南京航空航天大學
NANJING UNIVERSITY OF AERONAUTICS AND ASTRONAUTICS

大语言模型参数高效微调与偏好优化

时间： 2026.11.24

汇报人：张文良

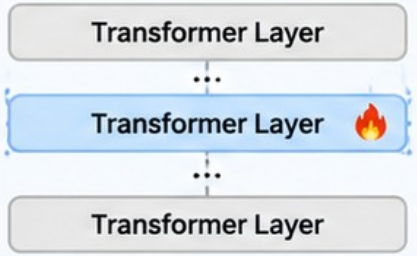
参数高效微调

南京航空航天大學 NANJING UNIVERSITY OF AERONAUTICS AND ASTRONAUTICS

参数高效微调（PEFT）

选择性微调 (Selective)

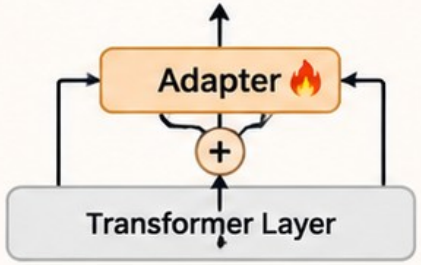
仅更新模型中的部分参数



- BitFit
- LayerNorm Tuning

附加式微调 (Additive)

冻结原模型，添加轻量可训练模块



- Adapter
- Parallel Adapter

提示式微调 (Prompt-based)

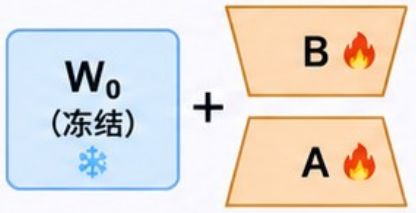
学习可训练的提示向量（软提示）



- Prompt Tuning
- Prefix Tuning
- P-Tuning / P-Tuning v2

重参数化微调 (Reparameterization)

通过低秩等结构重新参数化权重更新



- LoRA
- AdaLoRA
- QLoRA

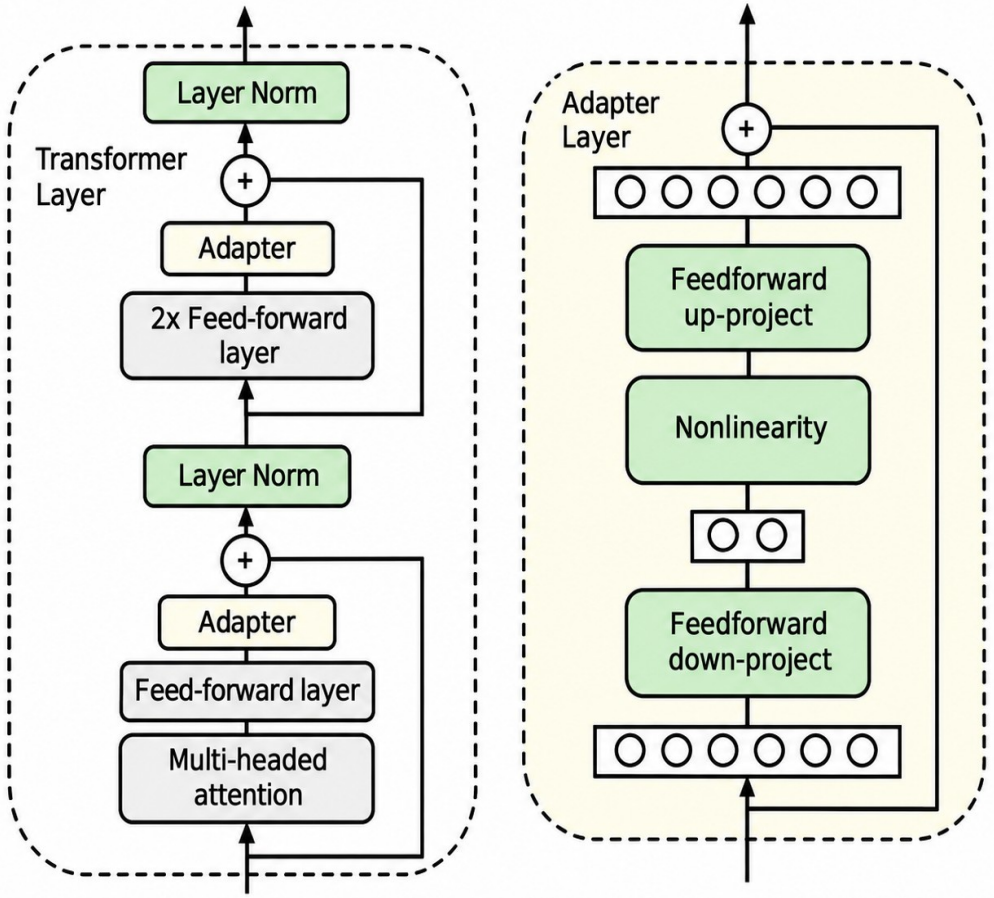
混合式微调 (Hybrid)

结合多种方法，按任务需求进行组合优化



- Adapter + Prompt
- UniPELT 等

Adapter Tuning：在 Transformer 内插入瓶颈模块



Adapter Tuning 是一种高效的参数微调方法：在冻结预训练模型大部分参数的基础上，在 Transformer 中插入小型可训练的 Adapter，通过降维–非线性–升维的瓶颈结构进行微调，从而在低成本下适配下游任务。

