RETAINING BY DOING: THE ROLE OF ON-POLICY DATA IN MITIGATING FORGETTING

Howard Chen Noam Razin
Karthik Narasimhan Danqi Chen

Princeton Language and Intelligence
Princeton University

November 24, 2025

Outline

- Introduction & Motivation
- SFT vs. RL Experiment
- Theoretical Expectation & Empirical Reality
- The Role of On-Policy Data
- Practical Alternative: Approximately On-Policy
- Conclusion

Introduction & Motivation

- Post-training is crucial for adapting LMs to new tasks.
- This adaptation risks "catastrophic forgetting"—degrading previously acquired capabilities
- Such forgetting has been reported to occur when training LMs to follow instructions via supervised fine-tuning (SFT) or aligning them with human preferences via reinforcement learning (RL)

Goal: Systematically compare the forgetting patterns of SFT and RL in order to identify principled guidelines for mitigating forgetting in LM post-training.

The Methods: SFT vs. RL

1. Supervised Fine-Tuning (SFT):

- How: Minimizes Cross-Entropy Loss on a fixed dataset of (prompt, response) pairs.
- Data Type: Off-Policy. The data is static and not from the model being trained.

2. Reinforcement Learning (RL):

- How: Maximizes a reward signal. The model generates its own response and gets a score.
- Data Type: On-Policy. The model learns directly from its own experience

Experimental Setup

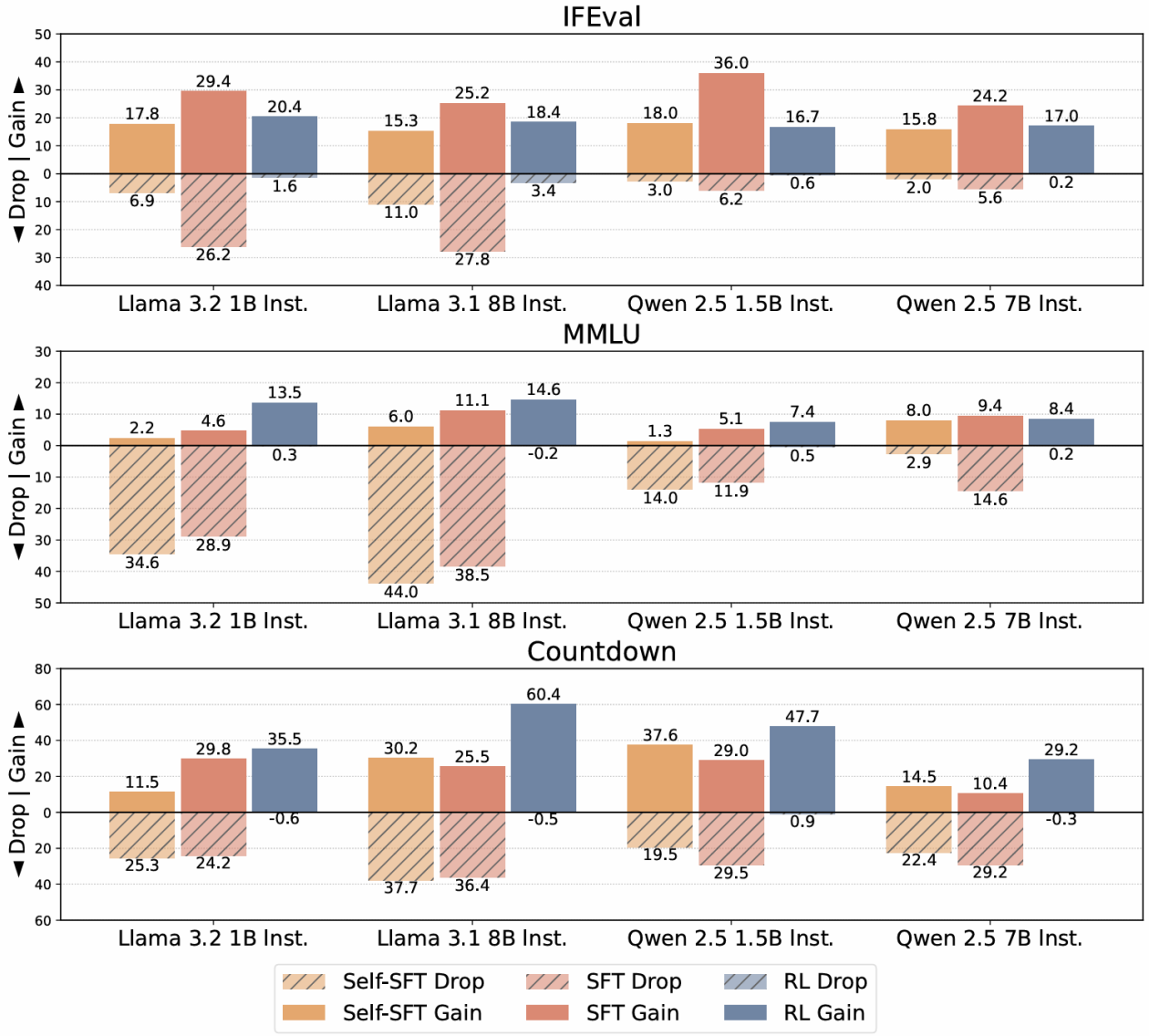
- Models: Llama 3 (1B, 8B) & Qwen 2.5 (1.5B, 7B)
- Target Tasks (Training): IFEval (Instructions), MMLU (Knowledge), Countdown (Arithmetic).
- Non-Target Tasks (Forgetting): All other tasks, plus MATH and Safety benchmarks
- Key Metrics:

