arXiv:2601.00578v1 [cs.LG] 2 Jan 2026

学习实现可复现性：鲁棒神经网络的自定义损失设计

Waqas Ahmed[1], **Sheeba Samuel**[2], **Kevin Coakley**[3],**Birgitta Koenig-Ries**[1],**Odd Erik Gundersen**[3]

[1]耶拿弗里德里希·席勒大学，德国耶拿 [2]开姆尼茨工业大学，德国开姆尼茨[3]挪威科技大学，挪威特隆赫姆
waqas.ahmed@uni-jena.de, sheeba.samuel@informatik.tu-chemnitz.de, kcoakley@sdsc.edu,
birgitta.koenig-ries@uni-jena.de, odderik@ntnu.no

摘要

为提升深度学习模型的可复现性和可靠性，我们针对当前训练方法中的一个关键缺口展开研究：缺乏确保模型在多次运行中表现一致且鲁棒的机制。实证分析表明，即使在受控初始化和训练条件下，模型的准确率仍可能出现显著波动。为解决这一问题，我们提出了一种自定义损失函数（CLF），以降低训练结果对权重初始化和数据打乱等随机因素的敏感性。通过微调其参数，CLF 明确地平衡预测准确率与训练稳定性，从而使模型表现更加一致、可靠。针对图像分类和时间序列预测任务中的多种架构开展的大量实验表明，我们的方法在不牺牲预测性能的前提下，显著提高了训练鲁棒性。这些结果证明，CLF 是一种有效且高效的策略，有助于开发更加稳定、可靠和值得信赖的神经网络。

引言

深度学习（DL）模型凭借从大规模数据中学习复杂表征的卓越能力，已成为医疗诊断、自主系统和金融预测等广泛应用领域的基础。尽管取得了这些成功，要让这些模型始终保持稳定且值得信赖的性能，仍是一项重大且尚未得到充分解决的挑战。即使架构、超参数和数据集等训练条件保持不变，模型在不同训练轮次中通常也会产生显著不同的结果。这种变化源于权重初始化、数据打乱和优化器行为等算法层面的随机性来源，它们会影响模型训练轨迹，导致结果不一致。近期研究强调了深度神经网络对这类随机因素的敏感性，并发现即使初始化发生细微变化，也可能导致模型最终性能出现大幅偏差（Summers 和 Dinneen（2021））。在深度学习中，控制随机影响的一种常见做法是在训练期间固定随机种子。这并不会直接降低随机因素的影响；相反，它通过伪随机数生成器（伪随机数生成器）来确定所有随机操作的序列，确保每次使用该种子运行时都会重现相同的序列。因此，使用相同种子重复实验会得到完全相同的结果。然而，使用不同种子时，随机操作序列会发生变化，进而改变训练轨迹，并可能导致结果出现显著差异。因此，模型性能仍然会受到种子选择的影响，随机因素带来的变化依然存在。虽然固定种子能针对特定实验设置实现有限形式的可复现性，但无法确保以更广泛意义上的鲁棒性，而这种鲁棒性对于在不同训练轮次中得出可靠结论至关重要（Pham 等（2020））。其次，另一种常见做法是报告使用不同种子进行多次训练所得到的平均指标。该方法通过对多个伪随机数生成器序列进行采样，更全面地展现模型性能，从而捕捉更广泛的变化。尽管从统计角度看，这种方法比依赖单一种子更为可靠，但其计算成本相当高。在许多情况下，要获得稳定估计，需要进行 25 次或更多完整训练（Renard 等（2020）；Bouthillier、Laurent 和 Vincent （ 2019 ））， 因此在大规模实验、资源受限环境或实时开发环境中并不实用。

我们的研究针对这一局限，提出了一种直接在学习目标中降低训练变异性的方法。受构建更可信、更稳定的深度学习系统这一需求的启发，我们引入了一种自定义损失函数（CLF），通过惩罚预测置信度和输出损失的波动来规范训练过程。CLF 并不彻底消除随机性，而是减轻其后续影响，从而实现更稳定的收敛并降低不同运行之间的变异性。这种方法开销小，且兼容现有架构和训练流水线。在开发该方法的过程中，我们系统地考察了变异性的算法来源，并采用固定且相同的训练条件来分离和衡量随机因素的影响。此外，我们还探究了训练过程对损失函数自身超参数的敏感性，发现对变异性惩罚的权重及其引入时机进行调节都至关重要。具体而言，我们发现，较早应用 CLF 并在整个训练过程中持续使用，能更好地平衡稳定性与学习效率。

版权所有 ©2026，人工智能促进协会（www.aaai.org）。保留所有权利。

我们在使用 CIFAR-10 和 CIFAR-100（Krizhevsky、Hinton 等，2009）开展的图像分类任务上进行了大量实验，以验证我们的方法，所用架构包括 ResNet（He 等，2016）、VGG（Simonyan 和 Zisserman，2015）以及 ShuffleNet（Ma 等，2018）。我们还将 CLF 应用于 ETTh1（Zhou 等，2021）数据集上的时间序列预测模型（Autoformer（Wu 等，2021）、NLinear（Zeng 等，2023）和 iTransformer（Liu 等，2023）），以展示其泛化能力。结果始终表明，CLF 能够降低模型性能的标准差，有时降幅超过 70%，且不会损害准确率。我们的主要贡献如下：