核心思想

冻结预训练模型参数，仅训练插入的 Adapter，利用少量可训练参数高效适配下游任务。

Adapter 结构

$$y = x + W_{\text{up}}\sigma(W_{\text{down}}x)$$

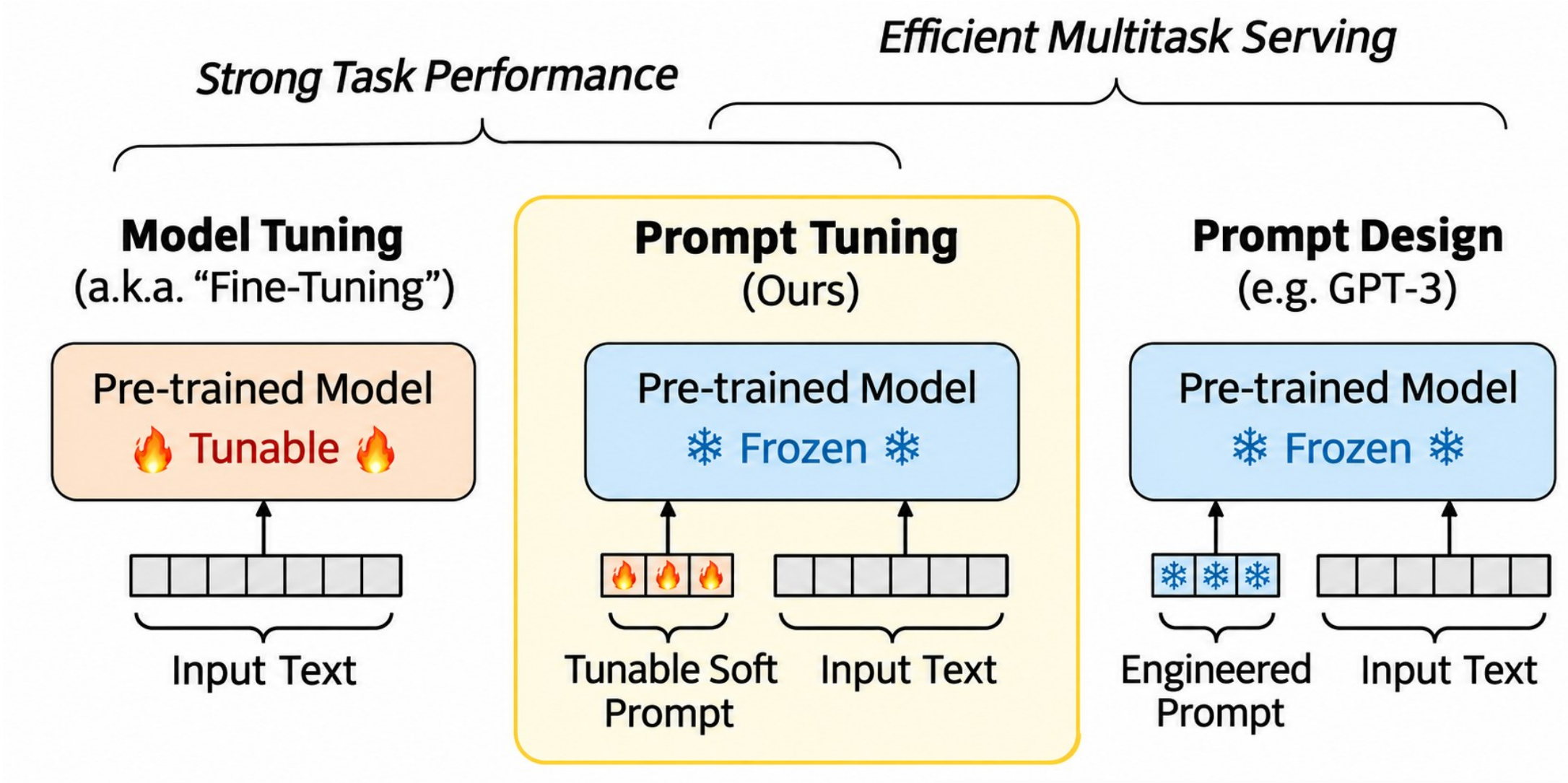
- W_{down}：降维（$d \to r$）
- σ：非线性激活（如 ReLU）
- W_{up}：升维（$r \to d$）

优势

- **参数高效**：仅新增少量参数
- **保留知识**：冻结原模型能力
- **灵活复用**：可组合多个 Adapter（如 AdapterFusion）

$$y = x + W_{\text{up}}\sigma(W_{\text{down}}x)$$

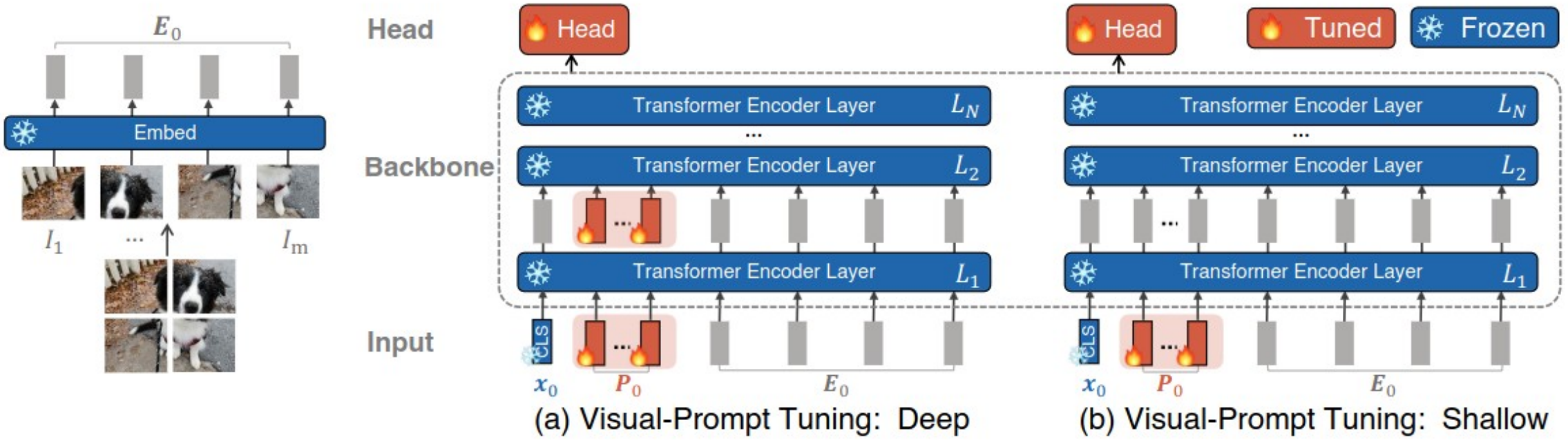
Prompt Tuning：冻结预训练模型，学习输入端软提示



在文本 Token Embedding 前拼接可学习的 Soft Prompt 向量，冻结预训练模型，仅更新 Soft Prompt

$$P^* = \arg\min_P \mathcal{L}_{\text{task}}\left(f_\theta([P; E(x)]), y\right)$$

Visual Prompt Tuning： 将可学习提示扩展到视觉模型



VPT-Shallow： 仅在输入层添加可训练提示，之后提示随特征一起经过后续 Transformer 层。

VPT-Deep： 在多个 Transformer 层分别引入可训练提示，而不是只在输入层加入一次。

$$Z_0 = [\mathrm{CLS}; P_v; E_{\mathrm{patch}}(I)]$$

Prefix Tuning： 为多层注意力加入前缀

南京航空航天大學 NANJING UNIVERSITY OF AERONAUTICS AND ASTRONAUTICS

前缀的生成方式（Reparameterization）

使用可学习的前缀向量经小型 MLP，生成各层的前缀 K/V。

$P' \in \mathbb{R}^{m \times d_p}$（可学习前缀向量） → MLP g_θ → $P \in \mathbb{R}^{L \times 2 \times m \times d}$（各层前缀 K/V）

第 l 层 Transformer Attention

$P_{K,l} \in \mathbb{R}^{m \times d}$ ⊕ $K_l \in \mathbb{R}^{n \times d}$ → $K'_l \in \mathbb{R}^{(m+n) \times d}$

$P_{V,l} \in \mathbb{R}^{m \times d}$ ⊕ $V_l \in \mathbb{R}^{n \times d}$ → $V'_l \in \mathbb{R}^{(m+n) \times d}$

关键公式（每层）

$$K'_l = [P_{K,l}; K_l], \quad V'_l = [P_{V,l}; V_l]$$

$$\text{Attn}_l(Q_l, K'_l, V'_l) = \text{softmax}\left(\frac{Q_l K'^T_l}{\sqrt{d}}\right) V'_l$$

