模型推理中的调度方法探究_250319

以下是本周的新进展，文档末尾附上了过去进展情况

实验部分推进

part1：CIFAR-100上的部署

我们之前参考其他文献得出CIFAR-100准确率训练到80%左右基本就可以用了，但是从60%到80%确实经历了一周。

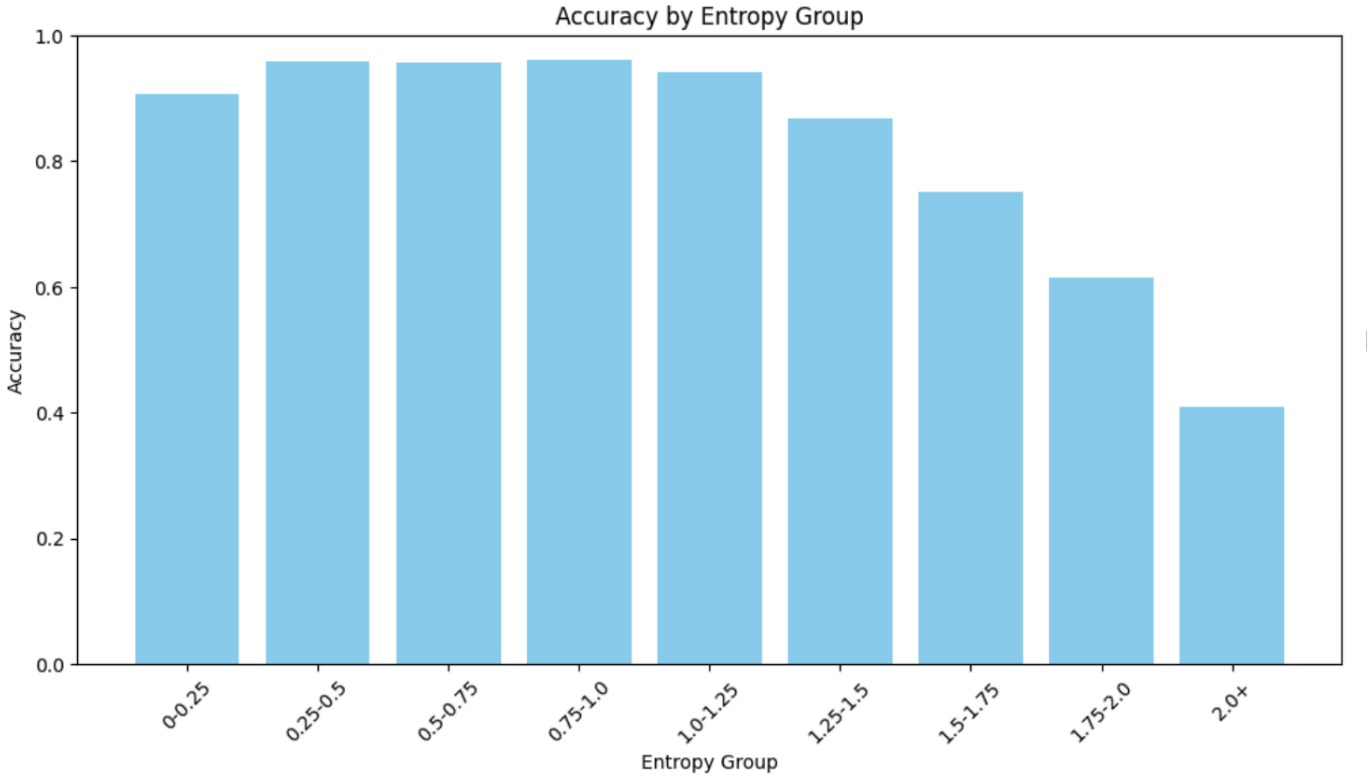
初始程序出现了对train_dataset的严重过拟合（99.9%）但是泛化能力及其差，在test_dataset上的表现大约就是60%（第二次训练调小batch_size达到了64%，但是总体还是很不准确）

