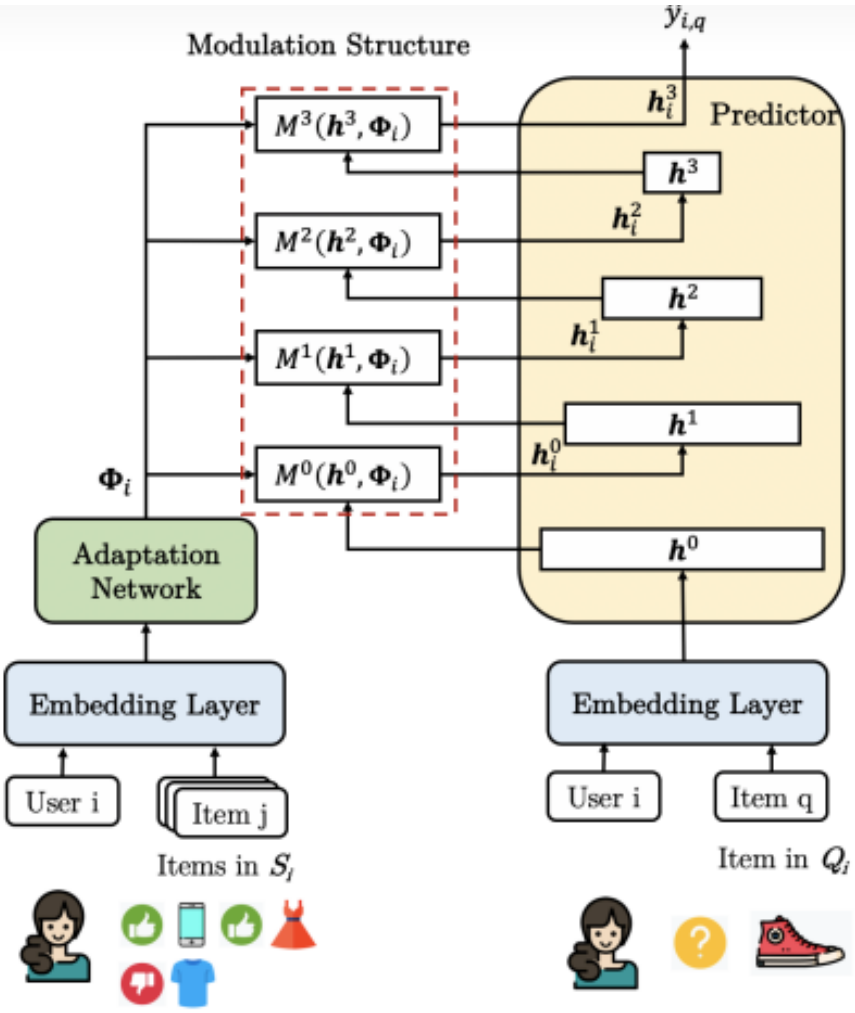
知乎




# 百度2023-ColdNAS：突破用户冷启动难题，论文解析创新架构搜索框架！

SmartMindAI
专注搜索、广告、推荐、大模型和人工智能最新技术，欢迎关注我
已关注



## Introduction

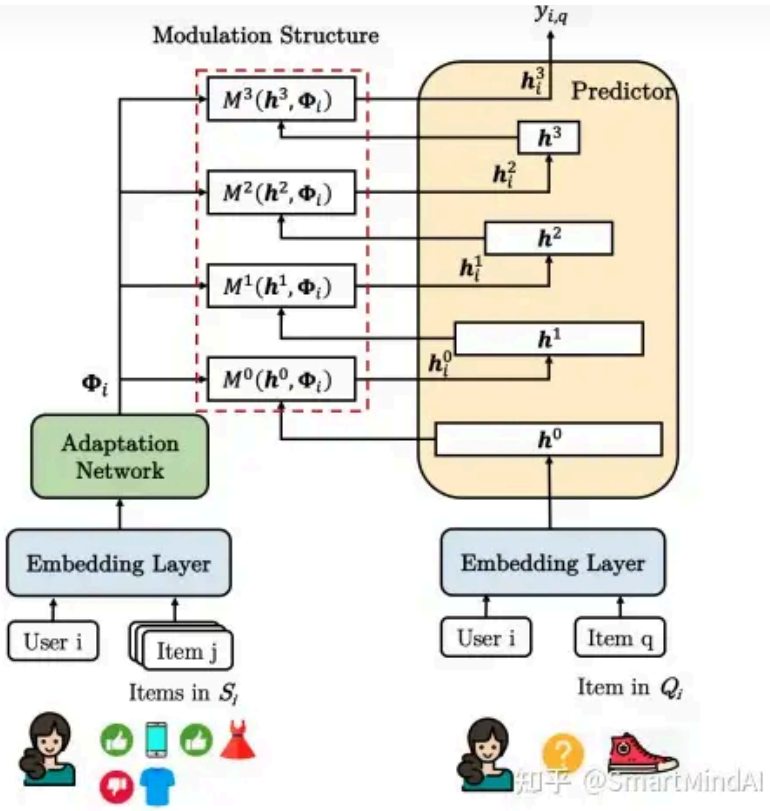
推荐系统旨在为用户提供相关的内容建议，如电影和书籍。尽管现有的在线推荐系统为大量用户提供了便捷的购物体验，但冷启动推荐问题仍然是其中的一个挑战。

这个问题可以被建模为一种少样本学习问题，其目标是通过少量训练样本快速泛化到新任务。许多论文采用了与模型无关元学习(MAML)策略，它从一组任务中学习良好的初始化参数，并通过在有限数量的带标签样本上进行几步骤梯度下降更新来适应新任务。然而，这种策略需要专业知识来调整优化过程以避免过拟合，而且耗时可能很长。

- 本文提出了一个名为ColdNAS的架构搜索框架，用于解决用户的冷启动问题。我们使用超网络将每个用户的历史交互映射到用户特定的参数，然后使用这些参数来调整预测器，并将如何调整和在哪里调整表述为一个NAS问题。
- 本文设计了一个架构搜索结构的搜索空间，该空间不仅可以涵盖现有的基于现有的用户冷启动模型，而且可以包含更丰富的表达结构。由于搜索空间可能很大，因此需要进行搜索空间转换，将原始空间转换为一个等效但更小的空间。我们提供了理论分析以验证其正确性。在转换后的空间上，我们可以使用可微架构搜索算法进行高效和稳健的搜索。
- 本文在用户冷启动问题的基准数据集进行了广泛的实验，并观察到模型始终获得了最先进的表现。我们还验证了搜索空间和算法的设计考虑，证明了ColdNAS的强大和合理性。

