Structure Without Uncertainty: What Reinforcement Learning Learns (and Doesn't Learn) in Synthetic Language

Anonymous
Anonymous Institution
Anonymous Location

***Abstract*—We investigate what reinforcement learning can and cannot learn about language-like structure without likelihood-based supervision. Through controlled experiments on synthetic grammars with ambiguous tokens (multiple valid continuations), we discover a fundamental asymmetry: RL learns *structure*—which tokens can follow which—but fails to learn *uncertainty*—how probability mass should be distributed over valid options. Specifically, RL achieves oracle-level accuracy (80%) but collapses to near-deterministic outputs (entropy 0.009 vs. target 0.693), while cross-entropy training preserves the full distribution. Crucially, representation analysis reveals that despite this behavioral divergence, RL and MLE learn highly similar internal representations (CKA 0.66–0.85), with *highest* similarity on ambiguous tokens (CKA 0.83). This demonstrates that distributional collapse occurs in the output layer, not in learned features. Our results explain why likelihood-based pretraining exists: not to learn structure (RL can do that), but to learn calibrated uncertainty over that structure.**

***Index Terms*—reinforcement learning, language modeling, uncertainty, representation learning, distributional collapse**

I. Introduction

Language models are trained with likelihood-based objectives: cross-entropy loss that directly supervises the probability distribution over next tokens. This paradigm underlies all modern large language models [1]. But *why* is likelihood supervision necessary? What specifically does it provide that reinforcement learning cannot?

A recent line of work [2] demonstrated that when prediction is framed as action ("prediction-as-action"), RL gradients can match supervised learning for continuous state prediction. This raises a fundamental question: if RL can learn prediction, why do we need likelihood-based pretraining for language?

A. The Central Paradox

This question leads to a paradox that motivates our investigation. Reinforcement learning can achieve the *same task accuracy* as supervised likelihood training on language-like prediction tasks, yet produces *radically different output distributions*—collapsing uncertainty despite matching structural correctness. Understanding this discrepancy—why two methods that appear equivalent on accuracy metrics diverge so dramatically on distributional metrics—is the central goal of this paper.

B. Our Core Finding

We provide a precise answer through controlled experiments on synthetic grammars:

> *RL learns linguistic structure but not linguistic uncertainty. This failure is objective-level, not representational.*

Concretely, when tokens have multiple valid continuations (ambiguity), RL learns *which* tokens are valid but not *how likely* each should be. It collapses to deterministic outputs despite ambiguous targets.

C. Three Key Results

1) **Distributional collapse** (Section IV-A): On ambiguous grammars with 50/50 token splits, RL achieves 80% accuracy (matching oracle) but policy entropy collapses to 0.009 (target: 0.693). RL picks one valid option with 99.97% confidence instead of maintaining the distribution.
2) **Representational similarity** (Section IV-B): Despite behavioral divergence, RL and MLE learn similar internal representations. CKA similarity is 0.66–0.85 overall, and *highest* (0.83) on ambiguous tokens where behavior differs most. This is the representation paradox: maximal representational alignment occurs exactly where behavioral divergence is greatest.
3) **Structural parity** (Section IV-C): On deterministic tasks, RL matches MLE perfectly (100% accuracy). Distributional collapse does not impair structural learning—only uncertainty learning.

D. Why This Matters

Our results explain a fundamental aspect of language model training:

> *Likelihood-based pretraining exists not to learn structure, but to learn calibrated uncertainty over structure.*

This also explains why RLHF uses KL-regularization against pretrained models [6]: RL fine-tuning would otherwise collapse the learned distributions.

This paper shows that likelihood-based training is not a historical accident or optimization convenience, but the only known mechanism that teaches models how uncertain they should be.

E. Scope and Limitations

We emphasize what this paper does **not** claim:

- This is **not** natural language learning—our vocabulary is 16 tokens with synthetic grammars.
- We do **not** claim RL is more efficient—cross-entropy converges $\sim 5\times$ faster.
- We do **not** test compositional generalization—that requires carefully designed tasks beyond our current scope.

Rather, we isolate a specific question: *what does likelihood supervision provide that RL cannot?* The answer is calibrated uncertainty.

