ChatGPT

Gemini 3 FlashのAgentic RLとGemini 3 Proとの技術比較

Gemini 3 Flashの高性能を支えるAgentic RL

Agentic RLの理論的背景（従来のRLとの違い）

Agentic RL（エージェント的強化学習）とは、大規模言語モデル（LLM）にエージェントのような自主的行動能力を持たせるための強化学習手法です。従来のRLHF（人間のフィードバックによる強化学習）は主に**静的な1ターンの出力**に対する人間評価を用いてモデルを調整していました。一方、Agentic RLではモデルが**対話的な環境で連続的に意思決定**を行います [1]。具体的には、モデルが**自ら「観察→思考→行動」を繰り返す**「エージェント的ループ」を形成し、長いシーケンスのステップ（マルチステップ）でゴール達成を目指す点が特徴です [2]。各ステップではモデルが**内部の思考（チェーン・オブ・ソート）**を展開し、それに基づいて環境に作用するアクション（例えばツールの呼び出しや回答の一部生成）を選択します。この**思考と行動の二段構え構造**により、LLMはより長い文脈で計画立案・実行が可能になります。

従来の強化学習と比較したAgentic RLの理論的な違いとして、**クレジット割当（credit assignment）の問題**があります。1ターン完結のタスクでは、出力全体に対する人間評価スコアをそのまま報酬とみなし、モデル全体を更新すれば良いですが、エージェントのように**一連の行動の末に遅延報酬**が得られる場合、どの行動が成功に寄与したかの評価が困難です [1]。また各ステップで**状態が部分観測（非Markov性）**であったり、報酬が極めて疎かつ最終結果にならないと判定できない（例：長い対話の最後に成功/失敗が判明）ことも多く、従来のRL手法をそのままLLMのマルチステップタスクに適用すると**高い分散と不安定性**が生じます [2]。Agentic RLでは、この問題に対処するため**途中経過に対する評価**（プロセススーパービジョン）を組み込んだり、長い軌跡を扱える新たな報酬設計を導入するなどの工夫が理論的背景として提案されています [3] [4]。例えば**Implicit Step Reward**のアイデアでは、軌跡全体の優劣から各ステップへの**暗黙的な部分報酬**を割り振り、長い試行の中でも逐次的に学習信号が得られるようにします [5] [6]。このようにAgentic RLは、LLMを**能動的なエージェント**として動作させるために、従来型RLとの差異（マルチステップ・長期的な意思決定、遅延報酬、非マルコフ性）に対処する理論的枠組みです。

Agentic RLの実装と学習アルゴリズム（訓練・最適化手法）

実装面では、Agentic RLを用いたLLMの訓練は**大規模Transformerモデル**に対し、強化学習のループを統合する形で行われます。まず基盤となるLLMを大規模データで事前学習し、次に**エージェントタスクのデモや模擬対話**で**教師あり微調整**（SFT）して基本的な振る舞い（ツールの使い方や思考様式）を身につけさせます [7] [8]。その後、本格的にAgentic RL訓練として**環境内でモデル自身に試行させる反復学習**を行います。モデル（ポリシー）は環境と対話して**軌跡（マルチステップの行動系列）**を生成し、各軌跡には最終的なタスク成功度に基づく**エピソード報酬**が与えられます [9]。さらに先述のように、軌跡を**比較評価**する報酬モデル（例：成功軌跡と失敗軌跡をランク付けする**アウトカム評価器**）を用いて**相対的な好み**を学習し、それを元に**各ステップへのスコア（ステップ報酬）**を割り振ります [10] [11]。これにより、**各行動ごとのメリット**と**軌跡全体としてのメリット**の両方を算出し、PPOなどの方策勾配法によってモデルパラメータを更新します [12] [13]。代表的なアルゴリズムとして、Google DeepMindが提案した**iStar**（implicit Step Reward）では、軌跡のペアに対する**DPO（Direct Preference Optimization）**に基づくロスで**プロセス報酬モデル**を学習し、そこから得たステップ報酬＋エピソード報酬で方策をアップデートする**自己強化ループ**を構築しています [5] [6]。この交互最適化により、方策（LLM）と報酬モデルが互いに改善し合い、安定した学習が可能になります [14]。学習アルゴリズムには**GRPO**（Generative PPO）や**RLOO**（Rollouts with Off-policy