## Proposed Method

知乎




### Problem Formulation

我们设用户集为$\mathcal{U} = \{u_i\}$，物品集为$\mathcal{V} = \{v_j\}$，用户$u_i$对物品$v_j$的评分为$y_{i,j}$. 用户冷启动推荐问题的目标是为只评分过少数物品的用户$u_i$提供个性化推荐。这个问题可以被视为少样本学习问题，我们需要学习一个模型，该模型能从一组训练用户冷启动任务$\mathcal{T}^{\text{train}}$中学习，并推广到新任务。每个任务$T_i$对应一个用户$u_i$，并有一个包含现有交互历史的支持集$\mathcal{S}_i = \{(v_j, y_{i,j})\}_{j=1}^{N}$和一个查询集$\mathcal{Q}_i = \{(v_j, y_{i,j})\}_{j=1}^{M}$，其中$M$表示查询集中的交互数量。

### Search Space

我们的用户冷启动方案包括嵌入层、适应网络和预测器。嵌入层将用户行为转化为向量表示，适应网络关联用户历史行为与目标行为，预测器预测用户的目标行为。

- 嵌入层E使用参数$\boldsymbol{\theta}_E$将特征嵌入到向量中，即

$$(\boldsymbol{u}_i, \boldsymbol{v}_j) = E(u_i, v_j; \boldsymbol{\theta}_E)$$

- 具有参数 的适应网络 接受特定用户 的支持集S 作为输入，并生成用户特定的自适应参数，即

$$\boldsymbol{\Phi}_i = \{\boldsymbol{\phi}_i^k\}_{k=1}^{C} = A(\mathcal{S}_i; \boldsymbol{\theta}_A),$$

