Robust SFT

Robust SFT

Fengnan Deng

Department of Statistics, George Mason University

Table of Contents

Current Section

LLM Post-Training: Categories

1. **Supervised Fine-Tuning (SFT):**
   Train on instruction-response pairs to enhance the model's ability in specific domains.
2. **Preference-Based Alignment:**
   - **Reinforcement Learning from Human Feedback (RLHF):** It learns a reward model on pairwise preference data, and optimizes LLMs using the Proximal Policy Optimization (PPO).
   - **Direct Preference Optimization (DPO):** It trains the model directly on (chosen, rejected) pairs without training the reward model.
   - **Group Distributional Preference Optimization (GDPO):** It aligns the model to the distribution of preferences within a group.

Epistemic uncertainty

- Uncertainty due to limited, biased, or unrepresentative training data.
- In LLMs: sensitivity to rare prompts or variations in how intent is expressed.
- This uncertainty can be reduced with more or better data, or with additional constraints.
- Example: A healthcare LLM trained mostly on common symptoms may give unreliable predictions for patients with rare or atypical symptom combinations.

Robust SFT

Accountant Setup

SFT Settings

1. **Training dataset:** $\mathcal{D} = \{\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}\}_{i=1}^{n}$, where $\boldsymbol{X}$ is the prompt sequence and $\boldsymbol{Y} = [Y_1, \cdots, Y_{T_{\boldsymbol{Y}}}]$ is the golden answer of length $T_{\boldsymbol{Y}}$.
2. **LLM model (token level):** Next token distribution is $K_\theta(y_j|\boldsymbol{X}, \boldsymbol{Y}_{1:(j-1)})$.
3. **LLM model (sequence level):** Output sequence distribution

$$K_\theta(\boldsymbol{Y}|\boldsymbol{X}) = \prod_{i=1}^{T_{\boldsymbol{Y}}} K_\theta(y_i|\boldsymbol{X}, \boldsymbol{Y}_{1:(i-1)}).$$

4. **Average log-likelihood:**

$$g(\boldsymbol{Y}; \boldsymbol{X}, \theta) = \frac{1}{T_{\boldsymbol{Y}}} \log K_\theta(\boldsymbol{Y}|\boldsymbol{X}) = \frac{1}{T_{\boldsymbol{Y}}} \sum_{j=1}^{T_{\boldsymbol{Y}}} \log K_\theta(Y_j|\boldsymbol{X}, \boldsymbol{Y}_{1:(j-1)}).$$

5. **SFT loss:** $L(\theta) = -\mathbf{E}_{(\boldsymbol{X},\boldsymbol{Y})\sim f_n}[g(\boldsymbol{Y}; \boldsymbol{X}, \theta)]$, where f_n is the empirical distribution.

Robust SFT

- We want the model to perform well even under the *worst-case* distribution Q close to F_n.
- We defined "closeness" using Kullback-Leibler (KL) divergence.
- We define the robust STF loss as

$$L_\epsilon(\theta) = - \inf_{Q: KL(Q||F_n) \leq \epsilon} \mathbf{E}_{(\boldsymbol{X}, \boldsymbol{Y}) \sim Q}[g(\boldsymbol{Y}; \boldsymbol{X}, \theta)]$$

  - F_n: empirical distribution.
  - Q: distribution within KL-divergence ball around F_n.
  - ϵ: uncertainty radius explicitly controlling robustness level.

