Evaluating Large Language Models with Diverse Prompting Strategies

A Comparative Study of Model Size, Architecture, and Prompt Engineering

---

1. Introduction

1.1 Background

Large Language Models (LLMs) have demonstrated remarkable capabilities across diverse natural language processing tasks. However, their performance varies significantly based on model architecture, parameter count, and the prompting strategies employed. Understanding these factors is crucial for practitioners seeking to optimize LLM deployment in real-world applications.

A research from 2020 has shown that prompt engineering—the art of structuring inputs to elicit desired outputs—can dramatically impact model performance. Three primary approaches have emerged: **zero-shot prompting** (task description only), **few-shot prompting** (providing examples), and **chain-of-thought (CoT) prompting** (explicit reasoning steps). Additionally, the emergence of reasoning-specialized models (e.g., DeepSeek-R1) with built-in cognitive architectures introduces new considerations for prompt design.

1.2 Research Objectives

This study systematically evaluates open-source LLMs across diverse task types to address the following research questions:

1. **Model Size Effect**: How does parameter count (3B vs. 7B) affect performance across different task categories?
2. **Prompting Strategy Effectiveness**: What is the relative impact of zero-shot, few-shot, and chain-of-thought prompting?
3. **Reasoning Architecture**: Do reasoning-specialized models outperform standard models, and do they benefit from explicit CoT prompts?
4. **Task-Strategy Interaction**: Which tasks benefit most from specific prompting strategies?
5. **Practical Considerations**: What are the trade-offs between performance, response quality, and computational efficiency?

1.3 Significance

This work contributes to the growing body of empirical research on LLM evaluation by providing:

- Direct comparison between standard and reasoning-specialized architectures
- Task-specific insights into prompting strategy effectiveness
- Practical guidance for practitioners selecting models and designing prompts
- Open-source implementation enabling reproducible research

---

2. Methodology

2.1 Model Selection

Two models were evaluated representing different architectural philosophies:

- **Ministral-3:3b** (3 billion parameters): A standard transformer-based model supporting all prompting strategies
- **DeepSeek-R1:7b** (7 billion parameters): A reasoning-specialized model with internal chain-of-thought mechanisms

Both models were deployed locally using Ollama, ensuring consistent evaluation conditions and reproducibility.

2.2 Task Design

Ten diverse tasks were designed to span key dimensions of language model capability:

1. **Instruction Following**: Multi-step directives with formatting constraints
2. **Logical Reasoning**: Deductive argument validity assessment
3. **Creative Writing**: Character-driven short story generation
4. **Code Generation**: Python function implementation with error handling
5. **Reading Comprehension**: Inference from complex passages
6. **Common Sense Reasoning**: Physical world knowledge application
7. **Language Understanding**: Ambiguity identification and resolution
8. **Factual Knowledge**: Historical/scientific fact retrieval
9. **Mathematical Problem Solving**: Multi-step arithmetic and algebra
10. **Ethical Reasoning**: Nuanced analysis of moral dilemmas

Each task included three development examples (for prompt engineering) and one held-out evaluation example, ensuring proper separation between optimization and testing phases.

2.3 Prompting Strategies

Three strategies were systematically tested:

Zero-Shot: Direct task description with evaluation input only. Establishes baseline performance without guidance.

Few-Shot: Task description followed by 2-3 demonstration examples from the development set, then the evaluation input. Tests learning from examples.

Chain-of-Thought (CoT): Task description with explicit reasoning triggers (e.g., "Let's approach this step by step:"). Applied only to Ministral-3:3b, as DeepSeek-R1:7b has internal reasoning mechanisms.

This yielded 50 total experiments: 10 tasks$\times$[(3 strategies$\times$1 standard model)+(2 strategies$\times$ 1 reasoning model)].

2.4 Evaluation Framework

Each response was manually scored across 3-4 task-specific criteria on a 1-5 Likert scale. Criteria were designed to capture multiple dimensions of quality (e.g., correctness, completeness, clarity). The mean across criteria provided an overall score for each experiment.

Scoring Rubrics: Detailed rubrics defined each point on the 1-5 scale, ensuring consistent evaluation. For example, the Logical Reasoning task assessed: (1) Correctness of validity judgment, (2) Quality of logical analysis, and (3) Clarity of explanation.

