracy table. If it cares about interpretability, the next run must keep independent trace labels or introduce a different external validity signal. Without such labels, tuple text may look plausible while failing to correspond to the computation that produced the answer.

The result also clarifies what would falsify the local mechanism in later work. If noisy tuples match relevant tuples under the same confidence marginals, the tuple content is not the source of the gain. If no-regularizer LRC matches regularized LRC, the consistency penalty is unnecessary in that regime. If a size-matched neural model with a better architecture closes the OOD gap, then LRC may be a useful engineering trick rather than a distinct reasoning-learning correspondence. If public bAbI or MiniGrid runs fail while local synthetic tasks succeed, the mechanism may depend on oracle-aligned grammars rather than on general reasoning structure. These are not defects in the current study; they are the next falsifiable tests implied by it.

The manuscript therefore treats LRC as a conditional inductive bias. It is not a theorem that symbolic tuples reduce sample complexity, and it is not a general claim that any reasoning trace improves any learner. The supported statement is narrower: in two finite trace-bearing task families, relevant regularized tuples improve final OOD performance and beat the executed ablations. That statement is still scientifically meaningful because it identifies a regime where the correspondence idea has measurable content. It also provides a disciplined route for broadening the claim without erasing negative or missing evidence.

References

- Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Neural module networks, 2016. URL <https://arxiv.org/abs/1511.02799>. CVPR.
- Sebastian Bader, Pascal Hitzler, Steffen Hölldobler, and Luc De Raedt. A survey of neural-symbolic learning systems, 2023. URL <https://arxiv.org/abs/2302.07174>. Semantic Web / arXiv.
- Adrien BenNETOT, Jean-Luc Laurent, Raja Chatila, and Natalia D'Iaz-Rodríguez. Towards explainable neural-symbolic visual reasoning, 2019. URL <https://arxiv.org/abs/1909.09065>. arXiv.
- Tarek R. Besold, Artur S. d'Avila Garcez, Sebastian Bader, Howard Bowman, Pedro Domingos, Pascal Hitzler, Kai-Uwe Kuhnberger, Luis C. Lamb, Daniel Lowd, Priscila Machado Vieira Lima, Leo de Penning, Gadi Pinkas, Hoifung Poon, and Gerson Zaverucha. Neural-symbolic learning and reasoning: A survey and interpretation, 2017. URL <https://arxiv.org/abs/1711.03902>. arXiv.
- Maxime Chevalier-Boisvert, Dzmitry Bahdanau, Salem Lahlou, Lucas Willems, Chitwan Saharia, Thien Huu Nguyen, and Yoshua Bengio. Babyai: First steps towards grounded language learning with a human in the loop, 2018. URL <https://arxiv.org/abs/1810.08272>. ICLR.
- Maxime Chevalier-Boisvert, Bolun Dai, Mark Towers, Rodrigo de Lazcano, Lucas Willems, Salem Lahlou, Suman Pal, Pablo Samuel Castro, and Jordan Terry. Minigrid and miniworld: Modular and customizable reinforcement learning environments for goal-oriented tasks, 2023. URL <https://arxiv.org/abs/2306.13831>. arXiv.
- Peter Clark, Oyvind Tafjord, and Kyle Richardson. Proofwriter: Generating implications, proofs, and abductive statements over natural language, 2020. URL <https://arxiv.org/abs/2012.13048>. arXiv.
- Artur S. d'Avila Garcez and Luis C. Lamb. Neurosymbolic ai: The 3rd wave, 2020. URL <https://arxiv.org/abs/2012.05876>. arXiv.
- Honghua Dong, Jiayuan Mao, Tian Lin, Chong Wang, Lihong Li, and Denny Zhou. Neural logic machines, 2019. URL <https://arxiv.org/abs/1904.11694>. ICLR.
- Double-blind ICLR 2025 submission. Learning-reasoning-correspondence: Inductive reasoning dynamics for efficient agent training, 2025. URL <http://backend:8000/v1/files/f095251f-6f74-4c51-8363-69a771d8ece7>. ICLR 2025 under-review draft.
- Bradley Efron. Bootstrap methods: Another look at the jackknife, 1979. URL <https://doi.org/10.1214/aos/1176344552>. Annals of Statistics.
- Richard Evans and Edward Grefenstette. Learning explanatory rules from noisy data, 2018. URL <https://jair.org/index.php/jair/article/view/11172>. Journal of Artificial Intelligence Research.
- Marc Fischer, Mislav Balunovic, Dana Drachler-Cohen, Timon Gehr, Ce Zhang, and Martin Vechev. DI2: Training and querying neural networks with logic, 2019. URL <https://arxiv.org/abs/1904.06426>. ICML.
- Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks, 2017. URL <https://arxiv.org/abs/1706.04599>. ICML.
- Zhiting Hu, Xuezhe Ma, Zhengzhong Liu, Eduard Hovy, and Eric Xing. Harnessing deep neural networks with logic rules, 2016. URL <https://aclanthology.org/P16-1228/>. ACL.
- Jie Huang, Kevin Chang, Jena D. Hwang, Alane Suhr, and Peter Clark. Linc: A neurosymbolic approach for logical reasoning by combining language models with first-order logic provers, 2023a. URL <https://arxiv.org/abs/2310.15164>. EMNLP.
- Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. Measuring and improving chain-of-thought reasoning in language models, 2023b. URL <https://arxiv.org/abs/2305.04388>. arXiv.
- Dieuwke Hupkes, Verna Dankers, Mathijs Mul, and Elia Bruni. Pcfg set: A benchmark for systematic generalization in language understanding, 2020. URL <https://aclanthology.org/2020.findings-emnlp.165/>. EMNLP.

