Generative and Agentic AI in Practice

DS 246 (1:2)

Lecture Topics

Phase 1: Foundations (Weeks 1-8, leading to Midterm)

Week 6 (Sep 17): LLM Training Methods: RLHF, DPO, GRPO & Beyond

- Topics: .

Learning Objectives

- Understand the evolution from supervised fine-tuning → preference-based optimization
- Explain the key differences between RLHF, DPO, and GRPO
- Connect reward optimization to reasoning capabilities in LLMs
- Recognize the role of process vs. outcome supervision
- Critically evaluate frontier alignment strategies (KTO, IPO, RLAIF, Constitutional AI)
- Apply knowledge to practical deployment trade-offs in industry contexts

Why Alignment Methods Keep Evolving

Why Alignment Methods Keep Evolving

Raw LLMs ≠ Aligned Assistants

- Great at language → poor at following preferences

RLHF solved first-generation problems... but:

- Expensive (3x compute vs. SFT)
- Reward model instability
- Human feedback bottleneck

Why Alignment Methods Keep Evolving

DPO, GRPO, and Variants

- Simpler pipelines, lower cost, more stable

Frontier Methods (2024–2025)

- AI feedback (RLAIF) replaces human labels
- Constitutional AI adds rule-based alignment
- Verifiers and debate scale oversight

Alignment is not a solved problem → methods evolve as models, tasks, and risks evolve.

Why Alignment Methods Keep Evolving

Why do some AI systems seem more helpful, harmless, and honest than others?

→ Because behind the scenes, they’re trained with different alignment strategies."

2.

Foundations

From Supervised Learning to Human Alignment

Alignment Challenge : Traditional Supervised Fine-Tuning

Traditional Supervised Fine-Tuning

- Supervised Fine-Tuning (SFT): Train on human-written prompts & responses
- Works well for capability learning
- But → does not guarantee alignment with human preferences

Alignment Challenge : The Gap Between Capabilities & Preferences

Models learn patterns, not values

Can generate:

- Harmful content
- Nonsensical answers
- Unhelpful / verbose responses
- Deployment Risk: Capability ≠ trustworthiness

Alignment Challenge : Alignment as a Requirement

Why Alignment Matters

- Trust & safety
- Industry adoption
- Regulatory compliance
- **Key Insight:** Improving LLMs for self-refinement tasks = alignment problem

Reward-Based Optimization Paradigm

Beyond Maximum Likelihood Training

SFT objective:

- maximize probability of training responses
- But: this ignores human preference signals
- Solution: Reward models as optimization signal

Reward-Based Optimization Paradigm

Core Reward-Based Framework

Pipeline:

- Base Model
- Human Preferences (pairwise/comparisons)
- Reward Model
- Policy Optimization

Reward-Based Optimization Paradigm

Key Principles of Reward Optimization

- Human feedback replaces “likelihood”
- Reward models approximate preferences
- Policy optimization aligns generation behavior
- Emerges as alignment paradigm

The Reasoning Connection

Why Reasoning Matters for Alignment

- Multi-step reasoning = hard to verify
- Need supervision at:
  - Process-level (steps)
  - Outcome-level (final answer)
- Same challenge faced in reward modeling

The Reasoning Connection

Process vs Outcome Evaluation

- Process Supervision: Check each reasoning step
- Outcome Supervision: Only final correctness matters
- Connection: Both mirror reward model training challenges in RLHF

Summary

- Traditional SFT → limited alignment
- Reward-based optimization → aligns with human preferences
- Reasoning supervision → the next frontier

3.
RLHF – Reinforcement Learning from Human Feedback

RLHF Architecture

What is RLHF?