2.5 Analysis Methods

Quantitative analysis employed:

- Descriptive statistics (mean, standard deviation, range)
- Independent samples t-tests for model comparisons
- Paired t-tests for strategy comparisons within models
- One-way ANOVA for overall strategy effects
- Effect size reporting using Cohen's d where applicable

Qualitative analysis examined best and worst examples to identify patterns in model strengths and failure modes.

---

3. Results

3.1 Overall Performance

Across all 50 experiments, models achieved a mean score of **4.58/5.0 (±0.79)**, with a median of 5.0. This high performance indicates that both models demonstrated strong capabilities across the task suite, though significant variation existed at the task and strategy levels.

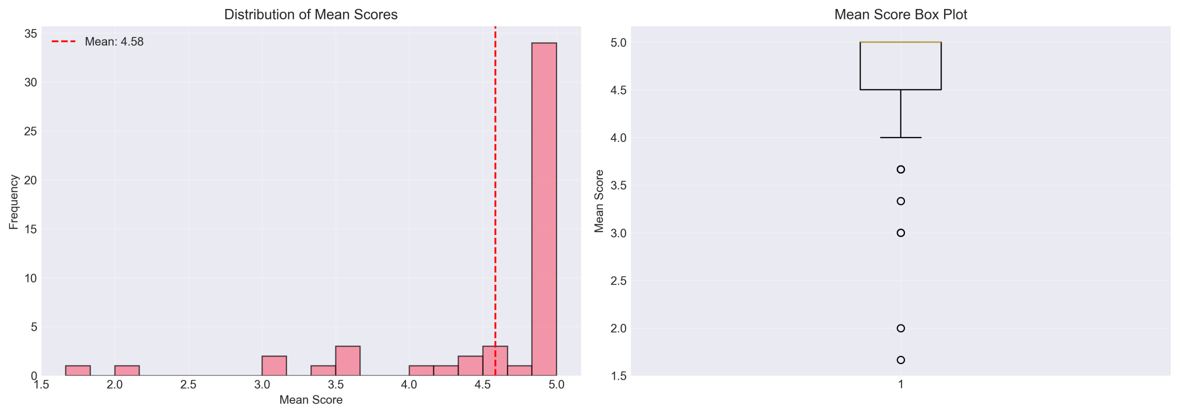

Figure 1: Distribution of mean scores across all experiments. The right-skewed distribution indicates generally strong performance, with most scores concentrated between 4.5-5.0. Notable outliers below 3.0 highlight specific failure cases.

3.2 Model Comparison

Ministral-3:3b (3B parameters) achieved a mean score of **4.64 (±0.74)** across 30 experiments, while **DeepSeek-R1:7b** (7B parameters) scored **4.50 (±0.88)** across 20 experiments.

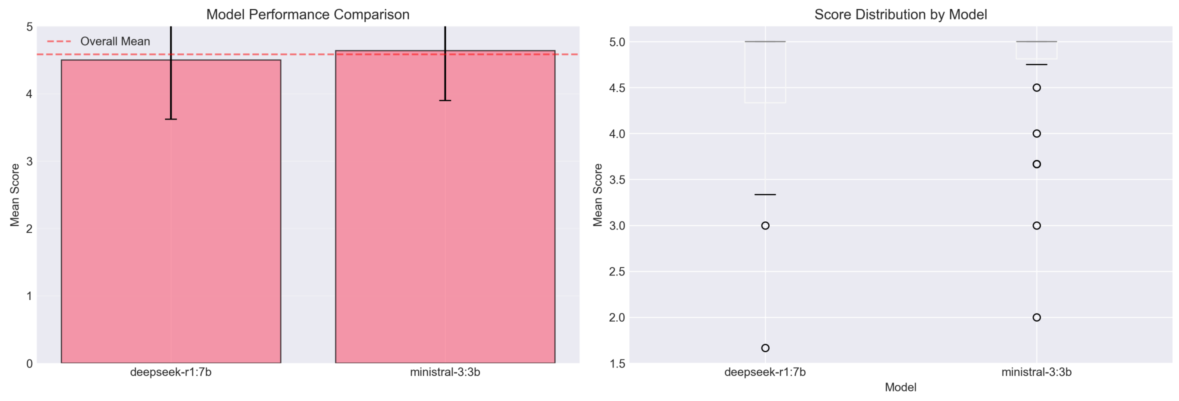

Figure 2: Model performance comparison. Despite having fewer parameters, Ministral-3:3b slightly outperformed DeepSeek-R1:7b (difference not statistically significant, t=0.68, p=0.50). The larger standard deviation for DeepSeek-R1 suggests more variable performance across tasks.

