arXiv:2303.05847v1 [cs.IR] 10 Mar 2023

用于量化并最大化多任务学习中知识迁移的梯度协调

Xuanhua Yang、Jianxin Zhao、Shaoguo Liu*、Liang Wang 和 Bo Zheng 阿里巴巴集团，
北京，中国
{xuanhua.yxh,zhaojianxin.zjx,shaoguo.lsg,liangbo.wl,bozheng}@alibaba-inc.com

摘要

多任务学习（多任务学习）已广泛应用于在线广告和推荐系统。为解决负迁移问题，近期研究提出了侧重于方向或幅度梯度对齐的优化方法。然而，已有研究证明，在有限的共享容量中，通用知识和特定知识并存，因此过分强调梯度对齐可能会挤占任务特定知识，反之亦然。本文提出一种由迁移驱动的方法**CoGrad**，通过**Co**ordinated**Grad**ient 修改，自适应地最大化知识迁移。我们将迁移明确量化为一个任务对另一个任务的损失降低，并通过对其进行优化推导出辅助梯度。我们将该梯度融入原始任务梯度中来执行优化，使模型自动最大化任务间迁移并最小化单个任务损失。因此，CoGrad 能够协调通用知识与特定知识，从而提升整体性能。此外，我们还提出了一种高效的海森矩阵近似方法，使 CoGrad 计算高效且易于实现。离线和在线实验均验证，CoGrad 的表现显著优于以往方法。

1 引言

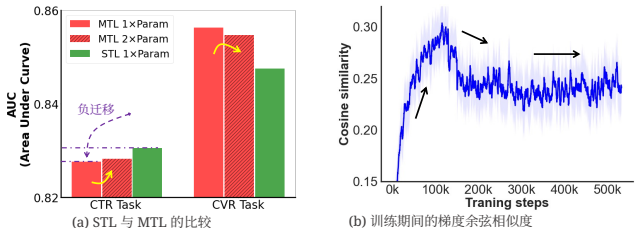
在在线广告和推荐系统中，多任务学习（MTL）已被证明在同时预测多种用户行为（例如点击、浏览和购买）方面极为有效 [16, 17]。如图 2 (a) 所示，MTL 的常见骨干网络由用于编码多行为表征的共享模块和若干用于输出任务特定预测的专用头组成。与单任务学习（STL）相比，共享模块被视为实现任务间互惠收益的关键组件。

编码通用知识带来的最关键问题是 负迁移 [12]。如图 1(a) 所示，与 STL 相比，点击率[1] 任务的性能有所下降（即负迁移），而转化率任务则有所提升。 [中也观察到了类似现象

8, 14]。近期，研究者提出了若干基于梯度对齐策略修改任务梯度的优化方法，以解决这一问题。PCGrad [18] 和 GradVac [15]通过调整梯度方向来保持方向一致性，从而增强迁移。GradNorm [2] 和 MetaBalance [5]通过统一梯度幅值，避免共享模块被梯度幅值较大的某些任务主导。MGDA [13] 和 CAGrad [9] 利用帕累托 解同时调整方向和幅值，以缓解梯度冲突。

[1]点击率：点击率，转化率：点击后转化率

* 通讯作者：Shaoguo Liu。



(a) STL 与 MTL 的比较　(b) 训练期间的梯度余弦相似度

图1：（a）负迁移问题及扩展模型容量的影响。（b）梯度余弦相似度的趋势。

然而，先前研究已表明，通用知识和特定知识在共享模块中处于相互竞争的状态 [14], 。过分强调梯度对齐，可能会挤占同样对单个任务有用的特定知识。例如，基于方向的方法通常通过提高余弦相似度来对齐梯度，因为余弦相似度被认为与迁移高度相关 [18]。然而，在处理两个弱相关任务时，这甚至可能降低整体性能。另一方面，在我们的设置中，我们观察到余弦相似度在训练早期上升，随后下降（如图 1(b) 所示）。这表明，多任务学习模型容易编码特定知识，而这可能会占用通用信息的容量预算。