Robust SFT

Optimization formulation:

$$\mathbf{E}_{(\boldsymbol{X},\boldsymbol{Y})\sim Q}[g(\boldsymbol{Y};\boldsymbol{X},\theta)] = \sum_{i=1}^{n} Q(\boldsymbol{x}^{(i)},\boldsymbol{y}^{(i)}) \cdot g(\boldsymbol{y}^{(i)};\boldsymbol{x}^{(i)},\theta)$$

Primal problem:

$$\begin{aligned} \text{Minimize over } Q: \quad & \textstyle\sum_{i=1}^{n} Q(\boldsymbol{x}^{(i)},\boldsymbol{y}^{(i)}) \cdot g(\boldsymbol{y}^{(i)};\boldsymbol{x}^{(i)},\theta) \\ \text{Subject to:} \quad & KL(Q||F_n) \leq \epsilon \\ & \textstyle\sum_{i=1}^{n} Q(\boldsymbol{x}^{(i)},\boldsymbol{y}^{(i)}) = 1. \end{aligned}$$

Robust SFT

Lagrangian:

$$\mathcal{L}(Q,\lambda,\nu) = \sum_{i=1}^{n} Q(\boldsymbol{x}^{(i)},\boldsymbol{y}^{(i)}) \cdot g(\boldsymbol{y}^{(i)};\boldsymbol{x}^{(i)},\theta) + \lambda\left[\sum_{i=1}^{n} Q(\boldsymbol{x}^{(i)},\boldsymbol{y}^{(i)}) \cdot \log\frac{Q(\boldsymbol{x}^{(i)},\boldsymbol{y}^{(i)})}{f_n(\boldsymbol{x}^{(i)},\boldsymbol{y}^{(i)})} - \epsilon\right] + \nu\left[\sum_{i=1}^{n} Q(\boldsymbol{x}^{(i)},\boldsymbol{y}^{(i)}) - 1\right]$$

Inner problem: Find $h(\lambda) = \inf_Q \mathcal{L}(Q,\lambda,\nu)$.

- Setting $\frac{\partial}{\partial Q(\boldsymbol{x}^{(i)},\boldsymbol{y}^{(i)})}\mathcal{L}(Q,\lambda,\nu) = 0$ gives

$$Q^*_\lambda(\boldsymbol{x}^{(i)},\boldsymbol{y}^{(i)}) \propto f_n(\boldsymbol{x}^{(i)},\boldsymbol{y}^{(i)}) \cdot e^{-\frac{g(\boldsymbol{y}^{(i)};\boldsymbol{x}^{(i)},\theta)}{\lambda}}.$$

- Normalizer: $Z(\lambda) = \sum_{i=1}^{n} f_n(\boldsymbol{x}^{(i)},\boldsymbol{y}^{(i)}) \cdot e^{-\frac{g(\boldsymbol{y}^{(i)};\boldsymbol{x}^{(i)},\theta)}{\lambda}}$.
- Hence: $Q^*_\lambda(\boldsymbol{x}^{(i)},\boldsymbol{y}^{(i)}) = \frac{1}{Z(\lambda)} f_n(\boldsymbol{x}^{(i)},\boldsymbol{y}^{(i)}) \cdot e^{-\frac{g(\boldsymbol{y}^{(i)};\boldsymbol{x}^{(i)},\theta)}{\lambda}}$.

Robust SFT

Dual problem:

$$\begin{aligned} \text{Maximize w.r.t. } \lambda: \quad & h(\lambda) = -\lambda\epsilon - \lambda \log Z(\lambda) \\ \text{Subject to:} \quad & \lambda > 0. \end{aligned}$$

- $h(\lambda)$ is concave in λ.
- We numerically solve the λ_ϵ that maximizes $h(\lambda)$.
- $Q^*_{\lambda_\epsilon}(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}) = \frac{1}{Z(\lambda_\epsilon)} f_n(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}) \cdot e^{-\frac{g(\boldsymbol{y}^{(i)}, \boldsymbol{x}^{(i)}, \theta)}{\lambda_\epsilon}}$ is the minimizer of the primal problem.

Robust SFT

For i.i.d. training samples, $f_n(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}) = \frac{1}{n}$.