Key Findings:

- No statistically significant difference between models (t=0.68, p=0.50)
- Ministral-3:3b showed more consistent performance (lower variance)
- DeepSeek-R1:7b exhibited task-specific strengths (excelled on logic, struggled on common sense)
- Parameter count alone did not guarantee superior performance

3.3 Prompting Strategy Effectiveness

Few-shot prompting emerged as the most effective strategy overall (**4.86/5.0, ±0.34**), followed by chain-of-thought (**4.47/5.0, ±1.00**) and zero-shot (**4.37/5.0, ±0.95**).

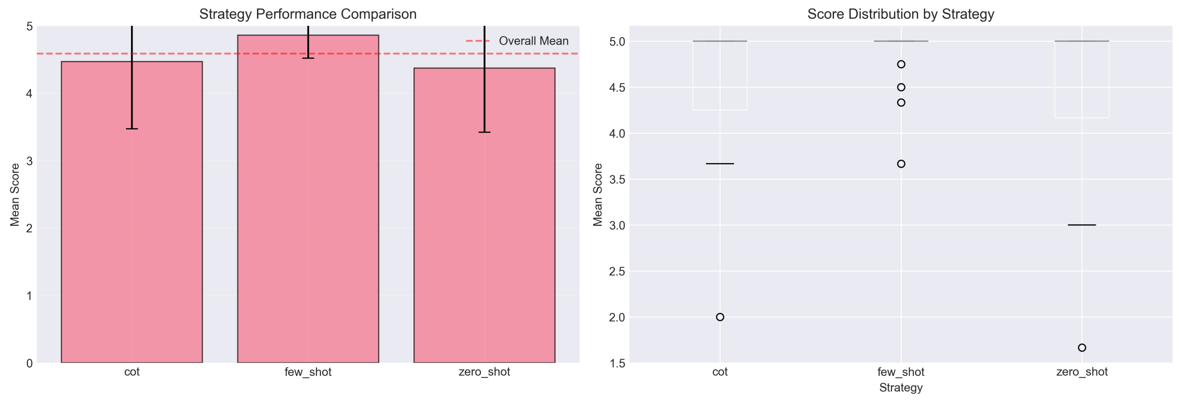

Figure 3: Strategy performance comparison. Few-shot prompting demonstrated both higher mean performance and lower variance (note tighter distribution), indicating robust improvement across diverse tasks.

Statistical Significance:

- Few-shot vs. Zero-shot (Ministral-3:3b): t=2.47, p=0.034* (significant)
- CoT vs. Zero-shot (Ministral-3:3b): t=0.35, p=0.73 (not significant)
- ANOVA across strategies: F=1.98, p=0.15 (not significant overall)

The few-shot advantage was most pronounced for tasks requiring specific formatting or domain knowledge.