- 预测器P使用参数$\boldsymbol{\theta}_P$，接收用户特定参数$\boldsymbol{\phi}_i$和$\boldsymbol{v}_q \in \mathcal{Q}_i$作为输入

$$\hat{y}_{i,j} = P((\boldsymbol{u}_i, \boldsymbol{v}_q), \boldsymbol{\Phi}_i; \boldsymbol{\theta}_P).$$

我们引入了一个额外的自适应网络来处理冷启动用户。对于每个 $\boldsymbol{u_i}$，我们将支持集 $\mathcal{S}_i$ 映射到用户特定参数 $\boldsymbol{\phi_i}$。然后，我们使用目标物品 $\boldsymbol{v}_q \in \mathcal{Q}_i$ 的特征、用户特征 $\boldsymbol{u_i}$ 和 $\boldsymbol{\phi_i}$ 进行预测。

最近，TaNP 提出了一种直接将式 (1) 中的 $\boldsymbol{M^l}$ 采用 FiLM 的形式应用于所有的 $\boldsymbol{L}$ 个 MLP 层的方法。这种方法可以控制每个用户在第 $\boldsymbol{l}$ 层中对 $\boldsymbol{h^l}$ 的个性化程度，同时可学习权重 $\boldsymbol{W^l}$ 和 $\boldsymbol{b^l}$ 也在第 $\boldsymbol{l}$ 层中进行更新。

$$\boldsymbol{h}_i^l = \boldsymbol{h}^l \odot \boldsymbol{\phi}_i^1 + \boldsymbol{\phi}_i^2.$$

该空间的大小为 $6^{C\times L}$。更大的 $C$ 会导致更大的搜索空间，虽然有更高的潜力包含适当的调整函数，但更难以有效地进行搜索。

### Search Strategy

我们的目标是设计一个高效的搜索算法✦，用于在大型搜索空间中进行搜索。为了解决这个问题，我们提出将原始搜索空间转换为等价的但更小的空间。在这个新的空间中，我们设计了一个超级网络结构，以进行高效且稳健的可微搜索。

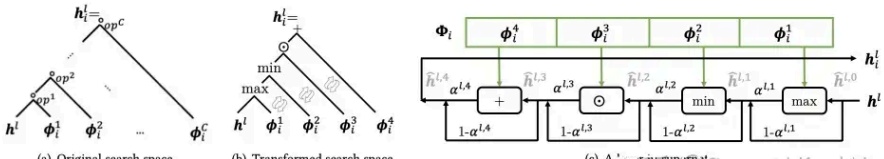





**Figure 2: Illustration of our proposition 3.1, and the structure of the supernet to search on the reduced space.**

### Search Space Transformation

尽管搜索空间可能很大，但我们可以将其转换为大小为$2^{4\times L}$的等效空间，该空间与C无关，如命题证明所示（证明见附录）。

假设自适应网络 $A$ 足够表达。任何形式为 的 $M^l$，其中 $\circ_{\mathrm{op}^k} \in \mathcal{O}$，$C$ 是任何非负整数，和 $\phi_i^k \in \Phi_i = A(\mathcal{S}_i, \theta_A)$，可以被 表示为 $h_i^l = \min(\max(h^l, \hat{\phi}_i^1), \hat{\phi}_i^2) \odot \hat{\phi}_i^3 + \hat{\phi}_i^4$，并且上述四个操作是 排列不变的。

为了更好地理解命题 ，让我们检查两个例子。

1. $\min(\max(h^l, \phi_i^1) + \phi_i^2 - \phi_i^3, \phi_i^4) \odot \phi_i^5$

其中 $\hat{\phi}_i^1 = \phi_i^1$, $\hat{\phi}_i^2 = \phi_i^4 - \phi_i^2 + \phi_i^3$, $\hat{\phi}_i^3 = \phi_i^5$, $\hat{\phi}_i^4 = (\phi_i^2 - \phi_i^3) \odot \phi_i^5$;

