****

**LocalMIM模型的研究与应用**

李 宁 2023600038 计算机技术（非全）

（安徽理工大学 计算机科学与工程学院 淮南 232000）

**摘 要:CVPR2023**会议中提出了一种新的掩码图像建模方法，称为局部多尺度重构（Local Multi-Scale Reconstruction, LocalMIM）。该方法旨在提高Vision Transformer（ViT）和Swin Transformer的图像理解能力。LocalMIM通过在多尺度上对局部区域进行掩码和重构，提高了模型的特征表达能力。与传统的全局重构方法相比，LocalMIM在减少计算成本的同时，提升了模型的性能。在多个基准数据集上的实验结果表明，LocalMIM在图像分类、目标检测和语义分割任务上均取得了显著的性能提升。

**关键词**：计算机视觉,深度学习,图像分类,目标检测,图像建模

# 动机与分析

近年来，从MoCo，SimCLR等对比式方法到MAE, BEiT等掩码图像建模（MIM）方法，视觉自监督表征学习的研究如火如荼。随着Vision transformer的发展，MIM方法凭借其在下游任务中更为优越的微调性能而成为关注的焦点。现有的MIM方法通常训练成本很高，而实践中我们期望其能够从海量无标记数据（如网络上随机爬取的图像）中学习通用的知识，所以高昂的预训练成本限制了其工业落地。

掩码图像建模（Masked Image Modeling，MIM）是 自监督视觉表示学习 领域的一类算法，通过对输入进行局部掩码，基于未被掩码的部分预测信号（如，归一化像素、离散标记、HOG特征、深层特征、频率特征），MIM可完成语义表示的学习，其受益于自然语言处理中的掩蔽语言建模（Masked Language Modeling，MLM）和ViT的发展。

MIM方法的计算量在于编码器和解码器，由于解码器可以很小，已有的加速预训练的方法都通过降低编码器的计算量来加速编码过程，可分为：

（1）编码器只处理可见patch，如MAE和GreenMIM；

（2）降低输入图像分辨率来减少总的patch，如LoMaR，UM-MAE和FastMIM。

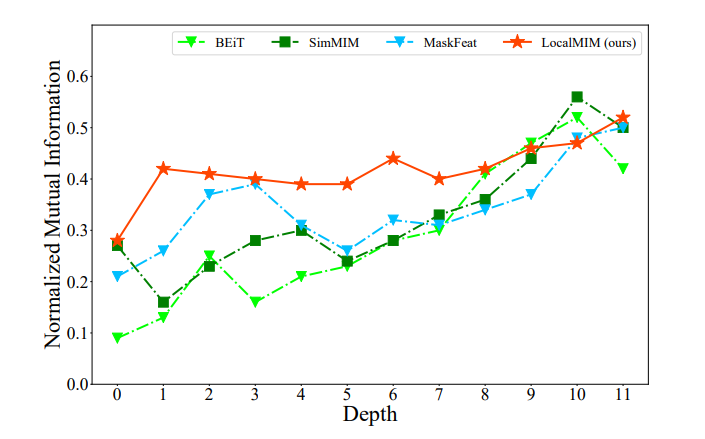
不同于已有的思路，我们转换视角，从表征学习过程本身来深入思考现有方法的不足。当我们对输入图像做片段划分并线性映射之后，由于映射过程的随机性，所获得的patch表征已经基本丢失patch之间的语义关联。

Vision Transformer（ViT）作为一种基于Transformer架构的图像分类模型，将图像分割成一系列的图像块（或称为图像补丁），然后使用 Transformer 的自注意力机制对这些图像块进行建模。ViT 的出现引发了对使用 ViT 作为主干网络的强大自监督学习框架的研究兴趣[2]。

Vision Transformer中的注意力机制通过后续patch之间的交互来学习出这种语义关联并获得比像素空间更优的表征空间。值得注意的是，处理patch之间关联的计算复杂度是O(N^2)的，所以模型想要较好地学到这种语义关联并非易事。现有全部MIM方法只在顶层引入重构任务，使得较低层无法获得直接的指导，从而只能通过缓慢的学习过程来学习patch表征及语义关联，拖累了整体的表征学习过程。尤其对于一些金字塔型主干网络，其较低层往往有着远多于顶层的patch（如Swin-224中3136（最底层）vs 49（顶层））。另一方面，实际上较低层在表征学习中扮演关键角色：

