



迷途之声：从PPO的喧嚣到GRPO的谐音

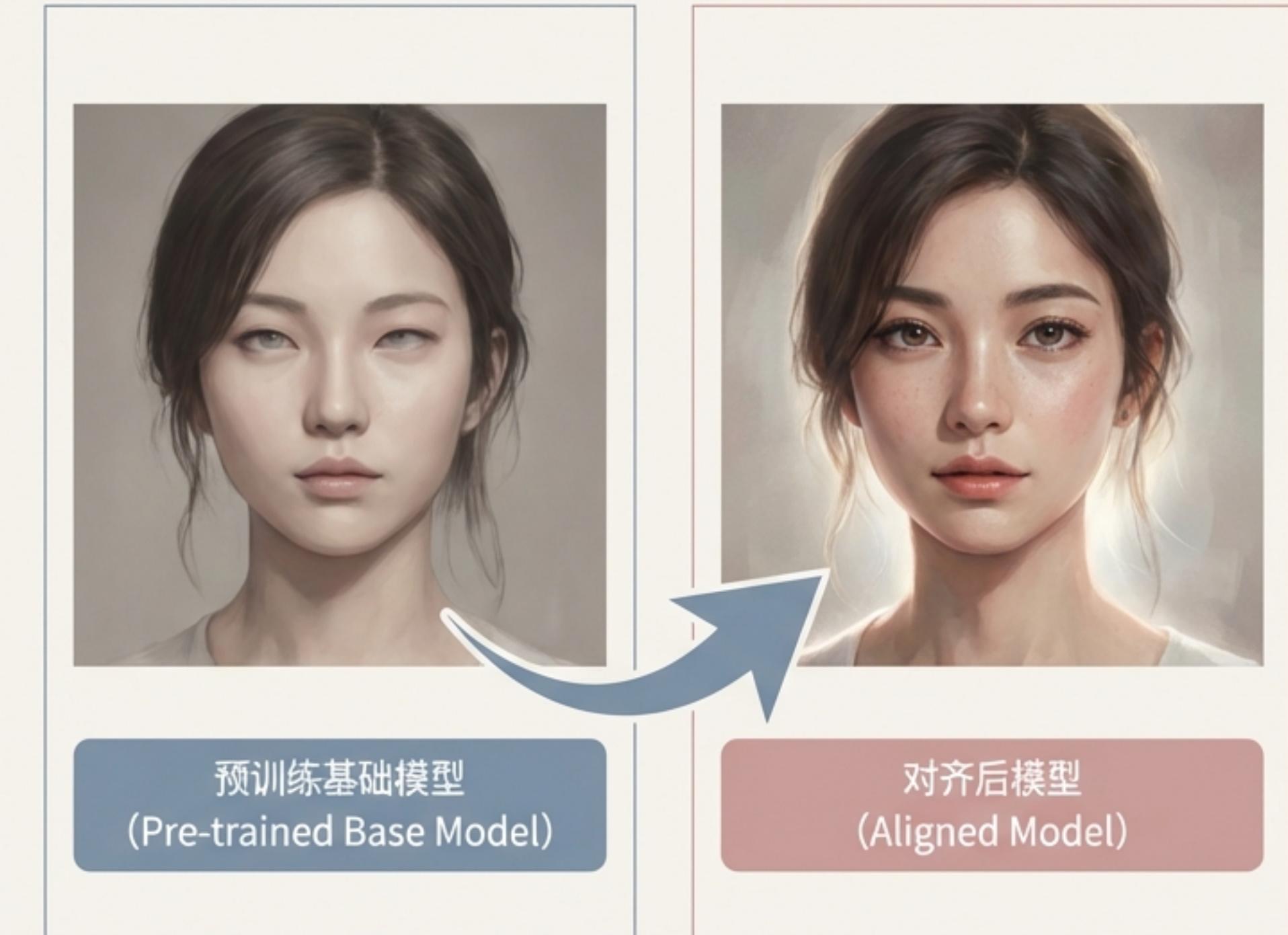
一位文生图工程师的对齐算法探索之旅

- 全面解析从PPO到GRPO的演进脉络
- 全面解析从PPO到GRPO的演进脉络
- 实战指南：将GRPO应用于Flow Matching文生图模型

我们为何需要对齐？原始模型的“无意识创作”