1. $\max(\min(h^l + \phi_i^1, \phi_i^2), \phi_i^3) \odot \phi_i^4$

其中 $\hat{\phi}_i^1 = \phi_i^3 - \phi_i^1$, $\hat{\phi}_i^2 = \phi_i^2 - \phi_i^1$, $\hat{\phi}_i^3 = \phi_i^4$, $\hat{\phi}_i^4 = \phi_i^1 \odot \phi_i^4$。

原始空间：$X \subseteq R^n$ 变换后的空间：$Y \subseteq R^m$ 其中：$m << n$ 举例来说，对于一个具有100个特征的问题，我们可以通过深度网络将其映射到一个只有10个特征的空间中。这样，我们可以大大降低问题的维度，同时保留其主要信息。

空间转换在ColdNAS中起着关键作用，表 有助于我们更好地理解命题在多大程度上帮助减少搜索空间，我们取层数 $L = 4$，比率计算为 $\frac{\text{original space}}{\text{transformed space}} = \frac{6^{C\times 4}}{2^{4\times 4}}$.

注意，当 $C = 1$ 时，转换不会减少空间。然而，这样的情况并无意义，因为由于调整函数的灵活性不足，其性能较差。转换后的空间通过将架构参数的数量从 $6 \times C \times L$ 减少到 $4 \times L$，实现了高效和稳健的可微搜索。同时，空间大小从 $6^{C\times L}$ 转换为 $2^{4\times L}$，对于任何 $C > 1$，这都是一种减少。

### Construction of the Supernet

对于每个 $M^l$，由于最多有 $4$ 个操作且它们是对称的，我们只需按任意顺序确定是否进行操作。我们通过引入可微参数来优化权重进行搜索。对于预测器的第 $l$ 层，我们有：

$$\hat{h}^{l,k+1} = \alpha^{l,k+1}(\hat{h}^{l,k} \circ_{\mathrm{op}^{k+1}} \phi_i^{k+1}) + (1-\alpha^{l,k+1})\hat{h}^{l,k},$$

衡量$M^l$中操作$\circ_{\mathrm{op}^{k+1}}$的权重，$k \in \{0,1,2,3\}$，并且$\{\circ_{\mathrm{op}^{k+1}}\}_{k=0}^{3} = \{\max, \min, \odot, +\}$。

为简化记号，我们设$\hat{h}^{l,0} = h^l$, $h_i^l = \hat{h}^{l,4}$。

知乎

Algorithm 总结了完整的算法。首先，我们优化超网并做出选择以确定结构，然后使用确定的结构重建模型并重新训练以进行推理。

**Algorithm 1** Training procedure of ColdNAS.

**Input:** Learning rate $\beta$, number of operations to keep $K$.

Construct the supernet by (8) and randomly initialize all parameters $\Theta = \{\{\alpha^{l,k}\}_{k=1,l=0}^{4,L-1}, \theta_E, \theta_A, \theta_P\}$.
**while** Not converge **do**
  **for** Every $T_i \in \mathcal{T}^{\text{train}}$ **do**
    Calculate $\Phi_i$ by (1).
    Calculate $\hat{y}_{i,j}$ for every $v_j$ in $\mathcal{Q}_i$ by (2).
    Calculate loss $\mathcal{L}_i$ by (10).
  **end for**
  $\mathcal{L}^{\text{train}} = \frac{1}{|\mathcal{T}^{\text{train}}|} \sum_{i=1}^{|\mathcal{T}^{\text{train}}|} \mathcal{L}_i$
  Update all parameters $\Theta \leftarrow \Theta - \beta \nabla_\Theta \mathcal{L}^{\text{train}}$.
**end while**
Determine the modulation structure by keeping operations corresponding to Top-$K$ $\alpha^{l,k}$ and remove the others.
Construct the model with determined modulation structure and randomly initialize all parameters $\Theta = \{\theta_E, \theta_A, \theta_P\}$.
Train the model in the same way as Step 2 ~ 10.
**Return:** The trained model.