（1）良好学习的较低层可以将知识传递给较高层以促进其学习；

（2）在下游任务微调时，较高层通常快速适应到新任务中，而较低层变化较慢，需要在预训练时就得到充分学习（无论是NLP还是CV下游任务中学习率layer decay通常效果最佳）。为了更直观地展现模型不同层对patch间语义关联的学习程度，我们考察了不同层的query patch与key patch之间的标准化互信息（Normalized Mutual Information，NMI[3]，常用在聚类中，度量两个聚类结果的相近程度。它的值域是[0,1]，值越高表示两个聚类结果越相似。归一化是指将两个聚类结果的相似性值 定量到0~1之间。）如图1所示。



**图（1） 预训练的 ViT-B 每一层的查询和关键 patches 之间的归一化互信息 (NMI)。**

较高的NMI值意味着注意力强烈地依赖于query patch，如图1所示已有的很多经典模型（BeiT，SimMIM和MaskFeat）在较低层的注意力并不像顶层一样强烈地依赖于query patch。

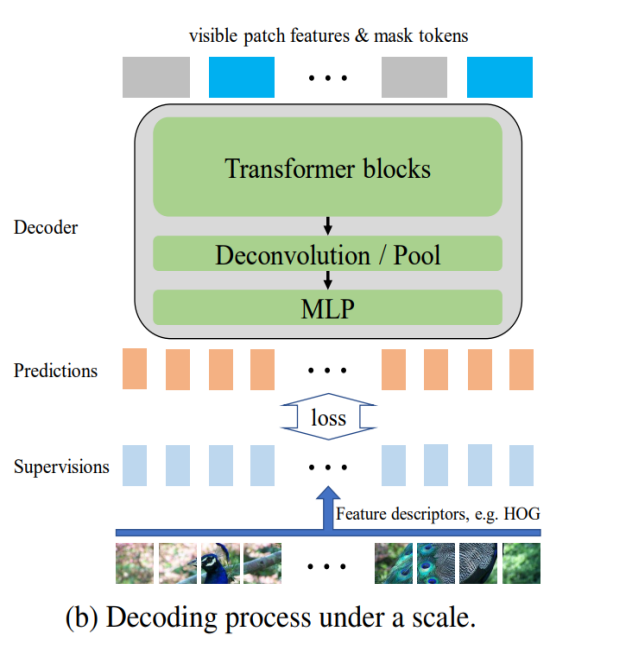
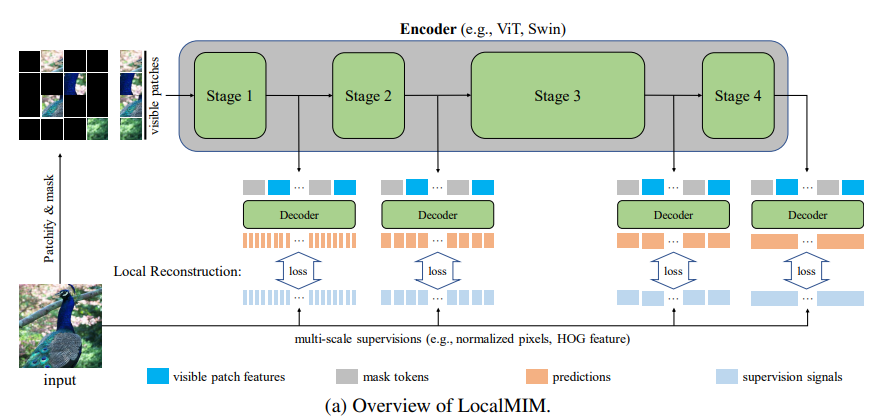
（1）编码器的低层在MIM中起关键作用对于预训练，上层特征是从下层计算的，学习良好的下层可以将语义知识传播到上层并促进知识的学习；对于微调，上层通常快速适应特定的下游任务，而下层任务变化较慢，需要在训练前充分学习。即使只微调几个上层并冻结其他层也可以获得相当的性能。

（2）低层 patches 很难学习 patches 间的关系，在patches化和线性投影之后，初始 patches 嵌入会丢失 patches 间的语义关系；ViTs中的自注意力机制负责通过patches 间交互来学习这些关系。由于自注意机制的计算复杂度与patches数量呈二次依赖，因此很难学习 patches 间的相互作用；在全局重建损失下，低层的 patches 间的交互缺乏明确指导。检查每一层查询和关键patches之间的归一化互信息，高NMI值意味着注意力图强烈依赖于查询patches。语义表示应该具有较高的查询自适应注意力，即不同的查询块忠实地关注其语义相关的区域。如图1 所示，现有的具有全局损失的MIM模型在较低层具有较小的NMI值，其 patches 的查询自适应注意力较少。

