

The Knowledge-Application Gap: Model Size Dictates Learning Paradigm

A study on Linguistic Reflection vs. Parameter Fine-Tuning

Authors: D. Rawson

Date: November 9, 2025

Institution: Independent Research

Abstract

We demonstrate a critical decoupling of knowledge from its application based on model size. Our experiment presented a complex scheduling task to a 3B parameter model, guided by a linguistic strategy previously validated on a 7B model (achieving 88% accuracy). The 3B model, despite having access to this correct knowledge via the prompt, failed to apply it, showing minimal improvement (+5.3%). However, when this *exact same knowledge* was embedded directly into its weights via LoRA fine-tuning, the model's performance skyrocketed to an identical 88% accuracy. This reveals a **capacity threshold for meta-cognitive learning**: smaller models (<~5B) lack the working memory to internalize and execute complex linguistic strategies, requiring knowledge to be "burned in" as reflexive patterns through parameter updates. This "knowledge-application gap" has profound implications, suggesting smaller models should be optimized via fine-tuning, while larger models can benefit from more interpretable, instruction-based learning.

Keywords: Linguistic Reinforcement Learning, LoRA, Model Scaling, Meta-Cognition, Knowledge-Application Gap, Parameter-Efficient Fine-Tuning

1. Introduction

1.1 Background

Two dominant paradigms exist for improving language model performance on specific tasks:

- Linguistic Learning:** Models improve through natural language instructions, strategies, and reflection (Brown et al., 2020; Wei et al., 2022). This relies on the model's capacity to *interpret and apply* knowledge.
- Parameter Learning:** Models improve through gradient-based weight updates (Hu et al., 2021). This relies on *embedding* knowledge directly into the model's parameters.

Recent work has shown that Linguistic Reinforcement Learning (LRL) enables models to learn through self-reflection without parameter updates, achieving significant performance gains through iterative strategy refinement (Rawson, 2025). However, all prior LRL experiments used models with $\geq 7B$ parameters.

1.2 Research Question

This paper investigates a critical question: **Is there a minimum model capacity required to *apply* complex, linguistically-expressed knowledge, and when does effective learning necessitate that knowledge be directly *embedded* into the model's parameters?**

We hypothesize that a "knowledge-application gap" exists in smaller models, where they can possess a correct strategy but lack the cognitive capacity to execute it, requiring parameter modification to bridge this gap.

1.3 Contributions

1. **First systematic comparison** of LRL vs. LoRA on identical tasks and test sets.
 2. **Discovery of a "knowledge-application gap"**: Smaller models fail to apply linguistic knowledge that they can successfully use once embedded via LoRA.
 3. **Identification of a capacity threshold**: The ability to learn from linguistic instruction emerges around 5-7B parameters.
 4. **Practical guidance**: Match the learning paradigm to the model's size—embed knowledge in small models ($<5B$) and instruct larger models ($\geq 7B$).
-

2. Related Work

2.1 Linguistic Reinforcement Learning

Prior work demonstrated that 7B models can learn through journaling and strategy distillation, achieving 78-88% accuracy on scheduling tasks through purely linguistic learning (Rawson, 2025). This approach:

- Requires no gradient updates
- Produces interpretable reasoning traces
- Enables strategy transfer across models
- Converges through reflective iteration

2.2 Parameter-Efficient Fine-Tuning

LoRA (Hu et al., 2021) enables efficient fine-tuning by learning low-rank weight updates. Benefits include:

- Minimal trainable parameters (typically $<1\%$ of model)
- Strong performance on downstream tasks
- Fast training and inference
- Model-specific learned patterns

2.3 Model Scaling Laws

Kaplan et al. (2020) and Hoffmann et al. (2022) established scaling laws for model performance, but focused on pre-training. Limited work exists on how model size affects *learning paradigm effectiveness* post-deployment.