II. Background and Related Work

A. Prediction-as-Action Architecture

Standard model-based RL separates prediction (world model) from control (policy). Recent work [2] proposed unifying these: the agent's action *is* its prediction, and reward measures prediction accuracy.

For continuous states:

$$a_t = \hat{s}_{t+1} \quad \text{(action IS prediction)} \tag{1}$$

$$r_t = -\|\hat{s}_{t+1} - s_{t+1}\|^2 \quad \text{(reward IS accuracy)} \tag{2}$$

We extend this to discrete tokens:

$$a_t = \hat{x}_{t+1} \in \mathcal{V} \quad \text{(predict next token)} \tag{3}$$

$$r_t = \mathbb{I}[\hat{x}_{t+1} = x_{t+1}] \quad \text{(binary correctness)} \tag{4}$$

B. The Entropy Collapse Problem

Policy gradient methods are known to collapse to deterministic policies [3]. The objective $\mathbb{E}[\sum r_t]$ has no term encouraging diversity—once a valid action is found, the policy sharpens around it. This mode-seeking behavior is well-documented in maximum entropy RL literature [4], where entropy regularization is explicitly added to prevent collapse.

For prediction tasks with *unique* correct answers, this is fine. But language has ambiguity: multiple continuations may be equally valid ("I went to the [bank/store/park]..."). A collapsed policy cannot represent this uncertainty. Expectation-based theories of language processing [5] suggest that humans maintain probabilistic expectations over continuations—precisely what RL fails to learn.

C. Why RLHF Uses KL Regularization

RLHF fine-tuning [6] always includes a KL penalty against the pretrained model:

$$\mathcal{L} = \mathbb{E}[r(x)] - \beta \cdot D_{KL}(\pi_\theta || \pi_{ref}) \tag{5}$$

Our results explain why this is necessary: without KL regularization, RL would collapse the distributional knowledge learned during pretraining.

D. Representation Similarity Analysis

Centered Kernel Alignment (CKA) [7] measures similarity between neural network representations:

$$\text{CKA}(X, Y) = \frac{\|Y^T X\|_F^2}{\|X^T X\|_F \|Y^T Y\|_F} \tag{6}$$

CKA is invariant to orthogonal transformations and isotropic scaling, making it suitable for comparing representations across different training runs. Recent mechanistic interpretability work [8] has used similar techniques to understand neural network internals; we apply these methods to compare RL and MLE representations.

III. Experimental Framework

A. Ambiguous Grammar Design

We design grammars where some tokens have multiple valid continuations:

TABLE I
Ambiguous Grammar Specification

Component	Specification
Vocabulary size	16 tokens
Ambiguous tokens	8 (50% of vocabulary)
Ambiguity level	High (50/50 splits)
Deterministic tokens	8 (unique successor)
Oracle accuracy	75% (ceiling)

Key property: For ambiguous tokens, two successors are equally valid. A policy matching the oracle should output 50/50 probabilities; a collapsed policy picks one with $\sim$100% confidence.

Metrics:

- **Accuracy:** Fraction of predictions matching sampled target
- **Policy entropy:** $H(\pi) = -\sum_a \pi(a) \log \pi(a)$
- **KL divergence:** $D_{KL}(\pi || \pi_{oracle})$

B. Model Architecture

Transformer backbone (2 layers, 64 hidden, 4 heads, $\sim$109K parameters):

- **Embedding layer:** Token to continuous representation
- **Transformer layers:** Self-attention + feedforward
- **Prediction head:** Categorical distribution over vocabulary
- **Value head:** Baseline for REINFORCE (RL only)

C. Training Protocols

Note on entropy regularization: We use a standard entropy bonus coefficient of 0.01. While this provides some exploration pressure, we verified that small entropy bonuses do not qualitatively alter collapse; preventing collapse entirely would require entropy regularization to dominate the objective, fundamentally changing the learning dynamics.