Optimization）などLLM向けに調整されたPPO派生が用いられ、加えてDPOやDAPOといった**直接方策最適化**の手法も組み合わせて安定性・効率を高めています [15]。

モデルアーキテクチャ自体は**Gemini 3のような基盤モデル**の場合、テキストと画像等を統合処理できる**マルチモーダルTransformer**ですが、Agentic RL用にいくつか拡張が加えられています。第一に、モデルが**内部思考を隠れた形で展開**できるよう、**Thought Signature**と呼ばれる特殊トークンを用いたプロンプト構造が実装されています [16] [17]。例えばGemini 3では、モデルが「考えている内容（理由や方針）」と実際の「行動（ツール呼び出しや最終回答）」とを区別し、内部思考には`<thought>`のようなタグ付きテキストを生成させて、それをユーザには見せず次の入力に引き継ぐ仕組みがあります [16] [18]。このような**隠れチェーン・オブ・ソート**により、モデルは自ら中間ステップを展開しつつ誤った思考内容をユーザに晒さない安全性も確保します。また**Function Calling API**（ツール呼び出し）も組み込まれており、モデル出力がJSON形式で「どのツールをどの引数で使うか」を表明できるようにアーキテクチャ上サポートされています [19] [20]。Gemini 3ではGoogle検索やコード実行、URL読込などいくつかの外部ツールが組込みでサポートされており、モデルは出力中にこれらを**API関数として呼び出す**ことができます [21]。Toolformerの研究のように、事前学習済みモデルに対してツール使用例を微調整することで、モデル自身が**「知識が不十分なときに検索する」「コードを生成して実行し結果を読む」**といった挙動を学習しています [22] [23]。このように**Agentic RLの学習アルゴリズムと実装**では、(a) 長期的な報酬設計（最終+中間報酬）、(b) 内部思考用のプロンプト構造、(c) ツール使用アクションの埋め込み、といった工夫を凝らすことで、エージェント的行動選択を安定して最適化できるようになっています。

GoogleにおけるAgentic RLの実用化・展開（Gemini 3での活用事例）

Google（DeepMind）はGeminiシリーズで培ったAgentic RL技術を**実用プロダクトに統合**しています。Gemini 2（前世代）で**エージェント時代の幕開け**を宣言し、Gemini 3ではそれをさらに発展させて**あらゆるアイデアを形にできる**汎用AIを目指しています [24] [25]。Gemini 3 Proの公開時には「高度な推論・マルチモーダル理解に加え、エージェント的なコード生成や対話能力でフロンティア性能を示した」とされ [26]、実際に**Box社**ではGemini 3による手書き文書解析の精度向上、**Bridgewater社**では非構造マルチモーダルデータの推論への活用など、様々な企業がGemini 3の高い推論力を活かしています [27] [28]。

中でも注目すべきは**自律エージェント応用**です。Googleは**Gemini Enterprise**という企業向けの安全なエージェント実行プラットフォームを提供し、複雑な業務フローをAIエージェントで自動化できるようにしています [29]。また開発者向けには**Google Antigravity**という新しいエージェント開発環境を公開し、コード生成エージェントなどを**対話的に構築・テスト**できるようになっています [30] [31]。Antigravityのデモでは、Gemini 3を用いたエージェントが「フライト追跡アプリ」を一から自動で作成する様子が紹介されています [32]。具体的には、エージェントがユーザの高レベル要求を受けて自らタスク分割し、必要なコードを**順次生成・実行**し、ブラウザ上でアプリをテストする、という**エンドツーエンドのワークフロー**を実現しています [32]。この過程で、Gemini 3のエージェントは内部で先述の**思考ステップを計画（プランニング）**し、適宜**検索ツールで情報補強**しつつ、**コード実行ツールで検証**するといった振る舞いを見せます。Googleによると、Gemini 3ではGemini 2で可能だったエージェント機能をさらに洗練し、**長い地平（long-horizon）でも破綻しにくい計画性**を向上させたとのことです [24]。実際、エージェントの長期タスク遂行能力を測る**Vending-Bench 2**というベンチマークでGemini 3はトップ性能を示しています [24]。