核心思想

- 在每层注意力中加入可学习的前缀 K/V。
- 仅训练前缀参数（P' 和 g_θ），冻结预训练模型。

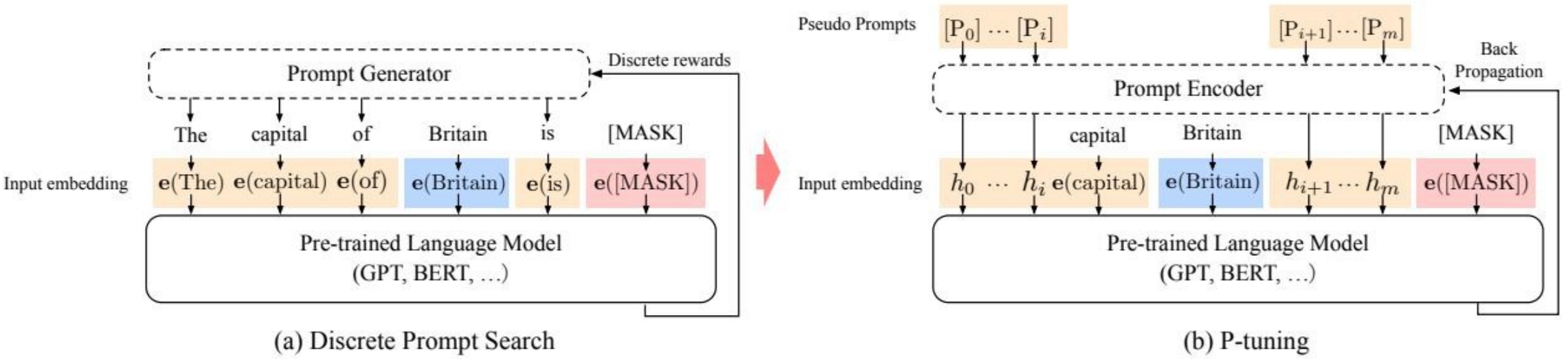
训练目标（文本生成）

$$\mathcal{L} = -\sum_{t=1}^{T} \log p_\theta(y_t | y_{<t}, x; P')$$

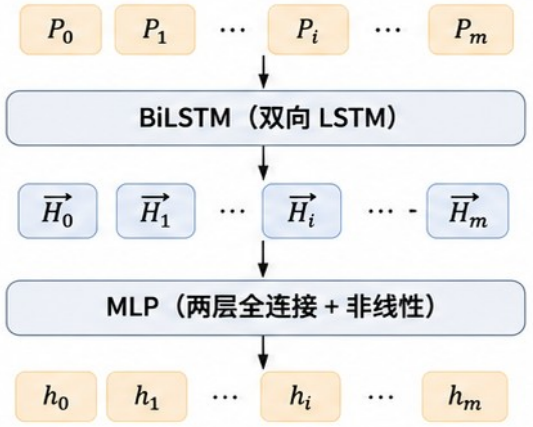
前缀参数 P（可学习） → 多层 Attention（含前缀 K/V） → 生成输出 & 计算 Loss → 更新前缀参数

Prefix Tuning： 为多个 Transformer 层提供额外的 Attention 前缀，通常表现为每层的前缀 K/V 。

P-Tuning： 将离散提示改为可学习的连续提示



Prompt Encoder：生成连续提示向量



插入方式：在输入的 Embedding 序列中灵活插入

- **前缀式（Prefix）**

$h_0 \cdots h_m$ $e(x_1) \cdots e(x_n)$ e([MASK])

- **中间插入（Infix）**

$e(x_1) \cdots e(x_k)$ $h_0 \cdots h_m$ $e(x_{k+1}) \cdots e(x_n)$ e([MASK])

- **多段插入（Multi-span）**

$h_0 \cdots h_i$ $e(x_1) \cdots e(x_k)$ $h_{i+1} \cdots h_m$ e([MASK])

关键公式

- **提示编码：**

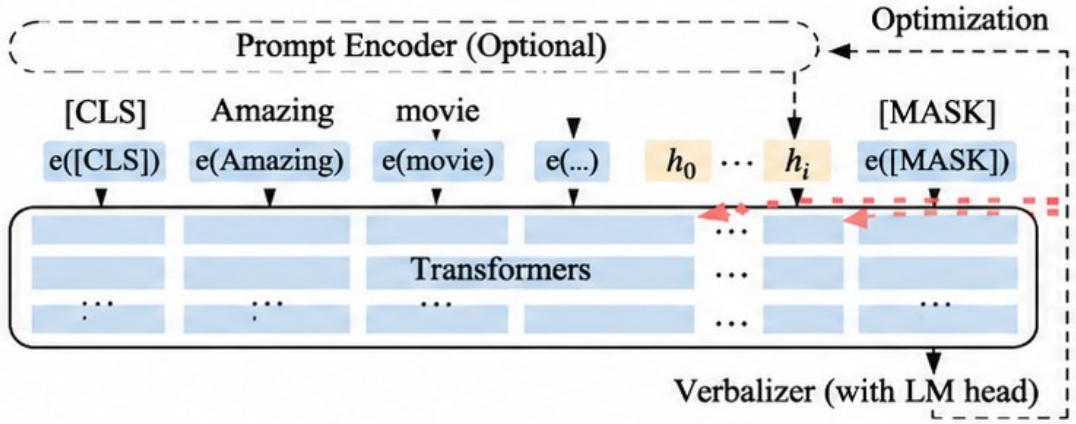
$$\begin{cases} \vec{H} = \mathrm{BiLSTM}(P) \\ h_i = \mathrm{MLP}(\vec{H}_i) \end{cases}$$

- **输入序列：** $Z = [h_{s1}, \ldots, h_{sm}, e(x_1), \ldots, e(x_n)]$
- **训练目标（以 MLM 为例）：** $\mathcal{L} = -\log p_\phi(y \mid Z)$

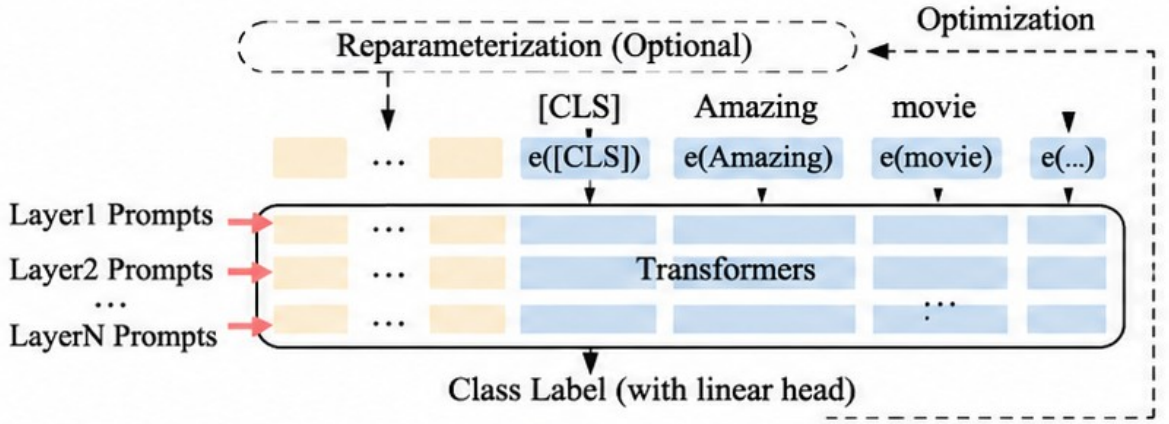
要点

- BiLSTM：建模不同提示位置的依赖关系（双向上下文）。
- MLP：非线性映射，将编码表示转换为模型输入的提示向量。
- 与 Prompt Tuning 的区别：引入 Prompt Encoder，并支持在输入中灵活插入多段连续提示向量。