TABLE II
TRAINING PROTOCOL COMPARISON

Property	RL (REINFORCE)	MLE
Loss	Policy gradient	Cross-entropy
Supervision	Scalar reward	Full distribution
Entropy bonus	0.01 (standard)	N/A
Training steps	20,000	20,000

TABLE III
AMBIGUOUS GRAMMAR RESULTS (20K STEPS, HIGH AMBIGUITY)

Metric	RL	MLE	Target
Accuracy	80%	80%	75% (oracle)
Policy entropy (ambig.)	**0.009**	0.68	0.693
KL from oracle (ambig.)	**6.28**	0.02	0.0

IV. RESULTS

A. Experiment 1: Distributional Collapse on Ambiguous Grammar

Key finding: RL collapses to deterministic outputs despite identical accuracy.

Both methods achieve oracle-level accuracy (80%), but their *distributions* are fundamentally different. Despite matching on the task metric (accuracy), the entropy differs by a factor of $77\times$ (0.009 vs. 0.68):

- **MLE** maintains near-oracle entropy (0.68 vs. 0.693 target)
- **RL** collapses to near-zero entropy (0.009)

Example prediction for an ambiguous token with oracle distribution [0.5, 0.5]:

- Oracle: $[0.0, ..., 0.5, ..., 0.5, ...]$
- MLE: $[0.0, ..., 0.48, ..., 0.52, ...]$
- RL: $[0.0, ..., 0.0003, ..., \mathbf{0.9997}, ...]$

RL picks one valid option with 99.97% confidence instead of the correct 50/50 split.

Interpretation: RL learns *which* tokens are valid (structure) but not *how likely* each should be (uncertainty). The policy gradient objective rewards finding *a* correct answer, not representing the full distribution of correct answers.

B. Experiment 2: Representation Similarity Analysis

Despite behavioral divergence, do RL and MLE learn similar internal representations?

TABLE IV
REPRESENTATION SIMILARITY (CKA, 500 SAMPLES)

Token Type	Linear CKA	RBF CKA
All tokens	0.662	0.735
Ambiguous tokens	**0.830**	**0.851**
Deterministic tokens	0.768	0.856

Key finding: Representations are highly similar, especially for ambiguous tokens.

The highest CKA (0.83–0.85) occurs for ambiguous tokens—exactly where *behavior* diverges most. This is the representation paradox: maximal representational alignment occurs precisely where behavioral divergence is greatest (entropy differs by $77\times$, but CKA is 0.83).

> *Distributional collapse occurs in the output layer, not in learned features.*

Both methods learn similar internal abstractions of token structure. The difference is entirely in how the output head maps these representations to probability distributions.

Implications:

1) RL successfully learns structural representations
2) The "failure" is specifically in the policy/output layer
3) Internal features could potentially be reused with a different output mechanism

C. Experiment 3: Structural Parity on Deterministic Tasks

Does distributional collapse impair structural learning?

TABLE V
DETERMINISTIC GRAMMAR RESULTS

Task	RL Accuracy	MLE Accuracy
Single-step (cyclic)	100%	100%
Single-step (permutation)	100%	100%
Multi-step (7-step delay)	100%	100%

Key finding: RL matches MLE perfectly on deterministic tasks.

When there is a unique correct answer, RL's tendency to collapse is not a problem—it converges to the correct deterministic mapping. This confirms that distributional collapse specifically impairs uncertainty learning, not structural learning:

- RL can learn token-level structure without likelihood supervision
- The limitation is specific to *uncertainty*, not structure
- For deterministic dynamics, the prediction-as-action framework works

D. Credit Assignment: Discrete vs. Continuous

An additional finding from multi-step experiments:

TABLE VI
CREDIT ASSIGNMENT COMPARISON

Action Space	Delay Steps	Accuracy
Continuous (prior work)	5	35%
Discrete (ours)	**7**	**100%**

Discrete action spaces show dramatically better credit assignment than continuous spaces. We hypothesize this is because discrete predictions are binary (correct/incorrect), providing sharper gradient signals than continuous predictions which can be "partially correct."

V. Discussion

A. Why Accuracy Masks Distributional Failure

A natural objection to our findings is: "If accuracy is high, why does distributional collapse matter?" This question reveals a fundamental confusion about what language models must do.

