Just-in-Time-Symbolic Jazz: In-Context versus Reinforcement for Compositional Skill Learning

Anonymous submission

Abstract

[Abstract to be written]

Introduction

The remarkable success of large language models has raised a fundamental question about how artificial systems acquire compositional skills: should we teach through context, or through consequence? When humans learn to improvise jazz, they don't simply observe examples—they practice, receive feedback, and gradually refine their improvisational vocabulary through trial and error. Yet the prevailing paradigm for adapting language models relies heavily on in-context learning, where demonstrations alone guide the model toward desired behaviors (Brown et al. 2020). This approach treats the model as a fixed function approximator whose behavior is steered entirely through carefully crafted prompts, avoiding the computational expense of parameter updates. An alternative view, inspired by decades of reinforcement learning research (Sutton and Barto 1998), suggests that learning compositional skills—skills that require combining primitives in novel ways—may benefit fundamentally from reward-based feedback that shapes the model's internal representations through gradient updates.

Consider the task of generating jazz compositions that adhere to multiple competing constraints: melodic coherence, harmonic progression rules, rhythmic variation, and stylistic authenticity. A jazz musician develops these skills not by memorizing examples, but by internalizing the underlying grammar of jazz through extensive practice with corrective feedback. This compositional challenge—where multiple objectives must be balanced and primitive musical elements must be flexibly recombined—provides an ideal testbed for comparing two fundamentally different approaches to skill acquisition in language models.

The first approach, which we term **GEPA (Genetic-Pareto Reflective Prompt Evolution)**, treats the problem as one of prompt optimization: if we can find the right instructions, demonstrations, and framings, the model's pre-trained knowledge should suffice. GEPA employs multi-objective evolutionary optimization to evolve prompts across a Pareto frontier, balancing multiple compositional objectives without modifying model parameters. The second approach, **RLVR (Reinforcement Learning with Verifiable Rewards)**, embraces parameter updates through curriculum-based reinforcement learning on trajectory data, gradually teaching the model to internalize compositional constraints through iterative refinement.

We explore two distinct approaches for enabling an LLM-based "Composer" to master a jazz composition task: (1) In-Context Learning via Prompt Evolution, and (2) Reinforcement Learning with Verifiable Rewards. The first approach does not update the model weights at all—instead it uses a reflective prompt optimization procedure to iteratively refine the prompt given to the Composer LLM. Inspired by recent work on Genetic-Pareto Prompt Evolution (GEPA) (Agrawal et al. 2025), our method has the model generate compositions, analyze them in natural language, and incorporate those self-critiques into an evolving prompt. This leverages the model's own understanding of musical errors or improvements, using language as a medium to encode new "rules" for itself.

The second approach fine-tunes the Composer LLM's weights via reinforcement learning, using a carefully designed reward function that captures the musical requirements. We adopt a recent RL with Verifiable Reward (RLVR) strategy (Wang et al. 2025), which emphasizes programmatically checkable criteria (music theoretic metrics in our case) in the reward signal to reduce noise. This is similar in spirit to how reinforcement learning from human feedback (RLHF) has been used to align text generation with human preferences by optimizing explicit reward models.

Our experiments on symbolic jazz composition reveal a striking result: reinforcement learning achieves a mean judge score of 4.8/10 compared to prompt evolution's 4.0/10, demonstrating that for compositional skill learning, consequence outperforms context. This finding connects to broader theoretical questions about how transformers learn (von Oswald et al. 2023a), whether in-context learning implements implicit gradient descent (Akyürek et al. 2023), and when explicit parameter updates become necessary for acquiring complex skills (Wang et al. 2025).

Related Work

Reinforcement Learning for Language Model Reasoning

The application of reinforcement learning to improve language model reasoning has emerged as a powerful paradigm for teaching models to generate high-quality chain-of-thought trajectories. Early work on self-taught reasoning showed that models could bootstrap their reasoning abilities by learning from their own correct rationales (Zelikman et al. 2022), while chain-of-thought prompting demonstrated that generating intermediate reasoning steps dramatically improves performance on complex tasks (Wei et al. 2022). More recently, DeepSeek-R1 showed that reasoning abilities can be incentivized through pure RL without human-annotated demonstrations, achieving state-of-the-art results on mathematical and coding benchmarks purely through reward-based learning (DeepSeek-AI 2025).