因此，对通用知识或特定知识进行过度编码都可能降低整体性能。为解决这一问题，一种直接的方法是扩展共享参数。然而，由于模型倾向于特定性，这可能会使模型更有可能学习特定知识，而非通用知识。图1（a）中的观察结果支持这一点：扩展参数会提升点击率性能，却会降低转化率。因此，协调通用知识和特定知识的编码颇具挑战。

本文提出一种由迁移驱动的方法**CoGrad**，通过 **Co**ordinated**Grad**ient modification，自适应地最大化知识迁移。具体而言，我们从理论上将任务间迁移量化为：由另一个任务梯度引起的更新所带来的某一任务损失降低。随后，我们优化这一量化指标以导出辅助梯度，并将其纳入原始任务梯度。通过这种方式，CoGrad 能够最大化任务间迁移，同时最小化各个单个任务损失。因此，CoGrad 实现了通用知识与特定知识之间的协调，从而提升整体性能。此外，CoGrad 包含海森矩阵，导致计算开销较高。我们还提出一种高效的海森矩阵近似，使 CoGrad 计算高效且易于在工业应用中实现。我们的贡献如下：

- 据我们所知，我们首次明确量化了任务间迁移，并将其用于梯度调制，这在多任务学习中具有广阔的应用前景。

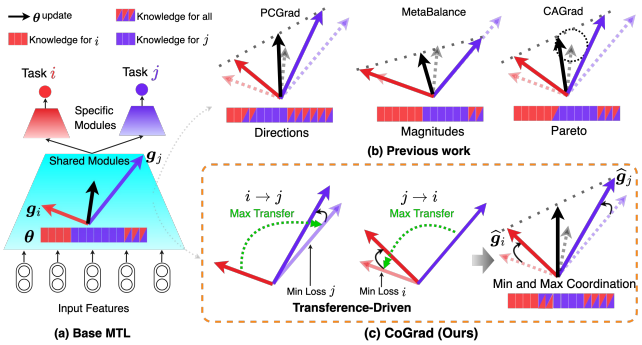


图 2：（a）展示了多任务学习范式和基础梯度下降。与通过统一梯度方向或幅值的既有工作（b）相比，CoGrad（c）采用迁移驱动的方式调整梯度，在最大化任务间迁移的同时，最小化单个任务损失。

- 我们提出了一种由迁移驱动的梯度调制方法（CoGrad），该方法能够自适应地最大化任务间迁移，同时计算效率高。
- 实验表明，CoGrad 优于以往的基线方法。我们的实证分析证实，CoGrad 能够有效协调通用知识和特定知识。

2 问题定义

在多任务推荐中，考虑一组 T 任务 $\mathrm{T}=\{t\}_{t=1}^{T}$，以及训练数据集 $\mathrm{D}=\{(x_n,\ \{y_n^t\}_{t\in\mathcal{T}})\}_{n=1}^{|\mathcal{D}|}$，其中$x_n$ 和 y_n^t 分别表示 n_{th} 实例的特征向量（包括用户特征、物品特征和上下文特征）以及用户二元反馈标签（用户是否点击、浏览、购买等）。我们将 t_{th} 任务数据集表示为 $\mathrm{D}_t=\{x_n,\ y_n^t\}_{n=1}^{|\mathcal{D}|}$。

令 $\boldsymbol{L}_t(\mathrm{D}_t;\theta,\ \phi_t)$ 表示任务 t 在 D_t 上的损失，其中 θ 和 ϕ_t 分别表示共享参数和特定参数。标准的多任务学习损失采用加权和形式进行聚合：

$$\mathcal{L}(\mathcal{D};\theta,\{\phi_t\}_{t\in\mathcal{T}})=\sum_{t\in\mathcal{T}}w_t\boldsymbol{L}_t(\mathcal{D}_t;\theta,\phi_t) \tag{1}$$

此处省略了正则化项，$\{w_t\}_{t=1}^{T}$ 表示用于缩放任务损失的权重。所有参数均按如下方式更新：

$$\theta^{k+1}=\theta^k-\eta\sum_{t\in\mathcal{T}}w_t\nabla_\theta\boldsymbol{L}_t(\mathcal{D}_t;\theta^k,\phi_t^k) \tag{2}$$

$$\phi_t^{k+1}=\phi_t^k-\eta w_t\nabla_{\phi_t}\boldsymbol{L}_t(\mathcal{D}_t;\theta^k,\phi_t^k),\forall t\in\mathcal{T} \tag{3}$$