学习率 $\beta$，保留的操作数 $K$。通过构建超网并随机初始化所有参数

$$\Theta = \{\{\alpha^{l,k}\}_{k=1,l=0}^{4,L-1}, \theta_E, \theta_A, \theta_P\}.$$

计算损失 $\mathcal{L}_i$。 $\mathcal{L}^{\text{train}} = \frac{1}{|\mathcal{T}^{\text{train}}|} \sum_{i=1}^{|\mathcal{T}^{\text{train}}|} \mathcal{L}_i$ 更新所有参数 $\Theta \leftarrow \Theta - \beta \nabla_\Theta \mathcal{L}^{\text{train}}$。

通过保留对应于Top-$K$ $\alpha^{l,k}$的操作并移除其他操作来确定调整结构。构建具有确定调整结构的模型并随机初始化所有 参数 $\Theta = \{\theta_E, \theta_A, \theta_P\}$。 以与步骤 $2 \sim 10$ 相同的方式训练模型。

空间转换带来了巨大的优化，使得双层目标函数⁺得以显著降低。传统的可微分架构搜索通过上层变量α和下层变量θ来优化双层目标函数。

$$\min_{\alpha} \mathcal{L}^{\text{val}}(\theta^*(\alpha), \alpha), \ \text{s.t.} \ \theta^*(\alpha) = \operatorname{argmin}_{\theta} \mathcal{L}^{\text{train}}(\theta, \alpha),$$

我们只需要训练超网络来优化训练集上的损失函数⁺ $\mathcal{L}^{\text{train}}$，并通过端到端的离散训练实现目标。对于训练集中的每个任务 $T_i \in \mathcal{T}^{\text{train}}$，我们首先将 $\mathcal{S}_i$输入到自适应网络 $A$，通过等式(eq:adapara)生成 $\Phi_i$ 然后对于每个项目 $v_j \in \mathcal{Q}_i$，我们通过等式(eq:forward)将 $(u_i, v_j)$ 和 $\Phi_i$ 输入到 预测器 $P$ 进行预测。最后，我们使用预测值 $\hat{y}_{i,j}$ 和 真实标签 $y_{i,j}$ 之间的均方误差⁺（MSE）作为损失函数。

$$\mathcal{L}_i = \frac{1}{M} \sum_{j=1}^{M} (y_{i,j} - \hat{y}_{i,j})^2,$$

我们通过梯度下降法⁺更新所有参数。当超网收敛后，我们选取$\{\alpha^{l,k+1}\}_{k=1,l=0}^{4,L-1}$中的前$K$个最大值所对应的操作，联合确定所有的$M^l$。随后，我们重新训练模型以构建最终的用户冷启动模型。在推理阶段，对于一组新的任务$\mathcal{T}^{\text{test}}$，我们将其中的$T_i$以及整个$\mathcal{S}_i$和$(u_i, v_j)$作为输入，获取每个物品$v_j \in \mathcal{Q}_i$的预测值$\hat{y}_{i,j}$。

**Discussion**

知乎

架构等价性，如ColdNAS。如果找到一组等价架构，评估其中任何一个架构就足够了。ColdNAS的搜索空间专为冷启动问题中的调整结构设计，我们已证明了原始空间和转换空间的等价性，从而大大减小了空间大小。

### Experiments

我们在三个基准数据集上进行了实验，以回答以下问题：

- RQ1: ColdNAS选择了何种调整结构？与当前最先进的冷启动模型相比，ColdNAS的表现如何?
- RQ2: ColdNAS的搜索空间和算法为我们提供了怎样的理解?
- RQ3: 超参数+对ColdNAS的影响如何?

### Datasets

- MovieLens：收集了包含1百万条用户电影评分的数据库，特征包括性别、年龄、职业、邮政编码+、发布年份、评分、类型、导演和演员。
- BookCrossing：BookCrossing社区用户对书籍评分的集合，特征包括年龄、地点、出版年份、作者和出版商。
- Last.fm：Last.fm在线系统中用户对艺术家听力的集合，特征仅包括用户和项目ID。 按照@lin2021task的说法，我们在Last.fm的查询集中生成了负样本。