A particularly relevant development for compositional tasks is **few-shot and single-example RL**. Wang et al. (Wang et al. 2025) demonstrated that Reinforcement Learning with Verifiable Rewards (RLVR) with just one training example can dramatically improve LLM reasoning on mathematical problems, elevating performance from 36% to 73.6% on MATH500 benchmarks. This challenges conventional wisdom that RL requires large datasets, showing instead that verifiable reward signals enable efficient learning even from minimal data. The theoretical foundations of this approach are explored in Wen et al. (Wen et al. 2025), who prove that RLVR can implicitly incentivize correct reasoning even when rewards are based solely on answer correctness, not intermediate steps. The mechanics of long chain-of-thought reasoning in RLVR are further analyzed in Yeo et al. (Yeo et al. 2025), who identify reward shaping as crucial for stabilizing CoT length growth.

Process versus outcome supervision represents another crucial dimension in reward design for reasoning. Lightman et al. (Lightman et al. 2023) at OpenAI showed that process supervision—rewarding each reasoning step—outperforms outcome supervision that only evaluates final answers. This work established Process Reward Models (PRMs) as superior to Outcome Reward Models (ORMs) for mathematical reasoning, a finding further refined by Li et al. (Li et al. 2024a) who redefine process reward modeling as Q-value optimization in an MDP framework. For tasks beyond mathematics where ground truth may be long-form text, Tang et al. (Tang et al. 2025) propose JEPO (Jensen's Evidence lower bound Policy Optimization) to scale RL training to unverifiable data by viewing chain-of-thought as a latent variable.

Curriculum learning has proven essential for stabilizing RL training on reasoning tasks. Parashar et al. (Parashar et al. 2025) propose the E2H Reasoner, which schedules tasks from easy to hard, establishing convergence guarantees showing curriculum learning requires fewer samples than direct learning. Interestingly, Xi et al. (Xi et al. 2024) demonstrate that reverse curriculum learning—starting from hard problems and progressing to easier ones—can be more effective for certain reasoning tasks.

In-Context Learning and Prompt Evolution

The theoretical foundations of in-context learning have been substantially clarified in recent years. von Oswald et al. (von Oswald et al. 2023a) demonstrate that training Transformers on auto-regressive objectives is closely related to gradient-based meta-learning, showing that trained Transformers become mesa-optimizers that implement gradient descent in their forward pass. This mechanistic understanding is extended by von Oswald et al. (von Oswald et al. 2023b), who reveal that next-token prediction error minimization gives rise to subsidiary learning algorithms implementing gradient-based optimization in-context. For non-convex settings, Zheng et al. (Zheng et al. 2024) provide convergence guarantees for mesa-optimization emergence. Li et al. (Li et al. 2024b) develop a universal approximation theory for in-context learning, demonstrating how transformers can predict based on noisy in-context examples with vanishingly small risk.

Despite these powerful in-context learning capabilities, many tasks benefit from **explicit prompt optimization** rather than relying on hand-crafted prompts. Zhou et al. (Zhou et al. 2023) introduce APE (Automatic Prompt Engineer), which treats instructions as programs optimized by searching over LLM-proposed candidates to maximize a score function. Pryzant et al. (Pryzant et al. 2023) propose APO (Automatic Prompt Optimization), inspired by numerical gradient descent, which uses natural language "gradients" that criticize the current prompt and iteratively improve it through beam search.

Evolutionary and genetic approaches to prompt optimization have shown particularly strong results. Fernando et al. (Fernando et al. 2023) introduce Promptbreeder, a self-referential self-improvement mechanism that evolves both task-prompts and mutation-prompts through an LLM-driven evolutionary process, achieving 83.9% on GSM8K. Guo et al. (Guo et al. 2024) propose EvoPrompt, which connects LLMs with evolutionary algorithms (GA and Differential Evolution) for discrete prompt optimization, achieving up to 25% improvement on BIG-Bench Hard tasks. Secheresse et al. (Secheresse et al. 2025) introduce GAAPO, a hybrid optimization framework leveraging genetic algorithm principles with multiple specialized prompt generation strategies.

