BÁO CÁO — PHƯƠNG PHÁP POST-TRAINING

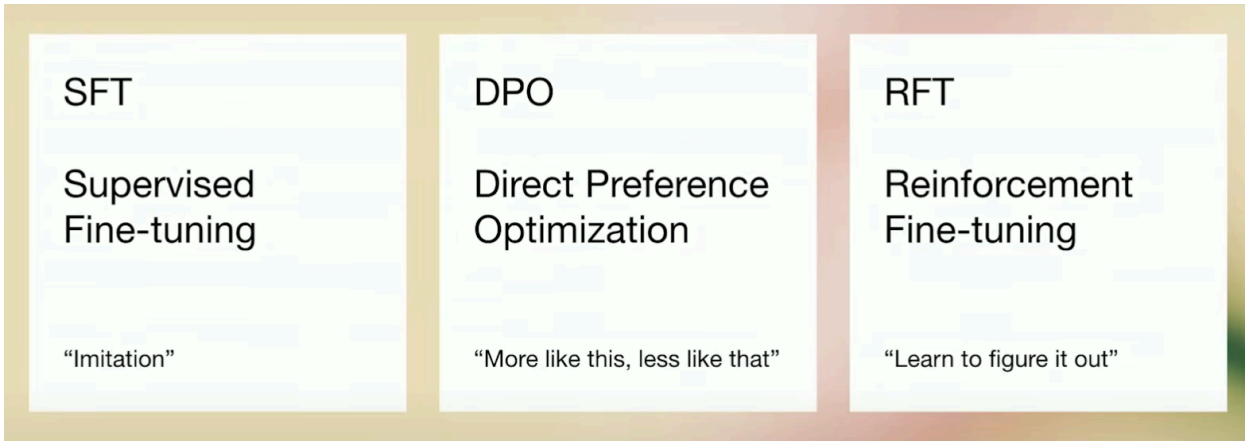
1. Tổng quan Post-training

Sau giai đoạn *pre-training* (mô hình học thống kê ngẫu nhiên từ kho dữ liệu cực lớn), **post-training** hay *alignment* là bước **đưa mô hình "về phía con người"**: bảo đảm trả lời đúng định dạng, hữu ích, an toàn, phù hợp với sở thích và giá trị của người dùng.