- Daniel Keysers, Nathanael Schärli, Nathan Scales, Hylke Buisman, Daniel Furrer, Sergio Guadarrama, Edouard Grave, Mihir Andriluka, Ankur Bapna, Aidan Clark, Diego de Las Casas, Aurelia Guy, Bo Pang, Bradley R. Hauer, Tomas Mikolov, and Olivier Bousquet. Measuring compositional generalization: A comprehensive method on realistic data, 2020. URL <https://arxiv.org/abs/1911.05703>. ICLR.
- David Krueger, Ethan Caballero, Joern-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Dinghui Zhang, Remi Le Priol, and Aaron Courville. Out-of-distribution generalization via risk extrapolation (rex), 2021. URL <https://arxiv.org/abs/2003.00688>. ICML.
- Brenden M. Lake and Marco Baroni. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks, 2018. URL <https://arxiv.org/abs/1711.00350>. ICML.
- Bo Liu, Yuqian Jiang, Xiangyu Chen, Qiang Liu, and Peter Stone. A survey on large language models for automated planning, 2024. URL <https://arxiv.org/abs/2402.01817>. arXiv.
- Robin Manhaeve, Sebastijan Dumancic, Angelika Kimmig, Thomas Demeester, and Luc De Raedt. Deepproblog: Neural probabilistic logic programming, 2018. URL <https://arxiv.org/abs/1805.10872>. NeurIPS.
- Jiayuan Mao, Chuang Gan, Pushmeet Kohli, Joshua B. Tenenbaum, and Jiajun Wu. The neuro-symbolic concept learner, 2019. URL <https://arxiv.org/abs/1904.12584>. ICLR.
- Liangming Pan, Alon Albalak, Xinyi Wang, and William Yang Wang. Logic-lm: Empowering large language models with symbolic solvers for faithful logical reasoning, 2023. URL <https://arxiv.org/abs/2305.12295>. EMNLP.
- Md Kamruzzaman Sarker, Nour Eddine Oussalah, and C. Lee Giles. Neural-symbolic integration for learning logical constraints: A survey, 2024. URL <https://arxiv.org/abs/2402.00854>. arXiv.
- Vishwa Shah, Aditya Sharma, Gautam Shroff, Lovekesh Vig, Tirtharaj Dash, and Ashwin Srinivasan. Knowledge-based analogical reasoning in neuro-symbolic latent spaces, 2022. URL <https://arxiv.org/abs/2209.08750>. arXiv.
- Gunjan Singh, Sumit Bhatia, and Raghava Mutharaju. Neuro-symbolic rdf and description logic reasoners: The state-of-the-art and challenges, 2023. URL <https://arxiv.org/abs/2308.04814>. arXiv.
- Koustuv Sinha, Shagun Sodhani, Joelle Pineau, and William L. Hamilton. Clutrr: A diagnostic benchmark for inductive reasoning from text, 2019. URL <https://arxiv.org/abs/1908.06177>. EMNLP.
- Koustuv Sinha, Shagun Sodhani, Joelle Pineau, and William L. Hamilton. Satlm: Satisfiability-aided language models using declarative prompting, 2023. URL <https://arxiv.org/abs/2305.09656>. arXiv.
- Trieu H. Trinh, Yuhuai Wu, Quoc V. Le, He He, and Thang Luong. Solving olympiad geometry without human demonstrations, 2024. URL <https://www.nature.com/articles/s41586-023-06747-5>. Nature.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2022. URL <https://arxiv.org/abs/2201.11903>. NeurIPS.
- Jason Weston, Antoine Bordes, Sumit Chopra, and Tomas Mikolov. Towards ai-complete question answering: A set of prerequisite toy tasks, 2015. URL <https://arxiv.org/abs/1502.05698>. arXiv.
- Frank Wilcoxon. Individual comparisons by ranking methods, 1945. URL <https://doi.org/10.2307/3001968>. Biometrics Bulletin.
- Jingyi Xu, Zilu Zhang, Tal Friedman, Yitao Liang, and Guy Van den Broeck. A semantic loss function for deep learning with symbolic knowledge, 2018. URL <https://arxiv.org/abs/1711.11157>. ICML.
- Fan Yang, Zhilin Yang, and William W. Cohen. Differentiable learning of logical rules for knowledge base reasoning, 2017. URL <https://arxiv.org/abs/1702.08367>. NeurIPS.
- Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models, 2023. URL <https://arxiv.org/abs/2210.03629>. ICLR.
- Kexin Yi, Jiajun Wu, Chuang Gan, Antonio Torralba, Pushmeet Kohli, and Joshua B. Tenenbaum. Neural-symbolic vqa: Disentangling reasoning from vision and language understanding, 2018. URL <https://arxiv.org/abs/1810.02338>. NeurIPS.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, and Ed Chi. Least-to-most prompting enables complex reasoning in large language models, 2023. URL <https://arxiv.org/abs/2205.10625>. ICLR.