多任务学习通过迭代公式 2 实现知识迁移。与大多数以往方法通过修改梯度幅值或方向不同，我们尝试明确量化任务间迁移，以指导公式 2 中的优化。

3 所提方法

在迭代 θ 过程中，我们期望来自某个任务的梯度更新既能最小化该任务自身的损失，也能尽可能地帮助降低另一个任务的损失。我们通过量化并最大化知识迁移来实现这一颇具挑战性的目标，以此进行优化。本节将详细阐述我们的方法。为简单起见，我们用 $\boldsymbol{L}_t(\theta)$ 和 $\boldsymbol{g}_t(\theta)$ 分别表示 $\boldsymbol{L}_t(\mathrm{D}_t;\theta,\ \phi_t)$ 和 $\nabla_\theta\boldsymbol{L}_t(\mathrm{D}_t;\ \theta,\ \phi_t)$。

算法1： CoGrad训练算法

输入： 训练数据集 D，初始参数 $\{\theta\}\cup\{\phi_i\}_{i\in\mathcal{T}}$，学习率 η，$\{w_i\}_{i\in\mathcal{T}}$用于缩放任务损失，$\{\gamma_i\}_{i\in\mathcal{T}}$用于控制最大化迁移的强度。

当 尚未收敛 **时**
- 从 D 中采样一批样本 B；
- 更新任务特定参数：
- $\phi_i^{k+1}=\phi_i^k-\eta\nabla_{\phi_i}\boldsymbol{L}_i(\mathrm{B};\theta^k,\ \phi_i^k)\ \forall i\in\mathrm{T}$;
- **共享参数的 CoGrad 优化**：
- 计算每个任务的任务梯度：θ
- $\boldsymbol{g}_i(\theta^k)=\nabla_\theta\boldsymbol{L}_i(\mathrm{B};\theta^k,\ \phi_i^k)\ \forall i\in\mathrm{T}$;
- 计算**CoGrad**，$\forall i\in\mathrm{T}$:
- $\widehat{\boldsymbol{g}}_i(\theta^k)=\boldsymbol{g}_i(\theta^k)-\sum_{j\neq i,j\in\mathrm{T}}\gamma_j\boldsymbol{g}_i(\theta^k)\odot\boldsymbol{g}_i(\theta^k)\odot\boldsymbol{g}_j(\theta^k)$;
- 更新共享参数 θ:
- $\theta^{k+1}=\theta^k-\eta\sum_{i\in\mathrm{T}}w_i\widehat{\boldsymbol{g}}_i(\theta^k)$;

结束

3.1 量化知识迁移

量化多任务学习中的知识迁移具有极大的提升泛化能力的潜力。先前的工作 [3]测量了任务间的亲和度，以便对任务进行分组。受此启发，我们将从任务 i 到 j $(i,\ j\in\mathrm{T})$ 的迁移量定义为任务 j 的损失降低，该降低由任务 i 的梯度更新所引起。具体而言，假设共享参数 θ 在时间步 k 以学习率 $\gamma_i>0$由任务 i 更新，则有：

$$\theta^{k+\tau_i}=\theta^k-\gamma_i\boldsymbol{g}_i(\theta^k) \tag{4}$$

然后，我们使用 $\theta^{k+\tau_i}$ 来考察其对任务 j 损失的影响，方法是比较任务 j 在前后的损失变化，公式如下：

$$\Delta^k\boldsymbol{L}_{i\to j}=\boldsymbol{L}_j(\theta^k)-\boldsymbol{L}_j(\theta^{k+\tau_i}) \tag{5}$$

将公式(4)代入公式(5)，并对 $\boldsymbol{L}_j(\theta^{k+\tau_i})$进行一阶泰勒级数展开，得到：

$$\Delta^k\boldsymbol{L}_{i\to j}=\boldsymbol{L}_j(\theta^k)-\boldsymbol{L}_j(\theta^k-\gamma_i\boldsymbol{g}_i(\theta^k))\approx\gamma_i\boldsymbol{g}_i^T(\theta^k)\boldsymbol{g}_j(\theta^k) \tag{6}$$