强大的预训练模型如同才华横溢但不受约束的音乐家。它们能生成技术上完美但缺乏“灵魂”的作品。对齐（Alignment）的目标，就是教会模型理解并表达我们心中真正的“艺术追求”——从美学、风格到特定指令的微妙之处。

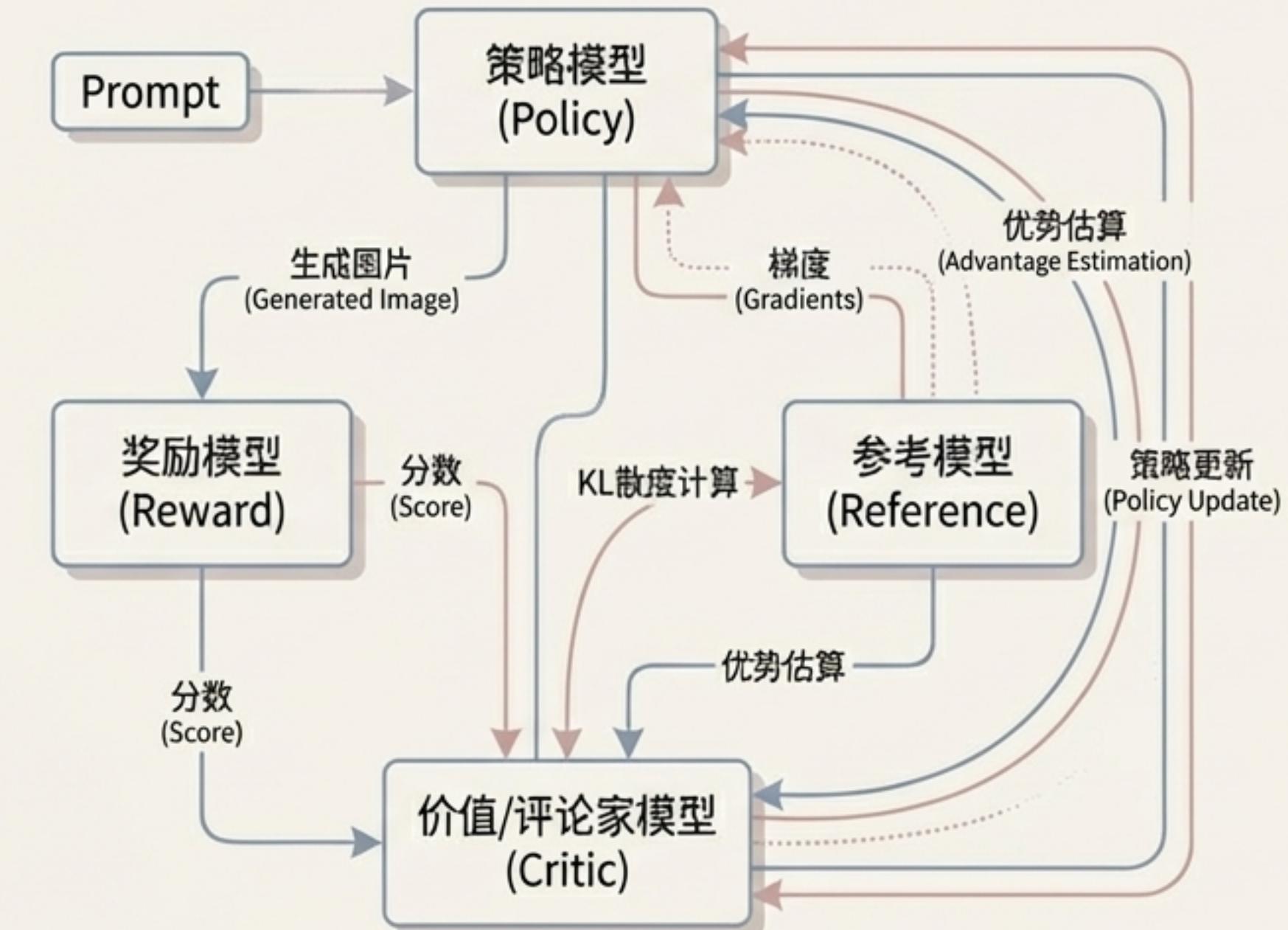
- **问题1：指令遵循不精确（Imprecise Instruction Following）：**模型可能忽略或误解Prompt中的复杂或否定性描述。
- **问题2：美学质量不稳定（Inconsistent Aesthetic Quality）：**默认生成结果的美学水平参差不齐。
- **问题3：风格控制困难（Difficulty in Style Control）：**难以精确复现或融合特定的艺术风格。