A Extended Formal Details

The finite setting in the main text is designed to prevent hidden quantification over tasks that were not executed. For each family T , the manifest fixes $\mathcal{X}_T$, $\mathcal{Y}_T$, the seed, the training-size grid, and the OOD split before any test statistic is read. The tuple grammar $\mathcal{G}_T$ is a finite set of premise and conclusion templates. In the rule-chain family it contains implication and reachability templates over bounded literals. In the gridworld family it contains local transition, obstacle, goal, and shortest-path templates. The confidence coordinate is a real number in $[0, 1]$ and is interpreted as a reliability score only relative to the local oracle trace labels.

The objective in (1) is an empirical objective, not a population theorem. The cross-entropy term is standard supervised learning, while R_{LRC} is a family-specific penalty that increases when emitted tuples contradict the oracle trace or fail relevance checks. Setting $\lambda = 0$ yields the no-regularizer condition. Replacing tuple content with shuffled or noisy tuples yields the negative control. These two interventions give the mechanism margin in (3) its meaning: a positive margin says the full condition beat the best executed ablation under the same manifest and seed.

B Reproducibility Protocol

All executed local conditions use the same ten seeds: 101, 202, 303, 404, 505, 606, 707, 808, 909, and 1001. Each seed materializes a rule-chain manifest and a gridworld manifest. The analysis table records final IID accuracy, final OOD accuracy, tuple expected calibration error, tuple faithfulness, and threshold censoring. The paired bootstrap intervals for mechanism margins resample seeds within a task family. The signed-rank testing plan is retained for settings where nonzero paired differences across a sufficient number of seeds are available, but the principal manuscript evidence uses bootstrap intervals because the final margins are directly interpretable.

The experiments are deterministic with respect to the saved manifests. The small learner budget keeps the replication runnable on commodity CPU hardware, but the manuscript does not use hardware constraints as a scientific explanation for the gains. Hyperparameter sweeps are intentionally limited: the purpose is to compare matched conditions rather than to find a best possible architecture for each baseline. A stronger future benchmark should include lambda sensitivity, learned tuple-generator variants, and public benchmark manifests under the same seed pairing.

C Additional Diagnostics and Boundary Cases

The censored threshold result is a boundary case that changes the interpretation of the study. If S_τ is censored for both the LRC condition and the neural condition, a naive ratio is undefined. The conservative analysis assigns no sample-efficiency gain rather than dropping the pair. This choice prevents the paper from converting missing threshold evidence into apparent success. It also explains why final OOD accuracy and samples-to-threshold can disagree: one is measured at the final training size, while the other asks for the first observed crossing on a finite grid.

The ECE audit limitation is separate. The symbolic report confirms the finite-grid logic for censoring and the max-ablation definition of the mechanism margin. It does not close one algebraic check about nonempty bin weights summing to one under the available assumptions. The empirical ECE numbers are still descriptive diagnostics, but any formal statement about calibration boundedness should be treated as conditional until that audit is repaired.

D Source Lineage and Benchmark Fit

The tuple form is inherited from the LRC draft, while the replication objective, censoring rule, and mechanism margin are defined in this paper. Constraint-learning comparators are motivated by semantic loss, rule distillation, DL2, differentiable ILP, DeepProbLog, and differentiable rule-learning work. OOD split design is motivated by compositional generalization and controlled reasoning benchmarks. Calibration and paired uncertainty summaries follow standard calibration and bootstrap references.