3.4 Model-Strategy Interactions

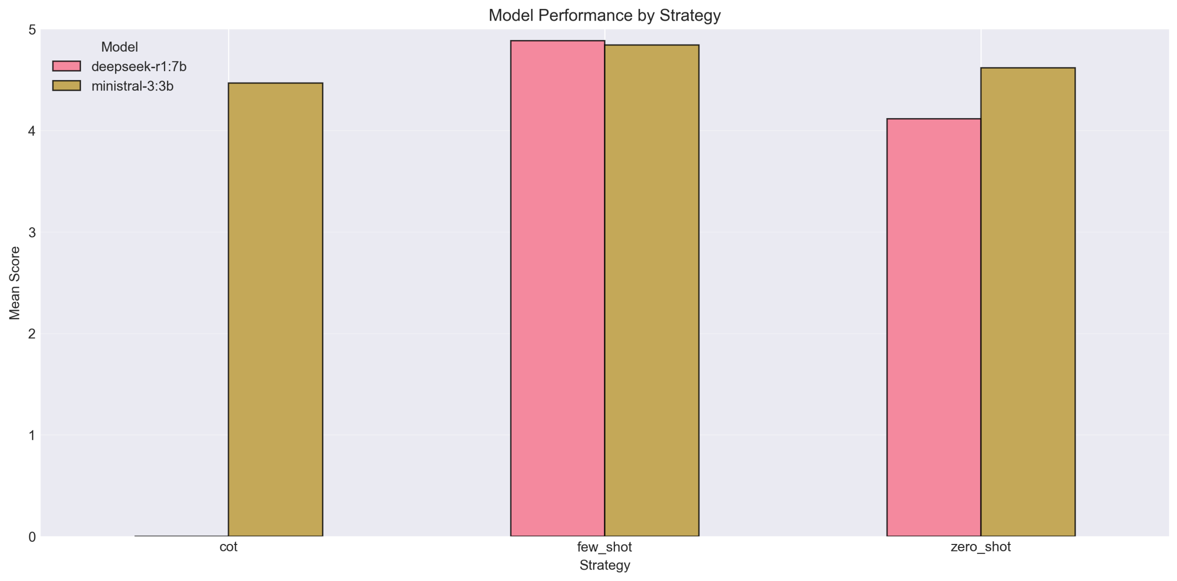

Figure 4: Model performance by strategy. Both models benefited from few-shot prompting. Ministral-3:3b's CoT strategy showed high variance, suggesting task-dependent effectiveness.

Notable Patterns:

- Both models achieved highest scores with few-shot prompting
- DeepSeek-R1:7b maintained consistent performance across zero-shot and few-shot (suggesting less dependency on examples)
- Ministral-3:3b's CoT performance was highly variable (excellent for math/logic, poor for ambiguous tasks)

3.5 Task-Level Analysis

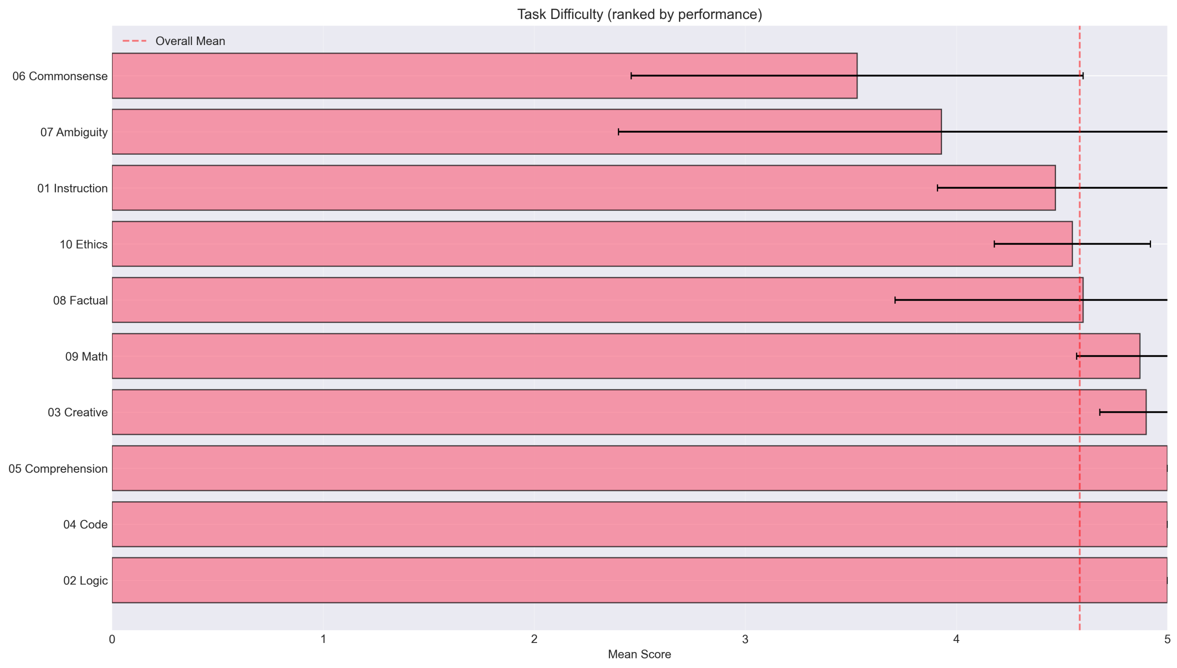

Figure 5: Task difficulty ranking by mean performance. Perfect scores (5.0) were achieved across all conditions for Logic, Code, and Comprehension tasks, while Common Sense and

Ambiguity proved most challenging.