**Table 4: Summary of datasets used in this paper.**

| Dataset | # User (Cold) | # Item | # Rating | # User Feat. | # Item Feat. |
|---|---|---|---|---|---|
| MovieLens | 6040 (52.3%) | 3706 | 1000209 | 4 | 5 |
| BookCrossing | 278858 (18.6%) | 271379 | 1149780 | 2 | 3 |
| Last.fm | 1872 (15.3%) | 3846 | 42346 | 1 | 1 |



数据分割 按照文献1的设置，我们将数据集分为训练集、验证集+、测试集+，比例为7：1：2。验证集用于判断超级网的收敛。训练集、验证集、测试集没有重叠的用户。

对于MovieLens和Last.fm，我们选择了交互历史长度在$40, 200$之间的用户。对于BookCrossing，我们将交互历史长度在[50,1000)之间的用户放入训练集，将交互历史长度在[2,50)之间的用户分为70%、10%和20%，分别放入训练集、验证集和测试集。 评估指标 我们使用平均绝对误差+(MAE)、均方误差(MSE)、归一化折扣累积增益nDCG3和nDCG5评估性能。

### Performance Comparison (RQ1)

我们将TheName与其他用户冷启动方法进行比较，包括：

(i)传统的深度冷启动模型DropoutNet

(ii) 基于FSL的方法如MeLU, MetaCS, MetaHIN, MAMO和TaNP。

我们还与ColdNAS的一个变体ColdNAS-Fixed进行了比较，该变体在每层都使用公式（2）中的固定FiLM函数，而不是我们搜索到的调整函数。

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| MovieLens | MSE | 100.90$_{(0.70)}$ | 95.02$_{(0.03)}$ | 95.05$_{(0.04)}$ | 91.89$_{(0.06)}$ | 90.20$_{(0.22)}$ | 89.11$_{(0.18)}$ | 91.05$_{(0.13)}$ | **87.96**$_{(0.12)}$ |
| | MAE | 85.71$_{(0.48)}$ | 77.38$_{(0.25)}$ | 77.42$_{(0.26)}$ | 75.79$_{(0.27)}$ | 75.34$_{(0.26)}$ | 74.78$_{(0.14)}$ | 75.65$_{(0.30)}$ | **74.29**$_{(0.20)}$ |
| | nDCG$_3$ | 69.21$_{(0.76)}$ | 74.43$_{(0.59)}$ | 74.46$_{(0.78)}$ | 74.69$_{(0.32)}$ | 74.95$_{(0.13)}$ | 75.60$_{(0.07)}$ | 75.11$_{(0.09)}$ | **76.16**$_{(0.03)}$ |
| | nDCG$_5$ | 68.43$_{(0.48)}$ | 73.52$_{(0.41)}$ | 73.45$_{(0.56)}$ | 73.63$_{(0.22)}$ | 73.84$_{(0.16)}$ | 74.29$_{(0.12)}$ | 73.89$_{(0.12)}$ | **74.74**$_{(0.09)}$ |
| BookCrossing | MSE | 15.38$_{(0.23)}$ | 15.15$_{(0.02)}$ | 15.20$_{(0.08)}$ | 14.76$_{(0.07)}$ | 14.82$_{(0.05)}$ | 14.75$_{(0.05)}$ | 14.44$_{(0.16)}$ | **14.15**$_{(0.08)}$ |
| | MAE | 3.75$_{(0.01)}$ | 3.68$_{(0.01)}$ | 3.66$_{(0.01)}$ | 3.50$_{(0.01)}$ | 3.51$_{(0.02)}$ | 3.48$_{(0.01)}$ | 3.49$_{(0.02)}$ | **3.40**$_{(0.01)}$ |
| | nDCG$_3$ | 77.66$_{(0.18)}$ | 77.69$_{(0.15)}$ | 77.68$_{(0.12)}$ | 77.66$_{(0.19)}$ | 77.68$_{(0.09)}$ | 77.48$_{(0.06)}$ | 77.65$_{(0.09)}$ | **77.83**$_{(0.01)}$ |
| | nDCG$_5$ | 80.87$_{(0.15)}$ | 81.10$_{(0.15)}$ | 80.97$_{(0.09)}$ | 80.95$_{(0.04)}$ | 81.01$_{(0.05)}$ | 81.16$_{(0.21)}$ | 81.12$_{(0.06)}$ | **81.32**$_{(0.10)}$ |
| Last.fm | MSE | 21.91$_{(0.38)}$ | 21.69$_{(0.34)}$ | 21.68$_{(0.12)}$ | 21.43$_{(0.23)}$ | 21.64$_{(0.10)}$ | 21.58$_{(0.20)}$ | 21.62$_{(0.16)}$ | **20.91**$_{(0.05)}$ |
| | MAE | 43.02$_{(0.52)}$ | 42.28$_{(1.21)}$ | 42.28$_{(0.76)}$ | 42.07$_{(0.49)}$ | 42.30$_{(0.28)}$ | 42.15$_{(0.56)}$ | 42.32$_{(0.34)}$ | **41.78**$_{(0.24)}$ |
| | nDCG$_3$ | 75.13$_{(0.48)}$ | 80.15$_{(2.09)}$ | 80.81$_{(0.97)}$ | 82.01$_{(0.56)}$ | 80.73$_{(0.80)}$ | 81.03$_{(0.37)}$ | 80.17$_{(0.32)}$ | **82.80**$_{(0.69)}$ |
| | nDCG$_5$ | 69.03$_{(0.31)}$ | 75.03$_{(0.68)}$ | 75.01$_{(0.64)}$ | 75.98$_{(0.33)}$ | 75.45$_{(0.29)}$ | 75.98$_{(0.41)}$ | 75.48$_{(0.21)}$ | **76.77**$_{(0.10)}$ |



