

JOLT vs GRPO: Matching MATH Accuracy with Fewer Tokens

TechRounder PDF Edition

Live article: <https://www.techrounder.com/ai/jolt-joint-on-policy-learning-teaching-token-efficiency/>

By Vipin PG | Published October 23, 2026 | Updated October 23, 2026 | Format: Comparison | 8 min read

Bottom line

JOLT reports matching GRPO's MATH accuracy with fewer tokens, but its 6.5× result comes from one experiment and a fitted curve.

JOLT reports matching GRPO's accuracy on the MATH benchmark while processing 6.5 times fewer training tokens. The figure comes from a fitted token-budget curve in a research paper, using Qwen3-0.6B. It measures token efficiency under the authors' experimental setup, rather than actual computing costs or training time. The result has not been independently replicated.

Joint On-policy Learning and Teaching (JOLT) trains a language model to act as both teacher and student. The teacher receives extra information about a task, while the student works without it. By training these two roles together, the researchers aim to give the student useful feedback at individual tokens instead of relying entirely on a reward for the final answer.

The paper evaluates two versions, JOLT and JOLT+. Their differences are important because the reward-augmented version performs substantially better on some benchmarks.

Why compare JOLT vs GRPO?

GRPO learns from rewards assigned to completed attempts. JOLT introduces token-level feedback from a teacher that has access to additional information.

When a model solves a problem, a final reward indicates whether its answer was successful. It does not directly identify which intermediate decisions were helpful. This becomes a limitation on difficult tasks where successful attempts are relatively uncommon or require lengthy responses.

The JOLT research paper investigates whether training a teacher and student together can make better use of the generated training data. Its experiments cover mathematical reasoning, coding, tool use and terminal tasks. However, the reported 6.5-fold token-efficiency improvement comes specifically from the MATH experiment with Qwen3-0.6B.

Understanding the method requires looking at how on-policy distillation works and why training the teacher may affect the feedback the student receives.

How on-policy distillation works

In on-policy distillation, the student generates its own responses. A teacher evaluates the tokens along those responses, allowing the student to receive feedback on the decisions it actually made.

This differs from conventional distillation, where a student may learn by imitating completed responses generated separately by a teacher. On-policy feedback follows the student's own attempts, including states and token sequences that might not appear in a collection of teacher-generated examples.

On-policy self-distillation (OPSD) goes further by using a single model for both roles. The teacher receives privileged information, such as a verified solution, while the student receives only the original task. The OPSD paper describes training through token-level divergence between the two roles on student-generated rollouts.

There is a potential problem with privileged information. A teacher that already knows the solution may produce token probabilities that are difficult for a student without that information to learn from. The teacher can be good at solving the task while providing poor guidance for the student's current capabilities.

JOLT addresses this by training the privileged teacher role as part of the same learning process. Its objective encourages successful teacher behavior while keeping the teacher's token distribution close to the student's policy.

What GRPO measures

Group Relative Policy Optimization (GRPO) is a reinforcement-learning method introduced in the DeepSeekMath paper.

GRPO generates multiple responses for a prompt and compares their rewards within a group. Each response receives an advantage estimate based on how its return compares with the group's average return.

In reinforcement learning, an advantage indicates how favorable an action or attempt is relative to a baseline. GRPO uses group-relative rewards to estimate advantages without requiring a separate value model.

The learning signal still depends on outcomes assigned to completed trajectories. Although GRPO updates token probabilities during training, the reward does not directly explain the contribution of each intermediate token.

JOLT uses a privileged teacher to provide that additional token-level signal. GRPO remains an experimental baseline, allowing the researchers to compare the two approaches under their selected training configurations.

How JOLT trains its teacher

JOLT uses one language model with shared parameters for the teacher and student roles. Each role generates trajectories under a different context. The teacher sees privileged information, while the student attempts the task using only the information normally available at inference time.

The teacher's training objective combines an outcome-based reward with token-level Kullback-Leibler (KL) regularization.

The reward encourages the teacher to generate successful responses. KL regularization penalizes excessive differences between the teacher's token distribution and the student's current policy. Together, these terms are intended to produce a teacher that can solve tasks while remaining useful to the student.

The researchers also derive a local condition under which the teacher's distillation guidance points in the same direction as the student's reward gradient. This analysis motivates the combined teacher objective. It does not guarantee that every update during joint training improves the student's performance.

Meanwhile, the student generates its own responses without privileged information. The teacher evaluates the token probabilities along those student-generated trajectories, providing dense feedback at the states the student encountered.

Because the roles share model parameters, updates affect the policy used by both teacher and student. As training progresses, the teacher changes alongside the student.

This shared training process is the basis of JOLT's approach. The paper also introduces a second variant that adds outcome rewards directly to the student's training objective.

JOLT, JOLT+ and related methods

Plain JOLT trains the student using dense distillation feedback from the privileged teacher. It does not add a direct outcome reward to the student's trajectories.

JOLT+ retains the same teacher objective but also incorporates the student's own centered outcome reward into its update. This gives the student another learning signal alongside the teacher's token-level guidance.

The distinction affects how the experimental results should be interpreted. Improvements achieved by JOLT+ cannot automatically be attributed to plain JOLT because the two variants use different student objectives.