P-Tuning v2：从浅层提示到深层提示



(a) Lester et al. & P-tuning (Frozen, 10-billion-scale, simple tasks)



(b) P-tuning v2 (Frozen, most scales, most tasks)

核心思想

- 将可学习提示从输入层扩展到 Transformer 的多层（深层提示），增强对模型的控制能力。

机制流程

① 为每一层学习一组前缀；
② 在该层 Self-Attention 的 K、V 前拼接；
③ 冻结预训练模型，优化提示参数（及任务头）。

关键公式

- 第 l 层注意力：

$$\text{Attention}(Q_l, K_l', V_l') = \text{softmax}\left(\frac{Q_l K_l'^{\top}}{\sqrt{d_k}}\right) V_l'$$

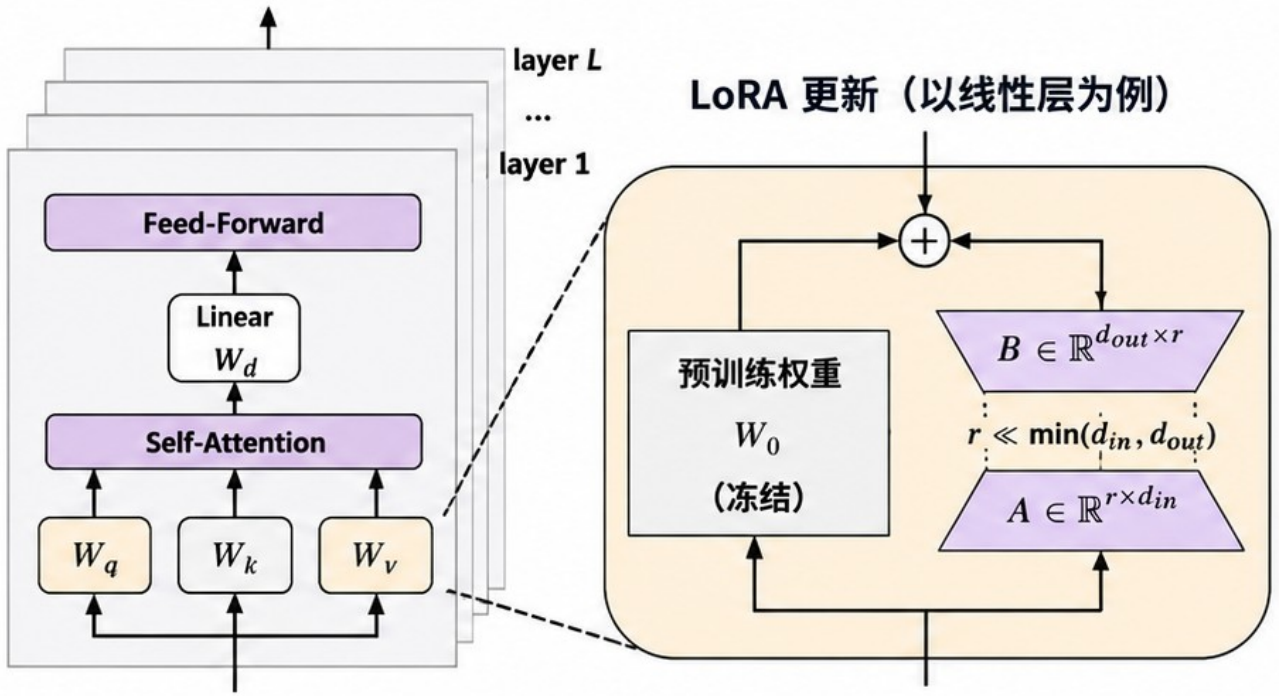
- 前缀的 K/V 拼接：

$$K_l' = [P_{K,l}; K_l], \qquad V_l' = [P_{V,l}; V_l]$$

与 Prefix Tuning 的关系

- **相同点**：均为多层 K/V 前缀的提示微调机制，训练时冻结主模型。
- **主要区别**：P-Tuning v2 聚焦 NLU 任务（分类、序列标注等），支持使用线性任务头；Prompt Encoder 可选。

LoRA：冻结基础权重，只训练低秩矩阵



核心思想

在冻结预训练权重 W_0 的基础上，
用两个**低秩矩阵** A、B 对权重增量 ΔW 进行参数化，实现高效微调。

核心公式

$$W = W_0 + \Delta W \ , \quad \Delta W = \frac{\alpha}{r} BA$$

- W_0：预训练权重（冻结）； $A \in \mathbb{R}^{r \times d_{in}}$， $B \in \mathbb{R}^{d_{out} \times r}$
- r：低秩秩（$r \ll d_{in}, d_{out}$）； α：缩放系数

训练与推理

- **训练：** 仅更新 A、B（新增参数量：$r(d_{in} + d_{out})$）。
- **推理：** 将 $\Delta W = \frac{\alpha}{r} BA$ 与 W_0 合并，
  无需增加推理开销。

主要优势

- ✓ **参数量小：** $r(d_{in} + d_{out}) \ll d_{in} \cdot d_{out}$。
- ✓ **效果好：** 接近全量微调的性能。
- ✓ **部署友好：** 可与原模型合并，推理无额外开销。

LoRA：冻结原模型，只训练低秩矩阵 A、B；用低秩更新 $\Delta W = (\alpha/r)BA$ 来适配新任务。

AdaLoRA： 动态分配不同层的低秩预算

核心动机

- LoRA 使用固定秩 r，但不同层 / 模块的重要性不同，存在参数分配不合理的问题。

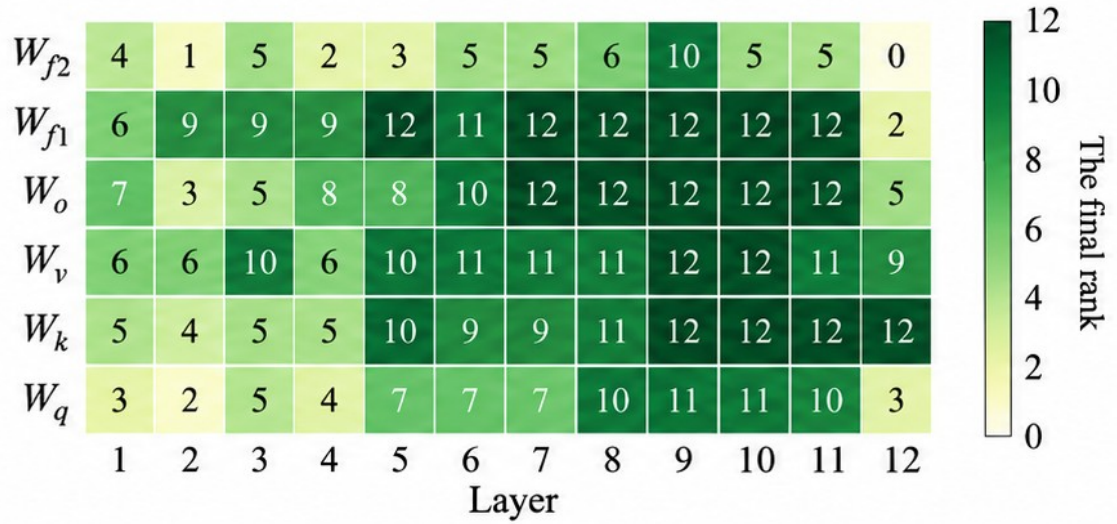
方法概述

- **基于 SVD 的参数化：**

$$\Delta W = P\Lambda Q, \quad (A = \sqrt{\Lambda}Q, \quad B = P\sqrt{\Lambda})$$

