DPO, ORPO, GRPO, and PPO - Comparison (Part-2)

Part-2 covers comparison of techniques. Refer part-1 for theoretical background and part-3 for the real world scenario.

5 Training Lineage

Standard Path for Each Method

PPO (Traditional RLHF):

1. Base Model → 2. SFT → 3. Reward Model Training → 4. PPO Optimization

DPO:

1. Base Model → 2. SFT → 3. DPO

ORPO:

1. Base Model → 2. ORPO (single stage)

GRPO:

1. Base Model → 2. SFT → 3. GRPO

Recommended Sequences

- **DPO**: Base → SFT → DPO
- **ORPO**: Base → ORPO (single stage)

- **GRPO**: Base → SFT → GRPO
- **PPO**: Base → SFT → RM → PPO

6 Trade-offs

6.1 Advantages & Disadvantages Matrix

Method	Key Advantages	Key Disadvantages	Best For
PPO	- Best complex task performance - Fine-grained control - Multiple reward signals - Extensive research support	- Highest compute cost (4 models) - Complex implementation (37+ details) - Training instability - Hyperparameter sensitive - Sample inefficient	Production systems with large compute budgets, complex reasoning tasks, code generation
DPO	- Simpler than PPO - No reward model needed - Stable training - Excellent for dialogue/safety - Low hyperparameter sensitivity	- Requires quality preference data - No online learning - Needs reference model - Limited exploration	Research/production with good preference data, dialogue systems, safety alignment, limited compute
ORPO	- Single-stage training - No reference model (20-30% memory savings) - Simplest implementation - Very stable - Fast training	- Slightly lower performance - Still needs preference data - Less theoretical foundation - Best for small models only	Resource-constrained environments, edge deployment, small model fine-tuning

GRPO	- Excellent at scale (7B+) - No critic network - Works with verifiable rewards - Online learning - Very stable	- Needs verifiable reward functions - High compute during training - Requires vLLM infrastructure - Limited to objective tasks	Math/code/reasoning tasks with verifiable outputs, large model alignment, DeepSeek-style systems

6.2 Decision Matrix by Constraints

Your Constraint	Recommended	Alternative
Limited GPU memory (<24GB)	ORPO	DPO with small model
Limited compute budget	ORPO or DPO	GRPO (if verifiable rewards)
Maximum performance needed	PPO	GRPO (for math/code)
Simple implementation required	ORPO	DPO
High-quality preference data available	DPO	PPO
Verifiable rewards (math/code)	GRPO	PPO
Dialogue/chat alignment	DPO	PPO
Complex reasoning tasks	PPO	GRPO
Small models (<7B)	ORPO	DPO
Large models (7B+)	PPO or GRPO	DPO
Fast iteration needed	ORPO	DPO
Fine-grained control required	PPO	GRPO

6.3 Cost-Benefit Quick Reference

Priority	Rank 1 (Best)	Rank 2	Rank 3	Rank 4
Lowest Cost	ORPO	DPO	GRPO	PPO

Fastest Training	ORPO	DPO	GRPO	PPO
Easiest Implementation	ORPO	DPO	GRPO	PPO
Best Stability	ORPO	GRPO	DPO	PPO
Best Complex Tasks	PPO	GRPO	DPO	ORPO
Best Overall Performance	PPO (with tuning)	GRPO (at scale)	DPO	ORPO
Most Flexible	PPO	GRPO	DPO	ORPO

6.4 Comparative Performance Findings

Recent comprehensive studies reveal:

- **PPO outperforms DPO** on challenging code generation tasks and complex reasoning
- **DPO performs comparably** to PPO on dialogue and summarization tasks
- **PPO achieves state-of-the-art results** on code competitions
- After hyperparameter tuning, rankings can shift significantly

6.5 Decision Framework

Choose PPO when:

- Working on **complex reasoning or code generation** tasks
- You need **maximum fine-grained control**
- Have **sufficient computational resources** (GPUs + infrastructure)
- Can invest in **careful hyperparameter tuning**
- Want to incorporate **multiple reward signals**

Choose DPO when:

- Have **high-quality preference data** and limited compute
- Working on **dialogue, summarization, or safety alignment**
- Need **simpler, more stable training**
- Want to avoid reward model training complexity

Choose ORPO when:

- Have **very limited memory** resources
- Want **simplest possible pipeline**
- Working with **smaller models** (125M-7B)

Choose GRPO when:

- Have **verifiable reward functions** (math, code correctness)
- Working with **larger models** (7B+)
- Want **online learning** without full PPO complexity
- Need **flexibility** in reward specification

6.6 Resource Requirements Summary

Compute Ranking (highest to lowest):

1. PPO - Requires 4 models, multiple training phases
2. GRPO - Requires vLLM server, online generation
3. DPO - Requires 2 models, single training phase
4. ORPO - Requires 1 model, single training phase

Implementation Complexity Ranking (hardest to easiest):

1. PPO - 37+ implementation details, highly sensitive
2. GRPO - Moderate complexity, online RL
3. DPO - Simple classification-like loss
4. ORPO - Simple monolithic approach

Stability Ranking (most to least stable):

1. ORPO - Very stable
2. GRPO - Excellent stability
3. DPO - Good stability
4. PPO - Prone to instability, requires careful tuning