  1.Gain (Δ_g):Accuracy increase on the target task.

  2.Drop (Δ_d): Average accuracy decrease on non-target tasks.

Result

RL Forgets Substantially Less than SFT

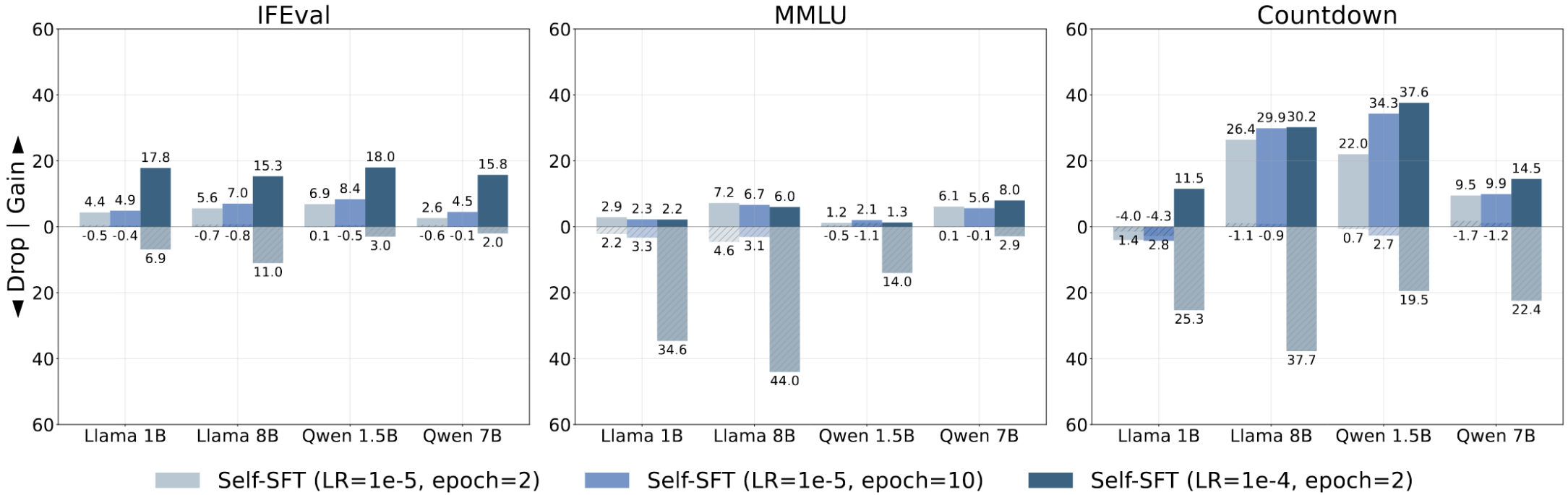


- For Self-SFT, achieving a similar target accuracy gain to RL induces a significantly larger drop on non-target tasks.
- While SFT can achieve a higher performance gain than RL on the instruction following task, it induces an even larger drop on non-target tasks relative to Self-SFT.

Result

SFT has a Performance-Forgetting Trade-off

- The paper investigated if this is just a learning rate problem for SFT.
- **High LR**: SFT learns the new task (High Gain) but suffers from severe forgetting (High Drop).
- **Low LR** : SFT reduces forgetting (Low Drop) but then fails to learn the new task (Low Gain).



The Theoretical Investigation

SFT and RL can be viewed as minimizing different directions of the KL divergence with respect to the optimal policy.

SFT as forward KL minimization (mode-covering).

- It is widely known that SFT is equivalent to minimizing the forward KL between the optimal and training policies

- $\mathcal{L}_{SFT}(\theta;\, x) = \sum_{\mathrm{y}} -\pi^*(y \mid x) \log \pi_\theta(y \mid x) = \sum_{\mathrm{y}} \pi^*(y \mid x) \log \frac{\pi^*(y \mid x)}{\pi_\theta(y \mid x)} +$