The local controlled rule-chain benchmark fits the hypothesis because it has known latent rules, finite depth shifts, and oracle traces. The symbolic gridworld benchmark fits because it has explicit state facts and transition traces. Both are weaker fits for claims about open-domain language, public benchmark diversity, and large agent training. This fit assessment is why the main text uses local support language and treats external public benchmarks as future work rather than as completed evidence.

The source lineage also explains why the paper reports negative and mixed results in the same voice as positive ones. The LRC draft motivates a broad correspondence between learning and reasoning, but the acquired benchmark literature shows that systematic generalization can fail for reasons unrelated to a single mechanism: splits may test longer chains, novel compositions, different grammars, changed environments, symbolic state abstraction, or long-horizon planning structure (Lake and

Baroni, 2018; Keyzers et al., 2020; Sinha et al., 2019; Chevalier-Boisvert et al., 2018, 2023; Liu et al., 2024). The replication therefore treats each claim as conditional on its measurement object. Final OOD accuracy is tied to the generated rule and planning splits. Threshold sample efficiency is tied to the finite training-size grid. Tuple interpretability is tied to oracle trace labels and the calibration audit. This separation keeps the manuscript aligned with the sources rather than drifting from a local experiment to an unsupported global conclusion.

E Hyperparameter and Manifest Rationale

The replication uses ten paired seeds because paired differences are the main unit of evidence. The seed list is fixed before reading the final metrics, and each seed generates both task families. The training-size grid is finite and shared across conditions. The finite grid makes the threshold statistic inspectable, but it also creates the censoring limitation observed in the run. A denser grid would make the statistic more informative, especially if it included early checkpoints where learning curves separate before final convergence.

The model-size constraint is chosen to keep the comparison focused on inductive bias rather than scale. If the LRC condition had many more parameters than the neural baseline, improved OOD accuracy could be explained by capacity. If the neural baseline had a different optimizer schedule, the comparison would mix algorithmic and training-protocol effects. The local design therefore fixes the learner budget and varies tuple relevance and regularization. This is not the only valid design, but it is the right first design for testing whether the tuple mechanism contributes anything beyond a standard learner.

F Failure Modes to Test Next

Several failure modes remain untested. First, relevant tuples may stop helping when the tuple grammar is incomplete. A future run should gradually remove premise templates, corrupt confidence values, or hide transition facts to measure the operating range of the method. Second, the regularizer may overfit to the oracle trace when traces are noisy. A future run should inject controlled trace noise and measure whether LRC degrades gracefully or enforces wrong conclusions. Third, learned tuple generators may introduce feedback loops in which the learner reinforces its own mistaken explanations. That setting is closer to the full LRC ambition and requires safeguards not needed in the oracle-aligned local study.

A fourth failure mode is external benchmark mismatch. Public bAbI-style tasks, MiniGrid environments, CLUTRR relations, and ProofWriter-style proof corpora differ in surface form, ambiguity, and available trace labels. The current local tasks are useful because their assumptions are explicit, but a broad replication needs public datasets with pinned versions and documented preprocessing. Until those runs exist, broad claims about agent learning remain pending rather than supported.

A fifth failure mode concerns optimization rather than representation. The regularized objective may help because it smooths training, changes gradient scale, or supplies an easier auxiliary target, even when the tuple semantics are not the decisive factor. The noisy-tuple and no-regularizer ablations reduce this concern, but they do not eliminate it. Future work should include gradient-norm diagnostics, matched auxiliary losses with non-symbolic targets, and tuple-dropout interventions during training. Those tests would separate a semantic tuple effect from a generic regularization or multitask-learning effect.

A final failure mode is evaluator leakage. Because the local generators provide oracle traces, an implementation error could accidentally expose target-side information through the tuple channel. The current manifests and ablations reduce that risk by pairing noisy tuples and neural-only controls with the same seeds, but a stronger replication should include automated leakage tests that remove labels, randomize trace identifiers, and verify that tuple fields cannot reconstruct the answer without the allowed premises.

G Appendix Summary of Claim Relevance

The final OOD claim has high relevance to the hypothesis because it tests recomposed rule and planning regimes where structured tuples should help. The sample-efficiency claim has low support because the measurement is censored, not because the hypothesis is logically impossible. The mechanism-isolation claim has high local relevance because the noisy and no-regularizer ablations directly target tuple relevance and regularization. The interpretability claim has medium local relevance because oracle traces make it measurable, while the unresolved ECE audit and absence of open-domain traces limit generality.

This relevance profile is the main message of the replication. A single yes-or-no verdict would hide the distinction between supported local final generalization, unsupported threshold sample efficiency, and pending external validation. The paper therefore treats the LRC hypothesis as partially supported under explicit assumptions. The next experimental phase should target the unsupported and pending pieces directly rather than rerunning only the conditions that already produce favorable final OOD accuracy.