1. **自定义损失函数**：我们提出了一种新的损失公式，将变异控制明确纳入其中，从而显著降低不同训练轮次之间的性能波动。2. **对持续时长敏感的有效性**：我们分析了 CLF 激活时长的影响，并发现，在训练过程中持续更长时间的激活始终能带来更好的性能和稳定性，而在训练后期才激活则收益有限。3. **跨领域泛化能力**：我们证明，CLF 能提高图像和时间序列领域的稳定性，证实了其在不同架构和任务中的广泛适用性。

这些发现共同为构建更可靠、更可信的深度学习系统铺平了道路，因为它们不仅解决了评估层面的变异性问题，也关注了训练动态本身的变异性。

相关工作

技术鲁棒性是可信人工智能的关键组成部分，尤其对于部署在动态或高风险环境中的系统而言更是如此。Kaur 等人（Kaur 等，2022）提供了一项全面的综述，强调系统需要能够抵御扰动和实现层面的变异，以确保行为可靠。Gundersen 等人（Gundersen 等，2023）专门研究了神经网络训练中的算法随机性所带来的鲁棒性问题。他们指出，权重初始化、数据打乱和非确定性操作等随机因素会导致模型性能出现显著变异。他们的研究结果表明，这种变异性往往被低估。他们提出了方法学标准，包括至少进行 25 次重复训练，以支持统计上可靠的结论。重要的是，他们认为，对随机性的鲁棒性并非次要的技术细节，而是得出可信的科学和实证结论的先决条件。这一观点与欧盟人工智能高级别专家组的立场一致；该组织通过鲁棒性、问责制和透明度等原则来界定可信度。因此，在深度学习中，实现对随机性的鲁棒性不仅意味着减少噪声，更是确保系统在现实世界部署中表现一致且可靠的基础。

基于这些概念基础，近期的实证研究凸显了对随机性缺乏鲁棒性在实践中的表现。Bouthillier 等人（Bouthillier、Laurent 和 Vincent，2019）对机器学习中的可复现性失败进行了批判性评估，并指出，其中许多问题源于对实验变异性控制不足。他们的研究说明，细微的实现细节、随机种子的选择或系统级因素，都可能导致显著的性能波动。

Zhuang 等人（Zhuang 等人，2022）沿着这一研究方向进一步考察了工具级随机性如何影响神经网络训练的轨迹，例如软件库、编译器和系统级抽象所引入的随机性。他们的结果表明，由于低层级非确定性来源的存在，看似完全相同的配置也可能产生不同的模型。同样，Summers 等人（Summers 和 Dinneen（2021））聚焦于随机梯度下降等优化过程所引入的不稳定性。他们 gnɥg明，即使初始化、数据和超参数保持不变，不同运行之间的模型结果仍可能存在显著差异，这表明深度学习优化动态中存在更深层次的技术不稳定性。Ahmed 等人（Ahmed 和 Lofstead，2022）提出了管理伪随机性的实用策略，包括一致地设定随机种子以及系统地记录随机状态。他们认为，这是提升可信度和实验可靠性的关键。然而，正如 Summers 等人（Summers 和 Dinneen（2021））所 gnɥg明的，仅靠这些控制措施并不能消除不稳定性；更深层次的算法敏感性仍然是一项挑战。Yi 等人（Ji 等人，2023）通过分析随机性对评估指标的影响，进一步印证了这一观点。他们建议采用多次运行报告和受控的随机种子策略，但也承认，确定足够的运行次数仍未解决。Picard（Picard，2021）提供了实证证据，说明随机种子的选择会如何显著影响计算机视觉模型中报告的结果。

针对这些持续存在的挑战，我们的工作将重点从外部缓解策略转向内部算法解决方案。我们提出了一种 CLF，可通过明确约束训练动态来降低模型结果的方差。CLF 不依赖重复训练或严格控制随机种子，而是直接稳定学习轨迹。这种方法提升了面对随机性的技术鲁棒性，并有助于开发在不同随机条件下表现一致的神经网络。

自定义损失函数

所提出的自定义损失函数（CLF）旨在缓解深度神经网络训练中常见的随机行为，包括因随机初始化、数据打乱和系统级非确定性而导致的不同训练运行间模型性能差异。CLF 通过同时增强梯度更新的稳定性以及小批次内和小批次间预测的一致性，提高优化过程的一致性。

它由三个部分组成：**(1) 交叉熵损失（CEL）**（第 节），**(2) 稳定损失（SL）**（第 节），以及**(3) 方差惩罚损失（VPL）**（第 节）。

