

摘要

视觉语言模型（VLMs）通过利用大规模预训练模型处理各类下游任务，推动了机器学习的变革。尽管在标签、训练和数据效率方面已有显著提升，但当前最先进的视觉语言模型仍需要针对特定任务进行超参数调优，且未能充分利用测试样本。为解决这些挑战，我们提出了一种基于图结构的标签高效适应与推理方法。该方法动态构建文本提示、少量示例和测试样本之间的图关系，利用标签传播进行推理，无需任务特定调参。与现有的零样本标签传播技术不同，本方法无需额外的无标签支持集，并能通过动态图扩展有效利用测试样本流形。我们还引入了一种上下文感知的特征重加权机制，以提高任务适应精度。此外，该方法支持高效的图扩展，可实现实时归纳推理。在细粒度分类和分布外泛化等下游任务上的大量实验验证了本方法的有效性。源代码已公开于：<https://github.com/Yushu-li/ECALP>

1 引言

新兴的基础模型已经改变了传统机器学习模型的开发范式。通过在海量图像-文本对上预训练视觉语言模型，这些模型展现出强大的跨下游任务推理能力，同时显著减少了对大规模数据收集和标注的依赖。近期关于预训练 VLMs 下游任务适应的研究主要集中在提升标签效率、训练效率和数据效率。DMN 是一种综合优化上述三个维度的代表性方法：它利用文本提示、测试样本和少量示例构建三支分类器，并通过分支预测融合生成最终结果。尽管 DMN 在细粒度分类任务中取得了最先进的性能，但其预测融合过程需调整任务特定的系数，且在缺乏独立验证集时往往依赖测试集进行超参数调优。此外，DMN 通过引入测试样本的记忆库为每个样本合成自适应分类器。但我们认为，这些测试样本的潜力尚未被充分挖掘——它们不仅可用于分类器合成，还能更有效地捕捉数据流形以支持直推推理，尤其在标注样本稀缺的场景下。

什么是细粒度分类？

细粒度分类是指在基本类别的基础上，对其子类进行更精细的划分，广泛应用于计算机视觉和图像处理领域。

为减少任务特定超参数调优的需求并更高效利用测试样本，我们提出了一种基于图结构的下游任务适应与推理方法。该方法动态构建由少量示例（如可用）、测试样本及语义标签关联文本提示组成的图结构，通过捕捉数据内在流形特征，并采用标签传播技术进行推理。与 DMN 相比，本方法无需任务特定超参数调优，且能更充分挖掘未标注测试样本中的潜在信息。近期提出的 ZLaP 方法基于标签传播实现了零样本 VLM 适应，其通过外部数据集构建流形结构，并采用闭式解将文本提示标签传播至测试样本。但我们认为 ZLaP 在标签高效 VLM

适应中存在三大局限：首先，**闭式解**在 ImageNet 等大规模数据集（含 5 万测试样本）中因拉普拉斯矩阵求逆计算代价过高，导致图连接性受限；其次，仅基于训练集的静态图结构未纳入测试样本，无法利用测试数据流形特性，在训练-测试集分布偏移时性能显著下降；第三，依赖余弦相似度度量测试样本与提示词亲和度存在偏差——由于 VLMs 预训练数据涵盖多样化的图文对，其视觉编码特征可能包含背景物体或图像风格等无关语义信息，影响下游任务判别性。

Zlap 方法、闭式解？

Zlap 方法通常用于信号处理、控制理论或数学建模中，将离散问题（如差分方程）通过 Z 变换或**拉普拉斯变换**转化为代数方程求解。其核心步骤包括：

- 1) 变换：对离散序列或方程应用 Z 变换（或拉普拉斯变换），将时域问题转为复频域问题。
- 2) 求解：在变换域中简化方程（如解代数方程）。
- 3) 逆变换：将解逆变换回时域，得到最终解。

闭式解指能用有限次标准运算（加、减、乘、除、指数、对数、三角函数等）和已知函数（如多项式、初等函数）明确表达的解。与数值解或迭代解不同，闭式解是精确的解析表达式。