注意，$\boldsymbol{g}_i^T(\theta^k)\boldsymbol{g}_j(\theta^k)$表明，内积值越大，迁移效果越好，这或许可以解释图 1(b) 中的观察结果：余弦相似度在训练初期呈上升趋势。公式 6 还表明，提高梯度内积可以增强迁移效果。这与先前的工作 [12, 15, 18], 相对一致，也验证了我们的量化方法。由于$\Delta^k\boldsymbol{L}_{i\to j}$反映了任务 i对 j的影响，因此可以将其视为任务间（$i\to j$）迁移的量化指标。接下来，我们将介绍如何利用这一量化方法进行协同优化，以协调通用知识和特定知识。

3.2 协调梯度调制

3.2.1 最大化任务间迁移。我们最大化任务间迁移 $\Delta^k\boldsymbol{L}_{i\to j}$（公式 5），以推导优化梯度。在此，我们固定任务 i，因为我们仅考虑任务i 对 j 的影响（即将 $\nabla_\theta\theta^{k+\tau_i}$ 近似为 1。0），并得到：

$$\nabla_\theta\Delta^k\boldsymbol{L}_{i\to j}=\boldsymbol{g}_j(\theta^k)-\boldsymbol{g}_j(\theta^{k+\tau_i}) \tag{7}$$

对 $\boldsymbol{g}_j(\theta^{k+\tau_i})$ 进行一阶泰勒级数展开，得到：

$$\begin{aligned}\nabla_\theta \Delta^k \boldsymbol{L}_{i\to j} &= \boldsymbol{g}_j(\theta^k) - \left(\boldsymbol{g}_j(\theta^k) - \gamma_i \boldsymbol{H}_j(\theta^k)\boldsymbol{g}_i(\theta^k)\right) \\ &= \gamma_i \boldsymbol{H}_j(\theta^k)\boldsymbol{g}_i(\theta^k)\end{aligned} \tag{8}$$

其中， $\boldsymbol{H}_j(\theta^k)$ 是 $\boldsymbol{L}_j(\theta^k)$ 的海森矩阵。

3.2.2 通用知识与特定知识协调. 这里，$-\nabla_\theta \Delta \boldsymbol{L}^k_{i\to j}$ 可以表示用于最大化从任务 i 向 j 迁移的梯度。接着，我们将其并入原始梯度 $\boldsymbol{g}_j(\theta^k)$（见公式 9），从而既最大化从任务 i 向 j 的迁移（公式 9 中的第二项），又最小化任务 j 的单独损失（公式 9 中的第一项），如图 2(c) 所示：

$$\widehat{\boldsymbol{g}}_j(\theta^k) = \boldsymbol{g}_j(\theta^k) - \gamma_i \boldsymbol{H}_j(\theta^k)\boldsymbol{g}_i(\theta^k) \tag{9}$$

其中，超参数 γ_i 可视为最大化迁移与最小化特定损失之间的平衡程度。事实上，如第 3.4.2 节所述，CoGrad 对 γ_i 不敏感。我们可以用相同的方式得到任务 i 的对应形式。基于此，我们提出一个更通用的形式：

$$\widehat{\boldsymbol{g}}_i(\theta^k) = \boldsymbol{g}_i(\theta^k) - \textstyle\sum_{j\neq i, j\in\mathcal{T}} \gamma_j \boldsymbol{H}_i(\theta^k)\boldsymbol{g}_j(\theta^k), \forall i \in \mathcal{T} \tag{10}$$

3.3 最大化迁移的近似

由于公式 8 中的海森矩阵 $\boldsymbol{H}_j(\theta^k)$，计算用于最大化迁移的梯度 $\nabla_\theta \Delta^k \boldsymbol{L}_{i\to j}$ 在存储和计算方面都代价过高。作为替代，我们在 [19] 中引入一种适用的近似器（见公式 11），其近似精度已通过理论和实证得到保证。

$$\boldsymbol{H}_j(\theta^k)\boldsymbol{g}_i(\theta^k) = \lambda^k \boldsymbol{g}_j(\theta^k) \odot \boldsymbol{g}_j(\theta^k) \odot \boldsymbol{g}_i(\theta^k) \tag{11}$$