JOLT also differs from other forms of self-distillation.

OPSD uses privileged and unprivileged contexts within a shared policy, but it does not optimize the privileged teacher role using JOLT's reward-plus-KL objective.

Another related approach, π -Distill, jointly trains teacher and student roles within one model. According to the π -Distill paper, its student update relies on teacher-generated trajectories. JOLT instead applies distillation feedback along trajectories generated by the student.

These methods share the goal of improving the feedback available during training, but they differ in how examples are generated and how the teacher is optimized.

What the experiments tested

The JOLT paper reports experiments involving mathematical reasoning, code generation, tool use and terminal-based tasks. The researchers use different models and forms of privileged information depending on the benchmark.

For MATH, the experiments use Qwen3-0.6B, with worked reference solutions provided to the teacher. The GSM8K setup uses DeepSeek-V2-Lite, whose teacher receives verified answers.

On LiveCodeBench v6, the experiments use Qwen3-8B. The teacher receives successful solutions or execution feedback, giving it additional information about the coding tasks.

The AppWorld experiments use Qwen3-4B-Instruct, with summaries of earlier attempts supplied as privileged context. AppWorld tests agents that interact with multiple applications and evaluates their behavior through state-based tests, as described in the AppWorld benchmark paper.

For Terminal-Bench 2.1, the researchers train Qwen3-30B-A3B-Instruct on TMax tasks. In this setup, the teacher has access to private verifier information.

The authors report tuning hyperparameters for the methods being compared. The published results therefore reflect specific training configurations and experimental procedures. They do not establish that every method received an identical hyperparameter-search budget or that independently implemented versions would produce the same rankings.

The main token-efficiency claim comes from the Qwen3-0.6B MATH experiment. Results from the other benchmarks provide evidence about the method's behavior across tasks, but they do not establish the same efficiency ratio.

What the token-efficiency claim actually measures

The paper reports that plain JOLT reaches a matched accuracy on MATH using 6.5 times fewer processed training tokens than GRPO. It also reports approximately 13 times fewer completion tokens.

These figures represent different measures of token usage.

Processed training tokens are the broader accounting measure used in the paper for tokens handled during training. Completion tokens count generated responses. Neither measure directly establishes how much accelerator time, energy or money a training run requires.

Practical training costs also depend on the hardware, batch sizes, token-scoring overhead and implementation. A method that processes fewer tokens may still require additional computation for other parts of its training objective.

The 6.5-fold figure also comes from a fitted curve rather than a direct comparison of measured computing costs.

The researchers fit a sigmoid relationship between accuracy and the logarithm of training-token budgets. They then use the fitted curves to estimate the token budgets required by each method at selected accuracy targets.

That analysis produces the reported 6.5-fold difference at a matched accuracy. The paper also reports different savings at other accuracy targets, so the ratio should not be treated as a fixed improvement across training budgets.

The result supports a narrower conclusion: under the authors' Qwen3-0.6B MATH setup and curve-fitting procedure, JOLT shows substantially better reported token efficiency than GRPO. Equivalent savings in actual training costs remain unestablished.

Results and limitations

On MATH, plain JOLT achieves a reported mean accuracy of 72.14%, compared with 68.36% for GRPO. JOLT+ records 72.59%.

Plain JOLT also achieves a higher reported mean accuracy than GRPO on LiveCodeBench v6. However, the results are less consistent on the agent-based benchmarks.

On AppWorld, plain JOLT scores 51.92%, below GRPO's 62.20%. JOLT+ performs better than both, reaching 72.09%.

Terminal-Bench shows a similar difference between the two JOLT variants. Plain JOLT trails GRPO at the reported pass@1 and pass@32 settings, while JOLT+ exceeds GRPO at both.

These results suggest that the value of dense teacher feedback depends on the task and the student's training objective. In some settings, adding a direct student outcome reward appears to improve performance substantially.

The findings also limit how broadly the method's performance can be described. Plain JOLT does not outperform GRPO across every benchmark in the paper, and some of the stronger results belong specifically to JOLT+.

The study is an author-reported preprint without independent replication. It covers several model sizes and tasks, but the 6.5-fold training-token comparison applies to Qwen3-0.6B on MATH, using the researchers' token accounting and fitted budget curves.

The paper does not establish equivalent reductions in accelerator-hours, electricity consumption or financial cost. It also leaves open how well the method's efficiency gains extend to larger models and production training workloads.

JOLT provides a way to train a privileged teacher and an unprivileged student within the same language model. Its reported MATH results offer evidence that dense token-level feedback can improve training efficiency under the tested conditions. Independent replication, direct matched-budget experiments and detailed compute accounting would help determine whether those improvements translate into lower training costs elsewhere.

References

1. arxiv.org - html / 2610.10447v1 - <https://arxiv.org/html/2610.10447v1>
2. arxiv.org - abs / 2601.18734 - <https://arxiv.org/abs/2601.18734>
3. arxiv.org - abs / 2402.03300 - <https://arxiv.org/abs/2402.03300>
4. arxiv.org - abs / 2602.04942 - <https://arxiv.org/abs/2602.04942>
5. arxiv.org - abs / 2407.18901 - <https://arxiv.org/abs/2407.18901>