Improving Extractive QA with Supervised Fine-Tuning and Group Relative Policy Optimization on MLQA

Martin Stamenov
Faculty of Computer Science and Engineering
Ss. Cyril and Methodius University in Skopje
martin.stamenov@students.finki.ukim.mk

***Abstract*—We present a three-stage pipeline for extractive question answering that combines retrieval-augmented generation (RAG), supervised fine-tuning (SFT), and Group Relative Policy Optimization (GRPO). Starting from `flan-t5-small` (80M parameters), we first fine-tune on the English split of the MLQA dataset, then apply GRPO—a reward-based reinforcement learning method that avoids training a separate reward model by using group-relative advantage estimation. Our approach improves F1 from 0.624 (base) to 0.668 (GRPO-best), a relative gain of 7.1%. We observe that GRPO exhibits non-monotonic validation performance, with the best checkpoint occurring early in training (step 200 of 774), highlighting the importance of periodic evaluation and early stopping in RL-based fine-tuning. We additionally evaluate zero-shot cross-lingual transfer on Spanish, German, and Macedonian, finding that English-only GRPO training transfers partially to typologically close languages (Spanish F1=0.425, German F1=0.290) but poorly to distant ones (Macedonian F1=0.039).**

I. Introduction

Extractive question answering (QA) is the task of selecting a text span from a given context that answers a posed question. While large language models have achieved strong performance on QA benchmarks such as SQuAD [1], smaller models remain attractive for deployment due to lower computational costs. The MLQA benchmark [2] provides a multilingual evaluation setting with parallel QA pairs across seven languages, enabling assessment of both monolingual and cross-lingual transfer capabilities.

Recent work has shown that reinforcement learning from human feedback (RLHF) can substantially improve language model outputs [3], [4]. However, RLHF requires training a separate reward model, which adds complexity and computational cost. Group Relative Policy Optimization (GRPO), introduced by Shao et al. [5], simplifies this by computing advantages from groups of sampled outputs relative to each other, using a task-specific reward function directly.

In this work, we apply a three-stage pipeline to `flan-t5-small`, a T5-based instruction-tuned model with 80M parameters [6], [7]:

1) **SFT**: Standard supervised fine-tuning on MLQA English QA pairs.
2) **GRPO**: Reinforcement learning using token-level F1 and exact match as the reward signal, with clipped surrogate objectives and KL regularization.
3) **RAG inference**: At test time, we optionally augment the input with retrieved context passages using dense retrieval [8], [9].

Our main contributions are: (1) we demonstrate that GRPO can improve a small model's extractive QA performance beyond SFT alone; (2) we provide a detailed analysis of GRPO training dynamics, including reward progression, KL divergence, and clip fraction behavior; and (3) we evaluate cross-lingual zero-shot transfer across four languages.

II. Related Work

1) Multilingual QA.: MLQA [2] is a benchmark for evaluating extractive QA across seven languages, constructed by crowd-sourcing parallel annotations on Wikipedia passages. Unlike translation-based approaches, MLQA contains naturally occurring questions, making it a challenging testbed for cross-lingual generalization.

2) Instruction-Tuned Models.: The Flan-T5 family [6] applies instruction tuning to the T5 architecture [7], training on a mixture of over 1,800 tasks phrased as natural language instructions. Even the smallest variant (`flan-t5-small`, 80M parameters) shows meaningful zero-shot and few-shot capabilities.

3) RL for Language Models.: Proximal Policy Optimization (PPO) [10] has been widely adopted for RLHF [3]. Direct Preference Optimization (DPO) [11] avoids explicit reward modeling but requires pairwise preference data. GRPO [5] takes a different approach: for each prompt, it generates a group of K candidate outputs, scores them with a task-specific reward, and computes advantages relative to the group mean. This eliminates the need for both a reward model and pairwise preference data.

4) Retrieval-Augmented Generation.: RAG [9] combines parametric knowledge in language models with non-parametric retrieval from an external corpus. Dense Passage Retrieval (DPR) [8] uses dual encoders for efficient retrieval. In our pipeline, we use Sentence-BERT embeddings [12] for context retrieval at inference time.