总损失定义如下：

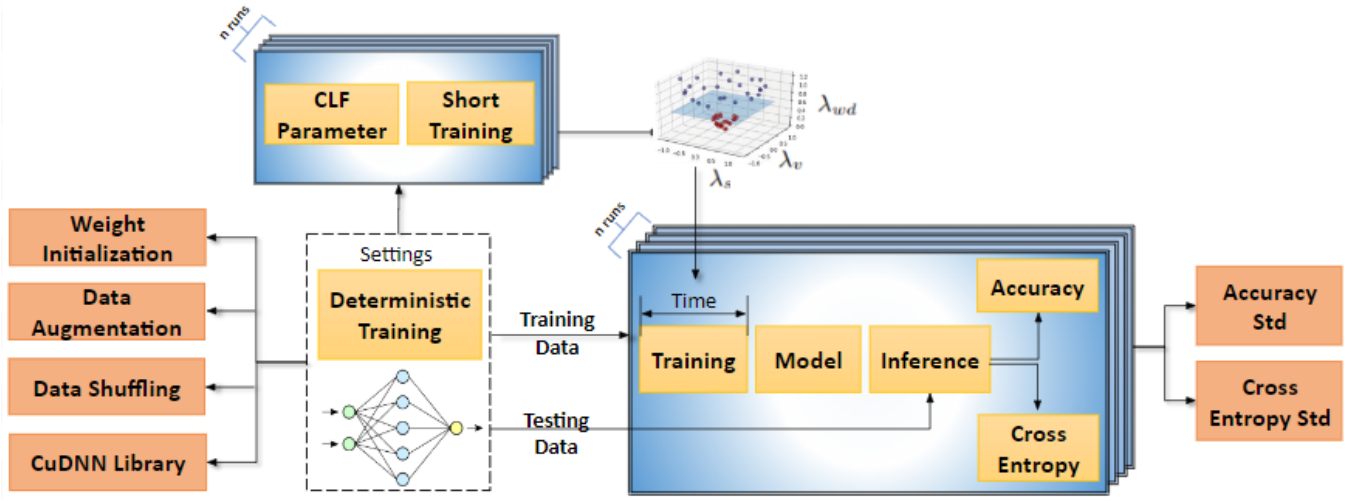

图1：方法概述（分类任务）：该图展示了在不同数据集和模型上进行 20 次固定且完全相同的训练（n=20），以评估 CLF 的有效性。我们通过控制随机源进行确定性训练。我们以准确率和交叉熵损失的标准差来评估模型的变异性，旨在量化训练过程的鲁棒性。

$$
\begin{aligned}
L(\theta; x, y) = \mathrm{CEL}(\theta; x, y) \\
+ \lambda_s \, \mathrm{SL}(\theta; x, y) \\
+ \lambda_v \, \mathrm{VPL}(\theta; x, y)
\end{aligned} \tag{1}
$$

其中，θ 是模型参数，$(x,\ y)$是输入样本和真实标签，而 λ_s、λ_v 是控制SL和VPL影响程度的标量超参数。

每个项都发挥着独特作用，用于控制不稳定性并提高对随机性的鲁棒性。

交叉熵损失（CEL）

交叉熵损失是分类任务的标准目标函数，用于衡量预测类别分布与真实标签分布之间的差异：

$$
\mathrm{CEL}(\theta; x, y) = -\frac{1}{N} \sum_{i=1}^{N} \log p(y_i | x_i; \theta), \tag{2}
$$

其中，$p(y_i|x_i;\theta)$ 是给定输入 x_i 时正确类别 y_i 的预测概率，N 是批大小。

虽然交叉熵损失有助于引导模型做出正确预测，但仅使用 CEL 并不会对优化轨迹的稳定性施加任何约束。在存在噪声标签或类别不平衡的情况下，较小的预测误差可能导致置信度分数偏低，进而产生幅度较大的梯度。这种敏感性体现在其梯度中：

$$
\nabla_\theta \mathrm{CEL} = -\frac{1}{N} \sum_{i=1}^{N} \frac{1}{p(y_i|x_i;\theta)} \nabla_\theta p(y_i|x_i;\theta), \tag{3}
$$

当 $p(y_i|x_i;\theta)$ 较小时，该值可能会迅速增大，从而导致更新不稳定。

为了解决这一问题，CLF 引入了两个额外的术语：SL 侧重于跨轮次稳定性，而 VPL 则确保批次内各类别的一致性。

稳定损失（SL）

SL 通过惩罚连续训练轮次之间损失值的突然变化，来确保训练过程中的时间稳定性。其定义如下：

$$
\mathrm{SL}(\theta; x, y) = \left| \mathrm{CEL}(\theta; x, y) - \mathrm{CEL}_{\mathrm{prev}} \right|, \tag{4}
$$

其中 $\mathrm{CEL}_{\mathrm{prev}}$ 是上一训练轮次的交叉熵损失。

其梯度为：

$$
\nabla_\theta \mathrm{SL} = \mathrm{sign}(\mathrm{CEL} - \mathrm{CEL}_{\mathrm{prev}}) \cdot \nabla_\theta \mathrm{CEL}, \tag{5}
$$