Gemini 3 FlashはこうしたAgentic能力を**低遅延で実現**することで、現実の応用をさらに拡大しました。Flashモデルは「ほぼリアルタイムの応答速度で複雑なエージェント・アプリケーションを支える」ことを狙っており [33] [34]、実際**対話型のカスタマーサポートエージェント**や**ゲーム内AIアシスタント**など**インタラクティブ用途**に適しています [35]。Flashの投入により、今まで大きなモデルでは遅すぎたマルチステップ推論が実用的な速度になり、**生産システムで高度なAIエージェント**を動かせるようになりました [36] [37]。例えば**JetBrains社**では、コーディングAIアシスタントにFlashを採用することで**Pro同等の賢さでレイテンシを大幅短縮**し、ユーザごとの使用クレジット内に収まる安定した**マルチステップAgentループ**を提供できるようになったと報告しています [36]。**Replit社**も「Flashは十分な性能と高速性を兼ね備え、初めてコーディング

エージェントの中核ループを実用レベルで駆動できるモデルだ。特にツール使用能力とコーディングスキルが非常に強力だ」と評価しています [37] 。このように、GoogleはGemini 3ファミリーを通じて**Agentic RLを現実世界のサービスに展開**しており、エージェント開発プラットフォーム（AntigravityやVertex AI Agent Builder等）や各種統合環境（AI Studio、Gemini CLI、Android Studio連携など）を整備することで、開発者や企業が**高度な自律エージェント**を構築・利用できるエコシステムを構築しています [38] [39] 。

Gemini 3 FlashとGemini 3 Proの違い（技術仕様比較）

Gemini 3には大きく**Pro**モデルと**Flash**モデルの2系統があります。Gemini 3 Proは**最高性能**を追求したフラッグシップモデルで、Gemini 2.5までの世代を大きく凌駕する**高度な推論力・マルチモーダル対応**を備えています [40] 。一方のGemini 3 Flashは、Proの持つ推論能力を維持しつつ**応答速度と効率を極限まで高めた**モデルで、**推論コストはProの4分の1以下**、レイテンシは飛躍的短縮を実現しています [23] 。以下の表に、Gemini 3 ProとFlashの主な技術的差異をまとめます。

比較項目	**Gemini 3 Pro** (3.0 Pro)	**Gemini 3 Flash** (3.0 Flash)
推論性能 （速度と応答品質）	大規模モデルによる最高水準の応答品質。複雑な推論や知識問題で最先端の精度を達成 [40] 。ただしモデルサイズが非常に大きいため**推論レイテンシは高め**で、リアルタイム処理には不向き。 ※Gemini 3 Pro Previewは1Mトークン文脈に対応し、高度な推論を行うが、Flash比で遅延が大きい。	Pro級の知能を持ちながら高速最適化されたモデル。**2.5 Proを上回る性能でありながら約3倍高速** (2.5 Pro比) を実現 [41] 。思考深度を低く抑えた場合でも旧世代の高思考設定より高精度を発揮する効率の良さ [41] 。低遅延で**ほぼリアルタイム応答**が可能 [35] 。 ※Flashは応答の初手が速く対話体験が軽快。高負荷下での処理並列性（レートリミット）もProより向上 [23] 。
学習効率 （モデル最適化とパラメータ）	大規模学習による最高精度志向。推定パラメータ数はFlashより多く、事前学習・RLHFともに膨大な計算資源を投入。人間のフィードバックも大量に使用し、**トレーニングコストは非常に高い**。その分、多様な知識と汎用能力を保持。	高効率チューニングによる性能向上。Proと**同じ基盤モデルを効率的に再最適化**したとされ、少ない追加学習でPro級の能力を達成 [42] 。特に**エージェント的タスクでのRL強化**により、パラメータ当たりの性能が向上し、より小規模でも高精度を発揮（詳細なサイズは非公開だが1兆+規模と推測 [42] ）。学習においてはiStarなど**サンプル効率の高いRL**手法を用いることで、学習安定性と迅速な性能向上を実現 [43] 。
マルチモーダル能力 （視覚・音声対応など）	最高クラスのマルチモーダル理解。画像・PDF・動画・音声入力に対応し、**視覚理解・空間認識**で最先端 [44] 。複数画像の比較やドキュメントの一貫性検証、大規模レポートの要約など**高度な視覚推論**が可能と報告 [45] 。画像生成は別モデル（Nano Banana Pro）に分離されているが、画像編集や解析では最も強力。 ※Proはドキュメント理解やUI画面解析、動画内容の要約などでSOTA性能 [46] 。	高度なマルチモーダル処理を維持しつつリアルタイム化。**画像・音声・動画入力の処理精度はProと同等級**ながら、Flash独自の**メディア解像度パラメータ**で処理コストを制御可能 [47] 。**最先端の視覚・空間推論能力**を備え、視覚入力に対するカウントや位置特定といった高難度タスクも実行 [23] 。加えて**コード実行による画像解析**機能を新搭載し、画像内のズームや物体カウント・編集をコード生成で行う先進機能も持つ [48] 。