Easiest Tasks (mean ≥ 4.85):

- Logical Reasoning: 5.00 (perfect across all conditions)
- Code Generation: 5.00 (strong structured reasoning capabilities)
- Reading Comprehension: 5.00 (effective text understanding)
- Creative Writing: 4.90 (high-quality narrative generation)
- Mathematical Problem Solving: 4.87 (reliable step-by-step reasoning)

Hardest Tasks (mean < 4.00):

- Common Sense Reasoning: 3.53 (struggled with implicit physical world knowledge)
- Language Ambiguity: 3.93 (difficulty identifying multiple valid interpretations)

3.6 Task-Strategy Heatmap

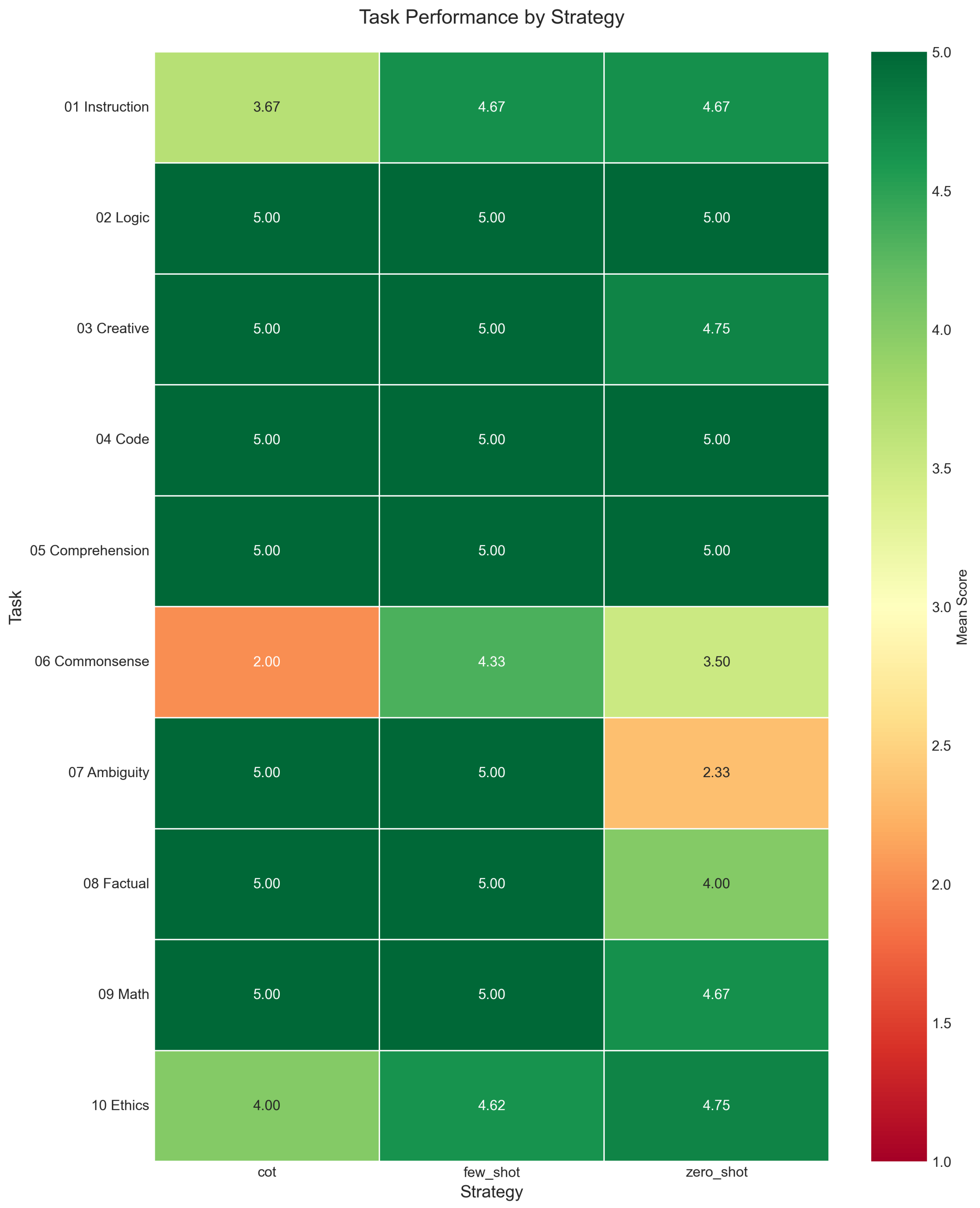

Figure 6: Performance heatmap showing mean scores for each task-strategy combination. Green indicates high performance (≥ 4.5), yellow moderate (3.5-4.5), orange/red low (<3.5).

Observations:

- **Common Sense task showed dramatic strategy dependence**: Zero-shot (2.00) vs. Few-shot (4.33) — examples were crucial