- Using $Q^*_{\lambda_\epsilon}$, we obtain the robust loss function

$$L_\epsilon(\theta) = -\frac{1}{n \cdot Z(\lambda_\epsilon)} \sum_{i=1}^{n} \exp\{g(\boldsymbol{y}^{(i)}; \boldsymbol{x}^{(i)}, \theta)\}^{-\frac{1}{\lambda_\epsilon}} \cdot g(\boldsymbol{y}^{(i)}; \boldsymbol{x}^{(i)}, \theta)$$

- We prove

$$\nabla_\theta L_\epsilon(\theta) = -\frac{1}{n \cdot Z(\lambda_\epsilon)} \sum_{i=1}^{n} \exp\{g(\boldsymbol{y}^{(i)}; \boldsymbol{x}^{(i)}, \theta)\}^{-\frac{1}{\lambda_\epsilon}} \cdot \nabla_\theta g(\boldsymbol{y}^{(i)}; \boldsymbol{x}^{(i)}, \theta).$$

- Interpretation: Samples with low log-likelihood (hard samples) get higher weight.

Certainty Constraints

Self-certainty for y_j:

$$C(\theta; \boldsymbol{x}, \boldsymbol{y}_{1:(j-1)}) = KL(U||K_\theta(\cdot|\boldsymbol{x}, \boldsymbol{y}_{1:(j-1)})) = -\frac{1}{|\mathcal{V}|}\sum_{i=1}^{|\mathcal{V}|}\log(|\mathcal{V}|K_\theta(V_i|\boldsymbol{x}, \boldsymbol{y}_{1:(j-1)})),$$

Certainty constraint: $\mathbf{E}_{(\boldsymbol{X},\boldsymbol{Y})\sim Q}[C(\theta; \boldsymbol{X}, \boldsymbol{Y})] \leq \gamma.$

New primal problem:

$$\begin{aligned} \text{Minimize over } Q: \quad & \mathbf{E}_{(\boldsymbol{X},\boldsymbol{Y})\sim Q}[g(\boldsymbol{Y}; \boldsymbol{X}, \theta)] \\ \text{Subject to:} \quad & KL(Q||F_n) \leq \epsilon \\ & \mathbf{E}_{(\boldsymbol{X},\boldsymbol{Y})\sim Q}[C(\theta; \boldsymbol{X}, \boldsymbol{Y})] \leq \gamma \\ & \textstyle\sum_{i=1}^{n} Q(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}) = 1. \end{aligned}$$

Robust loss function: $L_{\epsilon,\gamma}(\theta) =$

$$-\frac{1}{n \cdot Z(\lambda_{\epsilon,\gamma})}\sum_{i=1}^{n}\exp\{-\frac{1}{\lambda_{\epsilon,\gamma}}[g(\boldsymbol{y}^{(i)}; \boldsymbol{x}^{(i)}, \theta) + \beta_{\epsilon,\gamma}C(\theta; \boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)})\} \cdot g(\boldsymbol{y}^{(i)}; \boldsymbol{x}^{(i)}, \theta)$$

LLM Evaluation

On the domain of interest:

- **Large** $\lambda_{\epsilon,\gamma}$**:** The model behaves more reliably on rare or out-of-distribution prompts.
- **Small** $\beta_{\epsilon,\gamma}$**:** The model generates consistent output (high self-certainty).

Robust SFT

Robustness Evaluation

Implementation overview

- **Model:** Qwen3-0.6B.
- **Adapter:** LoRA with rank 16.
- **Training dataset:** 30k samples from HuggingFaceH4/ultrachat_200k train split.
- **Testing dataset:** 7k samples from HuggingFaceH4/ultrachat_200k test split.
- **Radius choice:** $\epsilon \in [0, 0.03, 0.05]$.