在添加了许多Data Augment 方法之后效果显著提升，具体表现在：训练过程中在训练集上的准确率总是小于测试集上的，并最终在测试集上的准确率达到了80%，可以说恰好符合需求

```
100%|████████| 98/98 [00:21<00:00,  4.60batch/s, acc=74.3, loss=1.28, lr=4.01e-5]
Epoch 196: Test Acc: 80.57% | Best Acc: 80.75%
100%|████████| 98/98 [00:21<00:00,  4.59batch/s, acc=74.3, loss=1.46, lr=2.26e-5]
Epoch 197: Test Acc: 80.54% | Best Acc: 80.75%
100%|██████████| 98/98 [00:21<00:00,  4.61batch/s, acc=73.2, loss=1.82, lr=1e-5]
Epoch 198: Test Acc: 80.50% | Best Acc: 80.75%
100%|████████| 98/98 [00:21<00:00,  4.60batch/s, acc=75.6, loss=1.08, lr=2.55e-6]
Epoch 199: Test Acc: 80.74% | Best Acc: 80.75%
100%|████████| 98/98 [00:21<00:00,  4.62batch/s, acc=70.7, loss=2.09, lr=8.03e-8]
Epoch 200: Test Acc: 80.44% | Best Acc: 80.75%
Training Complete. Best Accuracy: 80.75%
```

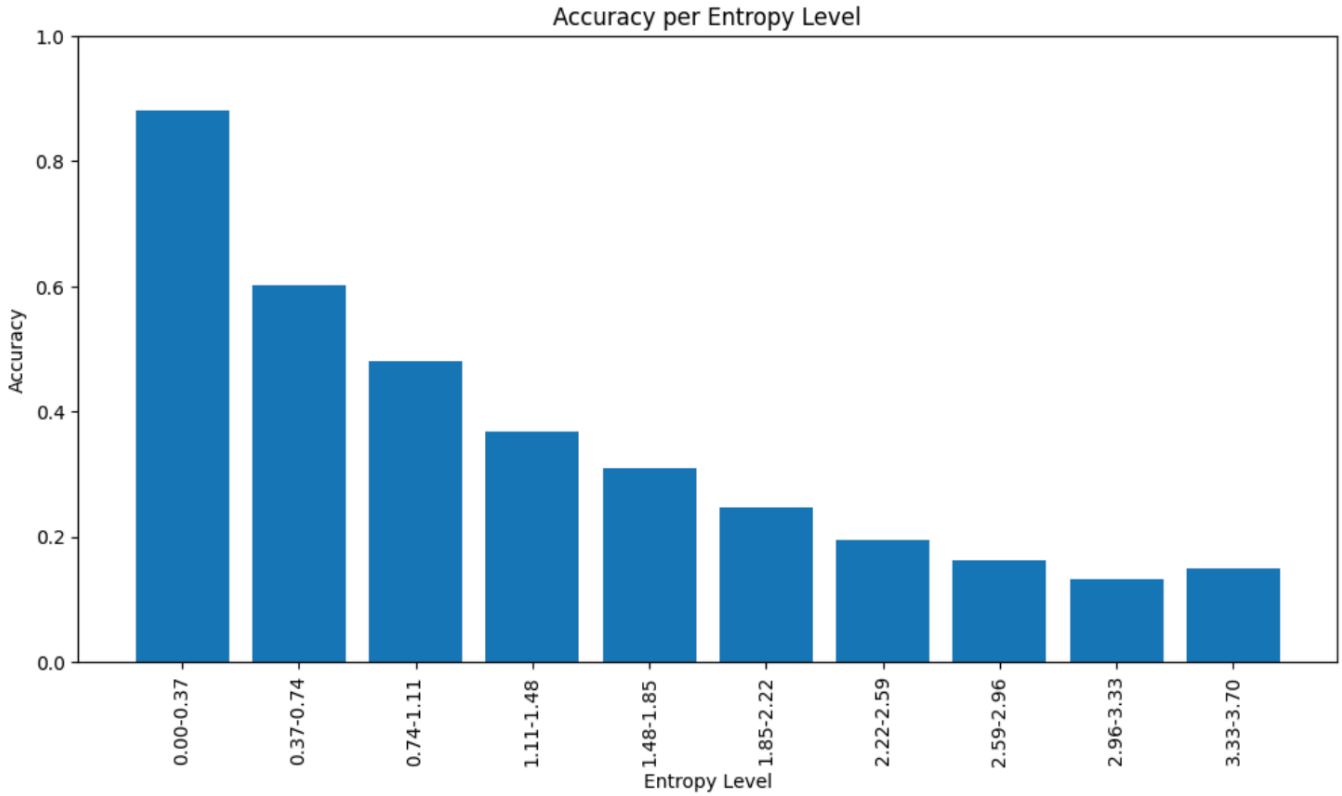
仿照之前代码的进展，我也做了对于该模型的熵等级划分与准确率情况：

```
Entropy Group 0–0.25: Accuracy = 0.9062
Entropy Group 0.25–0.5: Accuracy = 0.9595
Entropy Group 0.5–0.75: Accuracy = 0.9563
Entropy Group 0.75–1.0: Accuracy = 0.9602
Entropy Group 1.0–1.25: Accuracy = 0.9422
Entropy Group 1.25–1.5: Accuracy = 0.8682
Entropy Group 1.5–1.75: Accuracy = 0.7510
Entropy Group 1.75–2.0: Accuracy = 0.6155
Entropy Group 2.0+: Accuracy = 0.4096
```