- **动态秩分配：**
  - 计算每个模块的“重要性” $S_{k,i}$（由奇异值和梯度得到）；
  - 在总预算约束下，为不同层 / 模块分配秩 $r_{k,i}$（重要的分配更大）。

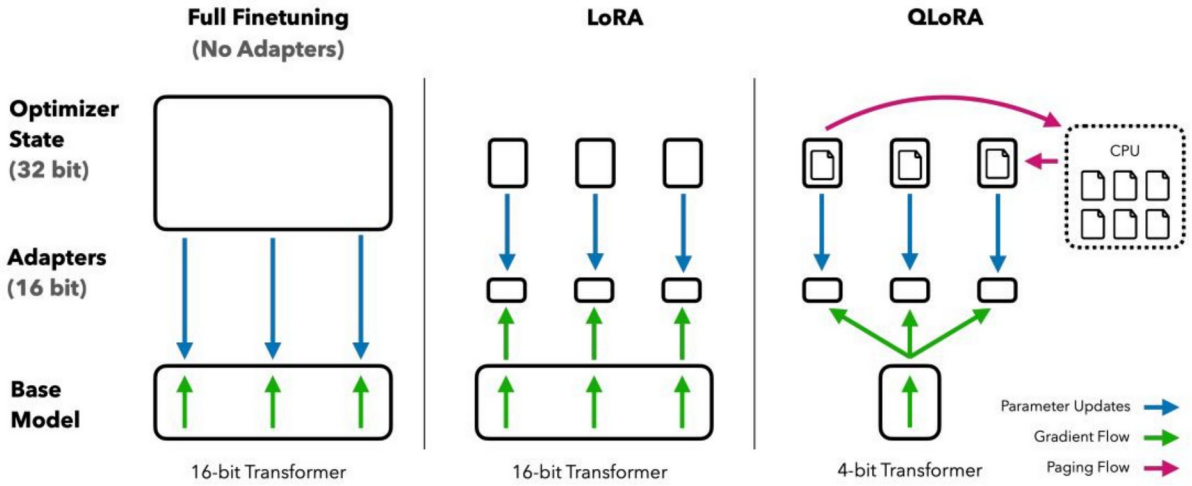
实验结果（秩分配）



结论

- 不同模块、不同层的最优秩显著不同（如中后层更大）。
- AdaLoRA 能根据重要性动态分配秩，在相同参数预算下获得更好的微调效果。

QLoRA：量化冻结基座 + LoRA，降低微调显存



QLoRA 的三项关键技术

1 NF4（4-bit NormalFloat）

关键作用 用 4-bit存储模型权重，并根据权重近似正态分布设计非均匀的量化值，尽量减少精度损失。

示例 4-bit 仅有 16 个量化值：



均匀量化
（量化值等间隔分布）

NF4
（0 附近更密集）

2 双量化（Double Quantization）

关键作用 将量化过程中使用的缩放系数（Scale）再量化，进一步减少存储开销。

示例 对 Scale 再量化：

Scale (FP32) 4 字节 → **Scale** (8-bit) 1 字节

对量化常数再次压缩，进一步节省显存。

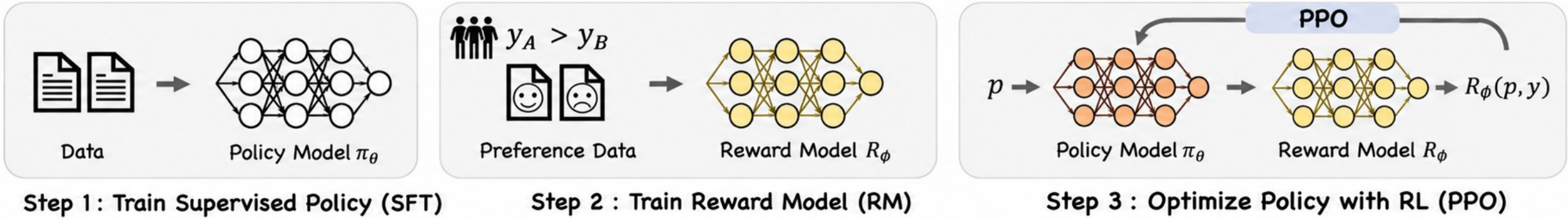
3 分页优化器（Paged Optimizers）

关键作用 利用 CPU 内存按需管理优化器状态，缓解训练中的显存峰值。

示例 显存峰值时的内存管理：

GPU 显存 (如 24 GB) 不足时 ↔ **CPU 内存** (临时存放部分优化器状态)

RLHF： 先学会生成，再学习人类偏好



Step 1 : Train Supervised Policy (SFT)　　Step 2 : Train Reward Model (RM)　　Step 3 : Optimize Policy with RL (PPO)