其中，符号项控制惩罚方向。与作用于参数的传统正则化项不同，稳定损失作用于损失层面，因此与架构无关。

方差惩罚损失（VPL）

SL 调节训练的时间维度，而 VPL 则关注小批次内的空间一致性。它会惩罚同一类别样本之间模型预测的方差，从而提升鲁棒性，并防止模型对批次特定噪声过拟合。VPL 定义为小批次内各类别方差的平均值：

$$
\mathrm{VPL}(\theta; x, y) = \frac{1}{|\mathcal{C}|} \sum_{j \in \mathcal{C}} \mathrm{Var}_j, \tag{6}
$$

其中，$\mathcal{C}$ 是小批次中出现的类别标签集合，Var_j 是类别 j 的方差，计算如下：

$$
\mathrm{Var}_j = \frac{1}{m_j} \sum_{i \in S_j} \left(f_j(x_i; \theta) - \bar{f}_j \right)^2, \tag{7}
$$

其中 $S_j = \{ i \mid y_i = j \}$ 表示具有真实标签 j 的样本集合，$m_j = |S_j|$，类别平均 logit f_j 定义为：

$$\bar{f}_j = \frac{1}{m_j} \sum_{i \in S_j} f_j(x_i; \theta). \tag{8}$$

VPL 的梯度。 VPL 关于 θ的梯度为:

$$\nabla_\theta \mathrm{VPL}(\theta; x, y) = \frac{1}{|\mathcal{C}|} \sum_{j \in \mathcal{C}} \nabla_\theta \mathrm{Var}_j, \tag{9}$$

其中:

$$\begin{aligned} \nabla_\theta \mathrm{Var}_j = & \frac{2}{m_j} \sum_{i \in S_j} \left(f_j(x_i; \theta) - \bar{f}_j \right) \\ & \times \left(\nabla_\theta f_j(x_i) - \frac{1}{m_j} \sum_{k \in S_j} \nabla_\theta f_j(x_k) \right) \end{aligned} \tag{10}$$

Since

$$\sum_{i \in S_j} \left(f_j(x_i; \theta) - \bar{f}_j \right) = 0, \tag{11}$$

（10）中的第二项相互抵消，将梯度简化为:

$$\nabla_\theta \mathrm{Var}_j = \frac{2}{m_j} \sum_{i \in S_j} \left(f_j(x_i; \theta) - \bar{f}_j \right) \nabla_\theta f_j(x_i). \tag{12}$$

最后，VPL 的梯度为:

$$\begin{aligned} \nabla_\theta \mathrm{VPL}(\theta; x, y) = & \frac{2}{|\mathcal{C}|} \sum_{j \in \mathcal{C}} \sum_{i \in S_j} \frac{1}{m_j} \left(f_j(x_i; \theta) - \bar{f}_j \right) \\ & \times \nabla_\theta f_j(x_i) \end{aligned} \tag{13}$$

最小化这一项可促使输出空间中同一类别的 logits 更紧密地聚集，从而降低类内差异，减轻随机训练因素的影响，并提高不同运行之间的鲁棒性。

方法论

为评估 CLF 的有效性，我们制定了一套系统的实验方案，用于分离随机效应，并量化性能和训练行为中的方差。图1概述了我们的方法。

我们首先通过严格控制已知的随机性来源，建立确定性基线。为此，我们在所有相关库中固定随机种子，并在框架层面禁用非确定性操作，例如 cuDNN 基准测试和并行内核执行。这样可以确保观察到的变异仅源于通过设定随机种子无法消除的固有随机效应。所有实验均在相同的硬件、软件和超参数设置下进行。为评估初始化的敏感性，我们使用 20 个不同的随机种子训练每个模型 $S = \{1, 2, \ldots, 20\}$。每次训练均采用相同的架构和优化配置，从而能够对不同运行之间的变异进行受控分析。我们采用从头训练协议，遵循相关建议这些建议来自 Summers 等人（Summers 和 Dinneen（2021））, 即每个模型均使用随机初始化权重独立训练。该方法能够捕捉模型初始化和优化中随机成分引入的全部变异，并避免由热启动模型或迁移学习造成的偏差。我们的实验包括图像分类和时间序列预测任务。我们通过测量分类任务测试准确率的标准差，以及时间序列预测中平均绝对误差（MAE）的标准差，量化不同随机种子之间的性能变异。

我们使用 Optuna 框架（Akiba 等，2019）对 CLF 超参数（λ_s 和 λ_v）进行了优化。调优目标侧重于最小化验证集方差，而非最大化原始性能，这与我们降低结果不稳定性的目标一致。Optuna 能够高效探索参数空间，而无需依赖穷举式的网格搜索评估。总体而言，我们的方法论旨在评估 CLF 能否提升深度学习训练的确定性和可靠性，尤其是在严格控制的实验条件下仍存在显著变异的情况下。

使用 Optuna 进行超参数优化:

我们使用 Optuna 对 CLF 参数进行了优化。Optuna 是一种超参数优化框架，能够通过贝叶斯采样高效探索搜索空间。与穷举评估所有组合的网格搜索不同，Optuna 会动态优先考虑有前景的配置，从而减少计算开销。算法1概述了调优过程，重点介绍以下关键参数: 方差惩罚权重（λ_v）用于控制输出方差的降低，稳定性权重（λ_s）用于调节稳定损失贡献，而方差惩罚权重衰减（λ_{wd}）则用于防止过度惩罚。该算法能够找出有前景的配置，以供我们在实验中进一步分析。