**2 方法**

**2.1 总体分析**  
 通过上面的分析，我们发现在掩码图像建模中较低层的学习很关键，然而目前所有MIM方法都只显式地指导顶层的学习。为此，考虑到重构任务需要patch之间的语义推理才能完成，我们将重构任务引入多个局部层以显式地进行有意义的指导。实际上，特征蒸馏也可以显式指导多个局部层，但是特征蒸馏需要提前训练好的或动量更新的teacher网络，会显著增加计算量，而且简单的特征匹配比不上推理任务对学习语义关联带来的增益。进一步地，我们发现直接地将顶层的重构任务引入到多个局部层增益不明显，原因可能是多个不同的局部层需要学习不同粒度的信息。

为此，我们考虑从原始输入中提取不同尺度的监督信号来指导多个局部层的学习。具体的，对于原始输入，为了获得监督信号，已有方法通常首先将x划分为不重叠的区域，该划分与构造编码器输入的划分对齐。然后使用恰当的特征描述算子（如像素标准化，HOG或预训练的codebook）提取每个区域的特征作为监督信号。我们定义此时监督信号的尺度为，通常可以认为在粗糙划分（p较大）下每个区域捕捉原始输入相对high-level的语义信息，比如目标的部分或整体形状，而精细划分（p较小）下每个区域捕捉相对low-level的语义信息，比如边、角或纹理。



**图（a)为LocalMIM[1]的概述图。随机对一些输入patches进行掩码操作，然后在多个局部层预测它们不同尺度的监督信号。多尺度监督首先划分不同尺度的输入，然后用一些特征描述符完成信号的提取。底层负责精细尺度重建，顶层负责粗尺度重建。本文还使用非对称编码器-解码器策略进行快速编码。**

**特定尺度的解码过程如图 (b) 所示。解码器由三部分组成：用于推理的 Transformer 块、（可选）卷积/Pool 用于重新缩放、多层感知器进行预测。**

另一方面，视觉主干网络设计的很多工作已经表明从底层到高层特征图的尺度由粗到细对各种视觉任务（检测，分割和分类）都有增益，通常较低层学习细粒度的信息，而较高层学习粗粒度的信息。为此，我们不妨从原始输入中构造多尺度的监督信号（选择不同的p）来分别用于多个局部层的重构，令较低层重构细尺度的监督信号而较高层重构粗尺度的监督信号。值得注意的是，我们的方法与“加速编码”方法兼容，特别的我们采用MAE和GreenMIM中的做法，即编码器只输入可见patch。算法的整体流程如图(a)所示。图(b)中演示了某一特定尺度下的解码过程，解码器由三部分组成：推理部分（transformer blocks）+缩放部分（Deconvolution/Pool）+预测部分（MLP）。推理部分负责基于可见patch的表征推理被遮挡patch的信息；缩放部分是处理特征尺度与监督信号尺度不一致的情况，比如ViT这种柱状结构每层特征尺度不变而监督信号尺度是变化的，当不匹配时需要使用反卷积或池化操作进行上/下采样；预测部分负责整合放缩后的预测来作为最终输出。对于金字塔结构的模型，通常已经划分为多个stage，我们将重构任务用于每个stage的末端，对于柱状架构，我们参照金字塔架构的经验，选择部分层进行重构。

总的来说，LocalMIM既能显式地指导较低层以加速整体表征学习，又能促进对输入图像的多尺度理解。而且，该方法是架构无关的，理论上与各种主干网络适配，可用于未来更先进的主干网络中。

**2.2 详细分析**

**2.2.1 训练过程**

（1）图像预处理：通常将图像分块后形成visual patches序列，作为 ViT 的输入。

（2）掩码：对于patches序列，MIM构造一个随机掩码 m∈{0,1}N来表示对应于mi=1的掩码patches。2种策略：随机掩码和块掩码。在金字塔结构中，由于patches的数量逐层减少的，需要进行上采样。