III. Method

Our pipeline consists of two training phases and an optional retrieval-augmented inference stage.

A. Phase 1: Supervised Fine-Tuning

We fine-tune `flan-t5-small` on the MLQA English validation split (1,148 samples, 90/10 train/eval split) using standard cross-entropy loss. The input is formatted as:

```
Answer the question based on the context.
Context: {context} Question: {question}
```

with the target being the gold answer span. We train for 3 epochs with AdamW (learning rate 5×10^{-5}, batch size 8, linear warmup over 10% of steps, gradient clipping at 1.0).

B. Phase 2: Group Relative Policy Optimization

Starting from the SFT checkpoint, we apply GRPO following the formulation of Shao et al. [5]. For each training prompt x, we sample a group of $K = 4$ candidate outputs $\{o_1, \ldots, o_K\}$ from the current policy π_θ using temperature sampling ($T = 0.8$, top-$p = 0.9$).

1) Reward Function.: Each candidate o_i is scored against the gold answer a using a composite reward:

$$r(o_i, a) = 0.7 \cdot \mathrm{F1}(o_i, a) + 0.3 \cdot \mathrm{EM}(o_i, a) - \lambda \cdot \max(0, |o_i| - L) \tag{1}$$

where F1 is token-level F1, EM is exact match, $\lambda = 0.001$ is a length penalty coefficient, and $L = 64$ is the maximum desired output length.

2) Advantage Estimation.: Advantages are computed via z-score normalization within each group:

$$\hat{A}_i = \frac{r_i - \mu_G}{\sigma_G + \epsilon} \tag{2}$$

where μ_G and σ_G are the mean and standard deviation of rewards within the group, and $\epsilon = 10^{-8}$ for numerical stability.

3) Policy Loss.: We use a clipped surrogate objective inspired by PPO [10]:

$$\mathcal{L}_{\text{policy}} = -\mathbb{E}\left[\min\left(\rho_i \hat{A}_i,\ \mathrm{clip}(\rho_i, 1 \pm \epsilon_c)\hat{A}_i\right)\right] \tag{3}$$

where $\rho_i = \pi_\theta(o_i|x)/\pi_{\theta_{\text{old}}}(o_i|x)$ is the importance sampling ratio and $\epsilon_c = 0.2$ is the clipping threshold.

4) KL Regularization.: To prevent the policy from diverging too far from the reference model (a frozen copy of the SFT checkpoint), we add a KL penalty:

$$\mathcal{L}_{\text{KL}} = \beta \cdot D_{\text{KL}}\left(\pi_\theta \,\|\, \pi_{\text{ref}}\right) \tag{4}$$

where $\beta = 0.04$. The full KL divergence is computed at the token level using a forward pass through both models, then averaged over non-padding tokens.

5) Total Loss.: The final training objective is:

$$\mathcal{L} = \mathcal{L}_{\text{policy}} + \mathcal{L}_{\text{KL}} \tag{5}$$

We train for 3 epochs with AdamW (learning rate 10^{-5}, batch size 4, cosine annealing schedule), evaluating every 200 steps and saving the best checkpoint based on validation F1.

C. RAG Inference

At inference time, we optionally augment the input context using dense retrieval. Given a question, we encode it with Sentence-BERT (`all-MiniLM-L6-v2`) [12], retrieve the top-k ($k = 3$) most similar passages from a chunked context store (chunk size 256 tokens), and prepend them to the model input.

IV. Experimental Setup

1) Dataset.: We use the MLQA benchmark [2]. For training, we use the English-English validation split (1,148 examples), divided into 1,033 training and 115 evaluation samples (90/10 split, seed 42). For final evaluation, we sample 500 examples from the English-English test split (11,590 total). Cross-lingual evaluation uses 100 samples each from German-German (de-de), Spanish-Spanish (es-es), and Macedonian-Macedonian (mk-mk, obtained via machine translation from English).

2) Models.: We compare four checkpoints:

- **Base**: `flan-t5-small` without any fine-tuning.
- **SFT**: After supervised fine-tuning (3 epochs).
- **GRPO-best**: Best validation checkpoint during GRPO training (step 200).
- **GRPO-final**: Final checkpoint after GRPO training concludes (step 774, early stopping triggered).