Pipeline

- For each ϵ, compute λ_ϵ from training dataset.
- Compute the weight $w_i = \exp\{g(\boldsymbol{y}^{(i)}; \boldsymbol{x}^{(i)}, \theta)\}^{-\frac{1}{\lambda_\epsilon}}$ for each sample in the training dataset.
- Define the robust loss function

$$L_\epsilon(\theta) = -\frac{1}{n \cdot Z(\lambda_\epsilon)} \sum_{i=1}^{n} w_i \cdot g(\boldsymbol{y}^{(i)}; \boldsymbol{x}^{(i)}, \theta).$$

- Use optimizer 'AdamW' with 0.03% of training samples as warm up.
- Evaluate perplexities of trained LLMs on whole testing dataset, on top 1% hardest samples from testing dataset, and on gsm8k testing dataset (unrelated to the training dataset).

Evaluation

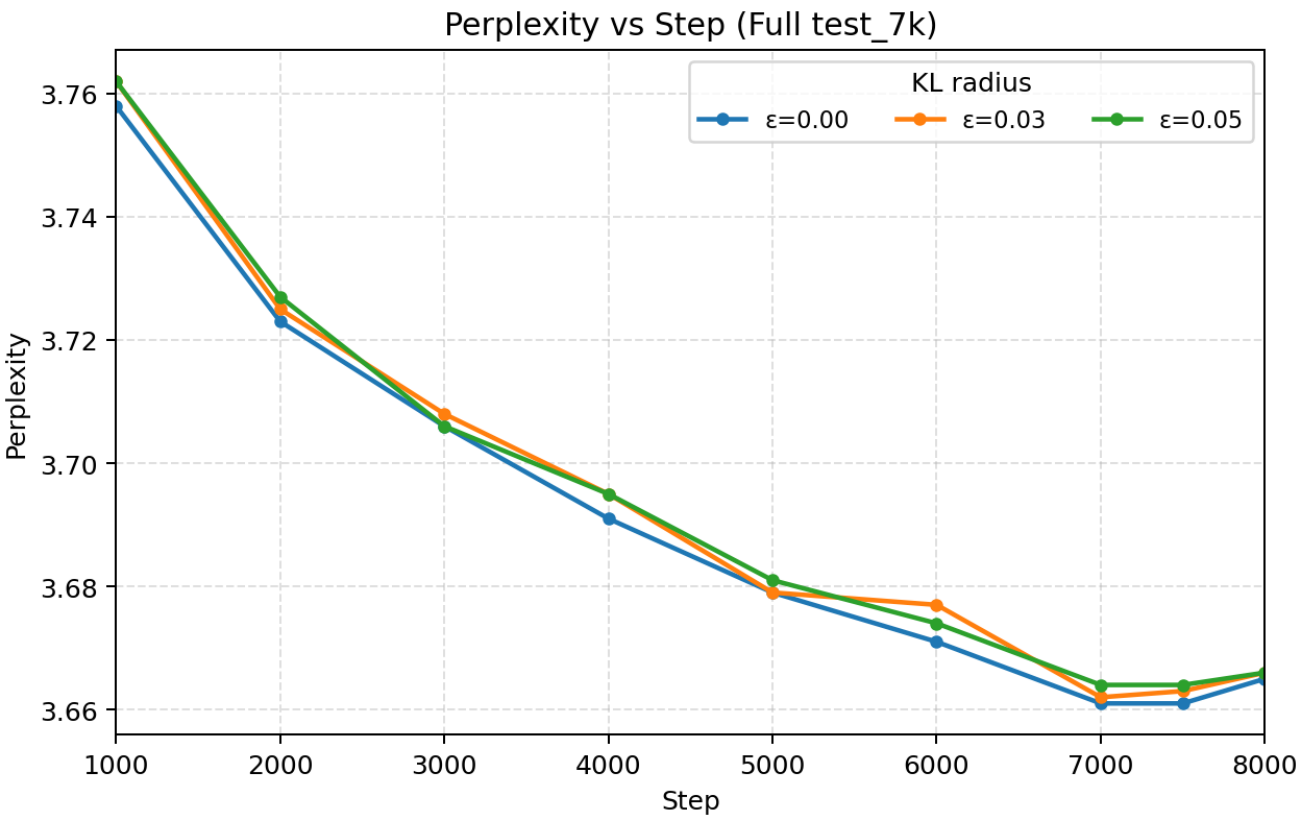