（3）编码器：仅将可见的patch 输入编码器，并映射到潜在特征，大大减少计算量和内存。

（4）解码器：解码器以编码后的可见patches和掩码标记（共享、可学习）作为输入，采用位置编码保留位置信息。

（5）全局重建：现有的所有MIM模型都基于编码器的最终输出特征来预测监督信号y，并最小化全局重建损失。训练损失是所选层重建损失的加权和：



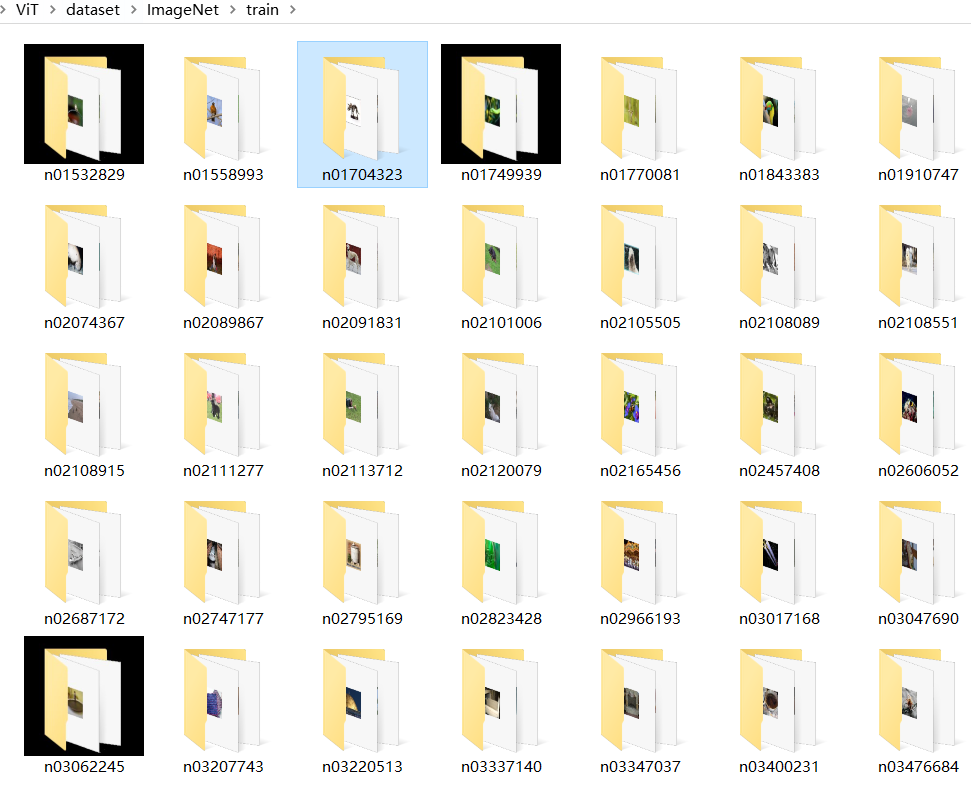
**其中L是所选层的集合，是每个局部损失的系数，掩码是通过对初始掩码m进行上采样/下采样来计算的。这些局部损失引导多个选定层的patches在不同的尺度上进行语义交互，这不仅加速了多层的学习，而且促进了对输入的多尺度语义理解。**