原预实验中采用了从teachermodel中蒸馏出的student_model，但是我们这里就直接采用了先前训练的准确率大概为60%左右的模型，记为student_net (实际与模型蒸馏无关)，下面是stu模型的不同熵等级的准确率情况

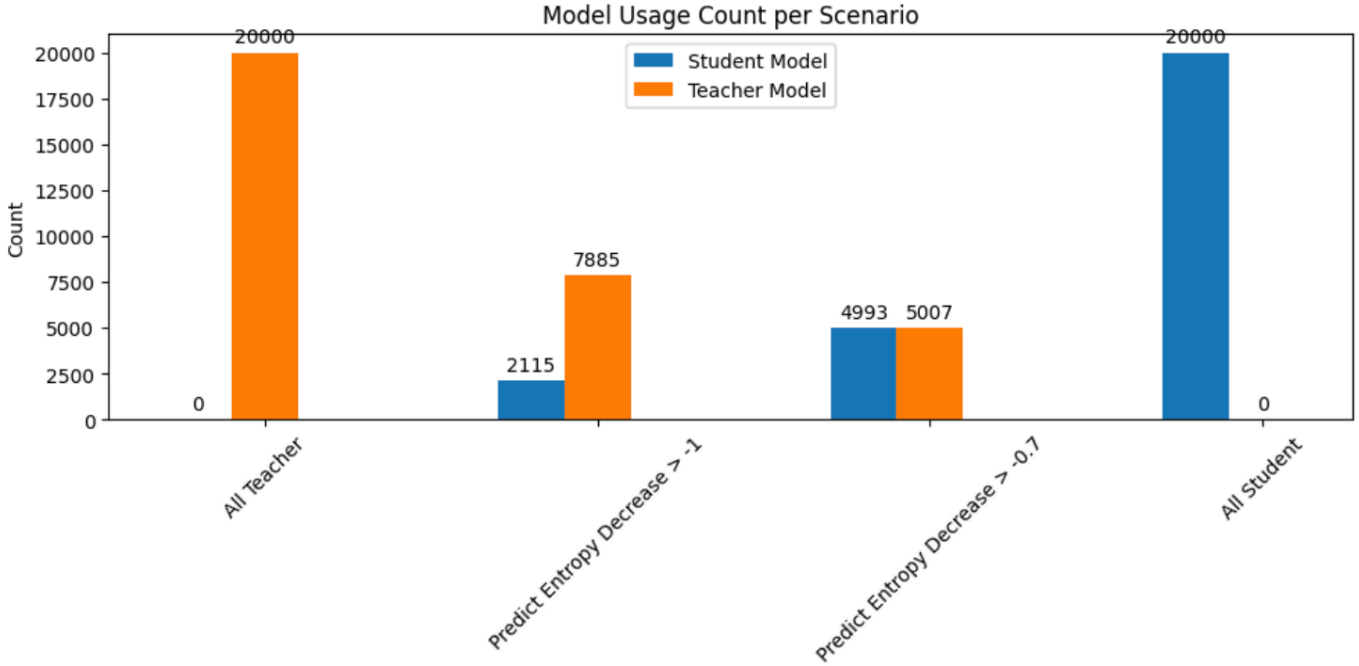
```
Model accuracy on test set: 60.63%
```