其中，$\odot$ 是哈达玛积（即逐元素乘积），λ^k 是一个超参数。根据 [19],，我们在方法中将 λ^k 固定为 1.0。将公式 11 代入公式 10，便可得到 CoGrad 简单高效的训练流程，如算法 1 所述。

3.4 讨论

3.4.1 与元学习的联系。 Sequential Reptile (Seq.Rept) [7] 在平衡一般信息与特定信息方面与我们的方法相似。Seq.Rept 基于内循环轨迹，采用元学习并按顺序处理所有任务，从而对齐任务梯度。非正式地说，在有两个任务和两个中间步骤的条件下，Seq.Rept 与 CoGrad 非常相似。这些联系也有助于理解我们的方法。然而，面对更多任务时，由于存在内循环和外循环，Seq.Rept 的计算开销会变得很大；而 CoGrad 仍然高效，计算量几乎不会增加。此外，明确量化并最大化任务间迁移也符合商业实际。因此，CoGrad 在工业应用中更具实用性。

3.4.2 超参数选择。 我们的方法在公式 10 中引入了新的超参数 $\{\gamma_j\}_{j\in\mathcal{T}}$，用于控制优化迭代期间最大化任务间迁移的力度。根据理论推导，我们可以将 γ_j 视为用于更新任务 j 参数的虚拟学习率。因此，选择一个足够小的值即可。此外，我们通过实证还发现，性能对这些超参数并不敏感，它们对收敛速度的影响也很小。

表1：两个数据集的统计信息。

数据集	# 曝光	# 点击	# 浏览	# 购买
Ali-CCP	8400万	340万	–	18 k
Ecomm	16亿	2500万	1000 万。	–

4 实验

4.1 实验设置

4.1.1 数据集。 我们在两个工业数据集上开展离线实验。第一个是名为**Ali-CCP** [11], 的公开数据集，其中包含点击和购买行为。第二个数据集来自我们的大规模电子商务广告平台 Ecomm，其中包含另一种转化行为（即浏览商品详情页），名为**Ecomm**。我们按时间戳以 4:1:1 的比例将每个数据集划分为训练集/验证集/测试集。表1列出了统计信息。

4.1.2 对比方法。 我们使用 Shared Bottom [1] 和 MMOE [10] 这两种基础多任务学习架构，将 CoGrad 与此前最先进的梯度调制技术进行比较：(1) PCGrad [18] 基于方向一致性，积极地投影任务梯度。(2) MetaBalance [5]（简称 MBalance）自适应地均衡梯度幅值，避免模型被某些任务主导。(3) CAGrad [9] 在平均梯度附近进行搜索，以最大化表现最差任务的性能。(4) Seq.Rept [7] 利用包含顺序任务的内循环轨迹，以最大化梯度对齐。

我们使用 AUC（Ali-CCP）和用户分组AUC（ [4] 中用于 Ecomm 的一项工业指标）作为评估指标。为确保公平比较，所有方法（包括 STL）的特征嵌入维度均固定为 8，每个网络包含三个隐藏层：Ecomm 为 {512,256,128}，Ali-CCP 为 {128,64,32}。多任务学习模型的前两层为共享层，最后一层为专属层。对于 MMOE，我们设置了三个专家。我们使用 Adam [6] 优化器，批量大小分别设为 1024（Ecomm）和 256（Ali-CCP），学习率分别设为 0.001（Ecomm）和 0.005（Ali-CCP）。损失缩放权重通过一种基于先验统计交叉熵的启发式方法设定（不包括 MBalance）。基线方法中的其他超参数均依据原始研究进行了仔细调优。我们的超参数 $\{\gamma_{ctr},\ \gamma_{cvr}\}$ 设为 {0.003，0.001}（Ecomm）和 {0.01，0.005}（Ali-CCP）。

4.2 结果与讨论

4.2.1 主要结果。 表2展示了所有方法在生产数据集和公开数据集上的结果。对于这两个数据集，所有多任务学习方法在转化率任务上的表现都优于单一深度神经网络，但在点击率任务上的表现略差。这是因为数据稀疏性问题 [11] 使得单一转化率模型难以拟合，而点击率模型则能凭借充足的数据进行有效学习。与基于共享底层的方法相比，基于MMOE的方法在点击率任务上表现略好，但在转化率任务上表现较差。这与我们的观点一致：显式的点击率专属专家有助于编码更多点击率专属知识，但会占用用于编码转化率专属知识或通用知识的容量预算。