- Reinforcement Learning from Human Feedback
- Trains models to follow human preferences
- Became the de facto standard in alignment after GPT-3.5 / ChatGPT

RLHF Architecture

Three-Stage Pipeline

- Supervised Fine-tuning (SFT) → base instruction following
- Reward Model Training → learn from pairwise preferences
- Policy Optimization → reinforcement learning on outputs

RLHF Architecture

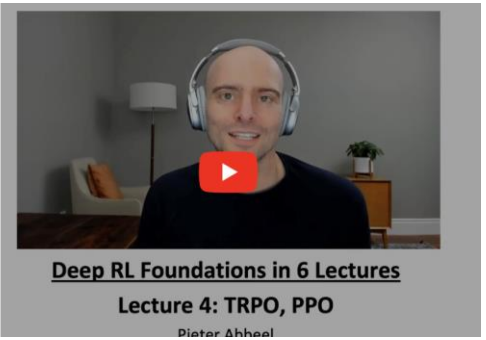
Reward Model

- Input: Two responses (A, B) to the same prompt
- Human labels: Which one is better?
- Reward Model = classifier that predicts preference probability

RLHF Architecture

Proximal Policy Optimization (PPO)

- Model generates → Reward model scores output
- Policy gradient update (often PPO, TRPO(Trust Region Policy Optimization))
  - to address a major challenge: taking a step that is too large and "catastrophically" changing the policy, which can lead to unstable training and a collapse in performance
- Kullback-Leibler (KL) -divergence penalty keeps model close to original SFT distribution
  - a measure of how one probability distribution (the new polic second, reference probability distribution (the old policy)



https://www.youtube.com/watch?v=2GwBez0D20A

RLHF Architecture

Why RLHF Works

- Moves beyond “imitating data” → optimizes for desired behavior
- Aligns with helpfulness, harmlessness, honesty (HHH)
- Empirically improved ChatGPT & Claude

RLHF Architecture: Practical Implementation

Data Collection: Human Preferences

- Methods:
  - Response ranking
  - Pairwise comparisons
- Critical: inter-annotator agreement and bias control

RLHF Architecture: Practical Implementation

Training Dynamics & Challenges

- Reward hacking risk (model exploits loopholes)
- Distribution shift (model drifts from pretraining data)
- Computational cost (~3× base model training)

RLHF in Reasoning Systems

RLHF for Complex Tasks

- Works especially well in:
  - Mathematical problem solving
  - Code generation & debugging
  - Multi-step logical inference

RLHF in Reasoning Systems

Example: Prompt Generator with RLHF

- Policy = LLM prompt generator
- Reward = downstream model success
- Feedback loop improves **prompt quality + reasoning**

Summary

- RLHF = cornerstone of alignment
- Bridges gap between raw capability and human values
- But → expensive, fragile, and spurred the rise of simpler alternatives (DPO, GRPO, etc.)

DPO: Direct Preference Optimization, GRPO: Group Relative Policy Optimization

4.

DPO – Direct Preference Optimization

Motivation for DPO

Why Look Beyond RLHF?

- RLHF is powerful but has limitations:
  - Complex 3-stage pipeline
  - Reward model instability (overfitting, bias)
  - High computational cost (~3× SFT)
  - Hyperparameter sensitivity

Enter DPO

Direct Preference Optimization (DPO)

- Simpler alternative
- Learns directly from preference data
- Eliminates:
  - Reward model training
  - Reinforcement learning loop
- Key Idea: Directly optimize policy from comparisons.

Enter DPO

Benefits of DPO

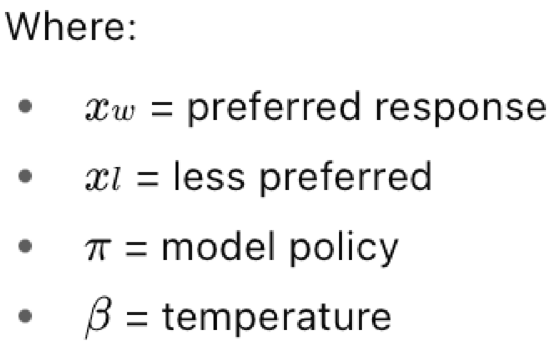
- Single-stage training → faster, easier
- Stable convergence (fewer hyperparameters)
- Lower compute (~close to SFT cost)
- Empirically comparable to RLHF

DPO Mathematical Framework

Core Insight

- Instead of explicit rewards → optimize preferences directly
- Human preferences = probability distribution over outputs
- Equation Insight:

$$L = -\log \sigma(\beta \log \pi(x_w|y) - \beta \log \pi(x_l|y))$$



Where:

- x_w = preferred response
- x_l = less preferred
- π = model policy
- β = temperature

DPO Mathematical Framework

Intuition Behind the Loss

- If model assigns higher probability to preferred output, loss ↓
- If not, loss ↑ → gradient pushes model toward better choice
- Logistic sigmoid (σ) ensures smooth preference scaling