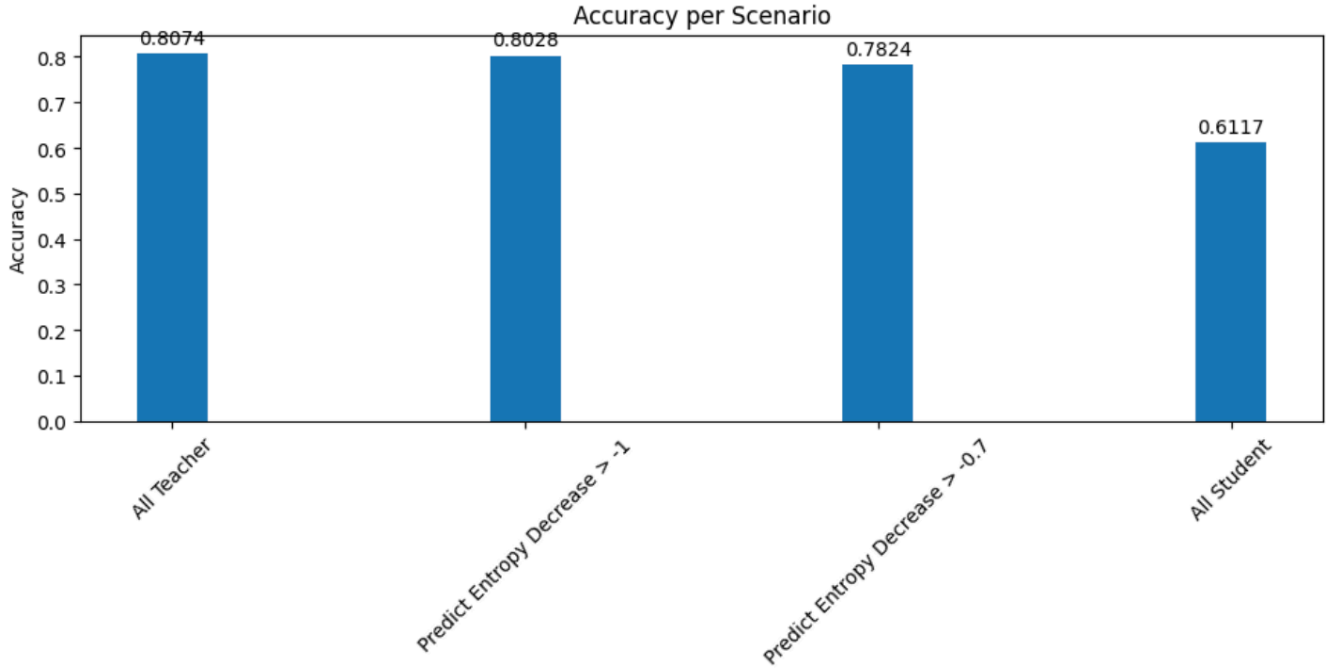


最后，与预实验一样，我们训练了熵下降预测机，但是，与预实验效果一样，这个熵下降的拟合程度并不是很好，loss比较大，后续对于这个的结构可能还需要进一步探索，目前50个epoch基本收敛

```
Epoch 43/50 | Loss: 0.0709
Epoch 44/50 | Loss: 0.0709
Epoch 45/50 | Loss: 0.0707
Epoch 46/50 | Loss: 0.0705
Epoch 47/50 | Loss: 0.0706
Epoch 48/50 | Loss: 0.0703
Epoch 49/50 | Loss: 0.0706
Epoch 50/50 | Loss: 0.0705
```

```
All Teacher:
  Student model output count: 0 (0.00%)
  Teacher model output count: 20000 (100.00%)
  Accuracy: 80.74%