Figure 1: Testing perplexity on whole testing dataset

Evaluation

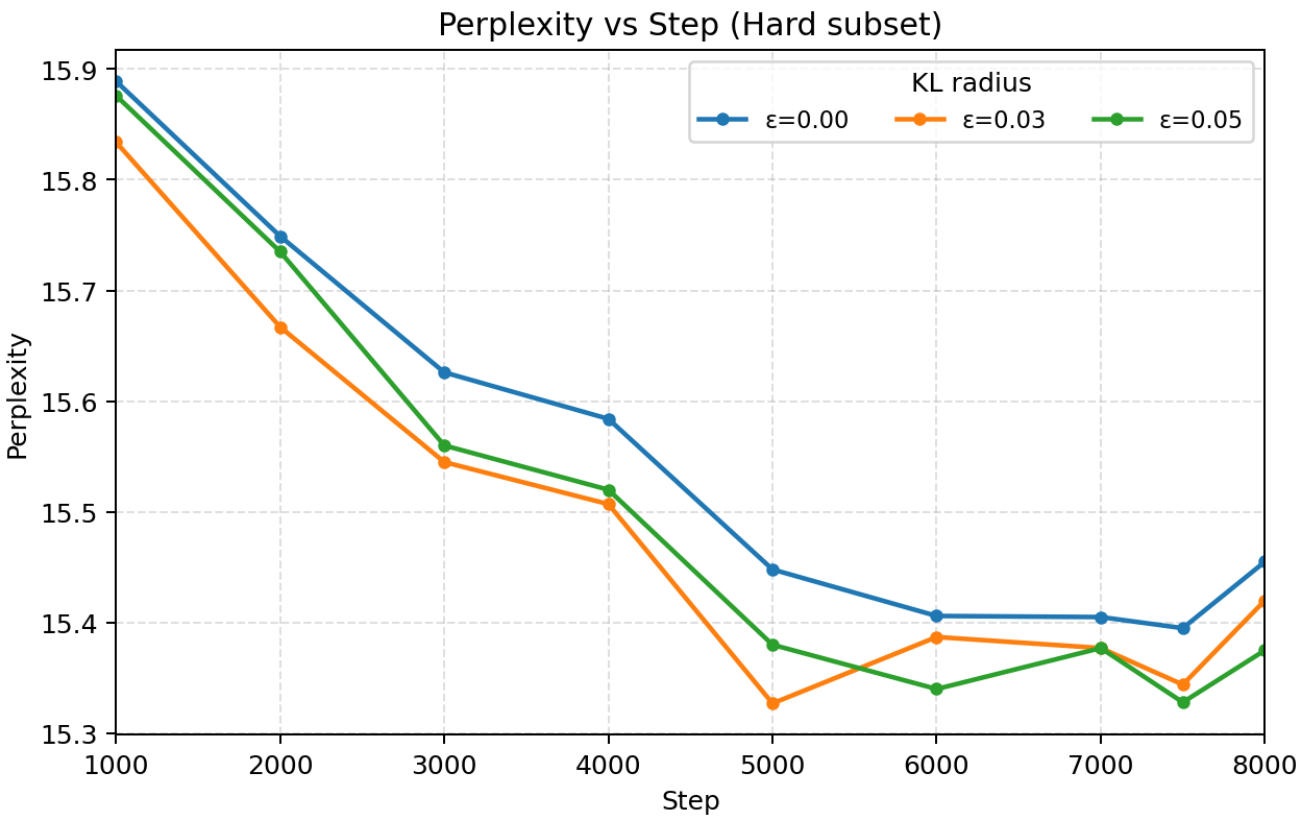

Figure 2: Testing perplexity on top 1% hardest samples from testing dataset

Evaluation

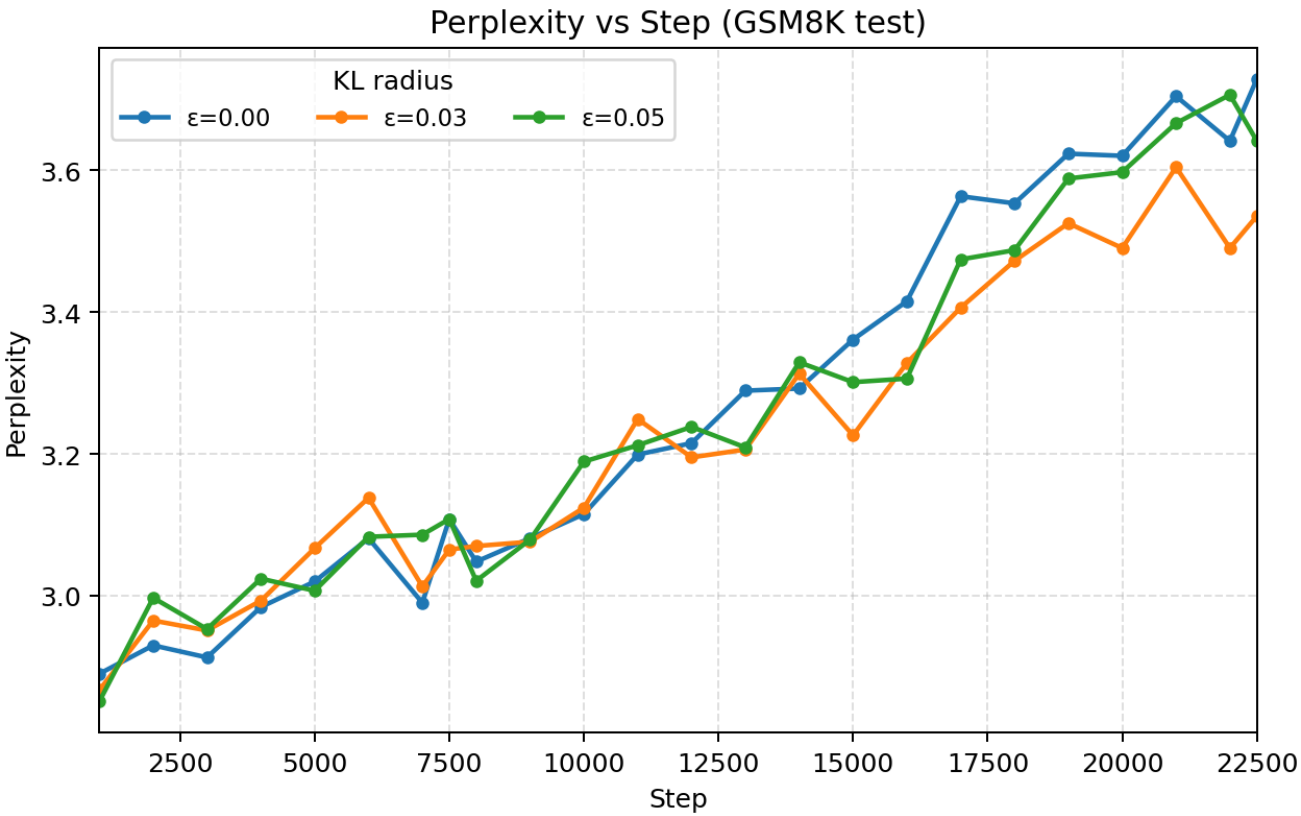

Figure 3: Testing perplexity on gsm8k testing dataset

Robust SFT

Model Comparison

Pipeline (Model Comparison Tests)