3. Methodology

3.1 Task: Meeting Room Scheduling

Problem Definition:

- Input: N meeting time ranges [start, end], M available rooms
- Output: YES if all meetings can be scheduled, NO otherwise
- Ground Truth: Max simultaneous meetings $\leq M$

Difficulty Levels:

- **EASY:** 2 meetings, 2-3 rooms
- **MEDIUM:** 3-4 meetings, 2-3 rooms
- **HARD:** 4-6 meetings, 2-4 rooms

Dataset:

- Training: 100 problems (seed=42), stratified by difficulty
- Test: 150 problems (seed=123), stratified by difficulty

3.2 Experimental Design

Four-Stage Experiment:

Stage 1: Baseline (No Learning)

- Model: Qwen2.5-3B-Instruct via Ollama
- Approach: Zero-shot inference
- Purpose: Establish baseline capability

Stage 2: Bootstrap Learning (LRL Training)

- Model: Qwen2.5-3B-Instruct via Ollama
- Approach: Solve problems → Journal reflections → Distill strategy
- Batches: 10 batches of 10 problems
- Strategy Updates: After batches 5 and 10
- Purpose: Learn through linguistic reflection

Stage 3: Test with LRL

- Model: Qwen2.5-3B-Instruct via Ollama
- Approach: Apply learned strategy from Stage 2
- Purpose: Measure linguistic learning effectiveness

Stage 4: LoRA Fine-Tuning

- Model: Qwen2.5-3B-Instruct via HuggingFace
- Training Data: 100 synthetic examples (seed=999) using learned strategy
- Configuration:
 - Rank: 8, Alpha: 16
 - Target modules: q_proj, v_proj
 - Quantization: 4-bit with CPU offloading
 - Epochs: 3, Batch size: 1 (8 gradient accumulation steps)
- Purpose: Measure parameter learning effectiveness

3.3 Comparison to Prior Work

Published 7B Results (Rawson, 2025):

- Baseline: 55.3%
- Bootstrap: 66.0%
- LRL Test: 88.0%
- **Improvement: +32.7%**

Our experiment uses identical methodology but with a 3B model to test size-dependence.

3.4 Evaluation Metrics

- **Overall Accuracy:** Correct predictions / Total problems
- **Accuracy by Difficulty:** Performance on EASY/MEDIUM/HARD subsets
- **Improvement over Baseline:** Δ accuracy from Stage 1

4. Results

4.1 Quantitative Results

Stage	Accuracy	EASY	MEDIUM	HARD	vs. Baseline
Stage 1: Baseline	16.0%	6.0%	16.0%	26.0%	—
Stage 2: Bootstrap	12.0%	12.1%	33.3%	20.6%	-4.0%
Stage 3: LRL Test	21.3%	—	—	—	+5.3%
Stage 4: LoRA	88.0%	100.0%	88.0%	76.0%	+72.0%

4.2 Narrative Summary of Findings

The results tell a clear and dramatic story. The 3B model, when given a proven linguistic strategy in its prompt (Stage 3), showed almost no ability to use it, gaining only 5.3% over its own weak baseline. However, when that *exact same strategic knowledge* was used to generate training data for a LoRA adapter (Stage 4), the model's performance became flawless on easy problems and reached 88% overall—precisely matching the performance of a much larger 7B model.

This demonstrates that the knowledge was correct and transferable, but the 3B model was incapable of applying it through active reasoning alone.

5. Analysis: The Knowledge-Application Gap

Our findings point not just to a difference in performance, but to a fundamental difference in *how* models of different sizes are capable of learning.

5.1 The Master Craftsman vs. The Apprentice

An analogy best explains this phenomenon:

- A **Master Craftsman (the 7B model)** can be handed a complex blueprint (the LRL strategy) and successfully build a sophisticated product. They have the cognitive capacity to read, interpret, and execute the instructions.
- An **Apprentice (the 3B model)** can be handed the same blueprint. They can read the words but lack the working memory and experience to hold the entire plan in their head while executing it. They fail. This is our Stage 3 result.
- However, if you use the blueprint to build a set of **custom jigs and tools (the LoRA adapter)** that guide every action, the apprentice can now produce a perfect product. They don't need to reason through the plan; the knowledge is embedded in the tools they use. This is our Stage 4 result.

This is the **Knowledge-Application Gap**: the gulf between possessing knowledge and being able to act on it.