  $\sum_{\mathrm{y}} -\pi^*(y \mid x) \log \pi^*(y \mid x) = KL\left[\pi^*(\cdot \mid x) || \pi_\theta(\cdot \mid x)\right] + \mathcal{H}(\pi^*(\cdot \mid x))$

- where $\mathcal{H}(\pi^*(\cdot \mid x))$ is the entropy of $\pi^*(\cdot \mid x)$, which does not depend on π_θ .

Intuition (mode-covering): If the target π^* contains both old and new knowledge, π_θ must "cover" both. Therefore, SFT should be robust to forgetting.

The Theoretical Investigation

SFT and RL can be viewed as minimizing different directions of the KL divergence with respect to the optimal policy.

RL as reverse KL minimization (mode-seeking).

- $J_{RL}(\theta;\, x) = \mathrm{E}_{y\sim\pi_\theta(\cdot\,|\,x)}[r(x,y)] - \beta \cdot KL[\pi_\theta(\cdot\,|\,x)\,||\pi_{\theta_0}(\cdot\,|\,x)]$

  $= -\beta \cdot KL[\pi_\theta(\cdot\,|\,x)\,||\,\pi^*(\cdot\,|\,x)] + \beta \cdot log\, Z(x)\,,$

- $\mathrm{Z(x)} = \sum_{\mathrm{y}} \pi_{\theta_0}(y\,|\,x)\exp(r(x,y)/\beta)$

- The optimal policy for the KL-regularized RL objective is given by

$$\pi^*(y\,|\,x) = \frac{1}{Z(x)}\pi_{\theta_0}(y\,|\,x)\exp(r(x,y)/\beta)$$

Intuition (mode-seeking): The model will abandon the old modes (old knowledge) to focus 100% on the new, high-reward mode (new task). Therefore, RL should be prone to forgetting.

The Contradiction

Theory Predicts:

- SFT (Mode-Covering) →Low Forgetting.
- RL (Mode-Seeking) → High Forgetting.

Experiment Shows:

- SFT → High Forgetting.
- RL → Low Forgetting.