首次尝试的复杂乐章：强化学习与PPO

Proximal Policy Optimization (PPO) 是从强化学学习 (RL) 领域引入的经典对齐方法。它通过一个外部的“评论家”（奖励模型）来为生成模型的“作品”打分，并据此优化模型。

- **策略模型 (Policy Model)**：我们要优化的生成模型本身。
- **奖励模型 (Reward Model)**：独立训练，用于评估生成图片与Prompt的匹配度及美学质量。
- **参考模型 (Reference Model)**：初始模型的副本，用于计算KL散度，防止策略“跑偏”太远。
- **价值/评论家模型 (Value/Critic Model)**：预测在当前策略下未来奖励的期望值，用于稳定训练。



PPO的“不和谐音”：为何实践如此痛苦？

尽管PPO在理论上可行，但在大规模文生图模型的实践中，其训练过程如同一次混乱的乐队排练，充满挑战。

不和谐音#1：训练流程复杂 (Complex Pipeline)

需要维护和协调四个独立模型，
系统搭建和调试成本极高。

不和谐音#2：超参数敏感 (Hyperparameter Sensitivity)

训练过程对学习率、KL惩罚因子等超参数极其敏感，难以调试，容易崩溃。



不和谐音#3：奖励模型依赖 (Reward Model Dependency)