Step 1： SFT（监督微调）

训练数据　指令 – 回答示范数据

$D_{SFT} = \{(x, y^*)\}$

训练对象　策略模型 π_θ

优化目标　最小化生成交叉熵（最大化参考回答概率）

$$L_{SFT} = -\sum_t \log \pi_\theta(y_t^* \mid x, y_{<t}^*)$$

Step 2： Reward Model（奖励模型）

训练数据　人类偏好比较数据

$D_{RM} = \{(x, y_w, y_l)\}\quad (y_w > y_l)$

训练对象　奖励模型 R_ϕ

优化目标　使优选回答的评分更高

$$L_{RM} = -\log \sigma\left(R_\phi(x, y_w) - R_\phi(x, y_l)\right)$$

Step 3： PPO（强化学习优化）

训练数据　Prompt + 模型生成的回答

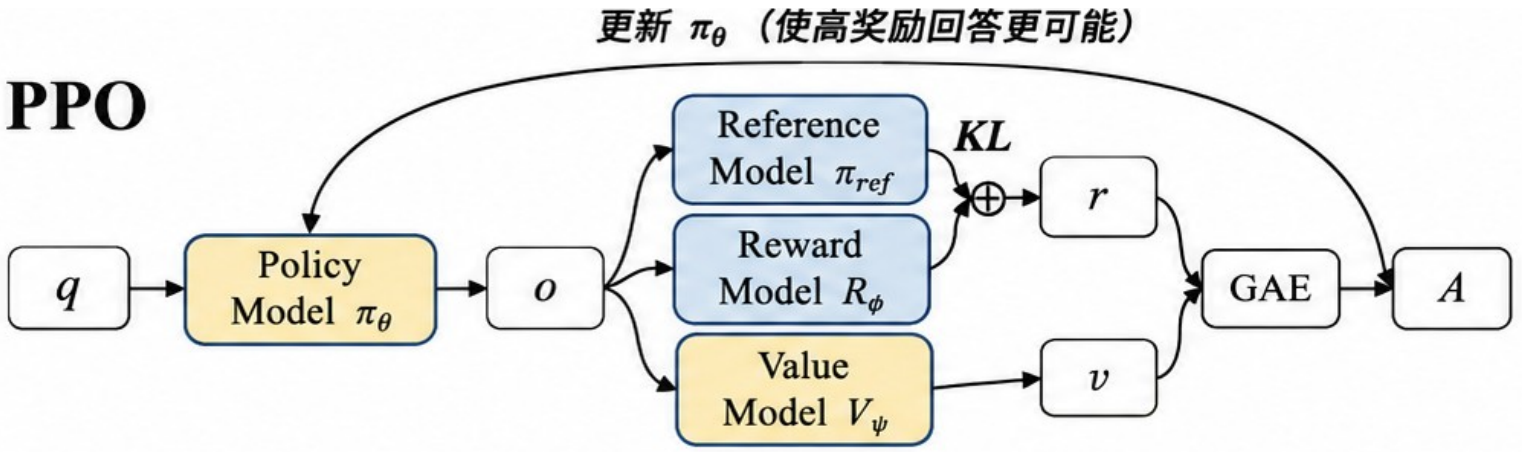
$y \sim \pi_\theta(\cdot \mid x)$

训练对象　策略模型 π_θ （奖励模型 R_ϕ 冻结）

优化目标　最大化奖励，并约束策略与参考模型的偏移

$$\max_\theta \; E_{x,y\sim\pi_\theta}[R_\phi(x, y)] - \beta\, E_x\left[D_{KL}(\pi_\theta \,\|\, \pi_{ref})\right]$$

PPO：奖励与价值预测如何转化为策略更新



PPO 目标函数（裁剪形式）

$$L_{PPO} = E_t\left[\min\left(\rho_t A_t,\ \text{clip}(\rho_t, 1-\varepsilon, 1+\varepsilon)A_t\right)\right]$$

- ρ_t：当前策略与旧策略的概率比；
- A_t：优势值（由奖励和价值预测通过 GAE 计算）；
- ε：裁剪范围，限制策略更新幅度。

从奖励到策略更新：一个例子（简化）

1 生成回答（记录概率）

q：请向六岁孩子解释月球。

y：月球绕着地球转。

Token	月球	绕着	地球	转
π_{old}	0.40	0.50	0.60	0.50

2 奖励与价值预测

- 奖励（RM）：$r = 8.0$
- 价值（V）：$V = [5.0, 5.5, 5.8, 6.0]$

3 计算优势（简化）

$$A_t = r - V_t$$

Token	月球	绕着	地球	转
A_t	3.0	2.5	2.2	2.0

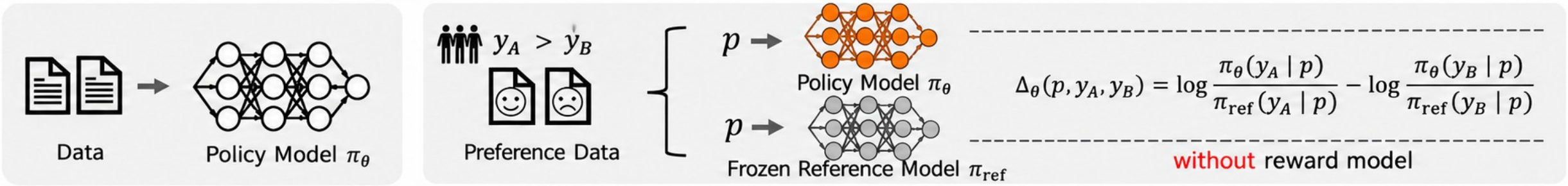
4 PPO 更新（以 $t=1$ 为例）

π_θ(月球) $= 0.50$，$\rho_1 = 1.25$

$$L_1 = \min(1.25 \times 3.0,\ 1.2 \times 3.0) = 3.6$$

Reward → Value + GAE → Advantage → PPO Clip → 更新 Policy

DPO： 通过偏好数据直接优化当前模型



Step 1：Train Supervised Policy（SFT）

Step 2：Optimize Policy with DPO

DPO 优化目标

利用偏好数据，直接优化策略，使模型更偏向优选回答。

$$\mathcal{L}_{\text{DPO}} = -\mathrm{E}_{(p,y_A,y_B)}\left[\log\sigma(\beta \cdot \Delta_\theta)\right]$$

- Δ_θ：两回答相对参考模型的对数概率差。
- β：偏好强度的超参数。

简单示例

模型	y_A（优选）	y_B（劣选）
π_θ（当前）	-1.20	-2.30
π_{ref}（参考）	-1.50	-1.80

$$\Delta_\theta = (-1.20 + 1.50) - (-2.30 + 1.80) = 0.80$$

$$\mathcal{L}_{\text{DPO}} = -\log\sigma(0.80) \approx 0.37$$

→ $\Delta_\theta > 0$，模型更偏向 y_A。

训练流程与优势

输入 问题 p + 偏好对 (y_A, y_B)

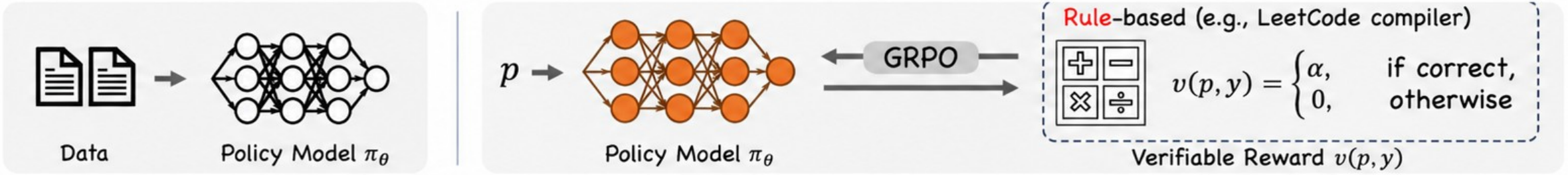
训练 冻结参考模型 π_{ref}，最小化 $\mathcal{L}_{\text{DPO}}$ 更新 π_θ。