- Intuitively, a mode-seeking objective such as reverse KL should be more susceptible to forgetting: it moves probability mass quickly from one mode to another, whereas mode-covering forward KL should better maintain probability mass on all modes. This intuition is invalidated in light of the evidence presented before.
- Address this discrepancy through an empirical analysis of a simplified setting with univariate Gaussian distributions.

The Uni-modal Setting

The Optimal Policy:

- The Target is modeled as a multi-modal distribution (a mixture of two Gaussian peaks), representing two distinct skills.
- $\pi^*(\mathrm{y}) = \alpha^* \cdot \mathrm{p}_{old}(\mathrm{y}) + (1 - \alpha^*) \cdot \mathrm{p}_{new}(\mathrm{y}),\ \theta = (\mu, \sigma)$
- Where p_{old} is the "Old Knowledge" mode and p_{new} is the "New Task" mode.

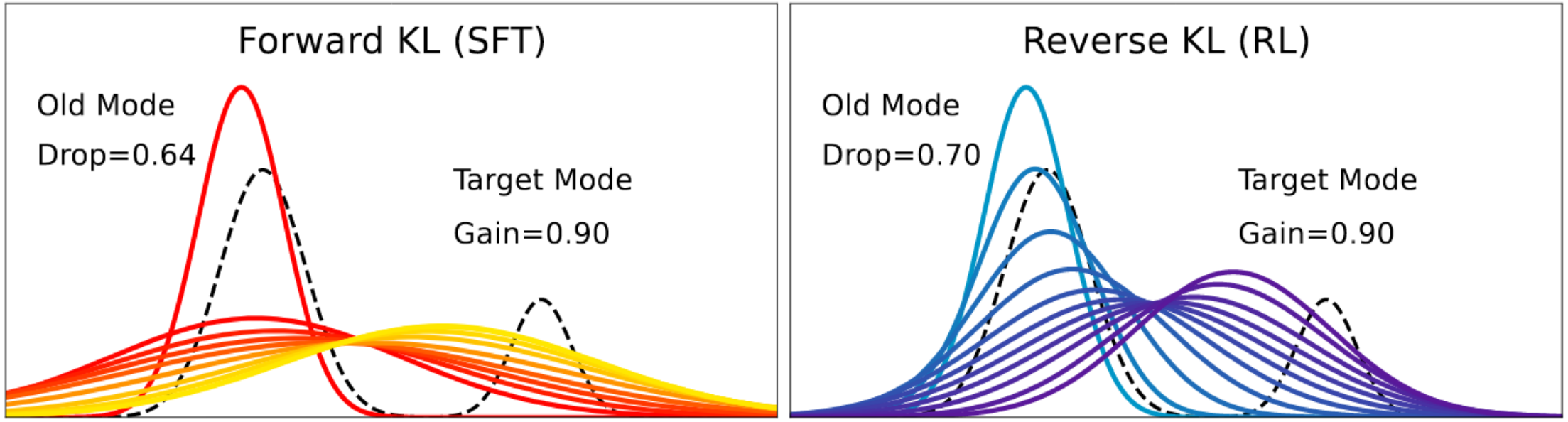
Model:

- Let the model (π_θ) be a uni-modal distribution (a single Gaussian peak)
- It is initialized to perfectly cover the "Old Knowledge" mode, p_{old}.

The Uni-modal Setting

Theory Predicts:

- Create a simulation where the model (π_θ) is forced to be a single Gaussian peak (a uni-modal policy) .
- The task is to learn a target that has two peaks (old and new).



Result: In this uni-modal setting, SFT forgets less than RL.