Accuracy treats all valid outputs as equivalent. If the oracle distribution is [0.5, 0.5] over tokens A and B, then predicting A with 100% confidence achieves 50% accuracy—the same as predicting B with 100% confidence. Accuracy cannot distinguish between these two collapsed policies, nor between them and the correct 50/50 distribution.

Language requires probability mass allocation. Downstream tasks depend on calibrated uncertainty:

- **Sampling:** Generation quality depends on the full distribution, not just the mode
- **Perplexity:** The standard evaluation metric directly measures distributional fit
- **Beam search:** Relies on relative probabilities across candidates
- **Uncertainty quantification:** Applications requiring "I don't know" responses need calibrated entropy

RL optimizes correctness, not uncertainty. The policy gradient objective $\nabla\mathbb{E}[r]$ rewards finding *a* correct answer. Once found, the gradient pushes probability mass toward that answer. There is no term in the objective that rewards spreading mass across multiple valid answers.

This is why RL "looks fine" on accuracy but fails generatively. Accuracy is the wrong metric for evaluating distributional learning.

B. What Likelihood Supervision Provides

Our results clarify a fundamental question: what does cross-entropy training provide that RL cannot learn?

TABLE VII
What Each Method Learns

Capability	RL	MLE
Token structure (what follows what)	✓	✓
Deterministic mappings	✓	✓
Internal representations	✓	✓
Calibrated uncertainty	×	✓

Key Claim: Likelihood-based training is not required to learn linguistic structure, but *is* required to learn calibrated uncertainty. Reinforcement learning optimizes for correctness, not probability mass allocation. The structural representations are equivalent—only the output distributions differ.

C. Contrast with Modern Practice: Pretraining + RLHF vs. RL-Only

Our findings directly explain why modern LLM pipelines use pretraining followed by RLHF, rather than RL alone. Table VIII makes this contrast explicit.

TABLE VIII
Modern Practice vs. RL-Only Language Learning

Property	Pretraining + RLHF	RL-Only
Structure learning	✓	✓
Uncertainty learning	✓(from pretraining)	× (collapses)
Sampling quality	High	Poor
Entropy control	KL-regularized	Absent
Perplexity	Meaningful	Undefined

The pretraining phase learns calibrated distributions; the RLHF phase refines behavior while the KL penalty *prevents* distributional collapse. Remove either component, and the pipeline fails:

- Without pretraining: No calibrated uncertainty to preserve
- Without KL penalty: RL collapses the learned distributions

Our synthetic experiments reproduce this phenomenon in miniature, explaining why this pipeline structure is necessary rather than merely conventional.

D. Why This Explains RLHF Design

RLHF [6] always includes:

1) **Pretrained initialization:** Start from likelihood-trained model
2) **KL regularization:** Penalize divergence from pretrained distribution

Our results explain both:

- **Pretraining** is necessary to learn calibrated uncertainty
- **KL penalty** prevents RL from collapsing learned distributions

Without these, RL fine-tuning would destroy the distributional knowledge that makes language models useful.

E. The Representation Paradox

Our most striking finding is that representation similarity is *highest* where behavioral divergence is greatest:

- Ambiguous tokens: CKA = 0.83, but entropy differs by 77×
- Deterministic tokens: CKA = 0.77, behavior identical

This suggests that both methods learn the same "what is possible" representation. The difference is entirely in "how likely is each possibility"—a property of the output layer, not the learned features. Figure 1 illustrates this objective-induced divergence.

F. Design Implications

Design Implication: Any RL-based language learner must include either:

(a) Likelihood supervision (pretraining or auxiliary loss), or
(b) An explicit distribution-matching regularizer (e.g., KL penalty)

if calibrated uncertainty is required. Scalar reward signals alone cannot induce distributional learning.