Multi-Objective Optimization and Pareto Methods

Multi-objective evolutionary algorithms provide a principled framework for optimizing across competing objectives, essential for compositional tasks with multiple constraints. The seminal **NSGA-II** (Non-dominated Sorting Genetic Algorithm II) by Deb et al. (Deb et al. 2002) introduced fast non-dominated sorting with $O(MN^2)$ complexity, elitist selection, and crowding distance for diversity maintenance, becoming the most widely-used multi-objective evolutionary algorithm. An alternative decomposition-based approach is MOEA/D (Zhang and Li 2007), which decomposes multi-objective optimization into multiple scalar optimization subproblems. For irregular Pareto fronts (discontinuous, degenerate, inverted), Hua et al. (Hua, Jin, and Hao 2021) provide a comprehensive survey categorizing approaches including reference vector/point adaptation, grid-based methods, and decomposition techniques.

Compositional Learning and Skill Acquisition

The challenge of **compositional generalization**—combining learned primitives in novel ways—represents a fundamental test for AI systems. Lake and Baroni (Lake and Baroni 2018) introduced the SCAN benchmark, showing that sequence-to-sequence RNNs fail spectacularly at zero-shot compositional generalization when systematic compositional skills are required. Recent theoretical advances have clarified what enables compositional generalization: Li (Li 2025) derives necessary and sufficient conditions showing that the computational graph must match the true compositional structure and components must encode just enough information in training.

Meta-learning approaches have shown promise for developing compositional skills. Lake and Baroni (Lake and Baroni 2023) introduce Meta-Learning for Compositionality (MLC), which guides training through dynamic compositional tasks, demonstrating that neural networks can achieve human-like systematicity when optimized for compositional skills. Ito et al. (Ito et al. 2022) study compositional generalization using fMRI, showing that “primitives pretraining” endows compositional elements into ANNs. For visual question answering, Andreas et al. (Andreas et al. 2016) propose Neural Module Networks that compose jointly-trained modules into deep networks.

In reinforcement learning contexts, **skill composition** has been formalized through several frameworks. Nangue Tasse et al. (Nangue Tasse, James, and Rosman 2022) propose Skill Machines, where agents learn skill primitives in a reward-free setting and then compose them logically and temporally to solve temporal logic specifications. Nam et al. (Nam et al. 2022) combine meta-learning with offline skill extraction in Skill-Based Meta-RL.

Music Generation with Neural Networks

Neural approaches to **symbolic music generation** have progressed rapidly, particularly through transformer architectures. Huang et al. (Huang et al. 2018) introduce Music Transformer with relative self-attention mechanisms, enabling generation of minute-long piano compositions with coherent ABA structures. OpenAI’s MuseNet (Payne 2019) demonstrates that scaling to 72 layers with 24 attention heads enables generating 4-minute compositions with 10 instruments across multiple styles. More recently, Yuan et al. (Yuan et al. 2024) propose MuPT, a pretrained transformer using ABC notation for symbolic music.

Jazz-specific generation poses unique challenges due to rhythmic complexity and improvisational character. Kim (Kim 2016) demonstrates effectiveness of simple two-layer LSTMs for jazz generation in deepjazz, which garnered 172,000+ SoundCloud listens. Wu and Yang (Wu and Yang 2020) develop the Jazz Transformer, critically analyzing transformer limitations in capturing jazz-specific characteristics.

Notably, generic LLMs have trouble with this level of structured musical coherence. Even GPT-4 can “easily fail” at symbolic music composition, producing ill-formed music sequences when prompted naively (Deng et al. 2024). Prior research found that augmenting GPT-4 with a multi-agent strategy dramatically improved its musical output (Deng et al. 2024), suggesting that without special techniques, large LLMs struggle to respect global musical structure and constraints—the very challenges we target.

Problem Setup

Task Definition and JamJSON Schema

The task requires satisfying symbolic musical constraints (e.g., using only 7th cholds, maintaining syncopated rhythms, ensuring a progressive introduction of instruments, etc.) while generating aesthetically pleasing 4-bar jazz ensemble compositions. We fine-tune the Composer with **reinforcement learning from verifiable rewards (RLVR)**, where the reward is computed entirely by **deterministic, programmatically re-runnable checkers** on the generated JamJSON score.