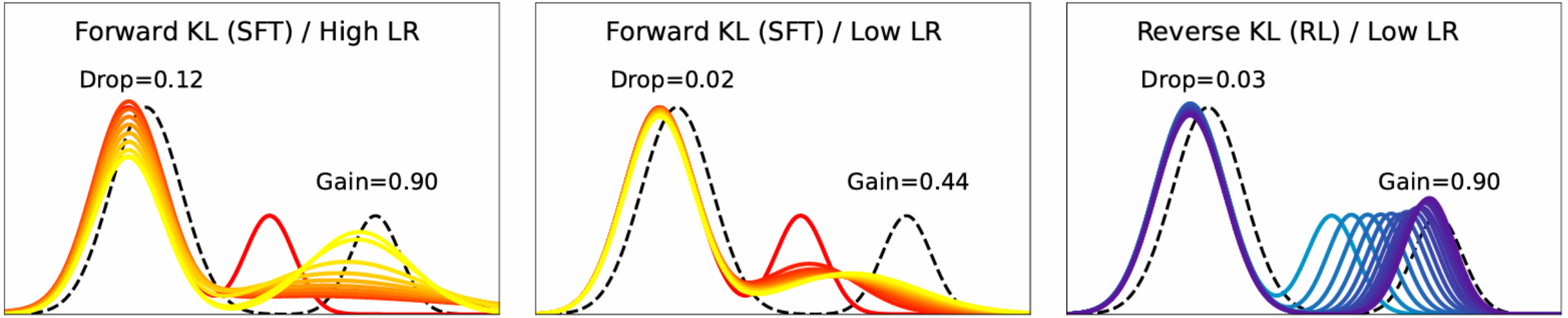
- RL forgets more (Drop=0.70) because its single peak moves entirely to the new mode.
- SFT forgets less (Drop=0.64) because it “stretches” its single peak to try and cover both.

The Multi-modal Hypothesis

New Simulation:

- Let the model (π_θ) also be a mixture of two peaks:
- $\pi_\theta = \alpha \cdot q_{old} + (1 - \alpha) \cdot q_{new}$
- This is a much better analogy for an LM that "knows" old things (q_{old}) and is learning new things (q_{new}).

The Multi-modal Hypothesis



Left/Middle (SFT):

- **High LR**: Achieve a target task gain of 0.9 with forward KL causes severe forgetting—the area overlap with p_{old} drops by 0.12.
- **Low LR**: Mitigate forgetting of the old mode, but leads to failure in learning the target p_{new}.

Right (RL):

- RL shifts q_{new} toward p_{new} while largely keeping the old mode intact. That is it can match a new target mode without redistributing probability mass from a mode that represents prior knowledge.

