# When Soft Labels Beat Perfect Rewards: Knowledge Distillation vs. GRPO

Tejas Alisyam, Prateek Sircar (✉), Deepak Gupta

Amazon
{tejasyam, sircarp, dgupt}@amazon.com

**Abstract.** In an e-commerce setup, product inconsistency detection flags listings where the title contradicts the bullets or attributes, for example, a "cotton shirt" title with "polyester" bullets. It matters for catalog quality, but large models and frontier LLM APIs are too expensive to run over a full catalog. While a 1.5B student trained with SFT on ∼101K labeled listings, the student reaches F1=0.833, a 4B model trained the same way reaches (F1=0.885). To close the gap, we curate 10,000 held-out samples the student misclassifies and compare three ways to use them: continue-SFT, targeted GRPO with binary ground-truth rewards, and SFT+KD with the 4B model's soft labels. KD reaches F1=0.864, beating both SFT (0.837) and GRPO (0.833). Notably, GRPO sees perfect ground-truth rewards on the student's exact failures, while the 4B teacher agrees with ground truth on only 73.25% of the same 10K, yet KD still leads by +0.031 F1 (non-overlapping bootstrap 95% CIs). The KD-trained student is now our production classifier, scoring full catalog of an e-commerce company, flagging inconsistent product pages for further review.

**Keywords:** Knowledge distillation · GRPO · Small language models · E-commerce · Model compression

## 1 Introduction

In an e-commerce setup, a *product inconsistency* occurs when a listing or a product's title contradicts other information on its page. For example, a title reads "Vivo V60 5G (Auspicious Gold, **8GB RAM**, 256GB Storage)" while a bullet point states "**12GB RAM** | 512GB Internal Memory." Such listings mislead customers, drive returns, and erode trust. Detecting these is not a string-matching problem: the model must distinguish genuine conflicts from acceptable extra detail (e.g., "Machine Wash" vs. "Hand & Machine Washable"). We treat inconsistency detection as binary classification over the title and bullet points.

**Why a Small Model.** An e-commerce catalog contains tens of millions of listings. Scoring the full catalog across marketplaces is infeasible with a 4B model or a frontier LLM API. A 1.5B classifier makes it tractable. Reaching it without sacrificing F1 is the focus of this paper.

Our student is Qwen2.5-1.5B-Instruct [7]. After SFT on ∼101K labeled listings, it reaches F1=0.833, short of a 4B reference (Qwen3-4B [8]) trained the

**Table 1.** Deployment comparison on a V100 GPU (p3.2xlarge EC2, fp16, 500 eval samples; run-level averages).

| Model | Params | Disk (GB) | VRAM (GB) | Throughput (samples/s) |
|---|---|---|---|---|
| Qwen3-4B reference | 4.02B | 7.6 | 10.5 | 30 |
| Qwen2.5-1.5B student | 1.54B | 2.9 | 6.25 | 75 |
| Ratio (4B / 1.5B) | 2.6× | 2.6× | 1.7× | 2.5× |

same way (F1=0.885). To close the gap, we run the trained student on a held-out labeled pool and keep the 10,000 samples it gets wrong. We compare three interventions on these 10K, all starting from the same SFT checkpoint: (i) **Continue SFT** on the 10K with the correct labels; (ii) **Targeted GRPO** [6,3] with binary ground-truth rewards on the *exact* cases where the student fails, making RL's reward both *perfect* and well-targeted; (iii) **SFT+KD** [1,5] using the 4B model's full output distribution as soft labels alongside the hard labels.

**What We Find.** KD reaches F1=0.864, with non-overlapping bootstrap 95% CIs against SFT (0.837) and GRPO (0.833). No GRPO variant we tried beats the SFT base of 0.833, across reward shapes, KL regularization, and a full-101K training run that rules out curriculum effects. When a larger model is available, distill its soft labels rather than investing in trajectory-level RL on the student.

## 2 Setup

**Task and Data.** We treat inconsistency detection as text classification with two labels, *consistent* and *inconsistent*. All labels are human-annotated and high quality. Evaluation set (13,184 held-out samples) is stratified 90/10 and the training corpus is upsampled to 75/25 for adequate minority-class signal. We report precision, recall, and F1 on the inconsistent class.

**Student and Reference.** The student is Qwen2.5-1.5B-Instruct [7] fine-tuned with LoRA [2] (rank 128, all linear layers, 3 epochs, lr= $2\times10^{-4}$), reaching F1=0.833. The 4B reference is Qwen3-4B-Instruct fine-tuned with SFT on the same corpus (F1=0.885), used as the soft-label source for KD. It is not the deployment target.

**The 10K Hard-Case Set.** After the SFT student converges (three epochs, ep3), we run it on a separate ground-truth-labeled pool from the same distribution as the training corpus and retain the 10,000 samples on which its prediction disagrees with the label.

**Three Interventions on the 10K.** All three start from the SFT ep3 checkpoint.

*SFT on 10K.* Continue fine-tuning with the correct labels, same optimizer and LoRA setup as the base SFT.

*SFT+KD on 10K.* Same setup, plus a KL term $\alpha \cdot D_{\mathrm{KL}}(p_{\mathrm{teacher}} \,\|\, p_{\mathrm{student}})$ with $\alpha$=0.1, using the 4B reference's full output distribution.