优势

- 无需奖励模型和价值模型（无 PPO/GAE）
- 训练更简单、稳定、成本更低
- 效果与 RLHF（PPO）相当

结论 DPO 将人类偏好直接转化为优化目标，以更简单的方式实现模型对齐。

GRPO：同题多回答，利用组内相对奖励



核心思想

- 针对同一问题生成多条回答，通过组内奖励的相对水平优化策略。
- 不需要训练奖励模型（RM），也不需要价值模型（Value Model）。

组内优势计算

对同一问题生成 G 个回答 $\{y_1, \cdots, y_G\}$，计算组内优势：

$$A_i = \frac{r_i - \bar{r}}{\text{std}(r_1, \cdots, r_G)}$$

- $\bar{r}$：组内平均奖励；std：组内标准差。
- $A_i > 0$ 表示该回答优于组内平均，反之为负。

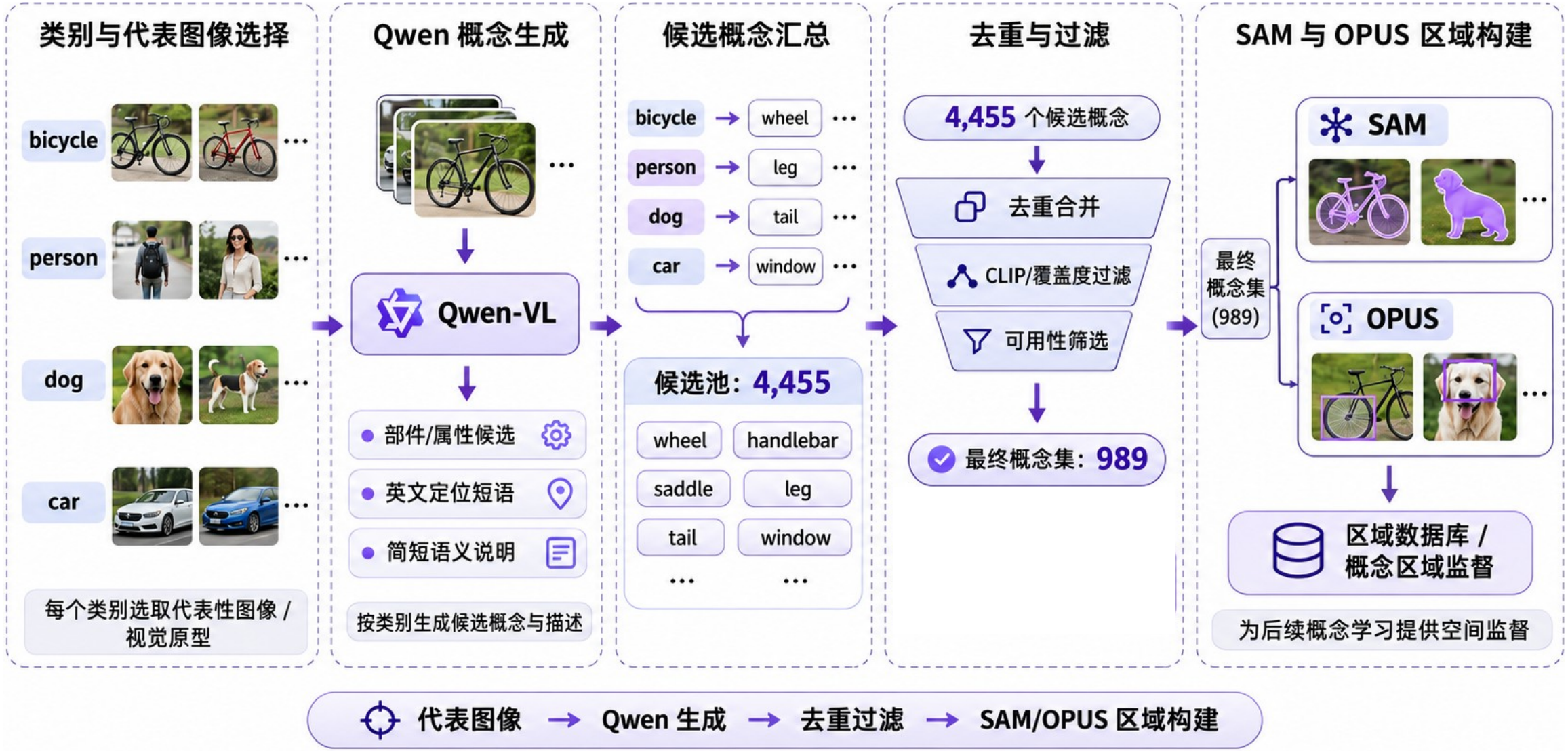
优化目标

使用与 PPO 相同的裁剪目标更新策略：

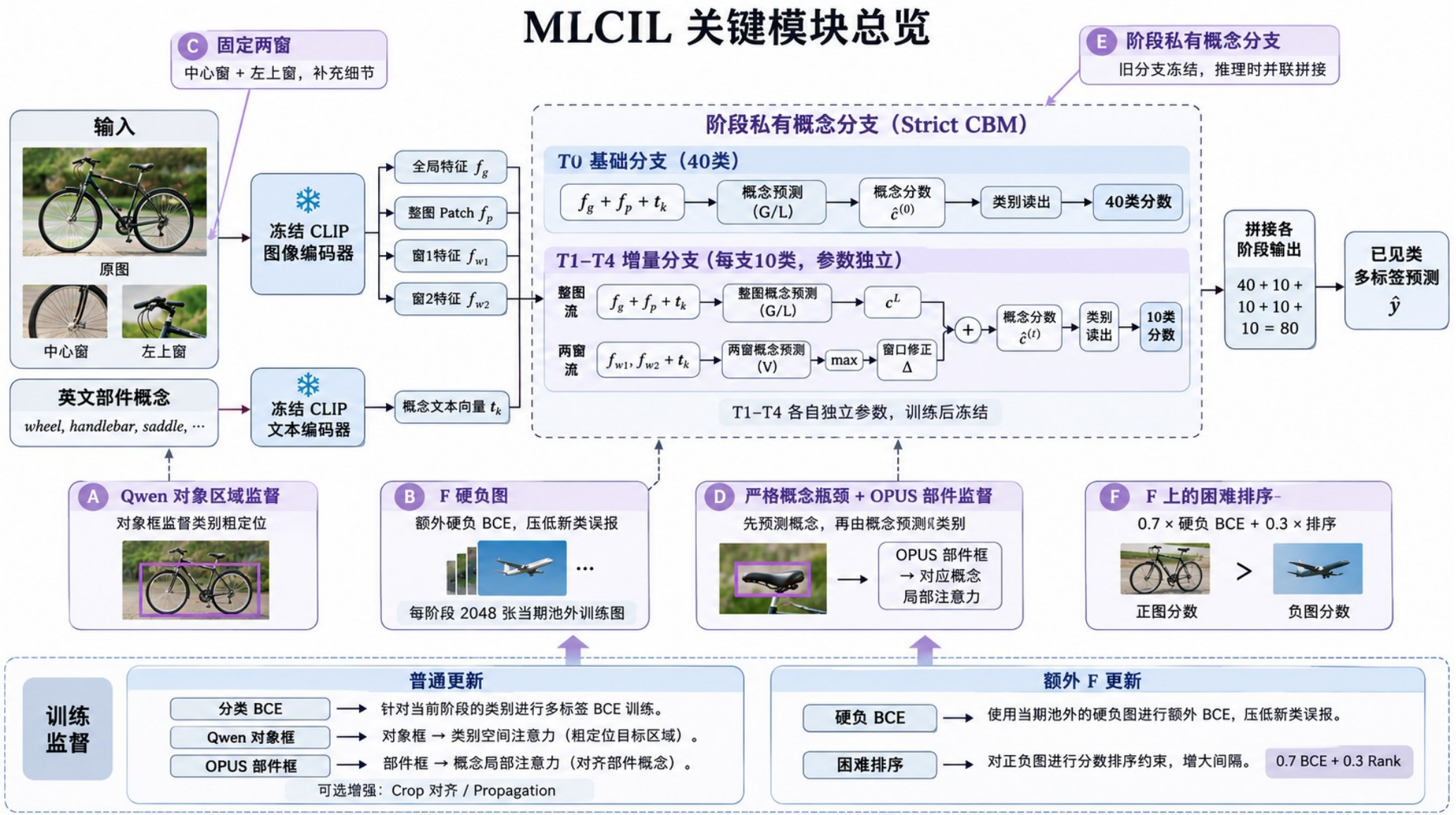
$$L_{\text{GRPO}} = -\text{E}_{i,t}\left[\min(\rho_{i,t}A_i, \text{clip}(\rho_{i,t}, 1-\epsilon, 1+\epsilon)A_i)\right]$$