聚焦于最差任务的 CAGrad，以及考虑单个任务的 Seq.Rept，在防止负迁移方面优于以往方法。这表明，通用知识和特定知识在共享模块中同等重要。得益于

表2：两个数据集上的平均结果（运行五次），粗体和下划线分别表示最佳和次佳结果。“*”表示在 $p < 0.05$ 显著性水平上的提升。 Δ 表示相对于单一深度神经网络的性能增益

方法	Ecomm （生产环境）		Ali-CCP （公开）	
	$GAUC_{ctr}(\Delta)$	$GAUC_{cvr}(\Delta)$	$AUC_{ctr}(\Delta)$	$AUC_{cvr}(\Delta)$
单一深度神经网络	**76.24**	78.11	63.97	65.85
共享底层	76.01 (↓0.23)	78.97 (↑0.86)	63.81 (↓0.16)	66.71 (↑0.86)
+PCGrad	76.07 (↓0.17)	79.21 (↑1.10)	63.84 (↓0.13)	67.25 (↑1.40)
+MBalance	76.04 (↓0.20)	78.92 (↑0.81)	63.78 (↓0.19)	66.99 (↑1.14)
+CAGrad	76.08 (↓0.16)	79.34 (↑1.13)	63.80 (↓0.17)	67.38 (↑1.53)
+Seq.Rept	76.10 (↓0.14)	79.50 (↑1.39)	**63.98 (↑0.01)**	67.56 (↑1.71)
+CoGrad	76.17 (↓0.07)	**79.61 (↑1.50)** *	63.97 (↓0.00)	**67.78 (↑1.93)** *
MMOE	76.05 (↓0.19)	78.93 (↑0.82)	63.84 (↓0.13)	66.49 (↑0.64)
+PCGrad	76.07 (↓0.17)	79.13 (↑1.02)	63.89 (↓0.08)	66.61 (↑0.76)
+MBalance	76.03 (↓0.21)	78.86 (↑0.75)	63.85 (↓0.12)	66.39 (↑0.54)
+CAGrad	76.10 (↓0.14)	79.33 (↑1.22)	63.91 (↓0.06)	66.85 (↑1.00)
+Seq.Rept	76.11 (↓0.13)	**79.45 (↑1.34)**	63.93 (↓0.04)	67.15 (↑1.30)
+CoGrad	76.21 (↓0.03)	79.42 (↑1.31)	**63.96 (↓0.01)**	**67.32 (↑1.47)** *

这种协调通用知识和特定知识的迁移驱动技术，CoGrad 在转化率方面显著优于所有基线方法，同时，其点击率表现也与 STL 相当。

值得注意的是，与 Seq.Rept 相比，CoGrad 在电商数据集上的提升似乎不大。首先，在工业场景中实现 0.1 的 AUC 增益已相当可观 [11]。其次，如第 3.4.1 节所述，CoGrad 在双任务设置下与 Seq.Reqt 相似。不过，CoGrad 的计算效率更高且易于实现，而这两点对于大规模应用至关重要。

4.3 进一步分析

4.3.1 容量大小的鲁棒性。 我们将第一个隐藏层的大小加倍，以考察共享底层对容量大小的鲁棒性。在基线模型中，扩大容量往往能够缓解点击率上的负迁移，但代价是转化率表现下降（见表3）。然而，CoGrad 在提升点击率表现的同时，不影响转化率。这验证了 CoGrad 具有很强的鲁棒性，能够自适应地最大化知识迁移。

4.3.2 通用知识和特定知识协调的可视化。我们借助预训练-微调来研究通用知识和特定知识协调的能力

范式。我们冻结训练好的模型中的全部共享参数，并将最后一个共享层连接到一个可训练的线性层。随后，我们针对每个任务微调这个线性层，以学习最后一个共享层的隐藏单元权重（128维）。其背后的直觉是，如果两个任务都认为某个单元很重要，那么它们归一化后的权重应该接近，反之亦然。最后，我们计算权重（归一化后）的差值（即点击率权重减去转化率权重），得到如图3(a)所示的平滑分布。当差值接近零时，知识被认为具有更强的通用性；差值离零越远，知识则越具体。