- Generated 99 semantically equivalent paraphrase rounds (dataset variants) via OpenAI Batch (GPT-5.2-nano).
- Fix $\epsilon = 3$ and $\gamma = 14.5$. For each round and each model (Qwen3-0.6B, Qwen3-8B, Llama3.2-1B, RNJ1-8B, Croissant-1B), compute log-likelihood and self-certainty and solve pair $(\lambda_{\epsilon,\gamma}, \beta_{\epsilon,\gamma})$.
- For each model pair, run two-sample t-tests on $\lambda_{\epsilon,\gamma}$ and $\beta_{\epsilon,\gamma}$ across rounds, and a Hotelling T^2 test on $(\lambda_{\epsilon,\gamma}, \beta_{\epsilon,\gamma})$.

Model Summary Across Paraphrase Rounds

Averages over 99 paraphrase rounds for sequence log-likelihood (higher is better), self-certainty (higher is better), λ^* (higher is better), and β^* (lower is better).

Model	**Avg log-likelihood**	**Avg self-certainty**	**Avg** (λ^*, β^*)
Croissant-1B	-2.103140	16.567155	(0.415640, 0.500944)
Llama3.2-1B	-2.135211	11.427630	(0.216279, 0.002997)
Llama3.1-8B	-1.920471	13.191394	(0.241138, 0.024425)
Qwen3-0.6B	-2.311708	14.753453	(0.231023, 0.142725)
Qwen3-8B	-2.478333	19.811302	(0.375483, 0.501449)
RNJ1-8B	-1.784117	13.407911	(0.764820, 0.002201)

Pairwise p-values: t-test on λ^*

Compare $\lambda^*_{\epsilon,\gamma}$ across models using t-tests over 99 rounds.

	Croissant-1B	Llama3.2-1B	Llama3.1-8B	Qwen3-0.6B	Qwen3-8B	RNJ1-8B
Croissant-1B	1.000	0	0.001	0.000	0.413	0.319
Llama3.2-1B	0	1.000	0.018	0.094	0	0.114
Llama3.1-8B	0.001	0.018	1.000	0.253	0	0.131
Qwen3-0.6B	0.000	0.094	0.253	1.000	0	0.124
Qwen3-8B	0.413	0	0	0	1.000	0.261
RNJ1-8B	0.319	0.114	0.131	0.124	0.261	1.000

Note: entries shown as 0 are below display precision (very small *p*-values).

Pairwise p-values: t-test on β^*

Compare $\beta^*_{\epsilon,\gamma}$ across models using t-tests over 99 rounds.

	Croissant-1B	Llama3.2-1B	Llama3.1-8B	Qwen3-0.6B	Qwen3-8B	RNJ1-8B
Croissant-1B	1.000	0	0	0	0.982	0
Llama3.2-1B	0	1.000	0.025	0	0	0.832
Llama3.1-8B	0	0.025	1.000	0	0	0.008
Qwen3-0.6B	0	0	0	1.000	0	0
Qwen3-8B	0.982	0	0	0	1.000	0
RNJ1-8B	0	0.832	0.008	0	0	1.000

Note: entries shown as 0 are below display precision (very small p-values).

Pairwise p-values: Hotelling T^2 on (λ^*, β^*)

Compare the pair $(\lambda^*_{\epsilon,\gamma}, \beta^*_{\epsilon,\gamma})$ across models over 99 rounds.

	Croissant-1B	Llama3.2-1B	Llama3.1-8B	Qwen3-0.6B	Qwen3-8B	RNJ1-8B
Croissant-1B	1.000	0	0	0	0.619	0
Llama3.2-1B	0	1.000	0	0	0	0.284
Llama3.1-8B	0	0	1.000	0	0	0.009
Qwen3-0.6B	0	0	0	1.000	0	0
Qwen3-8B	0.619	0	0	0	1.000	0
RNJ1-8B	0	0.284	0.009	0	0	1.000

Note: entries shown as 0 are below display precision (very small p-values).