3) Metrics.: We report token-level F1 and Exact Match (EM), computed after text normalization (lowercasing, article removal, punctuation stripping), consistent with the SQuAD evaluation protocol [1].

4) Hardware.: All experiments are conducted on a single NVIDIA GPU. Mixed-precision training (AMP) was disabled as it produced NaN losses with `flan-t5-small`; all training uses FP32 precision.

V. Results

A. Main Results

Table I presents the performance of each model variant on the English-English MLQA test set.

TABLE I
PERFORMANCE ON MLQA ENGLISH-ENGLISH TEST SET ($n = 500$). BEST RESULTS IN **BOLD**.

Model	F1	EM	Len	Δ F1
Base (flan-t5-small)	0.624	0.508	3.37	—
+ SFT	0.633	0.516	2.79	+1.4%
+ GRPO (best, step 200)	**0.668**	**0.548**	3.17	+7.1%
+ GRPO (final, step 774)	0.651	0.534	3.00	+4.3%

SFT provides a modest improvement of 1.4% relative F1 gain over the base model. GRPO-best achieves the strongest results, with a 7.1% relative improvement in F1 and a 4-point absolute improvement in EM over base. Notably, the final GRPO checkpoint performs *worse* than the best checkpoint, with a decline of 1.7 F1 points from step 200 to step 774. This non-monotonic behavior is characteristic of RL-based fine-tuning, where continued optimization of the reward signal can lead to reward hacking or distribution drift.

B. Cross-Lingual Transfer

Table II reports zero-shot cross-lingual transfer results using the GRPO-best model, which was trained exclusively on English data.

TABLE II
ZERO-SHOT CROSS-LINGUAL TRANSFER OF GRPO-BEST.

Language	F1	EM	n
English (en-en)	0.668	0.548	500
Spanish (es-es)	0.425	0.240	100
German (de-de)	0.290	0.170	100
Macedonian (mk-mk)	0.039	0.020	100

Spanish, as a Romance language with substantial representation in the Flan-T5 pretraining data, retains 63.6% of the English F1. German retains 43.4%. Macedonian, which uses Cyrillic script and has minimal representation in the pretraining corpus, achieves near-zero performance (F1=0.039), demonstrating the limitations of cross-lingual transfer for under-represented languages in small models.

C. Training Dynamics

Figure 1 shows the training pipeline overview. SFT loss decreases smoothly over 3 epochs, while the GRPO phase shows more complex dynamics.

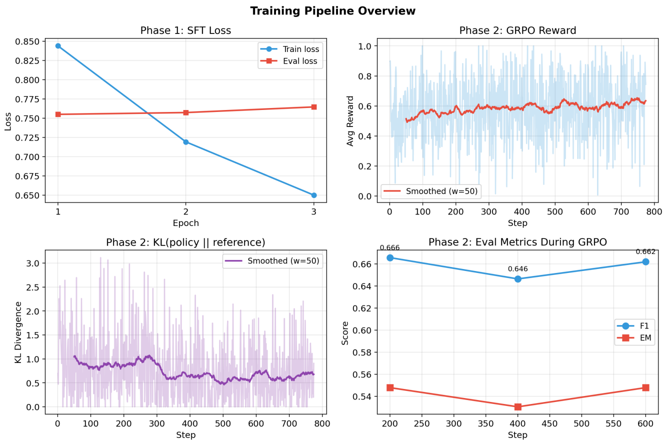

Fig. 1. Training pipeline overview. **Top left:** SFT train/eval loss over 3 epochs. **Top right:** GRPO average reward per step (smoothed). **Bottom left:** KL divergence between the policy and reference model. **Bottom right:** Validation F1 and EM during GRPO training.

Figure 2 compares all four model variants on F1 and EM, with relative improvement annotations.