DPO Mathematical Framework

DPO Pipeline

- Base Model
- Collect pairwise preference data
- Train directly with DPO loss
- Deploy aligned model

DPO vs RLHF

Performance Comparison

- Quality: Comparable in benchmarks
- Stability: DPO usually smoother training
- Speed: DPO converges faster
- Flexibility: RLHF handles richer reward signals

DPO vs RLHF

Trade-offs

- DPO Strengths:
  - Low cost, robust, simple pipeline
- DPO Weaknesses:
  - Limited to pairwise preferences
  - Cannot capture nuanced/multi-dimensional rewards
- RLHF Advantage: More expressive but costly

Summary

- DPO = lightweight alternative to RLHF
- Best for: general alignment tasks
- RLHF still relevant for: complex, multi-criteria optimization

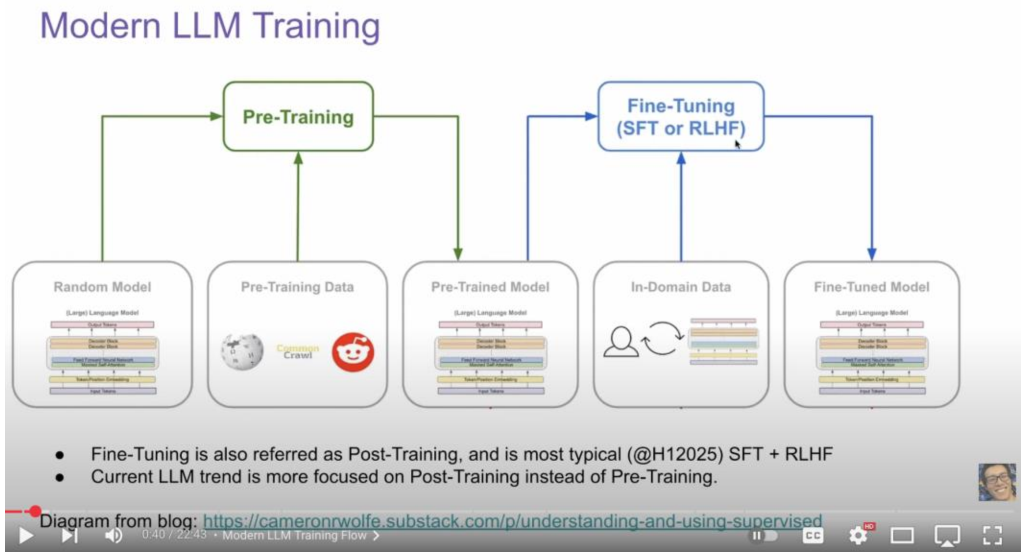
5.

GRPO – Group Relative Policy Optimization

GRPO Methodology

What is GRPO?

- Extends preference learning beyond pairs
- Uses groups of responses to a prompt
- Optimizes model by ranking within the group
- Developed by DeepSeek AI



Modern LLM Training

- Fine-Tuning is also referred as Post-Training, and is most typical (@H12025) SFT + RLHF
- Current LLM trend is more focused on Post-Training instead of Pre-Training.

Diagram from blog: https://cameronrwolfe.substack.com/p/understanding-and-using-supervised

https://youtu.be/aB7ddsbhhaU?feature=shared

GRPO Methodology

Core Innovation

- Unlike DPO (pairwise), GRPO learns from relative position
- Group-wise preference modeling:
  - Preferred > Neutral > Less-preferred
- Encourages more fine-grained signal extraction

Technical Approach

- Responses scored relative to one another
- Gradient updates depend on relative rank
- Objective encourages policy to maximize probability of higher-ranked outputs

GRPO: Advantages Over Pairwise Methods

Why Groups > Pairs?

- Pairwise limitation: ignores information in other responses
- GRPO advantage:
  - Uses all responses in a comparison set
  - Captures richer preference structure

Benefits of GRPO

- Sample efficiency → more info per human annotation
- Robustness → less sensitive to noise in single comparisons
- Better alignment for complex tasks (creative writing, multi-criteria problems)

Summary

Use Cases

- Creative generation: Stories, essays, design outputs
- Complex reasoning: Multi-step math & code tasks
- Multi-criteria optimization: e.g., helpfulness + safety + style

Summary

- GRPO generalizes preference learning beyond pairs
- More efficient & robust than DPO in feedback-limited settings
- Trade-off: Requires slightly more complex implementation

6.