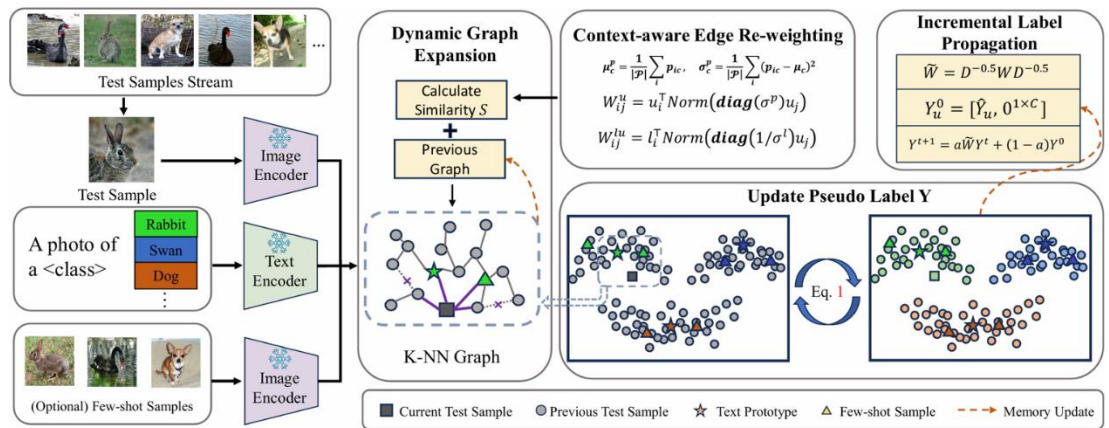


图 1：ECALP 整体框架示意图。本方法通过整合文本提示、少量样本及测试样本构建动态图结构。当新样本输入时，系统基于上下文感知的重加权边权重动态扩展图网络，并采用迭代式解决方案预测所有测试样本的类别标签。

2 贡献

本文提出了一种面向视觉语言模型（VLMs）标签高效适应的整体性标签传播方法（整体框架如图 1 所示）。针对计算效率的挑战，我们采用迭代式标签传播方案替代闭式解，该方法可通过增量式标签传播与标签重置优化计算过程。此外，我们提出**上下文感知的特征维度重加权机制**，以增强模型对下游任务的适应能力——仅需文本提示和或少量示例即可为特征重加权提供上下文信息。我们还设计了高效的图扩展机制，支持对测试样本流进行归纳推理，而无需依赖全部测试数据完成直推式标注。该方案被命名为**高效上下文感知标签传**

播 (ECALP)。我们在细粒度分类、分布偏移和少样本分类等多类下游任务上进行了实验验证，主要贡献如下：

- 提出基于标签传播视角的视觉语言模型统一高效适配框架。与现有最优方法相比，该方法在保持超参数固定的同时实现了更高的推理速度。
- 针对视觉编码器特征多样性问题，提出基于文本嵌入或下游任务少量样本特征的统计量对视觉特征维度进行动态重加权。
- 通过多场景下游任务验证方法有效性：在细粒度分类与分布外泛化等任务上的实验表明，本方法取得了当前最优性能。

3 方法论

问题定义

我们首先形式化定义视觉语言模型的零样本-少样本适应任务。将 CLIP 等 VLM 模型的图像编码器和文本编码器分别表示为 $f(x)$ 和 $g(z)$ 下游任务包含未标注测试数据的编码特征