Predict Entropy Decrease > -1:
  Student model output count: 2115 (21.15%)
  Teacher model output count: 7885 (78.85%)
  Accuracy: 80.28%
Predict Entropy Decrease > -0.7:
  Student model output count: 4993 (49.93%)
  Teacher model output count: 5007 (50.07%)
  Accuracy: 78.24%
All Student:
  Student model output count: 20000 (100.00%)
  Teacher model output count: 0 (0.00%)
  Accuracy: 61.17%
```





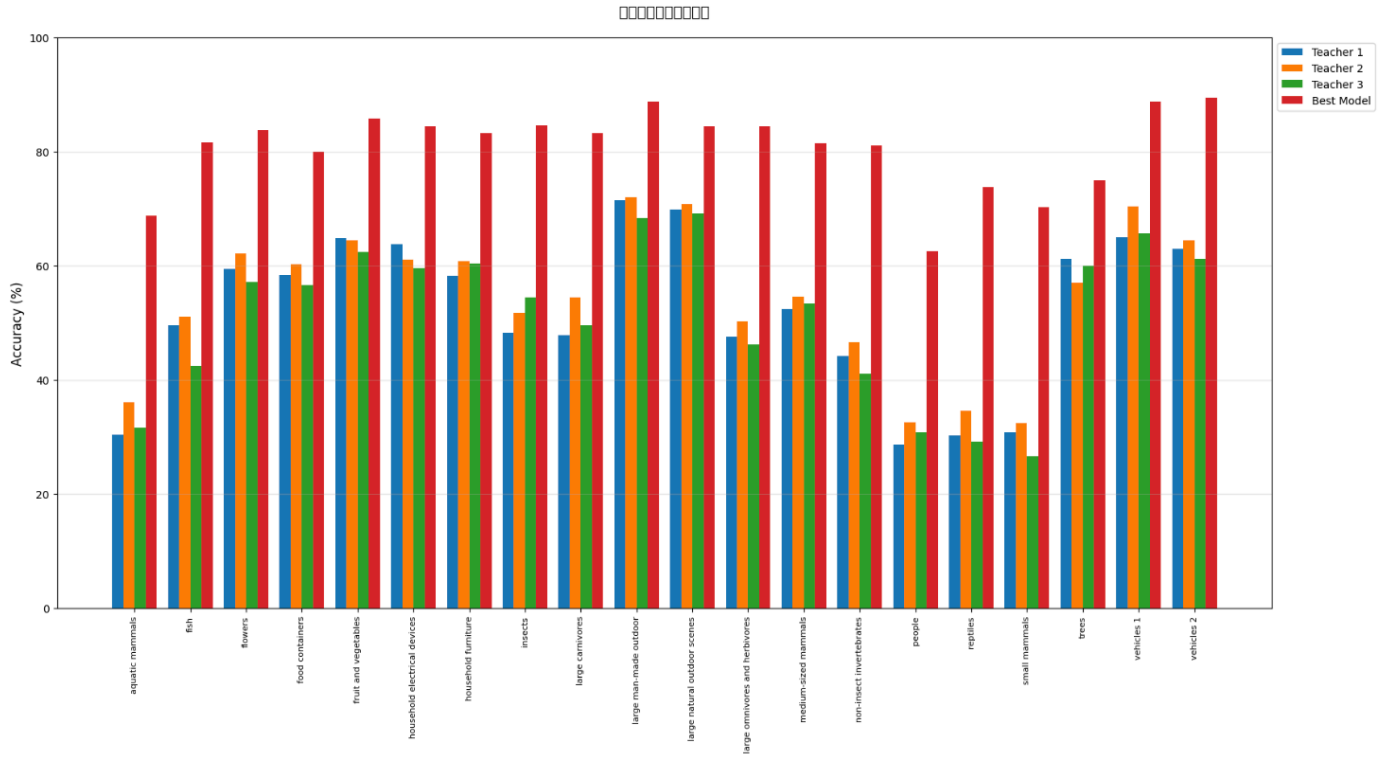
实验结果：在准确率下降小于1%的情况，我们可以过滤掉20%的输出，在准确率下降小于3%的情况，我们过滤50%的输出。