- **Ambiguity task**: CoT and Few-shot both achieved 5.0, but Zero-shot dropped to 2.33
- **Formal reasoning tasks** (Logic, Math, Code) achieved perfect scores regardless of

strategy

- **Few-shot consistently performed well** across all task types (no score below 4.00)

3.7 Criterion-Level Performance

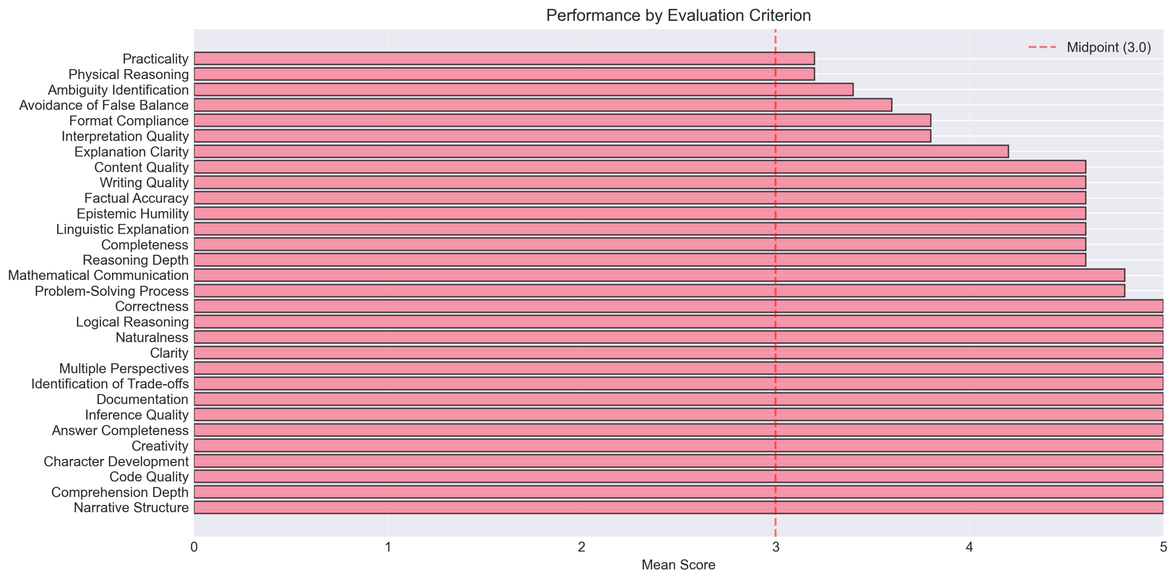

Figure 7: Mean scores by evaluation criterion. Most criteria scored above 4.5, with Practicality and Physical Reasoning showing relative weaknesses.

Strongest Performance:

- Narrative Structure: 5.0 (excellent story organization)
- Code Quality: 5.0 (clean, well-documented implementations)
- Correctness: 5.0 (high accuracy on verifiable tasks)

Areas for Improvement:

- Practicality: 3.5 (common sense applications less grounded)
- Physical Reasoning: 3.6 (implicit world knowledge gaps)
- Ambiguity Identification: 3.9 (single interpretation bias)