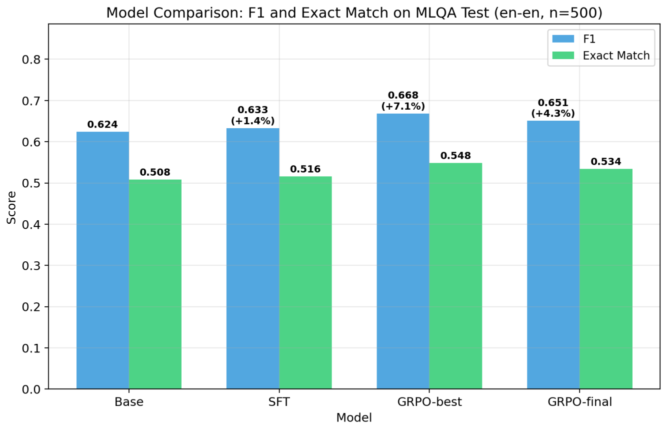

Fig. 2. F1 and Exact Match comparison across model variants. Percentages indicate relative F1 improvement over the base model.

Figure 3 visualizes the cross-lingual transfer gap across languages.

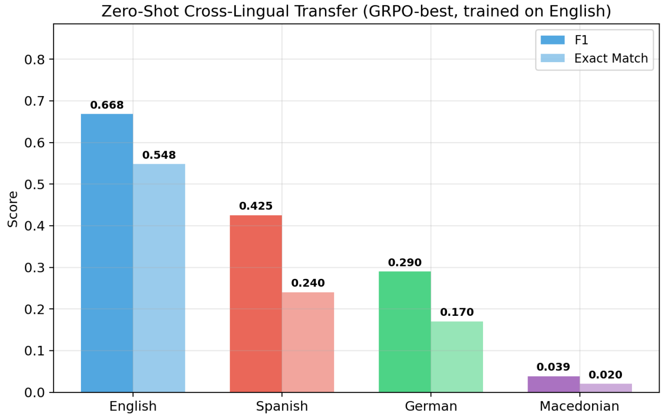

Fig. 3. Zero-shot cross-lingual transfer performance of GRPO-best across four languages.

VI. ANALYSIS

1) GRPO-best vs. GRPO-final.: The validation F1 peaks at step 200 (0.668) and declines afterward, reaching 0.651 at step 774 when early stopping triggers. This suggests that extended GRPO training leads to overfitting to the reward signal. Monitoring the KL divergence (Figure 1, bottom left) reveals a steady increase throughout training, indicating growing policy drift from the reference model despite the KL penalty ($\beta = 0.04$).

2) Clip Fraction.: The GRPO diagnostics (Figure 4) show that clip fractions remain high (0.75+) throughout training. This is likely due to a mismatch between “old” and “new” log-probabilities: both are computed from the same model within a single step (before and after the forward pass), but dropout introduces stochasticity that inflates the ratio variance. While this does not prevent learning, it suggests that the effective clipping is more aggressive than intended.

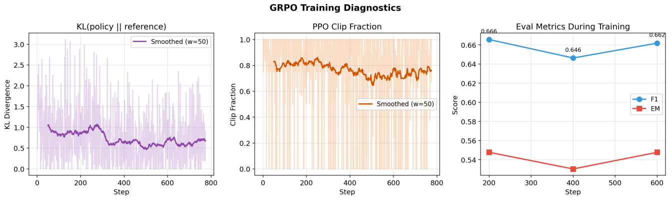

Fig. 4. GRPO training diagnostics. **Left:** KL divergence between policy and reference. **Center:** Clip fraction over training steps. **Right:** Validation F1 and EM at evaluation checkpoints.

3) Cross-Lingual Error Patterns.: The Macedonian results (F1=0.039) reveal a systematic failure mode: the model outputs garbled character sequences (e.g., “ ”) rather than coherent Macedonian text. This indicates that `flan-t5-small`'s tokenizer and embedding space have insufficient coverage of Cyrillic characters, rather than a reasoning failure. By contrast, Spanish and German errors tend to be semantic (wrong answer span) rather than orthographic.

4) Answer Length.: SFT slightly reduces average answer length (from 3.37 to 2.79 tokens), while GRPO partially reverses this trend (3.17 tokens for GRPO-best). The GRPO reward function includes an explicit length penalty ($\lambda = 0.001$ for outputs exceeding 64 tokens), but this rarely activates given the short average lengths.