算法1：用于自定义损失函数的 Optuna 超参数优化

要求：~~λ_v、λ_s、λ_{wd}~~ 的搜索范围 1：初始化 ~~Optuna~~ 研究，并设置最小化目标 2：对范围为 num_trials 的每个试验执行 3：使用对数均匀分布采样 λ_v、λ_s、λ_{wd} **4：使用采样得到的参数初始化模型 5：使用随机梯度下降和学习率调度器优化模型 6：在多个随机种子下评估测试准确率 7：计算归一化准确率和标准差 8：计算目标分数：score = norm_std –norm acc_ 9：如果 score < best score，则_ 10：更新** $best_params \leftarrow \{\lambda_v, \lambda_s, \lambda_{\text{wd}},$**score**$\}$ **11：结束如果 12：结束循环 13：返回** $best_params$

实验设置

我们的实验涵盖两个领域：图像分类和时间序列预测。对于图像分类任务，我们使用 CIFAR-10 和 CIFAR-100 数据集，如表 1 所示。我们测试了三种广泛使用的卷积神经网络架构：ResNet、VGG-16 和 ShuffleNet-V2，这些架构均因其在分类任务中的出色表现而被选用。训练采用余弦衰减学习率计划，初始峰值为 0.40，批大小为 512，动量设为 0.9，权重衰减为 5×10^{-4}。针对每个模型与数据集组合，分别调整 CLF 参数（λ_s 和 λ_v）。所有训练均从头开始，并使用 20 个不同的随机种子，以评估初始化变化下的鲁棒性。主要评估指标为这些训练运行中测试准确率的标准差。所有实验均使用 PyTorch（Paszke 等人，2019）在配备 64 个 CPU 核心、512 GB RAM 和 NVIDIA A100 GPU 的计算环境中进行。

对于时间序列预测，我们采用 ETTh1 数据集，它是电力变压器温度（ETT）数据集的一个子集。该数据集包含从中国两个县收集的两年逐小时测量数据，包括变压器油温和六项相关电力负荷特征。数据集按 12:4:4 的月份比例划分为训练集、验证集和测试集。预测任务是根据历史多变量输入序列预测未来的油温值。我们使用了三种针对长序列预测设计的神经网络架构：NLinear、Autoformer 和 iTransformer。对于每个模型，我们均采用各自原始论文中规定的训练配置和超参数。这一实验设计能够直接比较 CLF 在不同模态和模型类型上的表现，从而评估其在广泛条件下对鲁棒性的贡献。

表 1：数据集、网络和训练设置。

数据集	训练集/验证集/测试集划分	网络
CIFAR-10	50,000 /- /10,000	ResNet-14 VGG-16 ShuffleNet-V2
CIFAR-100	50,000 /- /10 ,000	ResNet-32 VGG-16 ShuffleNet-V2
ETTh1	12/4/4 个月	NLinear Autoformer iTransformer

结果与讨论

本节全面评估了 CLF 在图像分类和时间序列预测任务中的表现。总体而言，我们针对分类任务进行了 280 轮固定且完全相同的训练，并开展了少量超参数搜索，总计约耗费 230 小时的 GPU 时间。对于时间序列实验，我们进行了 400 次固定且完全相同的运行，耗时超过 250 小时的 GPU 时间。

CLF 对图像分类的影响

如表2所示，CLF 在不同设置下均能稳定降低变异性，但改善幅度因模型和数据集而异。对于 CIFAR-10 上的 ResNet-14，CLF 将准确率标准差从 0.14 ± 0.04 降至 0.08 ± 0.02，变异性降低了 42.2%。计算这一降幅时，综合考虑了误差上下界，以确保即使是最小的波动也纳入考量。具体而言，预期变异性估算为最大值与最小值的平均值：

$$\text{Avg. Var. Reduction} = \frac{\text{Upper Bound Var.} + \text{Lower Bound Var.}}{2} \tag{14}$$

这种方法能够更全面、更稳健地衡量训练变异性。其他配置也呈现出类似趋势。在 CIFAR-10 上，ShuffleNet-V2 的变异性降低了 33.9%，而 VGG-16 的降幅较小，为 12.2%。CLF 在 VGG-16 上影响有限，原因是非 CLF 设置下的基线变异性本就较低，报告的标准差为 0.12 ± 0.04。由于初始方差已经很小，进一步降低的空间自然有限。在 CIFAR-100 上，ResNet-14 的变异性降低了 39.4%，VGG-16 提升了 26.1%，而 ShuffleNet-V2 的降幅最大，达到 77.1%。ShuffleNet-V2 的降幅尤其显著，可能是因为其轻量级架构对随机种子变化和训练不稳定性往往更为敏感。在这类情况下，我们的方法似乎特别有益：即使稳定性略有提升，也能显著增强性能一致性。这些结果证实，CLF 在初始方差较高或优化轨迹不稳定的训练环境中尤其有效。通过降低对随机因素的敏感性，CLF 能够使模型在保持准确率的同时，在多次重复运行中表现出一致的行为，从而提升训练可信度。