比較項目	**Gemini 3 Pro** (3.0 Pro)	**Gemini 3 Flash** (3.0 Flash)
ツール使用能力 （Toolformer的設計）	高度なツール呼び出し能力を持つ。検索や電卓、コード実行など**外部ツールを適切に使いこなす**よう訓練されている [49] [50]。Terminal-Bench 2.0（ターミナル操作能力評価）では**54.2%**のスコアを達成し、一連のコマンド操作での課題解決を学習 [49]。デフォルトでは**高い思考レベルで計画→ツール実行を慎重に行う**傾向。	ツール使用に特化した調整で軽快な操作。**複数の内蔵ツール**（Google検索、ファイル検索、コード実行、URL読込）をサポートし、モデルが自発的にAPI呼出を挿入 [21]。とくに**コーディングエージェント**用途でツール利用性能が顕著に向上しており、Flashではツールを駆使したコードバグ調査や自動修正が高速かつ正確 [51] [52]。**低思考レベルでもツール連携が破綻しにくい**よう強化されており、プロンプト内で連続して複数ツールを呼ぶシナリオにも耐える設計（Sigトークンによる状態管理） [53] [18]。
エージェント的行動 （Agenticループ実行）	長大なタスクの遂行力に優れる。高思考モードでは入念にプランニングし、**複雑な目標を逐次達成**する能力が最高。例えば対話エージェントとして複数ユーザ要求を並行処理したり、ゲームAIとして先読み戦略を立てるなど**長期的最適行動**に秀でる。ただし**ステップ当たりの時間は長め**で、複数ステップを要すると遅延蓄積が大きい。	リアルタイムエージェント動作に最適。**高速な思考・行動サイクル**でエージェントループを回せるため、ユーザとやり取りしながら何度もツール実行・判断を繰り返す対話型シナリオで威力を発揮 [54] [35]。**Proに匹敵する深い推論力**を維持しているため、小刻みな思考でも重要ポイントを外さず行動できる [41]。結果として**低コストで安定したマルチステップ実行**が可能となり、商用サービスで複雑エージェントを**スケーラブルに運用**できる [36]。

上述の比較から分かるように、Gemini 3 Flashは**Gemini 3 Proの能力的土台を引き継ぎつつ、実用上ネックとなる速度・コストを大幅に改善**したモデルです [26] [23]。特に**推論のパフォーマンスあたりの効率**ではFlashが群を抜いており、「**低い思考レベルでも高精度**」というチューニングが施されています [41]。一方、Gemini 3 Proは**モデル規模と事前学習データの豊富さ**から来る包括的知識を持ち、極限まで難しいタスクや創造的応答では依然として最高性能を示す場面もあります。またProは**デフォルトで深い内部思考（Dynamic thinking）**を行う設計で、複雑な要求に対してはじっくり推論して最適解を導く傾向があります [55] [56]。Flashは**思考レベルを開発者が細かく指定可能**で、必要に応じて`minimal`から`high`まで**推論深度を調整**できます（Gemini 3 Proは現在`low`か`high`のみ対応） [56] [57]。この違いもあり、Flashは**用途に応じて軽量モードから高度推論モードまで柔軟**に使い分けできるのに対し、Proは常に高性能だがコスト固定という位置づけです。総じて、**Gemini 3 Proは最大能力が必要なケースに、Gemini 3 Flashはほとんどのケースで十分な能力を高速に提供するエンジンとして**使い分けられると考えられます [23] [41]。