- $\rho_{i,t} = \frac{\pi_\theta(y_{i,t}|p, y_{i,<t})}{\pi_{old}(y_{i,t}|p, y_{i,<t})}$：新旧策略的概率比。
- ϵ：裁剪阈值（如 0.2）。

概念生成方式和使用方式



MLCIL-CBM 方案


MLCIL 关键模块总览
C 固定两窗
中心窗 + 左上窗，补充细节
E 阶段私有概念分支
旧分支冻结，推理时并联拼接
输入
原图
中心窗
左上窗
冻结 CLIP 图像编码器
全局特征 f_g
整图 Patch f_p
窗1特征 f_{w1}
窗2特征 f_{w2}
英文部件概念
wheel, handlebar, saddle, ···
冻结 CLIP 文本编码器
概念文本向量 t_k
阶段私有概念分支（Strict CBM）
T0 基础分支（40类）
$f_g + f_p + t_k$
概念预测 (G/L)
概念分数 $\hat{c}^{(0)}$
类别读出
40类分数
T1–T4 增量分支（每支10类，参数独立）
整图流
$f_g + f_p + t_k$
整图概念预测 (G/L)
c^L
两窗流
$f_{w1}, f_{w2} + t_k$
两窗概念预测 (V)
max
窗口修正 Δ
+
概念分数 $\hat{c}^{(t)}$
类别读出
10类分数
T1–T4 各自独立参数，训练后冻结
拼接各阶段输出
40 + 10 + 10 + 10 + 10 = 80
已见类多标签预测 $\hat{y}$
A Qwen 对象区域监督
对象框监督类别粗定位
B F 硬负图
额外硬负 BCE，压低新类误报
每阶段 2048 张当期池外训练图
D 严格概念瓶颈 + OPUS 部件监督
先预测概念，再由概念预测类别
OPUS 部件框 → 对应概念局部注意力
F F 上的困难排序-
0.7 × 硬负 BCE + 0.3 × 排序
正图分数
>
负图分数
训练监督
普通更新
分类 BCE → 针对当前阶段的类别进行多标签 BCE 训练。
Qwen 对象框 → 对象框 → 类别空间注意力（粗定位目标区域）。
OPUS 部件框 → 部件框 → 概念局部注意力（对齐部件概念）。
可选增强：Crop 对齐 / Propagation
额外 F 更新
硬负 BCE → 使用当期池外的硬负图进行额外 BCE，压低新类误报。
困难排序 → 对正负图进行分数排序约束，增大间隔。 0.7 BCE + 0.3 Rank

近期进展及结果

南京航空航天大學 NANJING UNIVERSITY OF AERONAUTICS AND ASTRONAUTICS

B0-C10（10-80 类）

方法	视觉 Backbone	Backbone 状态	Last ↑	Avg ↑
无 CBM	CLIP ViT-B/16	冻结	82.1253	85.1873
有 CBM	CLIP ViT-B/16	冻结	**82.3879**	85.2018
CBM 辅助	CLIP ViT-B/16	冻结	82.1944	**85.3144**
DeCLIP	CLIP ViT-B/16	冻结	78.3 ± 0.4	84.0 ± 0.3
P2L-CA+	ViT-B/16（ImageNet-21K）	冻结	75.7	81.9
DPA	CLIP	冻结	73.3	80.7
L3A	TResNet-M	首批训练，后续冻结	77.6	81.5
HCP	TResNet-M	训练	71.2	77.9
CSC	TResNet-M	训练	72.8	78.0

方法	核心思路（详细）
DeCLIP	**基于 CLIP 的持续学习：用文本提示做增量训练，并用知识蒸馏保持旧类。** ■ 将图像编码器冻结，使用**类别文本 Prompt**（正/负提示）进行增量训练； ■ 在新类学习时，通过**知识蒸馏**（KD）约束旧类的预测，缓解遗忘。
P2L-CA+	**参数高效的增量学习：在冻结 backbone 上加入可学习提示并用注意力增强。** ■ 在冻结的 ViT/CLIP backbone 上，引入可学习的**提示**（Prompt/Adapter），以少量参数适配新类； ■ 利用**类别注意力**（CA）增强类别表征，提升新类判别能力并保持旧类性能。
DPA	**双路径适配器：结合全局与局部适配器学习新类，并保持旧类能力。** ■ 设计**全局适配器**（建模整体语义）和**局部适配器**（建模区域/细节）； ■ 两条路径联合训练以学习新类，同时通过**知识蒸馏**约束旧类，缓解遗忘。
L3A	**大规模增量分类：分阶段训练并在阶段结束后冻结，保持旧类稳定。** ■ 采用**分阶段**的增量训练策略，每个阶段仅学习新加入的类别； ■ 阶段结束后**冻结**当前模型（作为下一阶段的教师），在保留旧类的同时扩展新类。
HCP	**分层概念原型框架：利用类与概念的层次结构进行增量学习。** ■ 构建“**类—概念**”的层次化表示（如物体→部件/属性），为每个概念维护原型； ■ 通过**概念原型**的继承与更新实现知识迁移，增强新类学习并保持旧类稳定。
CSC	**类别语义约束的增量学习：通过语义关系对齐与约束缓解新类冲突。** ■ 利用**类别语义关系**（如相似性/排斥性）构建约束损失； ■ 在增量训练中通过语义对齐与约束，缓解新类与旧类的语义冲突和遗忘。

近期进展及结果

南京航空航天大學 NANJING UNIVERSITY OF AERONAUTICS AND ASTRONAUTICS

B40-C10（40–80 类）

方法	视觉 Backbone	Backbone 状态	Last ↑	Avg ↑
无 CBM	CLIP ViT-B/16	冻结	**82.9588**	**84.1964**
有 CBM	CLIP ViT-B/16	冻结	82.9349	83.9826
CBM 辅助	CLIP ViT-B/16	冻结	–	–
DeCLIP	CLIP ViT-B/16	冻结	81.4 ± 0.3	84.1 ± 0.2
P2L-CA+	ViT-B/16（ImageNet-21K）	冻结	81.4	84.1
DPA	CLIP	冻结	77.2	81.1
L3A	TResNet-M	基础阶段训练，增量阶段冻结	78.8	79.9
HCP	TResNet-M	训练	75.3	78.9
CSC	TResNet-M	训练	75.0	78.2

Thanks