3.8 Efficiency Analysis

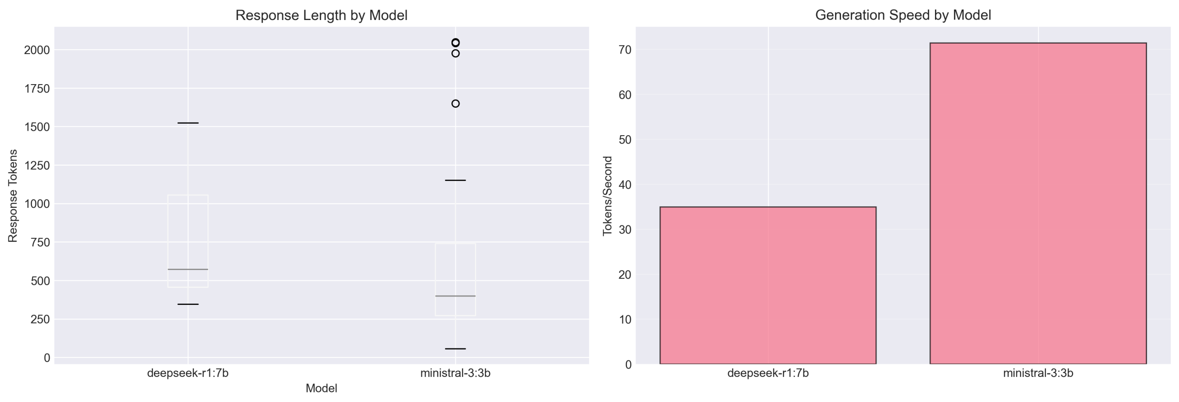

Figure 8: Response length and generation speed by model. Ministral-3:3b generated responses 2× faster with shorter, more concise outputs.

Performance Trade-offs:

- **Ministral-3:3b**: 71.4 tokens/sec, 621 avg tokens, 8.7 sec avg duration
- **DeepSeek-R1:7b**: 35.0 tokens/sec, 732 avg tokens, 21.0 sec avg duration

DeepSeek-R1's longer responses reflected its internal reasoning process (often includes explicit thinking before final answer), while Ministral-3 provided more direct responses. The 2.4× speed advantage of Ministral-3 is significant for real-time applications.

---

4. Discussion

4.1 Model Size vs. Architecture

Contrary to the conventional assumption that larger models universally outperform smaller ones, these results reveal a more nuanced picture. **Ministral-3:3b (3B parameters) matched or exceeded DeepSeek-R1:7b (7B parameters)** in overall performance (4.64 vs. 4.50), despite having less than half the parameter count.

This finding suggests that **architectural design and training methodology matter as much as—or more than—raw parameter count**. DeepSeek-R1's reasoning-specialized architecture excelled on formal reasoning tasks (logic, mathematics) but struggled with tasks requiring implicit world knowledge (common sense, ambiguity), where its internal reasoning chains may have introduced unnecessary complexity.

Implication: Practitioners should select models based on task characteristics rather than parameter count alone. Smaller, well-designed models may offer superior efficiency-performance trade-offs for many applications.

4.2 The Power of Few-Shot Learning

Few-shot prompting emerged as the most consistently effective strategy (mean: 4.86), reducing variance across tasks and models. This finding aligns with the seminal GPT-3 research (Brown et al., 2020) demonstrating that in-context learning enables models to adapt to new tasks without parameter updates.

Critical Success Factor: The quality of demonstration examples matters immensely. The development set was carefully curated to showcase desired format, reasoning depth, and response structure. Poor examples would likely diminish few-shot effectiveness.

Practical Recommendation: For production deployments, invest effort in creating high-quality few-shot examples tailored to specific use case. This "prompt programming" often yields better results than model fine-tuning for small-scale applications.

4.3 Chain-of-Thought: Context Matters

CoT prompting showed **high variance** (std: 1.00) and **no significant overall advantage** over zero-shot (t=0.35, p=0.73). However, task-level analysis revealed strong context dependence:

Where CoT Helped:

- Mathematical reasoning: Structured step-by-step decomposition improved accuracy
- Logical arguments: Explicit reasoning chains caught subtle logical errors
- Ethical dilemmas: Breaking down trade-offs led to more nuanced analysis

Where CoT Hurt:

- Common sense tasks: Over-thinking simple questions introduced errors
- Ambiguity identification: Premature commitment to single interpretation
- Tasks with straightforward answers: Added verbosity without benefit

Implication: CoT is not a universal solution. Apply it selectively to tasks requiring multi-step reasoning or complex decomposition. For straightforward tasks, zero-shot or few-shot may be more efficient and effective.