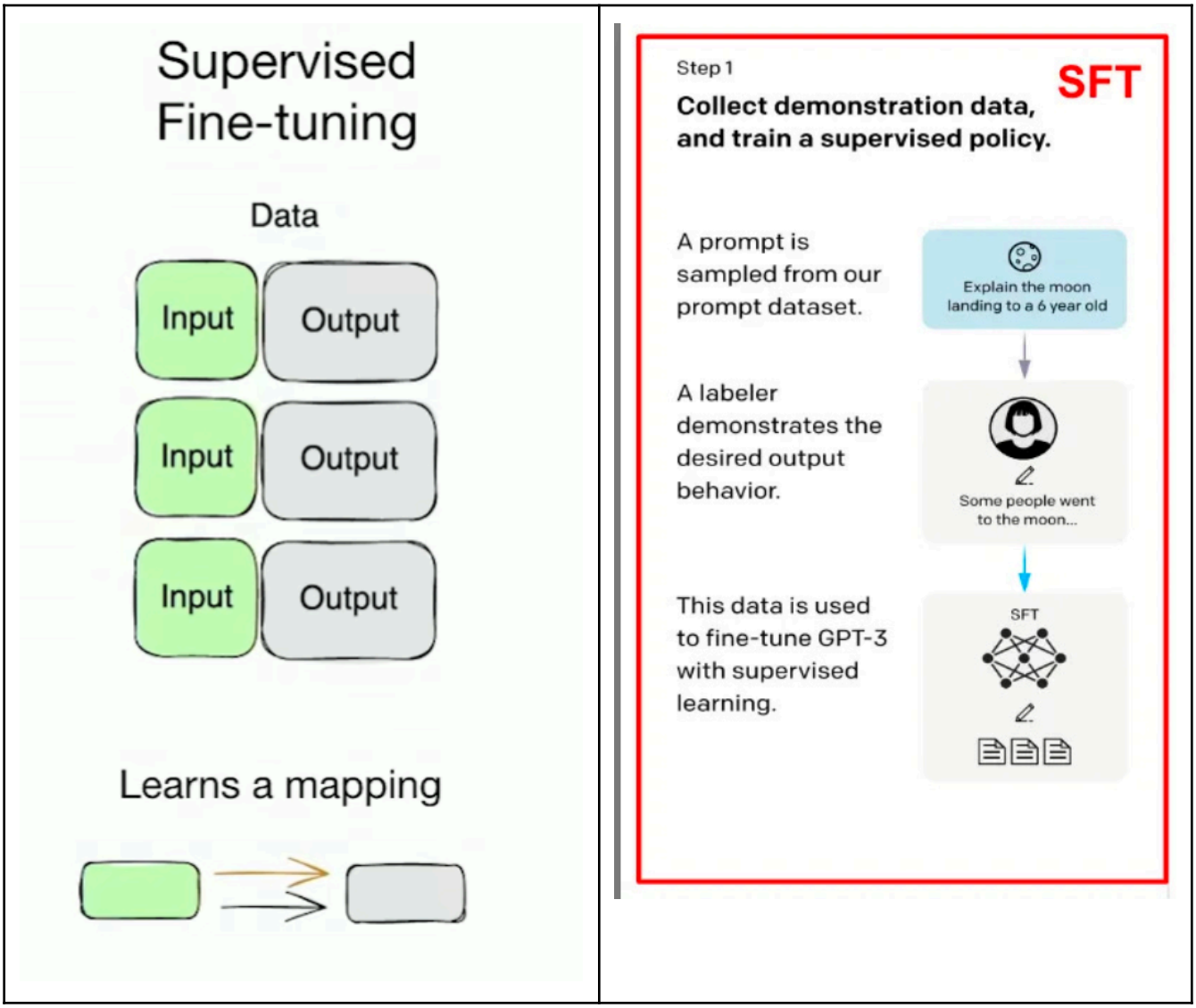
> **Mục tiêu chung**: Tinh chỉnh **chính sách sinh ngôn ngữ** để vừa tối đa hoá chất lượng (độ chính xác, tính logic) vừa tối thiểu rủi ro (nội dung độc hại, vi phạm chính sách).

Ba kỹ thuật phổ biến nhất hiện nay lần lượt xếp tầng như sau:

Thứ tự	Kỹ thuật	Vai trò chủ đạo
1	**SFT – Supervised Fine-Tuning**	Dạy mô hình *tuân thủ chỉ dẫn* (instruction)
2	**RLHF – Reinforcement Learning from Human Feedback**	Tối ưu **độ hài lòng** (preference) của người dùng
3	**DPO – Direct Preference Optimization**	Đơn giản hoá RLHF với hàm mất mát trực tiếp



2. Supervised Fine-Tuning (SFT)



2.1 Khái niệm & Mục đích

- **Khái niệm**: Fine-tune trên **cặp dữ liệu (prompt, response lý tưởng)** do chuyên gia biên soạn.
- **Mục đích**:

    - Buộc mô hình trả lời theo *định dạng chuẩn* (markdown, đoạn, gạch đầu dòng…).
    - Giảm “nói lạc đề”, tránh tiết lộ dữ liệu nhạy cảm.

2.2 Nguyên lý hoạt động

- **Loss**: Cross-entropy giữa chuỗi token sinh ra và nhãn vàng.
- **Regularization**: Weight decay, gradient clipping, dropout để giữ ổn định.
- **Kỹ thuật bổ trợ**:
    - **LoRA / QLoRA** cho phép fine-tune rẻ bằng *adapters*.
    - **R-Drop** giảm over-fit.

2.3 Quy trình triển khai

1. Thu thập & làm sạch tập chỉ dẫn (~10k – 100k cặp).
2. Chia train/valid; gán *conversation template*.
3. Fine-tune 1-3 epoch, batch size nhỏ, learning rate 1e^(-5) – 5^(e-5).
4. Đánh giá bằng *instruction test set*.