目前阶段最大的问题主要是对于entropy predictor的训练，还需探索。

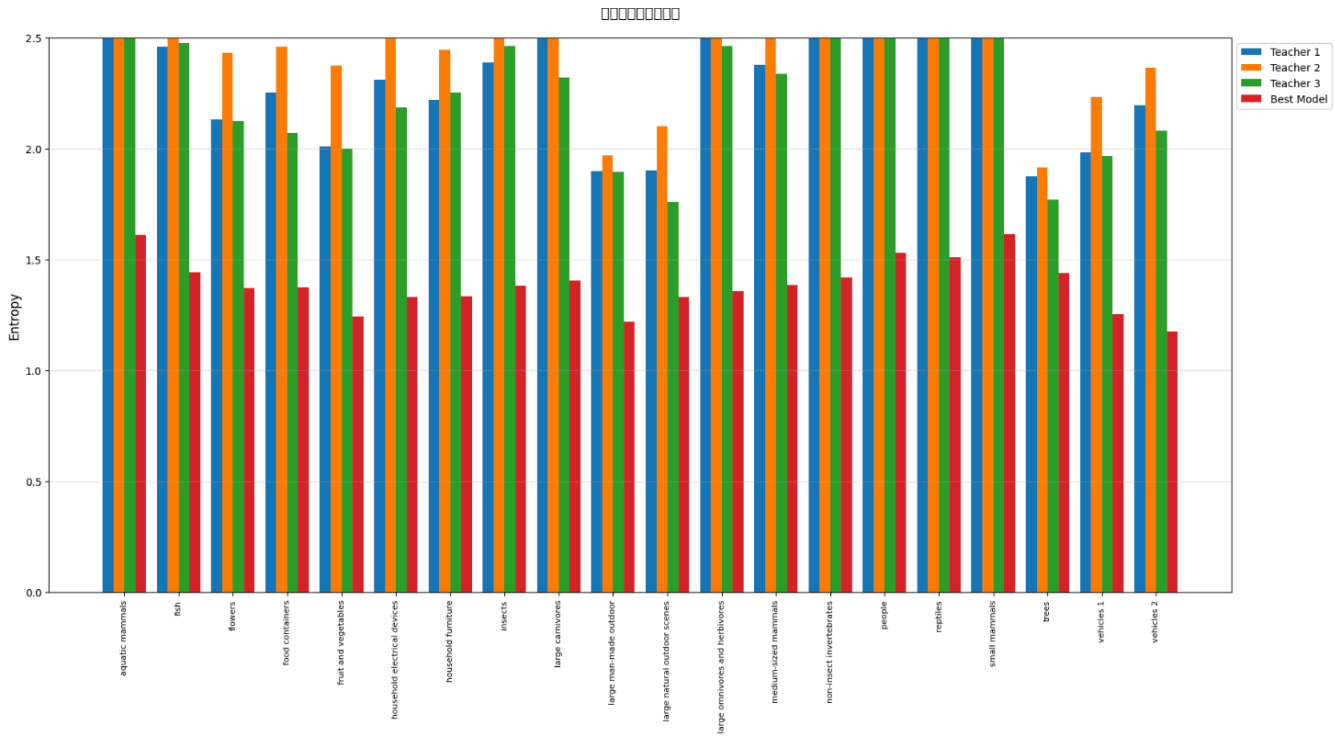
multi-task部分

下面讲述对于先前计划的多任务阶段的实验进展

先前计划对于同样的图像分类模型，仅仅做出对数据集大小的修改，就希望使其具有对于某个superclass的高表现，但是在第一轮实验中并未成功（约历时4h），第一轮实验结果表现为，模型对于所有类的识别都普遍偏低，无法从对于各个superclass的熵与准确率中表现出特定规律



该图为对于CIFAR-100所有superclass的准确率分布，teacher123分别为对某个superclass特化后的训练结果，best_model为第一个实验的模型训练结果。



该图为各个模型在各个superclass上的平均熵值。

于是在第二轮实验中，我们改变了策略，将三个模型的训练均划分为两个阶段，第一阶段，20个epoch，训练所有类别，第二阶段，180个epoch，仅训练特定target dataset，但是仍表现不佳

以下是过去进展

模型推理中的调度方法探究_250305

理论部分

费诺不等式Fano

https://eggplantisme.github.io/tags/#algorithm（有证明）

Fano不等式的定义

对于分类问题，假设输入 X 和输出 Y 是随机变量，其中 Y 有 K 个可能的类别。Fano不等式描述了条件熵 $H(Y|X)$ 与分类错误率 P_e 之间的关系：

$$H(Y|X) \leq H(P_e) + P_e \log(K-1) \tag{1}$$

其中：

- $H(Y|X)$ 是条件熵，表示在已知 X 的情况下 Y 的不确定性。
- $P_e = P(\hat{Y} \neq Y)$ 是分类错误率，即模型预测的类别 $\hat{Y}$ 与真实类别 Y 不一致的概率。
- $H(P_e)$ 是二元熵函数，定义为 $H(P_e) = -P_e \log P_e - (1-P_e)\log(1-P_e)$。