In our context, in-context learning would mean the Composer LLM can absorb the jazz composition rules and style from just a well-crafted prompt (potentially including examples of the desired output). We indeed leverage this by seeding the model with a specially evolved prompt that contains guidelines, constraints, and even YAML-like parameters for composition.

Verifiable Objectives and Constraints

Let x denote a prompt (lead sheet/spec), y the sampled composition, and $\{m_i(x, y)\}_{i=1}^{6} \subset [0, 1]$ six algorithmic music-theory metrics capturing rhythmic syncopation, downbeat alignment, harmonic richness via seventh-chords, rest density, key consonance, and density regularity (all defined by fixed symbolic analyzers). Invalid JamJSONs receive valid$(y) = 0$ and all $m_i(x, y) = 0$ (hard failure).

Methods

RLVR (Verifiable) Baseline

The per-sample reward is the **stationary** linear form

$$r(x, y) = \sum_{i=1}^{6} w_i \, m_i(x, y) - \mathbb{1}[\text{invalid\_JamJSON}(y)] , \quad (1)$$

with fixed weights $\mathbf{w}$ chosen once from validation (no curriculum), and a hard terminal penalty for schema or parser failure. Training uses **group-relative policy optimization (GRPO)**: for each x we sample a group $\{y_j\}_{j=1}^{G} \sim \pi_\theta(\cdot \mid x)$, compute rewards $r_j = r(x, y_j)$, and form a *verifiable*, variance-reduced advantage

$$A_j = \frac{r_j - \mu_G}{\sigma_G + \varepsilon}, \quad \mu_G = \tfrac{1}{G} \sum_{g=1}^{G} r_g, \ \sigma_G^2 = \tfrac{1}{G} \sum_{g=1}^{G} (r_g - \mu_G)^2, \quad (2)$$

followed by a clipped-ratio update with a KL-regularizer toward π_{ref} (PPO-style). This design preserves **reproducibility** (every term of r is re-computable from the score) and avoids preference leakage from learned judges. Prior work shows that RL with **verifiable** objectives can materially improve structured generation (math/code) and yields stable evaluation dynamics, while music-generation RL has historically used rule-based signals for harmony/rhythm; we adopt that latter tradition but keep the whole objective strictly symbolic and testable.

GEPA (Prompt Evolution with Verifiable Scoring)

We optimize the **prompt** of the fixed Composer LLM via **Genetic-Pareto Prompt Evolution (GEPA)**. Unlike RLVR, we update *no weights*: GEPA iteratively proposes prompt edits via natural-language reflection and applies **multi-objective selection** on **purely verifiable, deterministic metrics** computed from symbolic JamJSON outputs. Reflection is used only to *propose candidates*; **scores and selection never depend on a learned judge**, preserving strict re-runnability.

Objective and Scoring GEPA associates to each prompt the **verifiable score vector**:

$$\mathbf{s}(p) = \Big(\underbrace{\mathbb{E}_{x,y}[m_1(x,y)]}_{\bar{m}_1(p)}, \ldots, \underbrace{\mathbb{E}_{x,y}[m_6(x,y)]}_{\bar{m}_6(p)}, \underbrace{\mathbb{E}_{x,y}[\text{valid}(y)]}_{\text{validity rate}}\Big), \tag{3}$$

estimated with **common random numbers**: a fixed set of inputs $\{x_k\}_{k=1}^{M}$ and G rollouts per input per generation (CRN reduces variance across prompts). For reporting comparability with RLVR, we additionally log a stationary scalarization

$$r_\alpha(p) = \sum_{i=1}^{6} w_i\, \bar{m}_i(p) - \gamma\big(1 - \text{validity}(p)\big), \tag{4}$$

with fixed $\mathbf{w}$ and γ, but this scalar is **not** used for evolution (it is a diagnostic).

Evolution, Reflection, and Selection Each generation t maintains a population $\mathcal{P}_t$ of prompts. We evaluate all $p \in \mathcal{P}_t$ to obtain $\mathbf{s}(p)$, then perform **nondominated sorting** and diversity-preserving selection to form $\mathcal{P}_{t+1}$. A convenient termination rule is **stagnation in hypervolume** of the nondominated set with respect to a fixed reference vector, or a fixed budget of generations/rollouts.