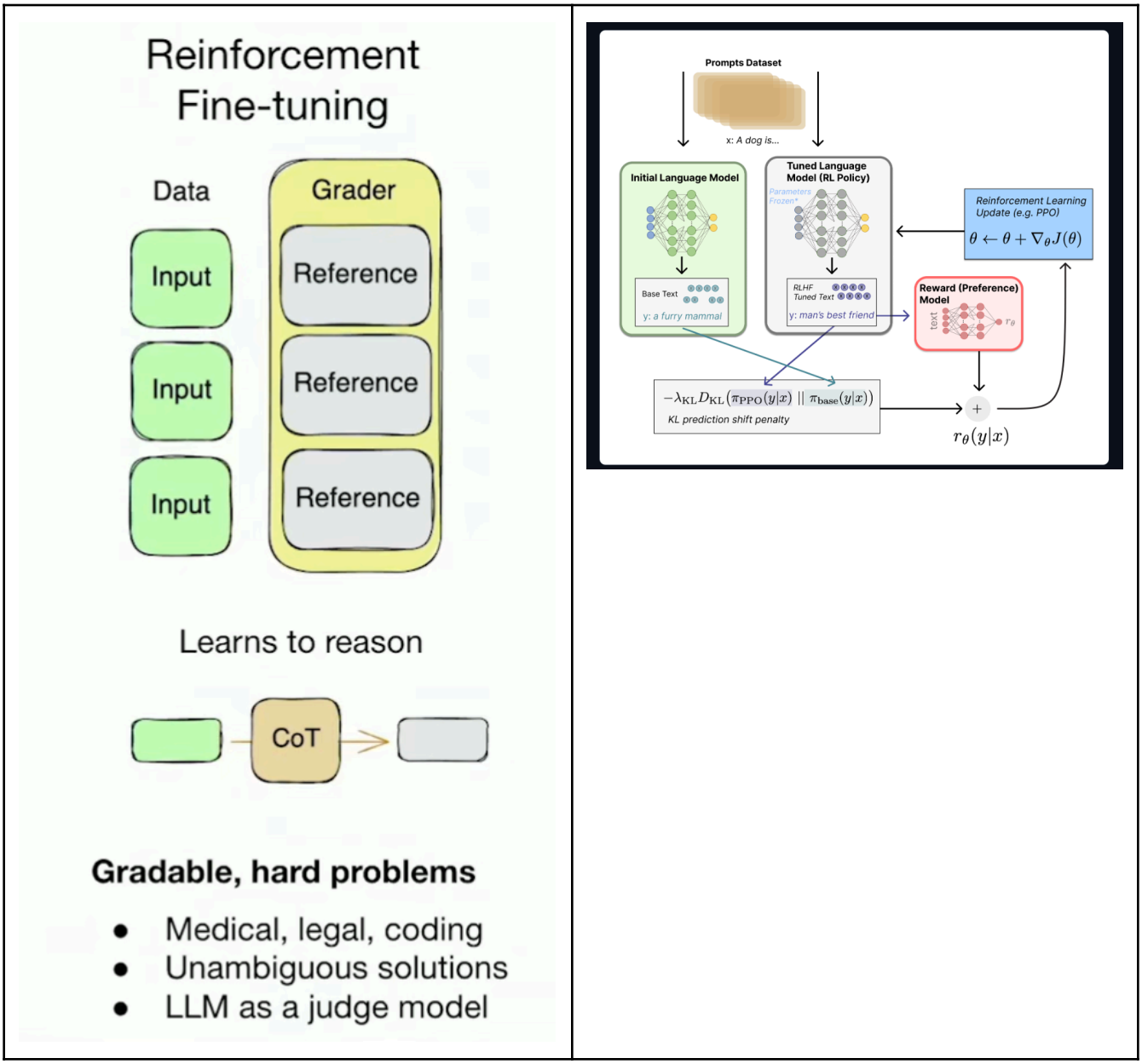
2.4 Ứng dụng tiêu biểu

- Chatbot CSKH, trợ lý viết email, tóm tắt cuộc họp, dịch máy đa ngôn ngữ.

2.5 Hạn chế & Cải tiến

Vấn đề	Cải tiến phổ biến
Exposure bias (mô hình chỉ nhìn reference)	**Data augmentation** (*Self-Instruct*, **Alpaca**), *Re-ranking* sample quality
Bao phủ hướng dẫn hạn chế	Kết hợp **crawled instructions**, dùng **LLM tự sinh prompt**
Chi phí GPU	**LoRA**, **PEFT**, **bits-and-bytes** quan trọng hoá thấp

3. Reinforcement Learning from Human Feedback (RLHF)



3.1 Khái niệm & Mục đích

- **Khái niệm**: Sử dụng **phản hồi ưu-khuyết** từ con người ⇒ huấn luyện **Reward Model (RM)** ⇒ dùng RL (thường **PPO**) để tối đa hoá điểm thưởng.