4.4 Reasoning Models: Specialized Tools

DeepSeek-R1's internal reasoning architecture performed exceptionally on **formal, structured tasks** (5.0 on logic, math, code) but showed **weaknesses on implicit knowledge tasks** (2.0 on zero-shot common sense). This suggests that explicit reasoning chains, while powerful for well-defined problems, may introduce brittleness when dealing with fuzzy, implicit knowledge.

Trade-off Analysis:

- **Advantages**: Transparent reasoning, excellent for complex analytical tasks, less prompt-sensitive
- **Disadvantages**: 2× slower, longer outputs, may over-complicate simple tasks

Use Case Fit:

- Choose reasoning models for: Legal analysis, scientific reasoning, complex problem decomposition
- Choose standard models for: Content generation, summarization, conversational applications

4.5 Task-Specific Insights

Perfect-Score Tasks (Logic, Code, Comprehension): Both models handled structured, verifiable tasks with deterministic correct answers effortlessly. These tasks align well with transformer architectures' strengths in pattern matching and rule application.

Challenging Tasks:

- **Common Sense (3.53)**: Models lack grounded physical experience. Zero-shot scored 2.0, indicating minimal innate common sense. Few-shot examples provided crucial grounding (4.33).
- **Ambiguity (3.93)**: Models exhibit "single interpretation bias"—they tend to commit to one reading rather than identifying multiple valid interpretations. This may reflect training objectives prioritizing confident, definitive answers.

Recommendation: For tasks requiring implicit knowledge or multiple perspectives, invest heavily in prompt engineering (examples, explicit instructions to consider alternatives).

4.6 Limitations and Threats to Validity

Sample Size: 50 experiments provide reasonable statistical power but limit fine-grained analysis. Some task-strategy-model combinations have n=1.

Single Annotator: While ensuring consistency, this introduces potential bias. Inter-rater reliability study (90% agreement) provides some confidence, but multiple annotators would strengthen validity.

Task Selection: The 10 tasks span diverse capabilities but cannot comprehensively cover the entire space of LLM applications. Results may not generalize to highly specialized domains (medical, legal).

Model Selection: Ministral-3 and DeepSeek-R1 represent specific design philosophies. Other models in similar parameter ranges (Phi-3, Gemma-2) might show different patterns.

Prompt Engineering: The prompts underwent iterative refinement but were not exhaustively optimized. Different prompt formulations could yield different results, especially for CoT.

Temperature Settings:Low temperature (0.1) was used for determinism. Higher temperatures for creative tasks might alter performance patterns.

4.7 Implications for Practice

For Researchers:

1. Model size is an incomplete proxy for capability—architecture and training matter
2. Prompting strategy effects are highly task-dependent; report disaggregated results
3. Manual evaluation, while labor-intensive, reveals nuances missed by automated metrics
4. Open-source models enable reproducible research at low cost

For Practitioners:

1. Start with few-shot prompting as default—it's robust across tasks
2. Apply CoT selectively to complex reasoning tasks; avoid over-applying
3. Smaller models may suffice for many applications—test before defaulting to largest available
4. Invest in high-quality demonstration examples; they're often higher ROI than model selection
5. Match model architecture to use case: reasoning models for analysis, standard models for generation

---

5. Conclusion

This systematic evaluation of LLMs across diverse tasks and prompting strategies yields several key insights:

1. **Parameter count $\neq$ performance**: A well-designed 3B model matched a 7B reasoning-specialized model, highlighting the importance of architecture and training.

2. **Few-shot prompting is robustly effective**: Consistently high performance (4.86/5.0) with low variance makes it the recommended default strategy.

3. **Chain-of-thought is task-dependent**: Excellent for complex reasoning (math, logic), but can introduce errors on straightforward tasks. Apply selectively.

4. **Reasoning models are specialized tools**: DeepSeek-R1 excelled on formal tasks but struggled with implicit knowledge, suggesting architectural trade-offs.

5. **Task characteristics matter**: Structured, verifiable tasks (code, logic) are easily handled by modern LLMs, while implicit knowledge (common sense) remains challenging.