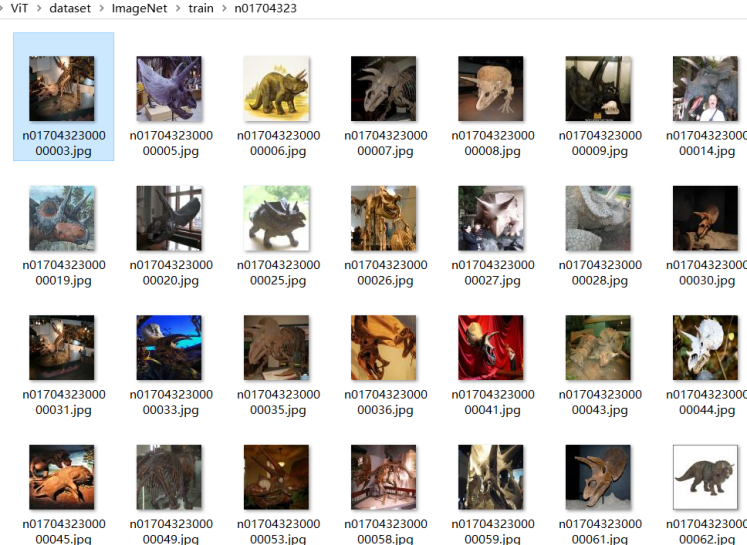
# 3 实验

# 由于是在自己笔记本电脑上进行实验，电脑配置受限，实验是在mini-ImageNet数据集上进行，使用单张NVIDIA GeForce GTX 1050 Ti显卡进行训练和测试。实验使用ViT模型。

**3.1 Mini-ImageNet 数据集简介**

# Mini-ImageNet数据集节选自ImageNet数据集。ImageNet是一个非常有名的大型视觉数据集，它的建立旨在促进视觉识别研究，训练ImageNet数据集需要消耗大量的计算资源，因此在2016年google DeepMind团队Oriol Vinyals等人在ImageNet的基础上提取出了Mini-ImageNet数据集。相比于CIFAR10数据集，Mini-ImageNet数据集更加复杂，但更适合进行原型设计和小样本分类研究[4]。





**图（2）数据集图片示例**

**3.2 实验过程**

**3.2.1 主要训练参数**

· **world\_size**: 1（单卡训练）

· **local\_rank**: 0（指定GPU）

· **batch\_size**: 128（每次迭代的样本数）

· **epochs**: 50（总训练轮数）

· **accum\_iter**: 1（梯度累积次数）

· **model**: MIM\_vit\_small\_patch16（使用ViT小模型）

· **input\_size**: 224（输入图像大小）

· **mask\_ratio**: 0.75（随机遮盖比例）

· **hog\_nbins**: 9（HOG特征的bin数）

· **weight\_decay**: 0.05（权重衰减）

· **blr**: 1e-3（基础学习率）

· **warmup\_epochs**: 5（学习率预热轮数）

· **data\_path**: ./dataset/ImageNet/（数据集路径）

· **output\_dir**: ./output/MAE\_ViT\_S（输出路径）

· **log\_dir**: ./logs（日志路径）

**3.2.1.1 学习率预热**

由于刚开始训练时,模型的权重(weights)是随机初始化的，此时若选择一个较大的学习率,可能带来模型的不稳定(振荡)，选择Warmup预热学习率的方式，可以使得开始训练的几个epoch或者一些steps内学习率较小,在预热的小学习率下，模型可以慢慢趋于稳定,等模型相对稳定后再选择预先设置的学习率进行训练,使得模型收敛速度变得更快，模型效果更佳。这种方法的目的是帮助网络更稳定地开始学习过程，特别是在训练初期，避免因为学习率过大导致的模型训练不稳定或者性能波动较大的问题[5]。

学习率预热通常包含以下几个关键步骤：

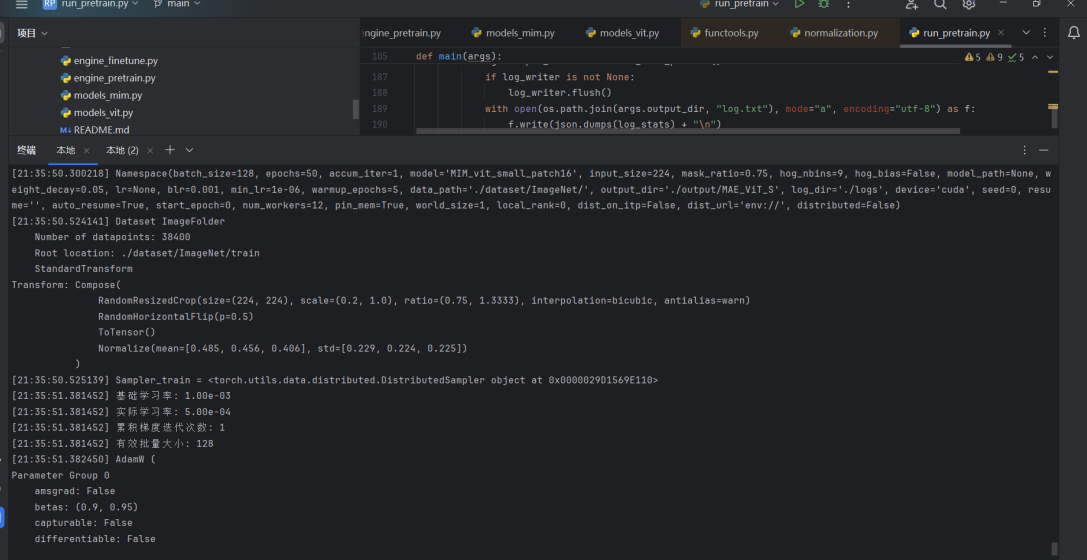
（1）初始学习率设定：将初始学习率设定为一个较小的值，通常远低于最终期望的学习率。