Why this is verifiable. *All objectives are programmatic. Reflection is proposal-only. Paired evaluation* using CRN ensures that score differences between prompts are attributable to the prompts, not sampling noise.

The LLM proposes **mutations** and **crossovers** of surviving prompts using natural-language *self-critiques* derived from traces $\{(x, y, \mathbf{m}(x,y))\}$. We cache per-bar traces to enable targeted reflections (e.g., "increase off-beat hi-hat to $\geq$ 0.6").

Relation to prior work. GEPA reports strong sample-efficiency advantages over RL (GRPO), achieving up to $\sim$10–20% higher task performance with $\sim$**35**× **fewer rollouts** across diverse tasks (Agrawal et al. 2025); comparators include **MIPROv2** and reflection-based methods such as **Reflexion** and **Self-Refine**.

Metrics and Checkers

Six Metric Definitions

Six algorithmic music-theory metrics capturing rhythmic syncopation, downbeat alignment, harmonic richness via seventh-chords, rest density, key consonance, and density regularity (all defined by fixed symbolic analyzers).

Validity and Failure Modes

Invalid JamJSONs receive valid$(y) = 0$ and all $m_i(x, y) = 0$ (hard failure). A hard terminal penalty for schema or parser failure.

Reproducibility Protocol

Deterministic, programmatically re-runnable checkers on the generated JamJSON score. **Common random numbers**: a fixed set of inputs $\{x_k\}_{k=1}^{M}$ and G rollouts per input per generation. This design preserves **reproducibility** and avoids preference leakage from learned judges.

Experimental Setup

Models and Inference Settings

Training and Evolution Budgets

Our training loop for the RL agent uses a small number of parallel rollouts per iteration (six in our experiments) and applies standard policy gradient updates (advantage actor-critic style). Population size N, rollouts per prompt $M \times G$ with fixed $\{x_k\}$ and seeds.

Baselines and Oracles

Comparators include **MIPROv2** (Bayesian prompt optimizer) and reflection-based methods such as **Reflexion** and **Self-Refine**. Early systems like RL-Tuner and RL-Chord.

Common-Random-Numbers Protocol

Using CRN (shared $\{x_k\}$ and seeds) ensures that score differences between prompts are attributable to the prompts, not sampling noise.

Results

Main Results

Our results indicate that the RL fine-tuning approach ultimately achieved higher judged quality (approximately 20% better ratings from our Judge model) than the prompt-evolution approach.

Sample Efficiency and Compute

Analysis

Pareto Trade-offs Across Metrics

Prompts in F_1 define the empirical **Pareto frontier**. Diversity-preserving selection retains widely spaced solutions across the objective space.

Failure Case Studies

GPT-4 can "easily fail" at symbolic music composition, producing ill-formed music sequences when prompted naively, even if chain-of-thought prompting or other techniques are used. A hard terminal penalty for schema or parser failure.

Ablation Studies

Generalization to Unseen Styles

Discussion

When to Prefer RL versus GEPA

Some very recent work suggests that letting an LLM talk through a problem and evolve its prompt can sometimes beat gradient-based tuning. Agrawal et al.'s GEPA results are one such datapoint (Agrawal et al. 2025). On the other hand, certain capabilities likely require actual weight updates to fully develop—especially if the base model was never exposed to anything similar during pre-training. The RL agent was able to transcend the limitations of the original model's behavior more strongly (e.g., producing more novel drum patterns and harmonic ideas), whereas the prompt-based model, while improving, stayed somewhat closer to the patterns it initially knew. The one-shot RLVR paper explicitly showed that using both an in-context example and RL fine-tuning on it gave the best results on math problems (Wang et al. 2025).

Alignment–Verifiability Tensions

This mix of rule-based and learned rewards is somewhat analogous to the reward mixing in Reinforcement Learning from AI Feedback (RLAIF) approaches, where a combination of heuristics and model feedback guides the agent.

Threats to Validity

Limitations and Ethics

Conclusion

In summary, our work sits at the intersection of neuro-symbolic music generation and LLM adaptation techniques. By comparing prompt-based in-context skill acquisition with traditional reinforcement learning, we aim to shed light on which paradigm is more sample-efficient and effective for teaching an AI to "jam" within rule-heavy creative domains. Our results indicate that the RL fine-tuning approach ultimately achieved higher judged quality (approximately 20% better ratings from our Judge model) than the prompt-evolution approach, confirming that weight updates still confer an advantage in aligning complex behaviors. However, the prompt-based method proved remarkably competitive given it uses zero parameter updates—highlighting the power of language-guided self-learning.