对齐效果的上限被奖励模型的准确性牢牢“卡住”。一个有偏差的奖励模型会“教坏”生成模型。

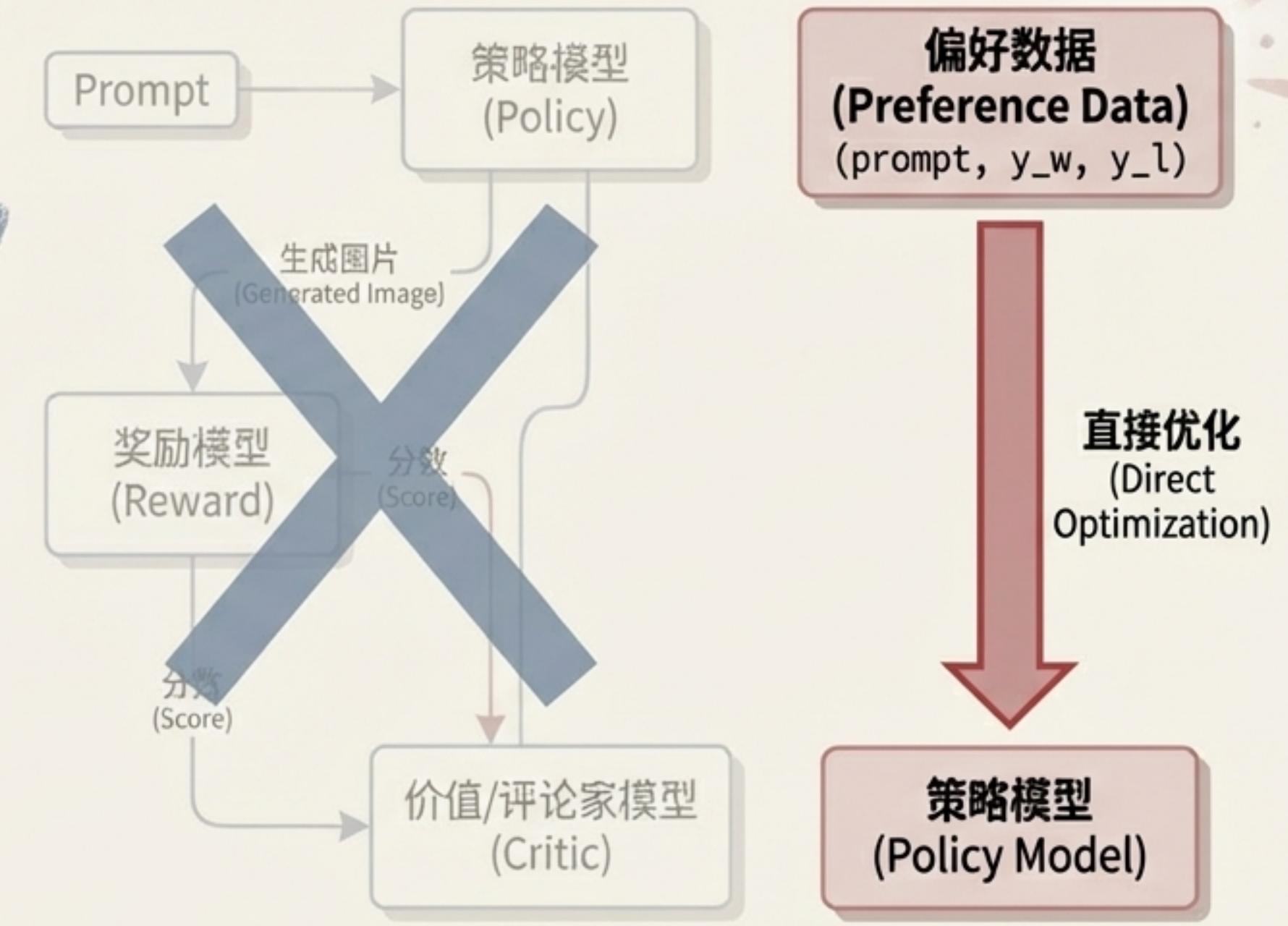
拨云见日：DPO如何用偏好数据直接谱曲

The Breakthrough: Direct Preference Optimization

(DPO) 发现，我们不需要先拟合一个奖励模型，再用它来优化策略。我们可以直接从“图片A比图片B好”这类成对的偏好数据中，推导出最优策略的更新方向。

How it Works

- 数据形式 (Data Format)** : (prompt, y_w , y_l) – 即对于同一prompt，人类更偏好的“获胜”图片 y_w 和“落败”图片 y_l 。
- 核心思想 (Core Idea)** : DPO通过一个简单的分类损失函数，直接最大化模型生成 y_w 的概率，同时最小化生成 y_l 的概率。
- 巨大优势 (The Big Advantage)** : 告别了独立的奖励模型训练和复杂的RL流程，将对齐问题转化为一个稳定、高效的监督学习问题。



华彩乐章：GRPO，融合直接优化与在线探索

GRPO (Generative Rejection Policy Optimization) 是一种更进一步的对齐算法，它将DPO的直接优化思想与在线数据生成和选择相结合。

核心机制：“即时创作，择优学习” (Core Mechanism: "Improvise, Select, and Learn")

它不再完全依赖于一个静态的偏好数据集。

在训练过程中，模型为每个Prompt即时生成 K 个候选样本。

使用一个奖励函数（可以是美学打分器等轻量级模型）在线选出“最佳”的那个样本。

这个被选中的“最佳”样本构成了动态的、高质量的训练目标。

这种方法让模型能够探索自身生成空间中的更优解，而不是仅仅模仿静态数据集中的偏好，从而达到更高的对齐天花板。



GRPO工作流：解构一首杰作的诞生过程



输入一个Prompt x 。当前策略模型 π_θ 生成 K 个候选图片 $\{y_1, y_2, \dots, y_K\}$ 。



一个奖励函数 $r(x, y)$ (如美学评分模型HPSv2) 为所有 K 个候选图片打分。选出得分最高的图片作为“获胜者” y_w 。



这个 y_w 隐式地定义了我们希望模型学习的目标分布。我们希望模型未来在遇到Prompt x 时，能更高概率地生成像 y_w 这样的图片。



通过一个类似于DPO的损失函数，更新模型参数 θ ，使其提高生成 y_w 的概率。



GRPO的“乐谱”：损失函数详解

$$\mathcal{L}_{\text{GRPO}}(\theta; x, \{y_i\}_{i=1}^K) = -\mathbb{E}_{y_w \sim p_w(y|x)} [\log \sigma(\beta (\log \pi_\theta(y_w|x) - \log \pi_{\text{ref}}(y_w|x)))]$$

其中 p_w 是由奖励函数排序决定的获胜样本分布。

$\log \pi_\theta(y_w|x)$ (**策略模型概率**)

解读：这是我们的模型生成“获胜”图片 y_w 的对数概率。

目标：我们希望最大化这一项，让模型更“喜欢”生成这张最好的图片。

β (**温度系数**)

解读：控制奖励信号强度的超参数。可以看作是“指挥的力度”，决定了模型学习的强度。

$\log \pi_{\text{ref}}(y_w|x)$ (**参考模型概率**)