Emerging Variants & Extensions

KTO – KL-regularized Preference Optimization

What is KTO?

- Extends DPO with explicit KL-divergence regularization
- Balances:
  - Staying close to base model
  - Moving toward human preferences

Why KL Regularization?

- Prevents mode collapse (over-optimizing narrow responses)
- Stabilizes training, especially with noisy feedback
- Bridges SFT-style MLE training with preference optimization

KTO – KL-regularized Preference Optimization

What is KTO?

- Loss = DPO loss + KL penalty
- Hyperparameter λ controls KL weight
- Produces models that are:
  - Stable (less overfitting to preferences)
  - Aligned without excessive drift

IPO – Implicit Preference Optimization

What is IPO?

- Learns preferences without explicit reward models
- Rewards emerge implicitly from training objective

Key Innovation

- Instead of defining a separate reward, IPO infers reward-like signals from preferences
- Approximates RLHF expressiveness but simpler
- Closer to DPO in efficiency

IPO – Implicit Preference Optimization

IPO in Practice

- Uses implicit gradient estimation
- Training remains end-to-end
- Benefits:
  - Avoids reward overfitting
  - More flexible than DPO
  - Retains stability advantages

Self-Learning from Contrastive Feedback

What is SLiC?

- Uses a contrastive loss instead of explicit reward model
- Learns from comparisons within data (synthetic or human-labeled)

Why Contrastive Objectives?

- Less human data required
- Generalizes well to new domains
- Efficient use of limited feedback → reduces annotation cost

Self-Learning from Contrastive Feedback

SLiC in Practice

- Model learns by pushing up good responses and pushing down poor ones
- Can leverage synthetic preferences (LLM-judged comparisons)
- Works as drop-in replacement for DPO in some settings

Other Contrastive / Synthetic Methods

Distillation from Judge Models

- Use a strong LLM (or verifier) to act as the “teacher”
- Generate preference labels automatically
- Student model learns from synthetic comparisons

Synthetic Pair/Ranking Generation

- Sample multiple outputs from model
- Use AI-based ranking to order them
- Train model with contrastive objective on generated rankings
- Pipeline: Model → Candidate Outputs → Judge → Ranked List → Training.

Emerging Methods Summary

- KTO: Adds KL-regularization → stability
- IPO: Implicitly models reward → balance of RLHF & DPO
- SLiC / Contrastive: Self-learning with synthetic contrastive signals
    - Reduce dependence on costly human annotation
    - Enable scalable preference data generation
    - Work best when paired with verifier/judge models

These methods are the "next wave" beyond DPO/GRPO — pushing toward scalable, stable alignment with less cost.

7.

AI Feedback & RLAIF

AI Feedback & RLAIF

What is AI Feedback?

- AI Feedback = using LLMs instead of humans to provide preference data
- Stronger models act as judges/verifiers
- Scales feedback generation across millions of samples

Why AI Feedback Matters

- **Cost Reduction:** Human annotations expensive & slow
- **Scalability:** AI can generate preference data at internet scale
- **Consistency:** Less noisy than human annotators (but biased!)

AI Feedback & RLAIF

AI Feedback Pipeline

- Generate candidate responses
- Judge model scores/ranks them
- Train student model using synthetic labels

AI Feedback & RLAIF

What is RLAIF?

- Reinforcement Learning from AI Feedback (RLAIF)
- Same principle as RLHF, but replace humans with AIs
- Popularized by Anthropic’s Claude and OpenAI internal systems

Benefits of RLAIF

- Scalable oversight → billions of preference data points
- Faster iteration → models can self-improve more rapidly
- Foundation for Constitutional AI (rule-based judging)

AI Feedback & RLAIF

Trade-offs & Limitations

- Bias propagation → judge inherits flaws of base model
- Echo chamber risk → models reinforce each other’s mistakes
- Scalability vs. Reliability trade-off: cheap but needs safeguards

Summary

- AI feedback = game changer for scaling alignment
- RLAIF in production (Anthropic, OpenAI) proves viability
- But → requires careful design to avoid bias amplification

8.