（2）逐步增加学习率： 在训练的前几个epoch中，逐步增加学习率，直到达到预设的初始学习率。这个过程可以是线性增加，也可以是指数增加，具体取决于具体的训练策略和模型的需求。

（3）稳定学习率：一旦达到预设的初始学习率，就继续保持这个学习率进行后续的训练。

通过学习率预热，可以有效地提高模型的稳定性和训练效率，尤其是对于大型、复杂的神经网络以及需要更精细调整学习率的训练任务。

在训练过程中，模型的学习率从0逐渐增加到设定的1e-3，并在前5个epoch内逐步热身。每个epoch处理300个批次，处理时间约为31分钟左右。实验总时间为50个epoch，耗时大约26小时左右。



**图（3) 预训练过程**

**3.2.2 结果分析**

在mini-ImageNet数据集上，模型在50个epoch后达到了较为稳定的损失值约为1.1300。以下是部分训练日志记录：

[01:00:04.158371] Epoch: [0] [299/300] eta: 0:00:06 lr: 0.000100 loss: 1.1302 (1.1303) time: 5.7254 data: 0.0003 max mem: 3601

[01:00:18.402788] Epoch: [0] Total time: 0:30:57 (6.1926 s / it)

[01:00:18.402788] True time 0:28:33

[01:00:18.403802] Averaged stats: lr: 0.000100 loss: 1.1302 (1.1303)

[11:02:41.710366] Epoch: [20] [299/300] eta: 0:00:06 lr: 0.000360 loss: 1.1298 (1.1303) time: 5.8852 data: 0.0001 max mem: 3601

[11:02:50.104403] Epoch: [20] Total time: 0:30:47 (6.1595 s / it)

[11:02:50.105407] True time 0:29:04

[11:02:50.105407] Averaged stats: lr: 0.000360 loss: 1.1298 (1.1303)

[02:04:20.582164] Epoch: [49] [299/300] eta: 0:00:06 lr: 0.000001 loss: 1.1303 (1.1301) time: 5.8843 data: 0.0001 max mem: 3601

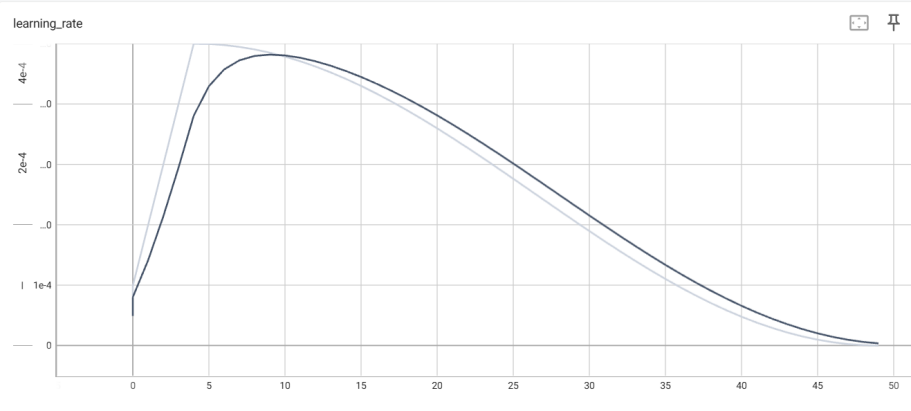
[02:04:24.875151] Epoch: [49] Total time: 0:30:58 (6.1956 s / it)

[02:04:24.876151] True time 0:29:20

[02:04:24.876151] Averaged stats: lr: 0.000001 loss: 1.1303 (1.1301)

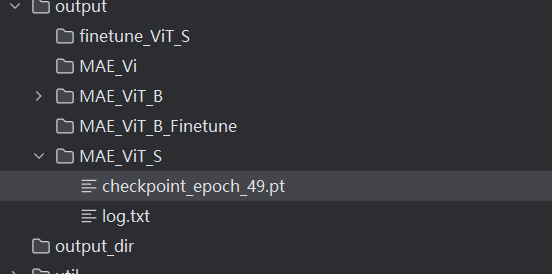
[02:04:25.337902] Model checkpoint saved at ./output/MAE\_ViT\_S\checkpoint\_epoch\_49.pt