$$D_u = \{u_i = f(x_i)\}_{i=1 \dots N}$$

以及可选的少样本标注数据特征与标签

$$D_l = \{l_j = f(x_j), y_j\}_{j=1 \dots N}$$

已知类别名称与提示模板组合形成文本提示词，每个类别包含多个文本提示

$$\{g(z_{cm})\}$$

取其平均值作为第 c 类的文本原型

$$\frac{1}{M} \sum_m g(z_{cm})$$

所有原型记为

$$P = \{p_c\}_{c=1 \dots C}$$

该任务的目标是利用文本原型 P 和可选的少样本标注数据 D_l 推断未标注数据 D_u

3.2 直推式标签传播方法重构

我们重新审视直推式标签传播技术，提出面向视觉语言模型高效适配的全局图构建方法。不同于直接计算未标注样本与类别原型的相似度，本方法旨在利用所有下游数据构成的流形结构。定义基于全部下游数据的图结构为 $G = (V, E)$ ，其中每个节点代表下游数据样本或原型，即 $v_i \in P \cup D_l \cup D_u$

图的邻接矩阵记为

$$W \in R^{N_p + N_l + N_u}$$

归一化邻接矩阵通过

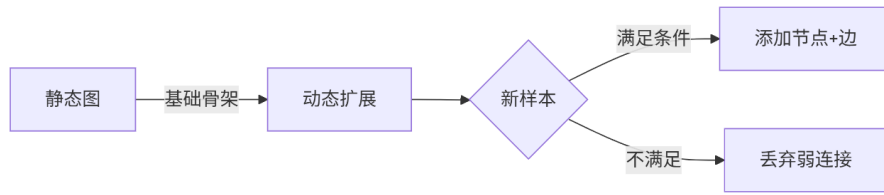
$$\tilde{W} = D^{-0.5}WD^{-0.5}$$

计算（ D 为度矩阵）。标签矩阵 Y 根据式(1)进行迭代传播与优化，其中 Y^0 为初始标签， t 为迭代次数， α 为加权超参数。当迭代次数趋近无穷时，传播过程将收敛至闭式解。

$$Y^{t+1} = \alpha \tilde{W}Y^t + (1 - \alpha)Y^0 \Rightarrow Y^\infty = (I - \alpha \tilde{W})^{-1}Y^0$$

在本研究的 VLM 适配框架下，初始标签矩阵定义如式(2)，其中 Y_p ， Y_l ， Y_u 分别表示文本原型、少样本标注数据和未标注数据的标签：

$$Y^0 = \{Y_p^0, Y_l^0, Y_u^0\}$$



构建初始的静态图，根据 KNN 原则动态更新标签

迭代更新算法，新节点到达时仅需局部迭代，无需全图重计算，每次标签更新的就是迭代更新过的邻接矩阵。

具体算法参考如下：

算法	视觉语言模型自适应迭代标签传播算法
1.	Input: D_u, D_l, P, T
2.	Output: $\{y_i\}$
3.	Initial Graph $W = 0$; initial label Y_p^0, Y_l^0, Y_u^0
4.	for $x_i \in D_u$ do
5.	Calculate W_{ij}^u & W_{ij}^{ul} // 计算初始图的邻接矩阵
6.	Update W_u, W_{up} , and W_{ul} // 动态更新
7.	$W = W + W^T, W = W^\gamma, \tilde{W} = D^{-0.5}WD^{-0.5}$
8.	for $t = 0$ to $(T - 1)$ do
9.	$Y^{t+1} = \tilde{W}Y^t$;
10.	$Y_p^{t+1} = Y_p^0, Y_l^{t+1} = Y_l^0$; // 先验知识保持不变，锚点
11.	end for
12.	$y_i = \operatorname{argmax}_c Y_{ic}^u$
13.	Update Y_0^u ;
14.	end for
15.	return $\{y_i\}$

异步收敛判断？

每个节点不是统一的更新的，当 ΔY_i 小于某个值时即可停止更新

方差感知重加权定理？

给定模态特征矩阵 $\mathbf{X} \in \mathbb{R}^{n \times d}$

维度方差

$$\sigma_c = \text{Var}(\{X_{1,c}, X_{2,c}, \dots, X_{n,c}\}), c = 1 \dots d$$

重加权相似度（邻接矩阵）

$$s(u, v) = u^T \text{diag}(w) v, w_c = f(\sigma_c)$$

其中 $f(x)$ 是映射函数