基于样本的评估：CLF鲁棒性

为评估 CLF 引入的鲁棒性是否能推广到特定种子分组之外，我们使用固定的 20 个训练后模型池开展了一项对照实验。我们随机抽取了 1000 个子集，分别包含 5、10 和 15 个模型，以模拟实际场景中可能只能部署有限数量模型的情况。对于每个组，我们分别计算使用和不使用 CLF 训练的模型的平均准确率和标准差。关键指标是某种配置取得较低标准差的频率。如表3所示，在大多数情况下，CLF 带来了更稳定的性能。例如，在组大小为 15 时，CLF 在 75% 的样本中取得了较低的标准差。这一趋势在所有组大小下都保持一致，表明 CLF 不仅能提升单个模型层面的性能一致性，还能提高模型在小型集成模型中的评估表现。这些结果支持 CLF 在部署场景中的广泛适用性，尤其是在可用的训练后模型子集有限时。

表2：引入 CLF 后变异性的降低。

训练	数据集	模型	平均准确率（%）	准确率标准差 (%)	交叉熵标准差 (%)	平均方差降低（%）
不使用 CLF	CIFAR-10	ResNet-14	88.2	0.14 ± 0.04	0.008 ± 0.002	-
		VGG-16	93.8	0.12 ± 0.04	0.007 ± 0.002	-
		ShuffleNet-V2	90.8	0.20 ± 0.07	0.008 ± 0.002	-
	CIFAR-100	ResNet-14	62.6	0.16 ± 0.05	0.01 ± 0.004	-
		VGG-16	74.0	0.17 ± 0.06	0.008 ± 0.002	-
		ShuffleNet-V2	67.9	0.30 ± 0.06	0.017 ± 0.007	-
使用 CLF	CIFAR-10	ResNet-14	88.3	0.08 ± 0.02	0.006 ± 0.001	42.2
		VGG-16	93.7	0.11 ± 0.03	0.006 ± 0.002	12.2
		ShuffleNet-V2	90.7	0.13 ± 0.04	0.003 ± 0.0008	33.9
	CIFAR-100	ResNet-14	62.1	0.10 ± 0.04	0.02 ± 0.007	39.4
		VGG-16	74.2	0.14 ± 0.03	0.006 ± 0.002	26.1
		ShuffleNet-V2	67.8	0.07 ± 0.02	0.020 ± 0.008	77.1

表3：在 CIFAR-10 上比较不同组大小下使用和不使用 CLF 的 ResNet-14。

组大小	平均准确率 (CLF)	标准差 (CLF)	平均准确率 (无 CLF)	标准差 (无 CLF)	较低标准差组
5	88.29478	0.1572	88.26957	0.1848	使用 CLF
10	88.29826	0.1827	88.26784	0.2024	使用 CLF
15	88.29689	0.1890	88.26738	0.2074	使用 CLF

表4：比较 ResNet-32 在 CIFAR-100 不同组大小下使用和不使用 CLF 的表现。

组大小	平均准确率 (CLF)	标准差 (CLF)	平均准确率 (无 CLF)	标准差 (无 CLF)	较低的标准差开发组
5	62.3355	0.2154	62.5287	0.1920	不使用 CLF
10	62.3356	0.2381	62.5300	0.2084	不使用 CLF
15	62.3341	0.2455	62.5295	0.2133	不使用 CLF

过拟合条件下的 CLF

为探究 CLF 在过拟合情形下的表现，我们使用 CIFAR-100 数据集上的 ResNet-32 架构开展了实验。该设置具有相对高容量的模型和复杂、细粒度的分类任务，因此适合用于考察 CLF 在模型可能发生过拟合时的表现。与在 ResNet-14 等较小架构中观察到的改进不同，本实验结果显示，CLF 未能有效降低变异性。在所有评估的组大小（5、10 和 15）下，与未使用 CLF 训练的模型相比，使用 CLF 训练的模型准确率标准差始终更高，如表4所示。这表明，在过拟合条件下，CLF 引入的方差正则化可能与模型固有的学习动态相冲突，从而非但没有抑制变异性，反而使其加剧。不过，平均准确率仍保持稳定，这一点值得注意，说明 CLF 并未对整体分类性能造成负面影响。这些发现表明，尽管 CLF 能有效控制规模适当的模型中的变异性，但在过拟合占主导的过参数化设置中，它无法带来同样的益处。

CLF 激活时长对训练性能的影响

我们研究了训练期间应用 CLF 的时长如何影响模型性能和变异性。由于 CLF 将交叉熵与额外的正则化相结合，关键问题在于它应在训练计划中保持激活多长时间。为评估这一点，我们在 CIFAR-10 上对 ResNet-14 进行了 500 轮训练，在最后 50、150、250、350 和 450 轮启用 CLF，而在前期阶段使用标准交叉熵。每种配置均采用 5 个随机种子重复实验，以评估变异性。延长训练计划确保模型充分收敛，从而能够公平评估 CLF 的影响。结果（图2）表明，延长 CLF 的应用时间可同时提升准确率和稳定性。当 CLF 应用于训练过程的较大部分时，模型能够受益于 StableLoss（SL）和方差惩罚损失（VPL）的稳定作用。相比之下，仅在训练末期短暂使用 CLF 几乎没有影响，因为模型已经基于交叉熵收敛，调整空间所剩无几。