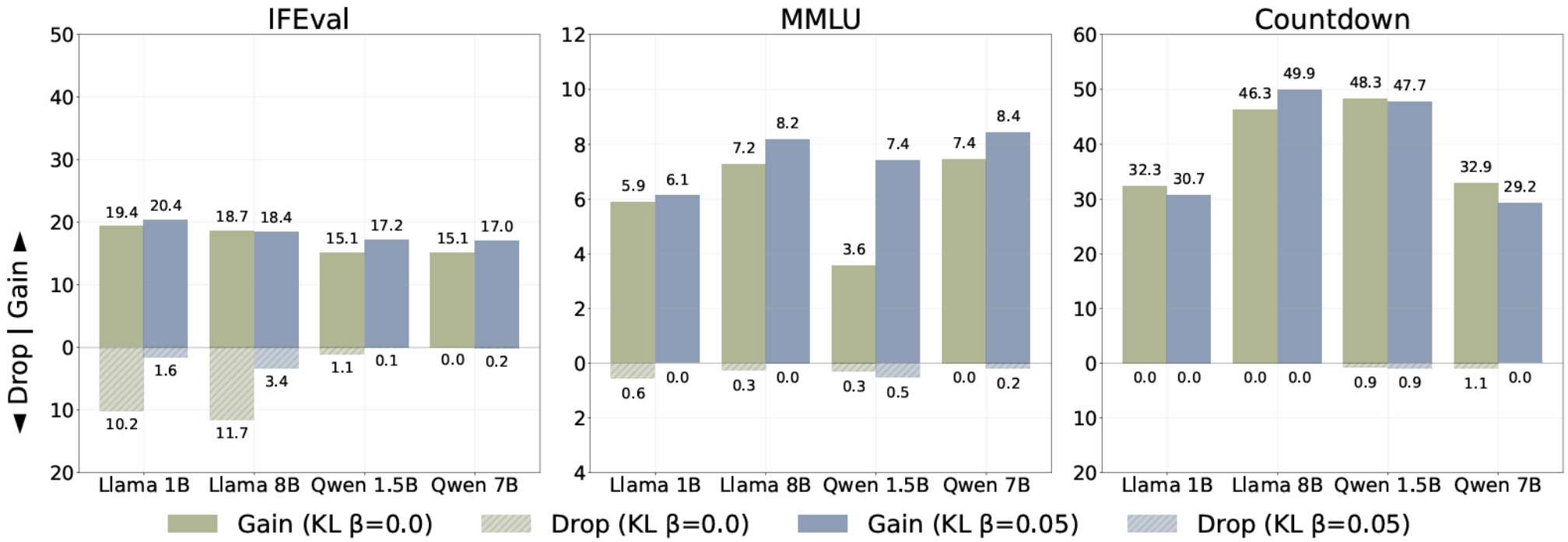
Validation: Ablation studies

The theory claims On-Policy Data is the key.

But the RL algorithm (GRPO) has 3 differences from SFT:

- KL Regularization
- Advantage Estimation
- On-Policy Data

Ablation 1: KL Regularization



- **Experiment**: Run RL with the normal KL term ($\beta = 0.05$) vs. RL with no KL term ($\beta = 0.0$).
- **Result** : The Drop (shaded bars) remains near-zero even with $\beta = 0.0$.
- **Conclusion**: KL regularization does not explain robustness to forgetting.