首先，这一可视化结果表明，共享模块确实包含通用知识和特定知识，这与 [14] 一致。其次，零值右侧的区域（即对点击率任务重要的知识）略小于左侧区域。这解释了为何转化率任务从

表3：Ecomm 上不同隐藏层大小的影响。 Δ表示相对于大小 512×的性能增益。

方法	$GAUC_{ctr}$		$GAUC_{cvr}$	
	大小 512×	大小 1024× (Δ)	大小 512×	大小 1024×(Δ)
共享底层	76.01	76.05 (↑0.04)	78.97	78.81 (↓0.16)
+ CoGrad	76.17	**76.23 (↑0.06)**	79.61	**79.63 (↑0.02)**

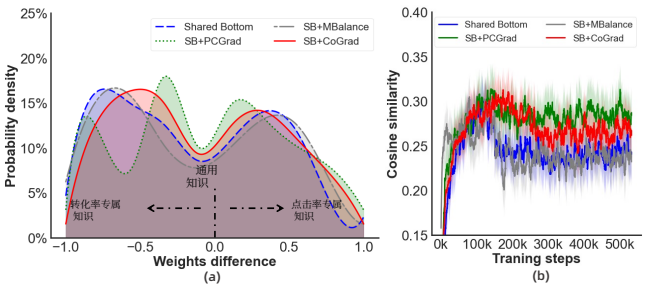


图3：(a) 最后一个共享层中隐藏单元的重要性权重差异分布。(b) CoGrad 与基线方法在梯度相似度上的比较。

多任务学习中显著获益，而点击率任务却受到负面影响。第三，基于方向的方法（PCGrad）确实通过强制实现梯度对齐来增加通用知识，但也阻碍了部分特定知识的编码。这支持了我们在引言中阐述的观点：强制实现梯度对齐可能会挤占特定知识。最终，与基础多任务学习相比，CoGrad 在提升通用知识的同时，也保留了足够的特定知识。这验证了 CoGrad 能够有效协调通用知识和特定知识。

4.3.3 对梯度相似度的影响。 图3(b)展示了所有方法在训练过程中的梯度余弦相似度。与基础模型相比，基于幅值的方法（MBalance）未能提升相似度，而基于方向的方法（PCGrad）在提升梯度相似度方面最为有效。结合表2中的主要结果，我们可以确认余弦相似度确实与性能相关。然而，两者之间的相关性并非完全正向。CoGrad 通过迁移驱动技术隐式对齐梯度，从而在一定程度上提高余弦相似度。结合图3(a)，我们可以确认，CoGrad 能够自动将通用知识的编码增强到适当程度，而不会过度编码。

4.4 在线A/B测试

我们在广告系统上进行了为期15天的在线实验，并与一个训练充分的多任务学习模型进行对比。我们使用的指标包括 CTR =#点击#曝光、CPC =广告主成本#点击、 CVR =#浏览#点击，以及CPA = 广告主成本#浏览。较高的点击率/转化率和较低的每次点击成本/每次转化成本表示性能更佳。在大规模工业应用中，提升1%已是显著改进。CoGrad分别将点击率和转化率提高了**2.03%** 和**4.75%**，并将每次点击成本和每次转化成本分别降低了**1.64%** 和**5.23%**，验证了其在工业领域的有效性。

5 结论

我们提出 CoGrad，这是一种迁移驱动方法，可通过协调梯度修改自动最大化任务间迁移

。它对迁移进行量化，并通过最大化这一量化指标、同时最小化任务特定损失来进行优化，从而协调共享模块中的通用知识与特定知识，提升整体性能。离线和在线实验均验证了其有效性。