5.2 Why LRL Failed: A Failure of Application

The 3B model's failure is not one of learning, but of execution.

- **Limited Working Memory:** Executing a multi-step linguistic strategy requires holding the rules, the current problem state, and the reasoning process in context simultaneously. The 3B model's smaller hidden dimensions likely cannot support this cognitive load.
- **Weak Meta-Cognitive Capabilities:** LRL requires "thinking about thinking"—monitoring one's own adherence to a strategy. The bootstrap phase, where the model's reflections failed to produce a generalizable strategy (dropping accuracy from 90% in-training to 12% on the test set), suggests the model cannot effectively self-correct through reasoning.

5.3 Why LoRA Succeeded: Cognitive Offloading

LoRA bridged the knowledge-application gap by performing a form of **cognitive offloading**.

- **Direct Pattern Encoding:** LoRA bypasses the need for active linguistic interpretation during inference. The logic of the strategy is encoded directly into the low-rank matrices, becoming the model's "muscle memory."
- **From Reasoning to Reflex:** The task is transformed. The model no longer needs to "reason" its way through the scheduling problem step-by-step; the optimal path is now an ingrained, reflexive response triggered by the problem's structure.

5.4 The Capacity Threshold for Meta-Cognitive Learning

This experiment reveals a critical threshold, likely between 3B and 7B parameters, where the ability to perform this cognitive offloading *internally* emerges.

- **Models below the threshold** require external offloading (i.e., LoRA fine-tuning) to embed complex knowledge.
 - **Models above the threshold** possess enough working memory and representational complexity to interpret and apply linguistic knowledge directly from their context. This is the emergent ability of meta-cognitive learning.
-

6. Limitations

6.1 Single Task Domain

- Results specific to constraint satisfaction (scheduling). May not generalize to language generation or other tasks. Future work should test on diverse task types.

6.2 Limited Model Size Range

- Only tested 3B and 7B (from prior work). The exact threshold location is uncertain. Need experiments at 4B, 5B, 6B to pinpoint the transition.

6.3 Architecture Specificity

- Both models from the Qwen family. Different architectures (e.g., Llama, Mistral) may have different thresholds.

6.4 Single LoRA Configuration

- Used rank=8. Different hyperparameters might change results, though the dramatic improvement suggests the effect is robust.
-

7. Related Phenomena

7.1 Emergent Abilities

Schaeffer et al. (2023) discuss "emergent abilities" in large models. Our findings suggest **meta-cognitive learning** is an emergent ability appearing around 5-7B parameters, enabling self-reflection and strategy internalization.

7.2 Instruction Following

Wang et al. (2022) found instruction-following quality scales with model size. Our results extend this: it's not just *following* instructions that scales, but *learning from* them that has a distinct threshold.

7.3 Chain-of-Thought Reasoning

Wei et al. (2022) showed CoT emerges in large models. Our findings suggest a distinction: CoT prompting works across sizes for inference-time reasoning, but CoT *learning* (strategy internalization) requires larger models.

8. Future Work

8.1 Threshold Characterization

- **Experiment:** Test 4B, 5B, 6B models with an identical protocol to pinpoint the exact capacity threshold.

8.2 Architecture Comparison

- Test Llama, Mistral, and Gemma families to identify architecture-specific factors and compare thresholds.

8.3 Hybrid Approaches

- Explore combining LRL and LoRA, for example, by using the LRL journal as the primary source for fine-tuning data in a continual learning loop.

8.4 Task Generalization

- Replicate the experiment on diverse tasks like math reasoning (GSM8K) and coding (HumanEval).
-

9. Implications

9.1 For Model Deployment

The choice of learning paradigm should be dictated by model size.

- **For Small Models (<5B):** Knowledge must be **embedded**. Use LoRA or full fine-tuning. Do not expect them to reliably execute complex strategies from a prompt. This sacrifices interpretability for cost-effective performance.
- **For Large Models (≥7B):** Knowledge can be **instructed**. Use LRL or sophisticated prompting for tasks that require flexibility and interpretability. This allows for auditable reasoning, even at a higher inference cost.