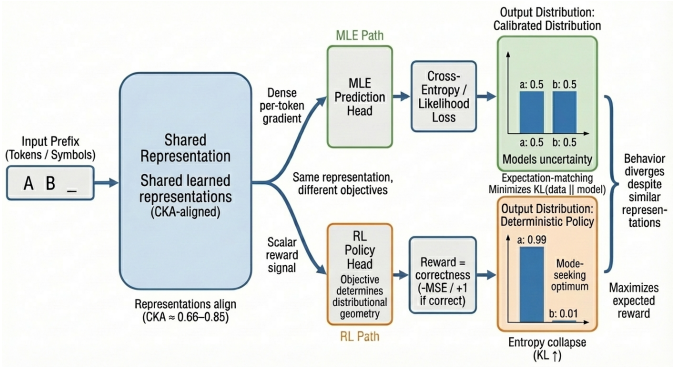

Fig. 1. Objective-induced divergence between representations and output distributions. Despite strong representational alignment (CKA ≈ 0.66–0.85), MLE and RL objectives induce fundamentally different output geometries: MLE minimizes KL(data ‖ model), producing expectation-matching distributions; RL maximizes expected reward, producing mode-seeking collapse. The objective determines distributional geometry—not the learned representations.

G. Implications for RL-Based Language Learning

Can RL learn language from scratch without pretraining? Our results suggest:

1) **Structure:** Yes, RL can learn token-level dynamics
2) **Uncertainty:** No, RL collapses to deterministic outputs
3) **Representations:** Largely preserved despite behavioral collapse

For language tasks requiring calibrated uncertainty (generation, sampling, perplexity), likelihood supervision remains necessary. For tasks requiring only structural knowledge (classification, deterministic prediction), RL may suffice.

H. Limitations

1) **Scale:** 16-token vocabulary, synthetic grammars
2) **Complexity:** No compositional structure, semantics, or long-range dependencies
3) **Algorithms:** Only tested REINFORCE; other policy gradient methods (PPO, SAC) may differ
4) **Entropy bonus:** Standard 0.01 coefficient; aggressive entropy regularization might prevent collapse but would fundamentally alter the learning objective

I. Future Work

1) **Entropy-regularized RL:** Can aggressive entropy bonuses prevent collapse while maintaining task performance?
2) **Distributional RL:** Methods that explicitly model return distributions [9]
3) **Larger scale:** Does the structure/uncertainty separation hold at 1000+ tokens?
4) **Compositional tasks:** Carefully designed tests of systematic generalization

VI. Conclusion

We investigated what reinforcement learning can learn about language-like structure without likelihood supervision. Our findings reveal a fundamental asymmetry:

1) RL learns **structure** (which tokens follow which) but not **uncertainty** (probability distributions over valid options)
2) This failure is **objective-level**, not representational—internal features are highly similar (CKA 0.83 on ambiguous tokens)
3) Distributional collapse occurs in the **output layer**, not learned representations

These results explain why likelihood-based pretraining exists: not to learn structure (RL can do that), but to learn calibrated uncertainty over structure. They also explain why RLHF requires KL regularization: to prevent RL from destroying learned distributions.

We emphasize that these are controlled experiments on 16-token synthetic grammars, not natural language. But they isolate a precise answer to the question: *what does likelihood supervision provide?* The answer is uncertainty—and that is what makes the difference for language.

Code Availability

Code and experiment logs will be made available upon publication.

References

[1] T. Brown et al., "Language Models are Few-Shot Learners," in *Advances in Neural Information Processing Systems*, vol. 33, 2020.
[2] [Anonymous], "Prediction as Action: When Reinforcement Learning Learns Predictive Representations Without Pretraining," *Under Review*, 2025.
[3] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine Learning*, vol. 8, pp. 229–256, 1992.
[4] B. D. Ziebart et al., "Maximum Entropy Inverse Reinforcement Learning," in *AAAI*, 2008.
[5] R. Levy, "Expectation-based syntactic comprehension," *Cognition*, vol. 106, no. 3, pp. 1126–1177, 2008.
[6] L. Ouyang et al., "Training language models to follow instructions with human feedback," in *NeurIPS*, 2022.
[7] S. Kornblith et al., "Similarity of Neural Network Representations Revisited," in *ICML*, 2019.
[8] N. Elhage et al., "Toy Models of Superposition," *Anthropic*, 2022.
[9] M. G. Bellemare et al., "A Distributional Perspective on Reinforcement Learning," in *ICML*, 2017.
[10] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, 1998.