解读：这是初始的、未经对齐的模型生成 y_w 的概率。

作用：作为基准线，衡量策略模型相对初始模型有多大的提升。

$\log \sigma(\dots)$ (**整体结构**)

解读：整个结构与DPO类似，将问题转化为一个对隐式奖励的逻辑回归，确保训练的稳定性。

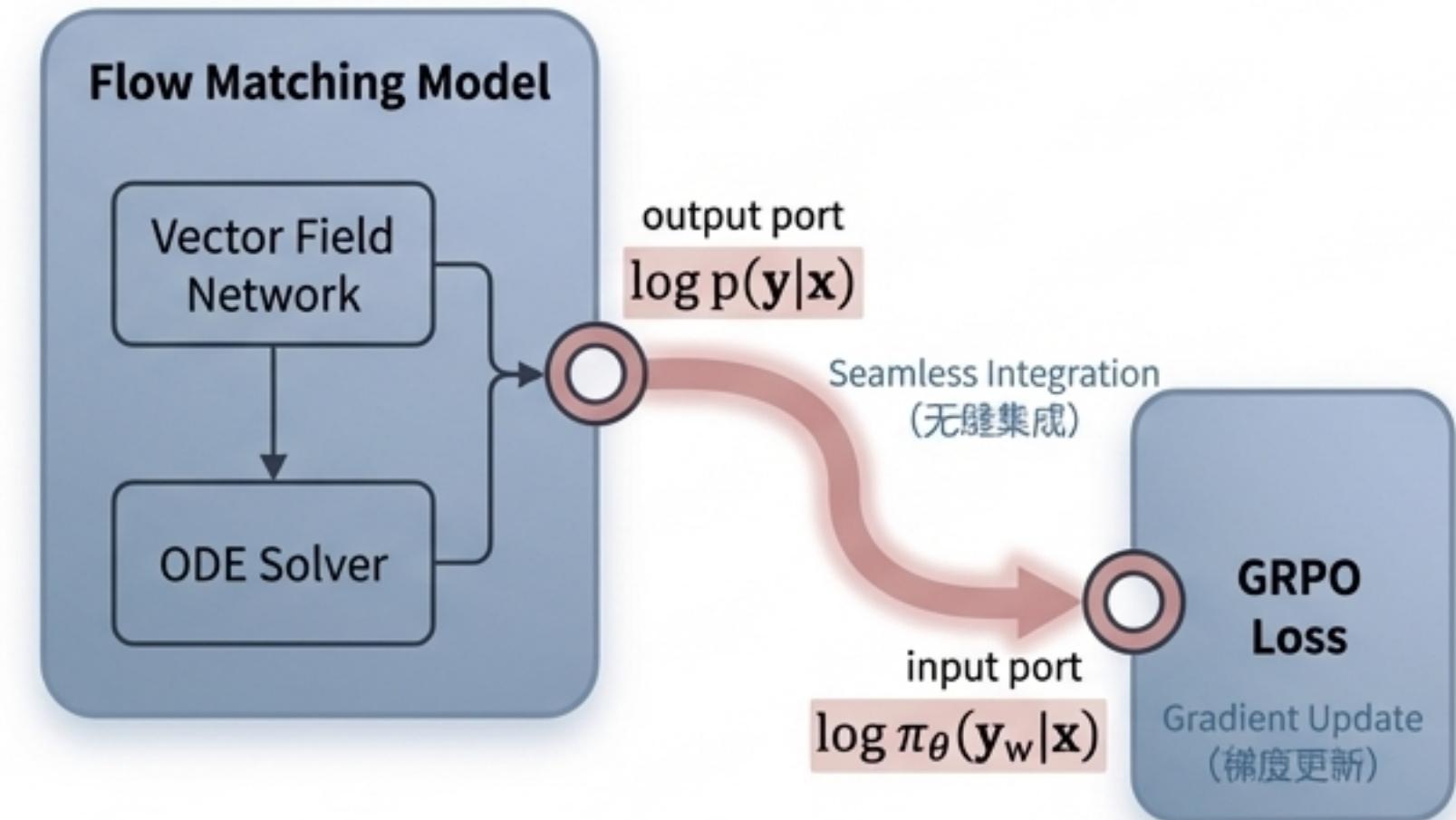
最终对决：PPO vs. DPO vs. GRPO

维度	PPO	DPO	GRPO
架构复杂度 (Arch. Complexity)	极高（4个模型）	低（无需奖励模型）	中（需在线奖励函数）
训练稳定性 (Training Stability)	低（对超参数敏感）	高（监督学习）	高（类似DPO）
数据依赖 (Data Dependency)	依赖高质量奖励模型	依赖大规模静态偏好数据	可利用在线生成数据，更灵活
对齐上限 (Alignment Ceiling)	受限于奖励模型质量	受限于静态数据集质量	更高（通过在线探索发现更优解）
核心隐喻 (Core Metaphor)	混乱的排练	精准的乐谱复刻	富有创造力的即兴演奏