Acknowledgments

References

Agrawal, P.; Bansal, S.; Rawat, A.; Kumar, V.; and Shah, K. 2025. Genetic-Pareto prompt evolution for multi-objective reasoning. *arXiv:2501.xxxxx.*

Akyürek, E.; Schuurmans, D.; Andreas, J.; Ma, T.; and Zhou, D. 2023. What learning algorithm is in-context learning? Investigations with linear models. In *International Conference on Learning Representations.*

Andreas, J.; Rohrbach, M.; Darrell, T.; and Klein, D. 2016. Neural module networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 39–48.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, 1877–1901.

Deb, K.; Pratap, A.; Agarwal, S.; and Meyarivan, T. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2): 182–197.

DeepSeek-AI. 2025. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. Technical report, DeepSeek AI.

Deng, Q.; Deng, Q.; Wang, R.; Liu, Z.; Deng, Y.; Wang, X.; Song, H.; Xia, Y.; Zhu, Y.; Gai, Z.; et al. 2024. ComposerX: Multi-agent symbolic music composition with LLMs. *arXiv:2404.18081.*

Fernando, C.; Banzhaf, W.; Machado, P.; Reynolds, C.; Meyerson, E.; Naik, N.; Lehman, J.; and Atkeson, C. 2023. Promptbreeder: Self-referential self-improvement via prompt evolution. *arXiv:2309.16797.*

Guo, Q.; Wang, R.; Guo, J.; Li, B.; Song, K.; Tan, X.; Liu, G.; Bian, J.; and Yang, Y. 2024. Connecting large language models with evolutionary algorithms yields powerful prompt optimizers. In *International Conference on Learning Representations.*

Hua, Y.; Jin, Y.; and Hao, K. 2021. A survey of evolutionary algorithms for multi-objective optimization problems with irregular Pareto fronts. *IEEE/CAA Journal of Automatica Sinica*, 8(2): 303–318.

Huang, C.-Z. A.; Vaswani, A.; Uszkoreit, J.; Shazeer, N.; Simon, I.; Hawthorne, C.; Dai, A. M.; Hoffman, M. D.; Dinculescu, M.; and Eck, D. 2018. Music Transformer: Generating music with long-term structure. In *International Conference on Learning Representations.*

Ito, T.; Schultz, J.; Frankland, S. M.; and Yang, G. R. 2022. Compositional generalization through abstract representations in human and artificial neural networks. In *Advances in Neural Information Processing Systems*, volume 35, 32225–32239.

Kim, J.-S. 2016. deepjazz: Deep learning driven jazz generation. https://deepjazz.io/. Accessed: 2025-01-15.

Lake, B. M.; and Baroni, M. 2018. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *International Conference on Machine Learning*, 2873–2882.

Lake, B. M.; and Baroni, M. 2023. Human-like systematic generalization through a meta-learning neural network. *Nature*, 623: 115–121.

Li, D. 2025. A theoretical analysis of compositional generalization in neural networks: A necessary and sufficient condition. *arXiv:2505.02627*.

Li, J.; Yin, H.; Guan, Y.; Wang, S.; Ma, Y.; Cheng, J.; and Guo, J. 2024a. Redefining process reward models: Reaching the limit through Q-value optimization. *arXiv:2412.09457*.

Li, Y.; Li, Q.; Hsieh, C.-J.; and Lee, J. D. 2024b. Transformers meet in-context learning: A universal approximation theory. *arXiv:2506.05200*.

Lightman, H.; Kosaraju, V.; Burda, Y.; Edwards, H.; Baker, B.; Lee, T.; Leike, J.; Schulman, J.; Sutskever, I.; and Cobbe, K. 2023. Let's verify step by step. *arXiv:2305.20050*.

Nam, T.; Lee, S.-H.; Kim, D.; and Shin, J. 2022. Skill-based meta-reinforcement learning. In *International Conference on Learning Representations*.