Ablation 2: Advantage Estimator

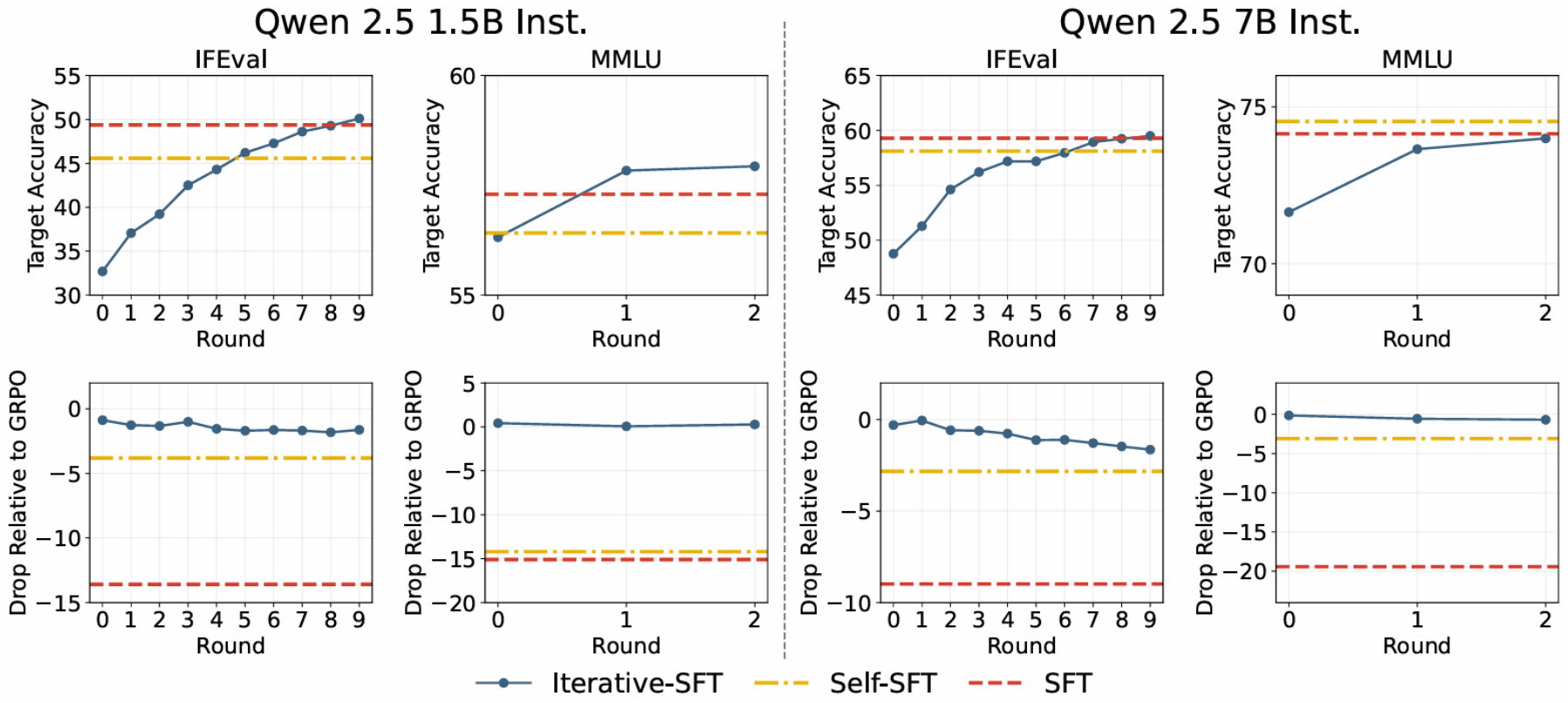
		IFEval		MMLU		Countdown	
		Gain (%) ↑	Drop ↓ (%)	Gain (%) ↑	Drop (%) ↓	Gain (%) ↑	Drop (%) ↓
Llama 3.1 8B Inst.	SFT	25.2	27.8	11.1	38.5	25.5	36.4
	REINFORCE	17.8	7.7	8.6	-0.1	7.5	-0.8
	GRPO	18.4	3.4	14.6	-0.2	60.4	-0.5
Qwen 2.5 7B Inst.	SFT	24.2	5.6	9.4	14.6	10.4	29.2
	REINFORCE	5.7	2.9	6.4	-0.6	11.9	-0.1
	GRPO	17.0	0.2	8.4	0.2	29.2	-0.3

- **REINFORCE** (Williams, 1992), a classical policy gradient RL algorithm that does not employ an advantage estimator.
- REINFORCE lags behind GRPO in optimizing the target task accuracy, yet maintains a similar low level of forgetting.
- This suggests that algorithmic differences, such as the advantage estimator used in RL, primarily affect the magnitude of performance gains, whereas the mitigation of forgetting can be primarily attributed to **the use of on-policy data**.

Approximately On-Policy

- **Problem**: Full On-Policy (RL) is computationally expensive. It requires generating new data at every single optimization step.
- **Question**: Can we get the same benefit with less on-policy data? What about "approximately on-policy"?
- **Solution**: Test whether “Iterative-SFT” , an approximately on-policy approach that iteratively trains on data generated at the start of each epoch, can suffice for mitigating forgetting.

Iterative-SFT Results



- Standard SFT/Self-SFT have a **high relative drop**.
- The Iterative-SFT line **stays** almost perfectly at the 0-line.
- Iterative-SFT is **almost as effective as** full RL at mitigating forgetting.

Conclusion

- RL consistently achieves strong target performance with substantially less forgetting than SFT.
- The robustness of RL to forgetting primarily stems from its use of on-policy data, rather than other algorithmic choices such as the advantage estimate or KL regularization.
- "Approximately on-policy" methods (like Iterative-SFT) provide a practical and efficient solution to reduce forgetting.