GRPO在保持DPO稳定性的同时，通过在线探索机制突破了静态数据的局限，是当前兼具性能、稳定性与灵活性的前沿选择。

实战演练：将GRPO的和谐旋律融入Flow Matching

- **Core Idea:** GRPO的优化框架是模型无关的，其核心在于计算策略模型 π_θ 对给定样本 y 的对数概率 $\log \pi_\theta(y|x)$ 。我们可以将其无缝对接到Flow Matching模型的训练流程中。
- **Why it Fits**
 - **Flow Matching (FM) 模型:** 通过学习一个向量场来定义从噪声到数据的连续轨迹。
 - **概率计算:** 我们可以利用FM模型的ODE（常微分方程）和概率流（Probability Flow）公式，精确计算出任何给定图片 y 在当前模型下的对数似然 $\log p(y|x)$ 。
 - **无缝集成:** 这个计算出的 $\log p(y|x)$ 就是GRPO损失函数中需要的 $\log \pi_\theta(y|x)$ 。因此，GRPO的更新梯度可以直接应用于训练FM模型的向量场网络。



部署GRPO的调音台：三步修改训练循环

// 步骤一：定义在线奖励函数 (Step 1: Define Online Reward Function)

选择一个高效的打分器，例如`Aesthetic Scorer`或`HPSv2`。它无需完美，只需能在`K`个样本中大致挑出最好的即可。

```
reward_fn = HPSv2_Scorer()
```

// 步骤二：生成与选择 (Step 2: Generate & Select)

在每个训练step中，为batch中的每个prompt生成`K`个候选图片。

```
candidates = model.generate(prompt, num_samples=K)
scores = reward_fn(prompt, candidates)
winner_image = candidates[argmax(scores)]
```

// 步骤三：计算损失并更新 (Step 3: Calculate Loss & Update)

[关键] 使用Flow Matching模型的概率流ODE计算`winner_image`的对数概率`log_prob_winner`。将`log_prob_winner`代入GRPO损失函数。

```
log_prob_winner = model.calculate_log_prob(winner_image, prompt) # [关键]
log_prob_ref = ref_model.calculate_log_prob(winner_image, prompt)
loss = calculate_grpo_loss(log_prob_winner, log_prob_ref, beta)
```

```
loss.backward()
optimizer.step()
```

成功的和声： GRPO对齐后的效果展示

"A majestic cat wearing a tiny crown, sitting on a velvet cushion,
photorealistic, cinematic lighting"

左侧：对齐前 (Before GRPO)



****评价 (Critique) ****

- 基本元素齐全
- 缺乏“史诗感”和真实感
- 光影平淡

右侧：GRPO对齐后 (After GRPO)



****评价 (Critique) ****

- ✓ 美学质量显著提升
- ✓ 更精准地捕捉了“cinematic”的氛围
- ✓ 细节丰富，质感真实

从喧嚣到和谐：我们学到了什么？

- 我们从PPO的**复杂性与不稳定性**出发，它代表了用传统RL硬解对齐问题的早期尝试。
- 我们见证了DPO的**简洁革命**，它证明了可以直接从偏好中学习，无需复杂的奖励建模。
- 我们最终抵达了**GRPO的在线探索范式**，它结合了DPO的稳定性与在线生成的灵活性，实现了更高的性能上限。



关键启示：模型对齐的未来在于更**直接、稳定、数据高效**的方法。GRPO不是一个孤立的算法，而是一种设计理念的体现：让模型在探索中自我完善，而不是被动地模仿。

我们的歌，才刚刚开始

算法的演进永无止境。PPO、DPO、GRPO... 它们不是终点，而是我们与模型共同寻找更美“旋律”的阶梯。掌握这些工具，意味着我们不再仅仅是Prompt的输入者，更是模型的“指挥家”和“制作人”，能够引导它们创造出前所未有的、真正属于我们时代的“音乐”。