Agentic RLと他の強化学習手法（RLHF、DPO）の比較

最後に、Agentic RLと代表的なLLM向け強化学習手法である**RLHF**（Reinforcement Learning from Human Feedback）および**DPO**（Direct Preference Optimization）を、いくつかの観点で比較します。

- **学習構造:**
- Agentic RL: モデルが**環境と対話しながら**マルチターンの軌跡を生成し、軌跡全体の成否に基づきポリシー（モデル）を更新する。エージェントは**逐次決定プロセス**（ステップごとに観察→思考→行動）を経るため、学習は**軌跡（エピソード）単位**の強化学習となる [1]。各ステップにも内部思考が絡む

複雑な構造で、学習には軌跡ごとのフィードバック信号を細分化する工夫（例: implicit step reward）が必要になる [2]。
- RLHF: モデルの**単発の出力**に対する人間の好みフィードバックを用いる手法。**学習は3段階構成**（(1)教師あり微調整→(2)人間の好みデータで報酬モデル学習→(3)その報酬を用いてモデルをRLで最適化）となり、基本的に**1ターン（プロンプト→応答）**を1エピソードとみなす [58]。マルチターン対話も可能だが、各応答ごとに評価するか会話全体を評価するか設計次第で、Agentic RLほど長い行動系列を直接最適化する構造ではない。

- DPO:　RLHFの最終RLステップを省き、**1段階で方策を直接最適化**する手法です [59]。学習構造は**人間の選好データに対する直接の方策勾配**で、ポリシーモデルを**教師あり分類**のように訓練します [60]。具体的には、あるプロンプトに対して人間が選んだ「好ましい応答」と「好ましくない応答」のペアを用意し、モデルが好ましい方を高確率で生成するように**確率分布を調整**します [60]。これを**単一の最適化ループ**で行うため、RLHFのような複雑な繰り返し構造がありません。

- **目的最適化関数:**

- Agentic RL: **環境内の長期的な目標達成**を最大化するのが目的です。最適化関数はエピソード全体の報酬（タスク成功度など）を期待値で最大化する典型的な強化学習目的ですが、Agentic RLではさらに**中間ステップの価値**も考慮します [9] [11]。すなわち、$J = E[\sum_t (r^{\text{step}}t + \mathbf{1}$を組み合わせて方策の期待報酬を定義します。これにより} \cdot r^{\text{final}})]$のように、各時刻$t$のステップ報酬$r^{\text{step}}_t$と最終時刻の結果報酬$r^{\text{final}}**グローバルな目標達成**と**ローカルな行動の良し悪し**の双方を最適化します。例えばiStarでは、軌跡の良否を判定する**アウトカム報酬モデル**と、そこから導いた**ステップごとの相対報酬**を組み合わせて方策を更新する**複合目的**を導入しています [6] [11]。
- RLHF: **人間の好みスコア**を最大化する目的関数です。具体的には、報酬モデルR_ϕが入力xとモデル出力yにスコア$R_\phi(x,y)$を与えるとして、方策π_θの目的は$E_{y\sim \pi_\theta}[\ R_\phi(x,y)\ ]$を高めることです [61] [62]。PPO実装ではこれに**KLペナルティ**を加えて、元のモデル分布からの乖離が大きくなりすぎないよう制御する目的を用います [63]。平たく言えば、「人間がより好む回答をする」ようモデルの確率をシフトさせる最適化です。