*GRPO on 10K* [6]. LoRA rank 16 (higher ranks underperformed in preliminary experiments, with rank-128 GRPO reaching only F1=0.811), $\beta$=0.04,

**Table 2.** Interventions on the 10K hard cases, starting from SFT ep3. Best student F1 in **bold**. 95% bootstrap CIs (1000 resamples) are reported for the three primary rows. Other rows are single-run values. GRPO on the 101K training corpus (also starting from ep3) rules out curriculum effects. Additional GRPO variants we tested (continuous teacher-confidence reward and other KL regularizations) all fell in [0.811, 0.834].

| Method | Prec. | Rec. | F1 (95% CI) |
|---|---|---|---|
| Qwen3-4B (SFT, reference) | 0.914 | 0.858 | 0.885 |
| Student SFT baseline (no 10K used) | 0.822 | 0.843 | 0.833 |
| *Starting from SFT ep3, using the 10K hard cases* | | | |
| + SFT on 10K | 0.832 | 0.842 | 0.837 [.829, .845] |
| + GRPO on 10K (vanilla, binary reward) | 0.830 | 0.837 | 0.833 [.825, .841] |
| + GRPO on 10K (teacher-confidence) | 0.827 | 0.839 | 0.833 |
| + GRPO on 101K (full training corpus) | 0.823 | 0.845 | 0.834 |
| + SFT+KD on 10K ($\alpha$=0.1) | 0.874 | 0.855 | **0.864 [.857, .871]** |

lr= $3\times10^{-5}$, 4 generations per prompt, binary ground-truth reward ($+1/-1$, $-0.5$ for unparseable outputs). We also tested several variants of this basic setup: alternative reward shapes (continuous teacher confidence), additional KL-regularization terms ($\alpha$-KL, KL-Distill+GRPO, teacher-as-reference, mixed-label). To rule out that the hard-case-only curriculum starves GRPO of positive rewards, we additionally ran vanilla GRPO on the full 101K training corpus.

## 3 Results

**Continuing SFT on the 10K gives a small bump.** Continuing SFT from the ep3 checkpoint on the 10K moves F1 from 0.833 to 0.837. The effect is small: cross-entropy on the correct labels is the same objective the student was trained with on the 101K corpus, now applied to 10,000 new hard cases.

**GRPO does not beat the SFT base.** Vanilla GRPO from ep3 reaches F1=0.833, identical to the starting checkpoint. We initially suspected reward sparsity, so we replaced the binary reward with a continuous teacher-confidence score. That variant also stops at 0.833. Nor is the hard-case-only selection to blame: running GRPO on the full 101K training corpus (which mixes easy and hard samples) yields F1=0.834, essentially indistinguishable from the SFT base. Across our sweep and the 101K run, F1 lands between 0.811 and 0.834, and no variant beats the SFT base.

**SFT+KD on the 10K gives the largest gain.** Adding the 4B reference's soft-label distribution to the SFT loss on the same 10K lifts F1 to 0.864, a +0.031 gain over every non-KD method, with non-overlapping bootstrap 95% CIs (KD [.857, .871] vs. SFT [.829, .845] and GRPO [.825, .841]). Since all three interventions use the same 10K samples and start from the same SFT ep3 checkpoint, the difference comes from the training signal itself, not from data or initialization. It is not label accuracy: on these 10K, the 4B reference agrees with ground truth only 73.25% of the time, so roughly a quarter of KD's soft-label targets point away from the correct class, while GRPO's reward is ground truth by construction. What KD provides that GRPO does not is density: a full

teacher distribution at every output token, versus a single scalar per generation. Here, that density dominates both label accuracy and failure-case targeting.

## 4 Discussion

**Why KD wins on the same 10K.** The 10K are cases where plain hard-label SFT has not produced a reliable classifier: even after three epochs on the 101K training corpus, the student still misclassifies these particular samples at inference. Re-running SFT on them (transferring only the teacher's top-1 choice, a scalar) nets +0.004 F1. Adding the teacher's full distribution (KD) nets +0.031. GRPO is an on-policy reward-driven update, but a scalar reward tells the model only whether a generation was correct, not where probability was misplaced or how the larger model would have distributed it. KD's per-token KL exposes that full teacher distribution at every step. A direct ablation we did not run would binarize the teacher's soft labels and retrain. If the KD gain disappears, density is the operative factor.

**Practical recommendation.** When a larger model is available, distilling its soft labels with KD was the most effective of the three approaches we tried. GRPO, despite its conceptual appeal, did not beat the SFT baseline. KD also avoids reward design, on-policy generation, and multi-GPU sampling. The KD student is now our production classifier, running every ten days over newly added listings on a workload the 4B reference would be too expensive to serve.

**Limitations.** Our results are from one binary classification task with one evaluation set. We do not claim they generalize to open-ended generation or to other structured-classification domains. The 10K come from the same distribution as the training corpus, so we cannot speak to distribution shift. Our RL result is specifically about *GRPO with trajectory-level rewards*: continuous teacher-confidence rewards and scaling to the full 101K both failed to help, but we did not evaluate denser reward shaping (e.g., step- or token-level variants) or RL formulations tailored to classification (e.g., DPO [4]), so we cannot rule out that such recipes would close the gap.

**Disclosure of Interests.** The authors have no competing interests.

## References

1. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv:1503.02531 (2015)
2. Hu, E.J., et al.: LoRA: Low-rank adaptation of large language models. In: ICLR (2022)
3. Ouyang, L., et al.: Training language models to follow instructions with human feedback. NeurIPS **35**, 27730–27744 (2022)
4. Rafailov, R., et al.: Direct preference optimization: Your language model is secretly a reward model. In: NeurIPS (2023)
5. Sanh, V., Debut, L., Chaumond, J., Wolf, T.: DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv:1910.01108 (2019)
6. Shao, Z., et al.: DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. arXiv:2402.03300 (2024)
7. Yang, A., et al.: Qwen2.5 technical report. arXiv:2412.15115 (2024)
8. Yang, A., et al.: Qwen3 technical report. arXiv:2505.09388 (2025)