结果显示，ColdNAS在所有数据集和指标上都显著优于其他方法。此外，ColdNAS相对于ColdNAS-Fixed的一致性能提升验证了搜索调整结构以适应数据集而不是使用固定结构的必要性。

**Table 3: Modulation structure with Top-4 operations searched on the three benchmark datasets respectively. We also show the modulation structure of ColdNAS-Fixed, which is the same regardless of the dataset used.**

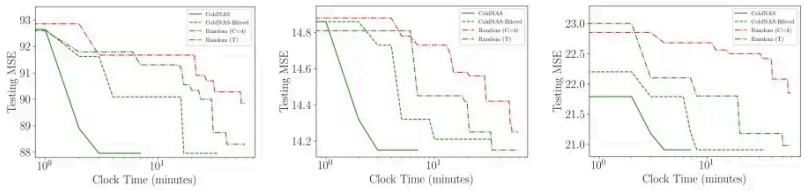
| | $M^0$ | $M^1$ | $M^2$ | $M^3$ |
|---|---|---|---|---|
| MovieLens | $\min(\max(\boldsymbol{h}^0, \boldsymbol{\phi}_i^{0,1}), \boldsymbol{\phi}_i^{0,2}) + \boldsymbol{\phi}_i^{0,3}$ | $\boldsymbol{h}^1 + \boldsymbol{\phi}_i^{1,1}$ | $\boldsymbol{h}^2$ | $\boldsymbol{h}^3$ |
| BookCrossing | $\min(\boldsymbol{h}^0, \boldsymbol{\phi}_i^{0,1})$ | $\boldsymbol{h}^1 + \boldsymbol{\phi}_i^{1,1}$ | $\boldsymbol{h}^2 \odot \boldsymbol{\phi}_i^{2,1} + \boldsymbol{\phi}_i^{2,2}$ | $\boldsymbol{h}^3$ |
| Last.fm | $\boldsymbol{h}^0 + \boldsymbol{\phi}_i^{0,1}$ | $\boldsymbol{h}^1 + \boldsymbol{\phi}_i^{1,1}$ | $\max(\boldsymbol{h}^2, \boldsymbol{\phi}_i^{2,1}) + \boldsymbol{\phi}_i^{2,2}$ | $\boldsymbol{h}^3$ |
| ColdNAS-Fixed | $\boldsymbol{h}^0 \odot \boldsymbol{\phi}_i^{0,1} + \boldsymbol{\phi}_i^{0,2}$ | $\boldsymbol{h}^1 \odot \boldsymbol{\phi}_i^{1,1} + \boldsymbol{\phi}_i^{1,2}$ | $\boldsymbol{h}^2 \odot \boldsymbol{\phi}_i^{2,1} + \boldsymbol{\phi}_i^{2,2}$ | $\boldsymbol{h}^3 \odot \boldsymbol{\phi}_i^{3,1} + \boldsymbol{\phi}_i^{3,2}$ |