9.2 For AI Safety

The knowledge-application gap is a critical safety consideration. A small model could "state" its adherence to a safety principle in its prompt but lack the capacity to faithfully execute it under pressure. For safety-critical applications, this implies either using larger models that can demonstrate reliable strategy adherence or using fine-tuning to embed safety constraints as non-negotiable reflexes.

10. Conclusion

This research reveals a fundamental principle of AI development: **the effectiveness of a learning paradigm is determined by model size**. We demonstrated that while a 7B model can master a task through linguistic instruction, a 3B model fails when given the exact same instruction. This failure is not due to flawed knowledge, but to an inability to apply it—a "knowledge-application gap."

The gap is bridged by LoRA fine-tuning, which succeeds by **embedding the linguistic knowledge directly into the model's parameters**, transforming the task from one of active reasoning to ingrained reflex. This act of "cognitive offloading" allows the 3B model to match the 7B model's peak performance.

Our central finding is the existence of a **capacity threshold for meta-cognitive learning**, located between 3B and 7B parameters, where models gain the ability to internalize and execute complex linguistic strategies. Below this threshold, knowledge must be embedded. Above it, it can be instructed. This size-paradigm interaction is a critical consideration for the efficient, interpretable, and safe deployment of artificial intelligence.

References

- Brown, T., et al. (2020). Language Models are Few-Shot Learners. *NeurIPS*.
- Hoffmann, J., et al. (2022). Training Compute-Optimal Large Language Models. *arXiv:2203.15556*.
- Houlsby, N., et al. (2019). Parameter-Efficient Transfer Learning for NLP. *ICML*.
- Hu, E. J., et al. (2021). LoRA: Low-Rank Adaptation of Large Language Models. *ICLR*.
- Kaplan, J., et al. (2020). Scaling Laws for Neural Language Models. *arXiv:2001.08361*.
- Rawson, D. (2025). Linguistic Reinforcement Learning: Emergent Occam's Razor Through Reflective Distillation. *Independent Research*.

Schaeffer, R., et al. (2023). Are Emergent Abilities of Large Language Models a Mirage?
arXiv:2304.15004.

Wang, Y., et al. (2022). Self-Instruct: Aligning Language Models with Self-Generated Instructions.
arXiv:2212.10560.

Wei, J., et al. (2022). Chain-of-Thought Prompting Elicits Reasoning in Large Language Models.
NeurIPS.

Appendices

Appendix A: Experimental Details

Hardware Configuration

- GPU: NVIDIA RTX (10GB VRAM limit)
- CPU: 30GB allocated for offloading
- Ollama: Local inference for Stages 1-3
- HuggingFace Transformers: Stage 4 LoRA training

Model Configurations

- **Ollama (Stages 1-3):** qwen2.5:3b, Temperature: 0.1
- **HuggingFace (Stage 4):** Base: Qwen/Qwen2.5-3B-Instruct, Quantization: 4-bit NF4

Prompt Templates

- Prompts for baseline, LRL, and LoRA training followed a structured format, detailed in the project repository.

Reproducibility

- All code, data, and results are available at: <https://github.com/DRawson5570/linguistic-rl-scheduling>
- Seeds: Training data (42), Test data (123), LoRA training (999)

Appendix B: Complete Results

- The complete results file, `scheduling_lrl_plus_lora_results.json`, containing over 5,000 lines of detailed experimental data, is available in the repository. This includes per-problem responses, reasoning traces, and timing information.
 - The full learned strategy (`scheduling_lrl_strategy.txt`) and the complete learning journal (`scheduling_lrl_journal.txt`) are also available.
-

Citation

code Bibtex

downloadcontent_copy

expand_less

```
@article{rawson2025model-size-learning,  
  title={The Knowledge-Application Gap: Model Size Dictates Learning Paradigm},  
  author={Rawson, D.},  
  journal={arXiv preprint},  
  year={2025},  
  note={Independent Research}  
}
```

Contact: rawson.douglas@gmail.com

Code: <https://github.com/DRawson5570/linguistic-rl-scheduling>