- DPO: **人間の選好確率を直接最大化**する目的です。報酬モデルを明示的に介さず、「与えられたプロンプトxに対し、好ましい回答y_wの確率が好ましくない回答y_lより高くなるようにする」という制約を目的に組み込みます [60]。具体的には**ブラッドリー・テリー模型**を応用し、$\sigma(\log \pi_\theta(y_w|x) - \log \pi_\theta(y_l|x))$を最大化（人間の選好に適合）するような損失を定義します [64]。結果として**RLHFの目的と同等**の「人間に選ばれる回答を高確率化する」目標を、**対数確率差の最大化問題**として直接解く形になります [65] [64]。このDPO目的は**凸に近く**、かつ**KLペナルティ相当も自動で内包**しているため、安定して最適化できる点が特徴です [64] [66]。

- **スケーラビリティ:**

- Agentic RL: **学習コスト・サンプル効率の両面で課題**があります。エージェントを環境内で訓練するには大量のインタラクション試行が必要で、1エピソードが長いため**1回の試行で消費する計算量も大きい**です。大規模LLMでこれを行うのは非常に重く、かつ環境によっては再現性が低い（同じ入力でも相手があると揺らぐ）ため**デバッグも難しい**です。しかし最近の研究では、iStarが示したように**ステップ報酬によるクレジット割当でサンプル効率を向上**でき、従来より**高速に性能改善**できることが報告されています [67] [68]。実際、iStarはベースラインのPPOより少ないステップで高い報酬に達し、最終性能も向上するなど**データ効率と安定性が向上**しています [43] [67]。それでも環境との相互作用を伴うAgentic RLは、基本的には**スケールさせにくい**手法であり、膨大な計算資源を投入できる大企業でこそ可能なアプローチと言えます。
- RLHF: **実用上スケール実績があるが複雑**な手法です。既にOpenAIのGPT-4などでも採用されているように、巨大モデルへの適用事例があります。ただし**学習パイプラインが3段階**と長く、人間による

フィードバックデータ収集や報酬モデル学習に大きなリソースを要します [64]。加えて、RLフェーズの**ハイパーパラメータ調整が難しく不安定**なため、大規模モデルではカスタムな工夫（例: KL制御の強化や経験的チューニング）が必要になります [64]。そのため組織的にはスケールできますが、技術的負債（調整コスト）が大きいと言えます。

- DPO: **シンプルさゆえにスケーラブル**な手法です。RLHFのような複雑なループが無いため、基本的に**通常のファインチューニングと同程度**の手間で実行可能です [65] [64]。報酬モデルを別途学習する必要もなく、モデルの出力確率を直接調整するだけなので、**計算資源や時間を節約**できます [65]。実験ではRLHF並み、あるいはそれ以上の性能を**少ない計算コスト**で得られるケースが報告されています [64]。従って、人手によるチューニング負荷や学習安定性の懸念も少なく、大規模モデルへの適用も比較的容易だと考えられます。

- **人間のフィードバック活用範囲:**

- Agentic RL: 環境から得られる**自動的な報酬信号**を活用する場合が多いです。例えばゲームやタスクにおける**スコア**や**成功/失敗フラグ**など、人手を介さず得られる報酬が定義できれば、人間の逐次フィードバック無しで学習可能です。しかし、対話エージェントのように**評価が主観的**な場合は、報酬モデルに人間ラベルで学習させたり、途中ステップに人間の介入（プロセスフィードバック）を入れることもあります [3]。もっともAgentic RLの目標は、極力**密な自動報酬**で人手のラベル付けを減らす点にあり [4]、iStarでは環境最終結果さえ定義できれば中間はラベル不要とするなど、**ラベル効率の向上**が図られています [4]。総じて、Agentic RLは**必要最低限の人間フィードバック**（タスク設計と最終評価基準の設定程度）で回せる場合もありますが、環境やタスクによっては人の介在が必要になる柔軟な立ち位置です。
- RLHF: **人間のフィードバックが中心**の手法です。報酬モデルの訓練データとして、**人間アノテータが行った出力同士の比較**や**評価スコア**を大量に使用します [61] [69]。フィードバック形式も**数値評価・ランク付け・テキスト修正**など多様なものを扱えるのが強みで [70]、必要に応じて詳細な指示や基準を人間が設けてモデルにフィードバックできます。ただし人間ラベルは主観やばらつきもあり得るため、大規模に集める際は品質管理が課題となります。
- DPO: **人間のフィードバックは二択比較のみ**を用います。基本的に**好ましい/好ましくないのバイナリ選好データ**が前提となり [70]、RLHFのようなスコアや自由記述のフィードバックは直接扱えません。したがって、人間から集めるデータも**ペア比較**が中心で、これはユーザにとって直感的でない場合もありますが、評価基準が明確で統一しやすい利点もあります [70]。DPOでは**LLM自体を報酬モデル代わりに利用**するとも言われ、追加の人間ラベルを大量に用意するのではなく、既存の比較データでモデルを直接最適化するため、必要なフィードバックデータ量も比較的少なくて済みます [71] [65]。