VII. Hyperparameters

Table III summarizes the key hyperparameters for both training phases.

TABLE III
Hyperparameters for SFT and GRPO.

Parameter	SFT	GRPO
Learning rate	5e−5	1e−5
Batch size	8	4
Epochs	3	3
Max grad norm	1.0	1.0
Warmup ratio	0.1	0.1
Group size (K)	—	4
Clip ϵ	—	0.2
KL coefficient β	—	0.04
Temperature	—	0.8
Top-p	—	0.9
Max input length	512	
Max output length	64	

VIII. Limitations

Our work has several limitations. First, we use `flan-t5-small` (80M parameters), which is substantially smaller than models typically used for QA. While this demonstrates that GRPO can benefit small models, the absolute performance remains well below state-of-the-art. Second, our training set is limited to the MLQA validation split (1,148 examples), as we reserve the larger test split for evaluation. Training on more data would likely yield further improvements. Third, the GRPO implementation processes each candidate sequentially (within a group of $K = 4$), making training computationally intensive relative to the model size. Batched candidate processing would significantly improve throughput. Fourth, the cross-lingual evaluation on Macedonian relies on machine translation of English data, introducing translation artifacts. Finally, the RAG component was evaluated separately and not jointly optimized with the GRPO training.

IX. Conclusion

We presented a pipeline combining SFT and GRPO for extractive question answering on MLQA using `flan-t5-small`. GRPO improves F1 by 7.1% relative to the base model, outperforming SFT alone. Our analysis reveals that periodic evaluation and early stopping are critical in GRPO training, as the best checkpoint occurs well before training completion. Cross-lingual experiments demonstrate partial transfer to related languages but expose fundamental limitations for languages with different scripts and low pretraining coverage. Future work could explore GRPO with larger models, joint optimization of retrieval and generation, and multilingual fine-tuning to improve cross-lingual transfer.

References

[1] P. Rajpurkar, J. Zhang, K. Lonstrup, and P. Liang, “SQuAD: 100,000+ questions for machine comprehension of text,” in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2016, pp. 2383–2392.

[2] P. Lewis, B. Oguz, R. Rinott, S. Riedel, and H. Schwenk, “MLQA: Evaluating cross-lingual extractive question answering,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2020, pp. 7315–7330.

[3] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray *et al.*, “Training language models to follow instructions with human feedback,” *Advances in Neural Information Processing Systems*, vol. 35, pp. 27 730–27 744, 2022.

[4] D. M. Ziegler, N. Stiennon, J. Wu, T. B. Brown, A. Radford, D. Amodei, P. Christiano, and G. Irving, “Fine-tuning language models from human preferences,” *arXiv preprint arXiv:1909.08593*, 2019.

[5] Z. Shao, P. Wang, Q. Zhu, R. Xu, J. Song, M. Zhang, Y. Li, Y. Wu, and D. Guo, “DeepSeekMath: Pushing the limits of mathematical reasoning in open language models,” *arXiv preprint arXiv:2402.03300*, 2024.

[6] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, Y. Li, X. Wang, M. Dehghani, S. Brahma *et al.*, “Scaling instruction-finetuned language models,” *Journal of Machine Learning Research*, vol. 25, no. 70, pp. 1–53, 2024.

[7] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, “Exploring the limits of transfer learning with a unified text-to-text transformer,” *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1–67, 2020.

[8] V. Karpukhin, B. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W.-t. Yih, “Dense passage retrieval for open-domain question answering,” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, 2020, pp. 6769–6781.

[9] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel *et al.*, “Retrieval-augmented generation for knowledge-intensive NLP tasks,” in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 9459–9474.

[10] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms,” *arXiv preprint arXiv:1707.06347*, 2017.

[11] R. Rafailov, A. Sharma, E. Mitchell, S. Ermon, C. D. Manning, and C. Finn, “Direct preference optimization: Your language model is secretly a reward model,” *Advances in Neural Information Processing Systems*, vol. 36, 2023.

[12] N. Reimers and I. Gurevych, “Sentence-BERT: Sentence embeddings using siamese BERT-networks,” *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, 2019.