使用 CLF 进行时间序列评估

我们使用 ETTh1 数据集和三种模型（Autoformer、NLinear 和 iTransformer），在预测跨度

表5：使用 CLF 进行时间序列预测时，预测变异性（MAE 的标准差）的降低。

模型	数据集	预测步长	MAE 标准差（不使用 CLF）	MAE 标准差（CLF）	降幅（%）
Autoformer	ETTh1	192	0.0151	0.0150	0.66
NLinear	ETTh1	192	0.0023	0.0021	8.70
iTransformer	ETTh1	192	0.0015	0.0018	−20.00
Autoformer	ETTh1	336	0.0146	0.0137	6.16
NLinear	ETTh1	336	0.0018	0.0013	27.78
iTransformer	ETTh1	336	0.0025	0.0026	–4.00
NLinear	ETTh1	720	0.0004	0.0003	25.00
iTransformer	ETTh1	720	0.0044	0.0040	9.09

为 192、336 和 720 的情况下，评估了 CLF 在时间序列预测中的有效性。由于时间序列预测的主要损失函数是均方误差（MSE），我们将 CLF 的稳定损失和方差惩罚损失（VPL）与 MSE 结合，以评估变异性的降低情况。如表5所示，在大多数情况下，CLF 显著降低了预测变异性，该变异性以平均绝对误差（MAE）的标准差衡量。NLinear 的改进最为显著，在预测跨度为 192、336 和 720 时，变异性分别降低了 8.70%、27.78% 和 25.00%。这可能是因为 NLinear 的设计简单且不依赖特定结构，缺少稳定机制，因此更容易受到随机变化的影响，也对 CLF 更为敏感。随着预测跨度延长，CLF 的益处也随之增加，因为更大的不确定性会加剧不稳定性。例如，Autoformer 在预测跨度为 192 时仅将 MAE 变异性降低了 0.66%，而在预测跨度为 336 时则降低了 6.16%。相比之下，iTransformer 受 CLF 的影响较小。在预测跨度为 192 和 336 时，变异性略有增加；在预测跨度为 720 时，则适度降低了 9.09%。这种有限的影响可能源于 iTransformer 的固有稳定性，其残差通路和时间建模能够缓解随机效应。

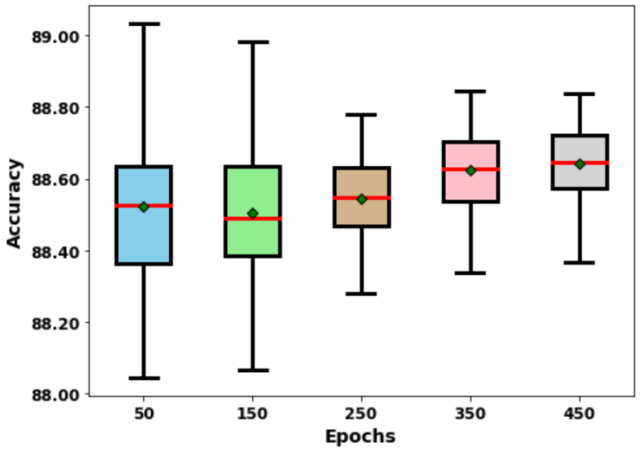

图2：在 CIFAR-10 上使用 ResNet-14 时，不同 CLF 激活时长下 5 次训练轮次的测试准确率分布。x 轴表示在固定的 500 轮训练计划中，CLF 处于激活状态的总轮数。

结论与未来工作

我们提出了一种自定义损失函数（CLF），通过减轻随机变异性来增强结果的鲁棒性。在图像分类和时间序列预测任务上的实验表明，该方法在不同架构下提升了性能一致性。在图像分类中，CLF 显著降低了准确率方差，尤其是在方差较大的模型中。例如，在 ShuffleNet-V2（CIFAR-100）上，CLF 将准确率标准差降低了 77%。相比之下，VGG-16（CIFAR-10）等稳定模型的提升较小。延长 CLF 的启用时间始终能提升鲁棒性，而在训练后期应用则影响有限。在时间序列预测中，CLF 降低了预测变异性，尤其是在较长的预测时域上。NLinear 的 MAE 变异性大幅下降，而 iTransformer 等较稳定模型的改善则较为有限。CLF 在训练动态不稳定或内部正则化较弱的场景中最为有效。此外，CLF 易于适配其他深度学习任务或架构，因此适用于广泛的实验设置。

局限性：每个数据集-架构组合都需要进行多次训练轮次，这限制了大规模测试，使这一领域尚未得到探索。此外，CLF 在训练期间使用固定的超参数，可能会降低性能，因为这些参数不会根据模型动态变化进行调整。根据训练信号动态调整 CLF，有助于模型更好地应对不断变化的条件，从而进一步提升稳定性和准确率。