- **Mục đích**:
    - Khiến mô hình **hữu ích, lễ độ, an toàn**.
    - Điều chỉnh phong cách: ngắn gọn, hài hước, trang trọng…

3.2 Nguyên lý hoạt động

Thành phần	Vai trò
Prompt	Trạng thái *s* trong RL
Output token	Hành động *a*
Reward	Điểm từ RM dự đoán (preference score)
Chính sách tham chiếu	Checkpoint SFT – giữ khoảng cách KL nhỏ
Thuật toán	**Proximal Policy Optimization** (PPO) thường dùng β để phạt lệch

3.3 Quy trình triển khai

1. **Dataset Pairs**: Thu thập ~50k cặp <answer_good, answer_bad>.
2. **Train RM**: MLP + LLM embeddings, fine-tune vài epoch.
3. **PPO Loop**:
    - Rollout n mẫu/thế hệ.
    - Tính reward & advantage.
    - Cập nhật policy, ràng buộc KL.
4. Kiểm thử *safety eval* (toxicity, bias, jailbreak).

3.4 Ứng dụng tiêu biểu

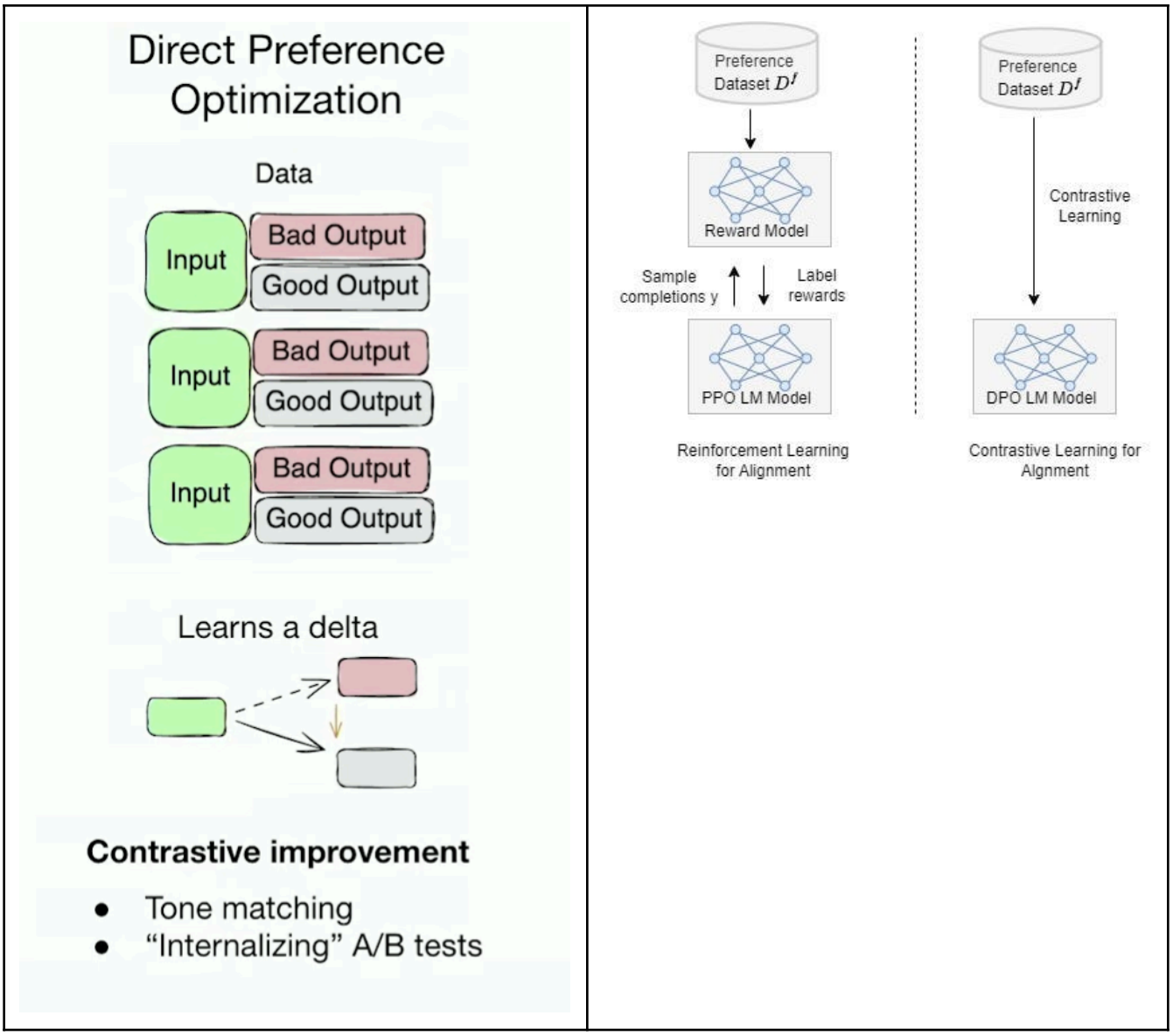
- ChatGPT, Claude, Copilot, trợ lý tư vấn y khoa, ghép đôi giáo dục.