[02:04:26.062651] 训练总时长 1 day, 1:35:05



**图（4) 实验学习率曲线，设置5个epoch作为学习率预热阶段。**

通过以上结果可以看出，模型在各个epoch中的损失逐渐趋于稳定，说明模型训练效果良好，达到预期性能。最终得到checkpoint\_epoch\_49.pt预训练模型文件。从训练结果表明，该模型具有良好的训练稳定性和高效性。



**3.3 实验过程分析总结**

在进行实验时，必须面对硬件资源的显著限制。原论文使用了8个Tesla V100-32G GPU，并且训练了长达1600个epoch。相比之下，受限于自身硬件环境不足以进行大规模数据的训练和微调。这一差异需要做出一些必要的调整和取舍：

首先，考虑到显卡的有限计算能力和内存资源，决定采用更简化的模型架构（MIM\_vit\_small\_patch16）。复杂的深度学习模型通常需要大量的计算资源和内存支持，而自身硬件条件明显无法满足这些需求。因此，在资源受限的情况下，选择不进行代码微调有助于降低计算复杂度，从而确保在硬件限制下仍能有效训练和实验。

其次，考虑到训练时间和epoch数目的限制。原论文的长时间训练可能不适用于本次实验的硬件环境。因此，必须根据实际情况调整训练的epoch数量，并且需要精确控制每个epoch的训练时间，通过观察模型在所选数据集上的短期训练表现，从而为进一步深入研究与优化提供数据分析基础。

此外，实验设计和对比分析也需按照硬件资源的实际情况进行合理的调整，需要仔细分析哪些实验设置对于研究问题至关重要，并相应地简化和调整实验设计，以确保实验结果的准确性和可比性。

# 4 结论与展望

LocalMIM模型通过引入局部多尺度重建策略，显著提升了自监督学习中的图像表示能力。实验结果表明，LocalMIM在mini-ImageNet数据集上的图像分类任务中表现优异，具有较高的准确率和较快的收敛速度。通过对部分图像进行随机遮盖并重建，被遮盖部分的策略，模型不仅能够有效地学习到图像的局部特征，还能捕捉到全局信息，提高了模型的整体性能。未来的研究可以在以下几个方面进一步探索和改进LocalMIM模型：

（1）扩展到更大规模的数据集：在更大规模的数据集上验证LocalMIM的性能，如ImageNet-1K或更大的图像分类数据集，以评估其在更复杂任务中的适用性和泛化能力。

（2）模型结构优化：探索更加优化的模型结构和参数配置，例如更深层次的Transformer架构或混合多尺度特征融合策略，以进一步提升模型性能。

（3）多任务学习：将LocalMIM应用于其他自监督学习任务，如目标检测、语义分割等，以验证其在不同任务中的通用性和有效性。

（4）硬件加速与优化：结合更强大的硬件设备和优化算法，如分布式训练和混合精度训练，进一步提高模型的训练效率和推理速度。

（5）实际应用探索：将LocalMIM应用于实际场景中，如医疗影像分析、自动驾驶、安防监控等，验证其在实际应用中的性能和稳定性，推动其在工业界的广泛应用。

通过持续的研究与优化，LocalMIM模型有望在自监督学习领域中取得更大的突破，并为各种视觉任务提供更为强大的技术支持。

# 参考文献

1. Masked Image Modeling with Local Multi-Scale Reconstruction. Wang et.al. CVPR 2023.
2. 骆迪,张乾,柏武贰.基于掩码自动编码器的图像修复研究[J].现代信息科技,2024,8(03):69-72+78.DOI:10.19850/j.cnki.2096-4706.2024.03.015.
3. 丁新宇,孔兵,陈红梅,等.路径掩码自编码器引导无监督属性图节点聚类[J/OL].计算机科学:1-15[2024-06-18].http://kns.cnki.net/kcms/detail/50.1075.TP.20240513.1447.029.html.
4. 王彩玲,魏清晨,仇真,等.基于多角度学生子注意力网络的小样本学习[J].南京邮电大学学报(自然科学版),2022,42(03):66-73.DOI:10.14132/j.cnki.1673-5439.2022.03.008.
5. 常禧龙,梁琨,李文涛.深度学习优化器进展综述[J].计算机工程与应用,2024,60(07):1-12.