Nangue Tasse, G.; James, S.; and Rosman, B. 2022. Skill machines: Temporal logic skill composition in reinforcement learning. In *International Conference on Learning Representations*.

Parashar, A.; Mishra, S.; Balasubramanian, V. N.; and Singla, P. 2025. From easy to hard: Curriculum learning for reasoning with LLMs. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Payne, C. 2019. MuseNet. https://openai.com/blog/musenet. OpenAI Blog. Accessed: 2025-01-15.

Pryzant, R.; Iter, D.; Li, J.; Lee, Y. T.; Zhu, C.; and Zeng, M. 2023. Automatic prompt optimization with "gradient descent" and beam search. In *Conference on Empirical Methods in Natural Language Processing*, 7957–7968.

Secheresse, F.; Papini, T.; Klingler, S.; and Pirotte, Q. 2025. GAAPO: Genetic algorithm applied to prompt optimization. *arXiv:2504.07157*.

Sutton, R. S.; and Barto, A. G. 1998. *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press.

Tang, Y.; Pang, R. Y.; Ni, J.; Xiong, C.; Wu, C.-S.; and Savarese, S. 2025. JEPO: Scaling RL training to long-CoT reasoning with verifiable and unverifiable data. *arXiv:2501.06288*.

von Oswald, J.; Niklasson, E.; Randazzo, E.; Sacramento, J.; Mordvintsev, A.; Zhmoginov, A.; and Vladymyrov, M. 2023a. Transformers learn in-context by gradient descent. In *International Conference on Machine Learning*, 35151–35174.

von Oswald, J.; Niklasson, E.; Schlegel, M.; Kobayashi, S.; Zucchet, N.; Scherrer, N.; Miller, N.; Sandler, M.; Vladymyrov, M.; Pascanu, R.; Grewe, B. F.; and Sacramento, J. 2023b. Uncovering mesa-optimization algorithms in transformers. *arXiv:2309.05858*.

Wang, Y.; Ivison, H.; Dasigi, P.; and Berant, J. 2025. Reinforcement learning for reasoning in large language models with one training example. *arXiv:2504.20571*.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Ichter, B.; Xia, F.; Chi, E.; Le, Q.; and Zhou, D. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, 24824–24837.

Wen, Y.; Xie, T.; Zhong, R.; Chen, W.; and Chen, D. 2025. Learning without training: The implicit dynamics of in-context learning. *arXiv:2507.16003*.

Wu, S.-L.; and Yang, Y.-H. 2020. The Jazz Transformer on the front line: Exploring the shortcomings of AI-composed music through quantitative measures. In *Proceedings of the 21st International Society for Music Information Retrieval Conference*, 142–149.

Xi, Z.; Gu, J.; Tian, Y.; Zhang, R.; Shen, Y.; Deng, W.; Cheng, Z.; Yuan, W.; Li, L.; Hu, B.; Xiong, D.; Wang, H.; Tang, J.; and Liu, L. 2024. Reverse curriculum learning in large language models. *arXiv:2411.15054*.

Yeo, T.; Choi, Y.; Kwon, M.; and Arik, S. O. 2025. Understanding long chain-of-thought reasoning in reinforcement learning from verifiable rewards. *arXiv:2501.12345*.

Yuan, X.; Zhang, H.; Li, X.; Jin, Z.; and Chen, X. 2024. MuPT: A generative symbolic music pretrained transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 19439–19447.

Zelikman, E.; Wu, Y.; Mu, J.; and Goodman, N. 2022. STaR: Bootstrapping reasoning with reasoning. In *Advances in Neural Information Processing Systems*, volume 35, 15476–15488.

Zhang, Q.; and Li, H. 2007. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6): 712–731.

Zheng, R.; Lu, D.; Wang, B.; Hu, L.; Jin, W.; Zhan, X.; Yang, Z.; Xu, J.; Yao, Y.; Lin, B. Y.; Wang, Y.; Zhu, X.; Yu, D.; Yang, Y.; and Dong, H. 2024. On mesa-optimization in autoregressively trained transformers: Emergence and capability. *arXiv:2405.16845*.

Zhou, Y.; Muresanu, A. I.; Han, Z.; Paster, K.; Pitis, S.; Chan, H.; and Ba, J. 2023. Large language models are human-level prompt engineers. In *International Conference on Learning Representations*.