3.5 Hạn chế & Cải tiến

Vấn đề	Kỹ thuật cải tiến
Tốn công gán nhãn con người	**RLAIF** (AI Feedback), **Constitutional AI** (luật đạo đức)
Unstable training (collapse)	**Adaptive KL Penalty**, **Reward normalization**, **P3O**
Bias của RM	**Active Sampling** – ưu tiên prompt gây tranh cãi

---

4. Direct Preference Optimization (DPO)



4.1 Khái niệm & Mục đích

- **Khái niệm**: Bỏ vòng RL; tối ưu hàm mất mát **log-sigmoid** trực tiếp trên cặp (good, bad) với ràng buộc KL (β).
- **Mục đích**:
    - Giảm 2-3× GPU / thời gian so với RLHF.
    - Giữ chất lượng cạnh tranh.

4.2 Nguyên lý hoạt động

$$\mathcal{L}(\theta) = -\log \sigma\Big(\beta\,[\log p_\theta(y^+|x) - \log p_\theta(y^-|x)]\Big)$$

- Cập nhật gradient **giống SFT**, dễ song song & tích hợp LoRA.

4.3 Quy trình triển khai

1. Sử dụng **cùng tập cặp ưu-khuyết** như RLHF.
2. Không cần train RM; trực tiếp fine-tune vài epoch trên loss DPO.
3. Đánh giá: HELM, MT-Bench, Alpaca-Eval.

4.4 Ứng dụng tiêu biểu

- Model mã nguồn mở: **Zephyr-β-7B-DPO, Open-Hermes-2.5-DPO, Qwen-1.8B-DPO**.

4.5 Hạn chế & Cải tiến

Vấn đề	Biến thể / Giải pháp
Chưa tận dụng trajectory dài	**IPO, KTO** thêm term nhiệt độ κ
Cần nhiều cặp ưu-khuyết	**Rank-debiased DPO**, **Sparsity-aware loss**
Độ kiểm soát style hạn chế	Kết hợp **DPO + RLHF** (2-stage)

---

5. So sánh & Lộ trình tiếp nối

Thuộc tính	SFT	RLHF	DPO
Phụ thuộc gán nhãn con người	Thấp (chỉ cần prompt-response)	Rất cao (so sánh cặp)	Cao (như RLHF)

Độ phức tạp triển khai	Dễ	Khó (RM + PPO)	Trung bình
Chi phí GPU	Thấp	Cao	Thấp-Trung
Kiểm soát an toàn	Trung bình	Cao	Cao-trung
Ổn định huấn luyện	Cao	Dễ *collapse*	Cao

Xu hướng 2025-2026:

- **AI Feedback + Constitutional AI** ⇒ giảm chi phí human label.
- **Retrieval-Augmented RLHF/DPO** ⇒ mô hình "thưởng" dựa knowledge hiện thời.
- **Multi-agent RL** ⇒ LLM tự chia nhiệm vụ & đánh giá lẫn nhau.