Self-Play, Debate & Multi-Agent Alignment

Self-Play Reinforcement

What is Self-Play?

- Self-Play = models generate challenges for themselves
- Inspired by game AI (e.g., AlphaZero)
- Creates adversarial training loop → models improve by competing

Self-Play for LLMs

- Model generates a task/problem
- Same or another model attempts to solve it
- Evaluation → reward signal → refinement

Self-Play Reinforcement

Why Self-Play Works

- **Robustness:** Model learns from hardest adversarial examples
- **Exploration:** Expands training distribution
- **Less reliance on humans:** feedback loop is automated

Debate Models

What is Debate in AI?

- Two (or more) LLMs argue opposing sides of a problem
- A judge (human or AI) evaluates arguments
- Goal: make truth easier to surface through structured debate

Debate in Practice

- Applications:
  - Factual Q&A Ethical dilemmas
  - Complex reasoning tasks
- Judge can be:
  - Human evaluator
  - Another LLM acting as referee

Debate Models: Benefits & Limitations

Benefits

- Scalable oversight → multiple perspectives
- Encourages models to expose weaknesses in reasoning

Limitations

- Debate quality depends on judge reliability
- Risk of persuasive but incorrect arguments

Summary

- Self-Play → models improve via adversarial tasks
- Debate Models → scalable oversight with multi-agent setup
- Both methods = frontier strategies for robust, aligned reasoning

9.

Process Supervision & Verifier Models

Process vs Outcome Supervision

Why Supervision Matters in Reasoning

- Traditional training = judge final output only
- Complex reasoning requires step-by-step oversight
- Example: math proof → each step must be valid, not just the final answer

Outcome Supervision

- Checks only final answer
- Simple, efficient
- But: errors in reasoning path may go undetected

Process vs Outcome Supervision

Process Supervision

- Verifier model scores each reasoning step
- Detects hallucinations, faulty logic early
- Crucial for:
    - Math proofs, Code execution & Scientific reasoning

Process vs Outcome: Comparison

Outcome Supervision

- Easier, cheaper
- Final correctness focus

Process Supervision

- Richer training signal
- Better generalization in reasoning tasks

Verifier-Based Optimization

What Are Verifiers?

- Specialized models trained to judge correctness
- Can evaluate:
  - Single step validity
  - Entire reasoning path quality
- Act as automated reviewers for LLM outputs

Process Reward Models

- Use verifier scores as reward signals
- Replace human step-checking with automated scoring
- Enables scalable supervision for reasoning-heavy domains
- Pipeline: LLM → Reasoning Steps → Verifier → Reward → Updated LLM.

Verifier-Based Optimization

Iterative Refinement

- Model generates reasoning path
- Verifier checks → flags errors
- Model self-corrects iteratively until consistent

Verifier-Based Optimization: Benefits & Challenges

Benefits

- Stronger reasoning performance
- Scalable without excessive human labor

Challenges

- Training reliable verifiers is difficult
- Risk: verifier inherits biases/errors

Summary

- Outcome supervision alone is insufficient for reasoning
- Process supervision + verifiers = richer, scalable feedback
- Enables step-level training signals → stronger alignment in reasoning tasks

10.

Constitutional AI & Rule-Based Alignment

Constitutional AI

What is Constitutional AI?

- Alignment via principles, not humans
- AI trained with a written “constitution” (ethical, safety, policy rules)
- Model self-critiques and refines outputs using those rules

Core Methodology

- Provide model with principles/constitution
- Generate responses
- Model critiques itself against principles
- Refines response without human label

Constitutional AI

Benefits of Constitutional AI

- Less human labeling → scalable
- Consistent ethical framework across tasks
- Deployable in sensitive domains (safety, law, medicine)

Limitations

- Quality of alignment depends on constitution quality
- Risk of rigid or biased principles
- Requires careful principle selection & balancing

Rule-Based Reward Shaping

What is Rule-Based Reward Shaping?