参考文献

Ahmed, H.; 和 Lofstead, J. 2022. 管理随机性以实现可复现的机器学习。发表于第 *5* 届计算机系统实用可复现评估国际研讨会论文集，15–20。

Akiba, T.; Sano, S.; Yanase, T.; Ohta, T.; and Koyama, M. 2019. Optuna：新一代超参数优化框架。载于第 *25*届 *ACM SIGKDD*知识发现与数据挖掘国际会议论文集

挖掘，KDD ’19，2623–2631。美国纽约州纽约：计算机协会。ISBN 9781450362016。Bouthillier, X.; Laurent, C.; and Vincent, P. 2019。不可复现的研究是可复现的。载于国际机器学习会议，725–734。PMLR。Gundersen, O. E.; Shamsaliei, S.; Kjærnli, H. S.; and Langseth, H. 2023。关于如何报告涉及神经网络和随机性的模型比较研究所得的稳健可信结论。载于*2023*年 *ACM*可复现性与可重复性会议论文集，37–61。He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016。用于图像识别的深度残差学习。载于*IEEE*计算机视觉与模式识别会议论文集，770–778。Ji, Y.; Kaestner, D.; Wirth, O.; and Wressnegger, C. 2023。随机性是万恶之源：更可靠地评估深度主动学习。载于*IEEE/CVF*冬季计算机视觉应用会议论文集，3943–3952。Kaur, D.; Uslu, S.; Rittichier, K. J.; and Durresi, A. 2022。可信人工智能：综述。*ACM*计算调查，55(2)。Krizhevsky, A.; Hinton, G.; 等。2009。从微型图像中学习多层特征。Liu, Y.; Hu, T.; Zhang, H.; Wu, H.; Wang, S.; Ma, L.; and Long, M. 2023。iTransformer：反转 Transformer 对时间序列预测十分有效。*arXiv*预印本 *arXiv:2310.06625*。Ma, N.; Zhang, X.; Zheng, H.; and Sun, J. 2018。ShuffleNet V2：高效 CNN 架构设计实用指南。载于 Ferrari, V.; Hebert, M.; Sminchisescu, C.; and Weiss, Y., 编，计算机视觉—*ECCV 2018*——第*15*届欧洲计算机视觉会议，德国慕尼黑，*2018*年*9*月*8*日至*14*日，论文集，第*XIV*部分，第11218卷计算机科学讲义，122–138。施普林格。Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; 等。2019。PyTorch：一种命令式风格、高性能的深度学习库。神经信息处理系统进展，32。Pham, H. V.; Qian, S.; Wang, J.; Lutellier, T.; Rosenthal, J.; Tan, L.; Yu, Y.; and Nagappan, N. 2020。深度学习软件系统训练中的问题与机遇：方差分析。载于第*35*届 *IEEE/ACM*自动化软件工程国际会议论文集，771–783。Picard, D. 2021。Torch。manual_seed（3407）就是你所需要的一切：随机种子对计算机视觉深度学习架构的影响。*arXiv*预印本 *arXiv:2109.08203*。Renard, F.; Guedria, S.; Palma, N. D.; and Vuillerme, N. 2020。医学图像分割深度学习中的变异性与可复现性。《科学报告》，10(1)：13724。Simonyan, K.; and Zisserman, A. 2015。用于大规模图像识别的超深卷积网络。载于 Bengio, Y.; and LeCun, Y., 编，第*3*届国际学习表征会议（*ICLR 2015*），美国加利福尼亚州圣迭戈，*2015*年*5*月*7*日至*9*日，会议论文集。

Summers, C.; and Dinneen, M. J. 2021. 神经网络优化中的非确定性与不稳定性。载于国际机器学习会议，9913–9922。PMLR。Wu, H.; Xu, J.; Wang, J.; and Long, M. 2021. Autoformer：用于长期序列预测的自相关分解 Transformer。载于第*35*届神经信息处理系统国际会议论文集，NIPS ’21。美国纽约州雷德胡克：Curran Associates Inc. ISBN 9781713845393。Zeng, A.; Chen, M.; Zhang, L.; and Xu, Q. 2023. Transformer 是否适用于时间序列预测？载于第三十七届 *AAAI* 人工智能大会、第三十五届人工智能创新应用大会和第十三届人工智能教育进展研讨会论文集，AAAI’23/IAAI’23/EAAI’23。AAAI出版社。ISBN 978-1- 57735-880-0。Zhou, H.; Zhang, S.; Peng, J.; Zhang, S.; Li, J.; Xiong, H.; and Zhang, W. 2021. Informer：超越高效 Transformer 的长序列时间序列预测。载于第三十五届人工智能大会（*AAAI 2021*），线上会议，第35卷，11106–11115。AAAI出版社。Zhuang, D.; Zhang, X.; Song, S.; and Hooker, S. 2022. 神经网络训练中的随机性：工具链影响的特征分析。机器学习与系统论文集，4：316–336。