以上の比較をまとめると、**Agentic RL**はエージェント行動の獲得に不可欠な手法であり、環境相互作用を通じて**自律性と連続的推論能力**を付与しますが、学習の複雑さ・コストは高めです。一方、**RLHF**は対話モデルの**出力品質やユーザ嗜好への合わせ込み**に威力を発揮し、既に多くの巨大モデルで採用されていますが、訓練が不安定で手作業も多いという課題があります [64]。そして**DPO**はRLHFの有効性を維持しつつ**シンプルさと安定性**を追求した新手法で、今後RLHFに代わって主流になる可能性も示唆されています [64] [66]。Google DeepMindもAgentic RLの文脈でDPO的な最適化（trajectory DPO）を取り入れており [72]、こうした手法の組合せによって**Geminiシリーズの飛躍的な性能向上**（特にFlashモデルの効率改善）を実現していると考えられます。 [41] [73]

---

[1] [2] [3] [4] [5] [6] [9] [10] [11] [12] [13] [14] [15] [43] [67] [68] [72] [73] Agentic Reinforcement Learning with Implicit Step Rewards
https://arxiv.org/html/2509.19199v3

[7] [8] [59] Simplifying Alignment: From RLHF to Direct Preference Optimization (DPO)
https://huggingface.co/blog/ariG23498/rlhf-to-dpo

16 17 47 Gemini 3 Flash | Generative AI on Vertex AI | Google Cloud Documentation
https://docs.cloud.google.com/vertex-ai/generative-ai/docs/models/gemini/3-flash

18 19 20 21 22 53 55 56 57 Gemini 3 Developer Guide | Gemini API | Google AI for Developers
https://ai.google.dev/gemini-api/docs/gemini-3

23 38 39 41 48 Build with Gemini 3 Flash: frontier intelligence that scales with you
https://blog.google/technology/developers/build-with-gemini-3-flash/

24 25 30 31 32 40 49 50 Gemini 3: Introducing the latest Gemini AI model from Google
https://blog.google/products/gemini/gemini-3/

26 27 28 29 33 34 35 36 37 51 52 54 Gemini 3 Flash for Enterprises | Google Cloud Blog
https://cloud.google.com/blog/products/ai-machine-learning/gemini-3-flash-for-enterprises

42 Google releases Gemini 3 Flash: Ranks #3 on LMArena ... - Reddit
https://www.reddit.com/r/singularity/comments/1pp0ncw/google_releases_gemini_3_flash_ranks_3_on_lmarena/

44 Gemini 3 Pro Preview – Vertex AI - Google Cloud Console
https://console.cloud.google.com/vertex-ai/publishers/google/model-garden/gemini-3-pro-preview

45 Gemini 3 Pro Vision: The AI Upgrade That Changes Everything We ...
https://medium.com/activated-thinker/gemini-3-pro-vision-the-ai-upgrade-that-changes-everything-we-know-about-multimodal-intelligence-3a943ad63c94

46 Gemini 3 Pro: the frontier of vision AI - Google Blog
https://blog.google/technology/developers/gemini-3-pro-vision/

58 60 61 62 63 64 65 66 69 70 RLHF and DPO compared: user feedback methods for LLM optimization | by Carlo C. | . | Medium
https://medium.com/aimonks/rlhf-and-dpo-compared-user-feedback-methods-for-llm-optimization-44f4234ae689

71 Direct Preference Optimization (DPO): a lightweight counterpart to ...
https://toloka.ai/blog/direct-preference-optimization/