- Combines:
  - Human feedback
  - Rule-based signals (policies, safety rules)
  - AI feedback
- Produces hybrid reward models for alignment

Practical Implementation

- Encode rules: “Don’t reveal private info,” “Stay non-toxic,” etc.
- Rule-checkers assign scores automatically
- Combined with preference optimization for training

Rule-Based Reward Shaping: Benefits & Challenges

Benefits

- Practical in safety-critical systems
- Reduces reliance on scarce human annotations
- Provides hard constraints where needed

Challenges

- Rules can be overly rigid
- Hard to anticipate all real-world cases

Summary

- Constitutional AI: alignment guided by principles instead of human labels
- Rule-based reward shaping: hybrid method for safety domains
- Together → scalable, principle-driven oversight

11.

Integration into Reasoning Frameworks

Reasoning Paradigms

Why Reasoning Matters

- LLMs often fail on multi-step reasoning
- Alignment methods can boost:
  - Accuracy, Reliability and Transparency

Core Reasoning Paradigms

- Chain of Thought (CoT): Step-by-step decomposition
- Logic of Thought: Formal logic rules integrated
- Program of Thought: Code/execution-based reasoning
- Self-Refinement: Iterative improvement of outputs

Reasoning Paradigms

Example: Self-Refinement in Practice

- Model generates initial solution
- Critiques own reasoning
- Revises until consistent

Optimization Synergies

Preference Optimization + Reasoning

- Use preference models to favor better reasoning chains
- Train LLMs to:
  - Prefer correct step orderings
  - Avoid shortcuts or hallucinations

Process Verifiers + Debate

- Process verifiers: check each step
- Debate frameworks: multiple agents expose weaknesses
- Combined → stronger correctness guarantees

Summary

- Alignment methods integrate with reasoning frameworks
- Synergy: Preference optimization + verification + debate
- Critical for math, coding, science, high-stakes decisions
- Roadmap: SFT → RLHF/DPO → Reasoning Integration.

12.

Industry Practice & Deployment

Industry Practice & Deployment

Overview

- Choosing the right optimization method
- Cost-benefit trade-offs in practice
- MLOps integration & safety evaluation

When to Use Which Method?

- RLHF → rich, multi-criteria alignment (but costly)
- DPO → default choice for most applications (simple & efficient)
- KTO / IPO → when stability & implicit reward signals are needed
- RLAIF → large-scale alignment with reduced human labeling

Industry Practice & Deployment

Cost-Benefit Trade-Offs

- Cost Dimensions:
  - Human labeling effort
  - Compute requirements
  - Training complexity
- Benefit Dimensions:
  - Model quality & stability
  - Alignment with human preferences
  - Safety & oversight robustness

Industry Practice & Deployment

Real-World Lessons

- Data quality is the bottleneck, not just size
- Iteration cycles more important than one-off training
- Trade-off: cheaper pipelines (DPO/RLAIF) vs more flexible ones (RLHF/KTO/IPO)

MLOps Integration

- Alignment methods → must fit into pipelines:
  - Data collection & versioning
  - Training orchestration
  - Continuous monitoring
- Integration with evaluation benchmarks across capability & safety

Industry Practice & Deployment

Safety Evaluation Pipelines

- Red-teaming & adversarial testing
- Automated toxicity, bias, and jailbreak detection
- Continuous feedback loops for real-world monitoring

Summary

- No “one-size-fits-all” alignment method
- Practical deployment = balance of cost, quality, safety
- Alignment success depends on tool choice + pipeline integration

12.

Future Directions

Future Directions

Recap

- We explored:
  - Core methods: RLHF, DPO, GRPO
  - Variants: KTO, IPO, SLiC
  - Scalable oversight: RLAIF, Self-Play, Debate, Verifiers, Constitutional AI
- Key Message: Alignment is an ongoing process, not a solved problem.

Future Directions

Future Research Frontiers

- Iterative Preference Learning → continuous refinement loops
- Multi-Objective Optimization → balance capability, safety, efficiency
- Agentic Training Pipelines → multi-agent systems improving one another
- Scalable Oversight Challenges → automated judges, bias control, robustness

Future Directions

Practical Takeaways

- Start Simple: DPO/IPO for most use cases
- Scale Carefully: RLHF/KTO for complex trade-offs
- Automate Oversight: AI feedback, verifiers, debate
- Keep Iterating: Continuous alignment = continuous improvement

The future of AI alignment will not be solved in one breakthrough – it will be an iterative partnership between humans (domain expert), AIs, and the principles we encode.