参考文献

[1] Rich Caruana。1997。多任务学习。机器学习 28, 1 (1997), 41–75。[2] Zhao Chen、Vijay Badrinarayanan、Chen-Yu Lee 和 Andrew Rabinovich。2018。GradNorm：深度多任务网络中用于自适应损失平衡的梯度归一化。收录于国际机器学习大会。PMLR，794–803。[3] Christopher Fifty、Ehsan Amid、Zhe Zhao、Tianhe Yu、Rohan Anil 和 Chelsea Finn。2021。高效识别多任务学习中的任务分组。收录于神经信息处理系统。[4] Ruining He 和 Julian McAuley。2016。起起落落：利用单类协同过滤对时尚趋势的视觉演变进行建模。收录于第*25*届国际万维网大会论文集。507–517。[5] Yun He、Xue Feng、Cheng Cheng、Geng Ji、Yunsong Guo 和 James Caverlee。2022。MetaBalance：通过调整辅助任务的梯度幅值来改进多任务推荐。收录于*ACM Web Conference 2022*论文集。2205–2215。[6] Diederik P Kingma 和 Jimmy Ba。2014。Adam：一种随机优化方法。*arXiv*预印本 *arXiv:1412.6980*（2014）。[7] Seanie Lee、Hae Beom Lee、Juho Lee 和 Sung Ju Hwang。2021。Sequential Reptile：面向多语言学习的任务间梯度对齐。*arXiv*预印本 *arXiv:2110.02600*（2021）。[8] Zihan Lin、Xuanhua Yang、Shaoguo Liu、Xiaoyu Peng、Wayne Xin Zhao、Liang Wang 和 Bo Zheng。2022。用于CTR与CVR联合估计的个性化任务间对比学习。*arXiv*预印本 *arXiv:2208.13442*（2022）。[9] Bo Liu、Xingchao Liu、Xiaojie Jin、Peter Stone 和 Qiang Liu。2021。面向多任务学习的冲突规避梯度下降。神经信息处理系统进展 34 (2021), 18878–18890。[10] Jiaqi Ma、Zhe Zhao、Xinyang Yi、Jilin Chen、Lichan Hong 和 Ed H Chi。2018。利用多门混合专家模型对多任务学习中的任务关系进行建模。收录于第*24*届*ACM SIGKDD*知识发现与数据挖掘国际会议论文集。1930–1939。[11] Xiao Ma、Liqin Zhao、Guan Huang、Zhi Wang、Zelin Hu、Xiaoqiang Zhu 和 Kun Gai。2018。全空间多任务模型：一种有效的点击后转化率估计方法。收录于第*41*届*ACM SIGIR*信息检索研究与开发国际会议。1137–1140。[12] Sebastian Ruder。2017。深度神经网络多任务学习概述。*arXiv*预印本 *arXiv:1706.05098*（2017）。[13] Ozan Sener 和 Vladlen Koltun。2018。作为多目标优化的多任务学习。神经信息处理系统进展31（2018）。[14] Zirui Wang、Zachary C Lipton 和 Yulia Tsvetkov。2020。多语言模型中的负干扰：研究发现与元学习应对方法。*arXiv*预印本 *arXiv:2010.03017*（2020）。[15] Zirui Wang、Yulia Tsvetkov、Orhan Firat 和 Yuan Cao。2020。梯度疫苗：研究并改进大规模多语言模型中的多任务优化。*arXiv*预印本 *arXiv:2010.05874*（2020）。[16] Penghui Wei、Weimin Zhang、Zixuan Xu、Shaoguo Liu、Kuang-chih Lee 和 Bo Zheng。2021。AutoHERI：用于点击后转化率估计的自动化层次表示整合。收录于第*30*届*ACM*信息与知识管理国际会议论文集。3528–3532。[17] Hong Wen、Jing Zhang、Fuyu Lv、Wentian Bao、Tianyi Wang 和 Zulong Chen。2021。通过多任务学习对微观与宏观行为进行层次化建模，以实现转化率预测。收录于第*44*届*ACM SIGIR*信息检索研究与开发国际会议论文集。2187–2191。[18] Tianhe Yu、Saurabh Kumar、Abhishek Gupta、Sergey Levine、Karol Hausman 和 Chelsea Finn。2020。面向多任务学习的梯度手术。神经信息处理系统进展 33（2020），5824–5836。[19] Shuxin Zheng、Qi Meng、Taifeng Wang、Wei Chen、Nenghai Yu、Zhi-Ming Ma 和 Tie-Yan Liu。2017。带延迟补偿的异步随机梯度下降。收录于国际机器学习大会。PMLR，4120–4129。