## Searching in ColdNAS (RQ2)

### Choice of Search Strategy

我们在《TheName》中优化了方程（3）并与其他搜索策略进行了比较。这些策略包括在原始空间中进行随机搜索（C=4），在转换空间中进行随机搜索，以及使用双层目标优化超网的"ColdNAS-Bilevel"。我们发现，在转换空间中进行搜索比在C=4时进行随机搜索更有效，而且ColdNAS和ColdNAS-Bilevel的性能都优于随机搜索。ColdNAS和ColdNAS-Bilevel的测试MSE收敛到相似值，但ColdNAS更快，这验证了在ColdNAS中直接对所有参数使用梯度下降的有效性。



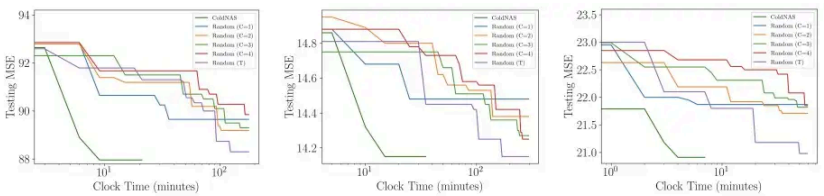

(a) MovieLens. (b) BookCrossing. (c) Last.fm.



**Figure 3: Testing MSE vs clock time of different search strategies. ColdNAS, ColdNAS-Bilevel and Random-t operate on the transformed space, while Random operates on the original space with $C = 4$ in every $M^l$.**

### Necessity of Search Space Transformation

我们已经选定了搜索算法，接下来我们需要特别考察搜索空间变换在时间和性能方面的必要性。




(a) MovieLens. (b) BookCrossing. (c) Last.fm.



**Figure 4: Testing MSE vs clock time on various search spaces: transformed space, and original spaces with different $C$.**

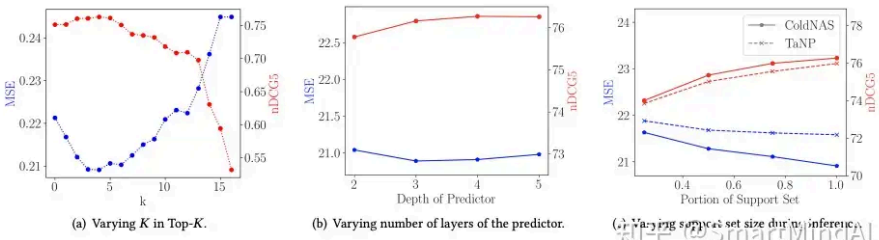
### Understanding Proposition

我们首先证明了适应网络A的表现力足够强大。如图所示，尽管A中的层数不同，但搜索到的调整操作相同，且性能差异小。因此，我们选择了2层，既具有足够的表现力，又具有较小的参数大

知乎

### Sensitivity Analysis (RQ3)

我们还探讨了预测器深度的影响。图绘制了使用具有不同L层数的预测器的效果。可以看出，在一定范围内选择不同的L对性能的影响不大，而选择L=4已经足够好，可以获得如表中所示的最佳结果。




(a) Varying $K$ in Top-$K$. (b) Varying number of layers of the predictor. (c) Varying support set size during inference.

**Figure 5: Model sensitivity analysis of ColdNAS on Last.fm.**



### Conclusion

我们提出了ColdNAS，一个解决用户冷启动推荐问题的调整框架。我们使用超网络将用户历史交互映射到用户特定参数，用于调整预测器。我们设计了一个搜索空间，能够找到更有效的调整结构。理论证明了我们可以将搜索空间转换为一个更小的空间，以高效搜索调整结构。实验结果表明，ColdNAS在基准数据集上表现最佳，且易于部署在推荐系统中。

原文《ColdNAS: Search to Modulate for User Cold-Start Recommendation》