Fano不等式表明：

- **条件熵 $H(Y|X)$ 越高，分类错误率 P_e 的下界越高。**

$$\mathrm{Acc} \leq 1 - \frac{H(Y|X) - H(P_e)}{\log(K-1)} \tag{2}$$

实验计划

参考类比文献

Effective Data Selection and Replay for Unsupervised Continual Learning

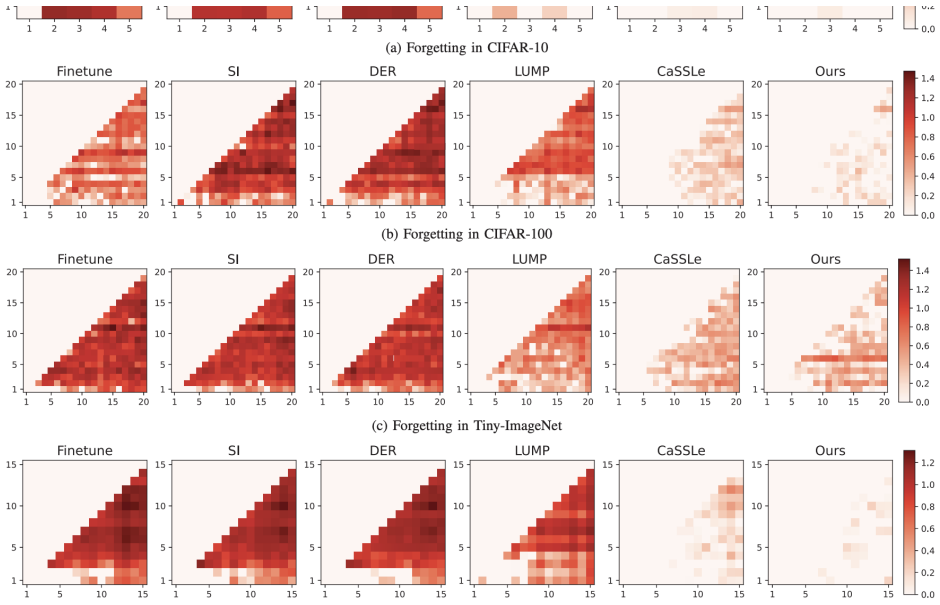
https://ieeexplore.ieee.org/document/10598102

TABLE II
DATA SET SUMMARY. *Positive Rate* IS THE RATIO OF TABULAR DATA WITH POSITIVE CLASSES.

	Name	#Train data	#Test data	#Classes	Image size
Image Data	CIFAR-10	50,000	10,000	10	32*32
	CIFAR-100	50,000	10,000	100	32*32
	Tiny-ImageNet	50,000	10,000	100	64*64
	DomainNet-real	120,906	52,041	345	64*64
	Name	**Size**	**#Input dim.**	**#Classes**	**Positive ratio**
Tabular Data	Bank	45,211	16	2	11.70%
	Shoppers	12,330	17	2	15.47%
	Income	32,561	14	2	24.08%
	BlastChar	7,043	20	2	26.54%
	Shrutime	10,000	10	2	20.37%

TABLE III
THE MODEL COMPARISON ON FOUR BENCHMARK IMAGE DATA SETS. FOR METHODS WITH MEMORY, THE LIMIT IS 256 FOR CIFAR-10, 640 FOR CIFAR-100 AND TINY-IMAGENET, AND 960 FOR DOMAINNET-REAL. MULTITASK IS EXCLUDED FROM COMPARISON SINCE IT ACCESSES OLD DATA.

Model	**CIFAR-10**		**CIFAR-100**		**Tiny-ImageNet**		**DomainNet-real**	
	Acc ↑	*Fgt* ↓	*Acc* ↑	*Fgt* ↓	*Acc* ↑	*Fgt* ↓	*Acc* ↑	*Fgt* ↓
Multitask	95.76 ± 0.08	-	86.31 ± 0.38	-	85.09 ± 0.01	-	75.37±0.07	-
Finetune	89.02 ± 0.05	5.79 ± 0.07	75.88 ± 2.18	5.23 ± 3.96	71.03 ± 1.31	10.01 ± 0.73	68.46 ± 0.16	7.10 ± 0.07
SI [54]	91.06 ± 0.08	3.79 ± 0.11	78.93 ± 1.15	8.37 ± 1.30	71.37 ± 0.82	9.99 ± 0.47	68.81 ± 0.06	6.57 ± 0.10
DER [60]	90.17 ± 0.62	5.15 ± 0.78	76.70 ± 0.45	9.21 ± 0.69	72.78 ± 0.59	8.58 ± 0.36	68.96 ± 0.23	6.79 ± 0.19
LUMP [24]	91.05 ± 0.37	2.11 ± 0.23	83.41 ± 0.14	4.12 ± 0.17	77.58 ± 0.24	4.24 ± 0.34	66.54 ± 0.06	6.11 ± 0.57
CaSSLe [33]	92.28 ± 0.13	0.62 ± 0.05	83.67 ± 0.35	1.33 ± 0.15	78.76 ± 0.25	2.48 ± 0.40	70.78 ± 0.23	0.55 ± 0.12
Our EDSR	**93.14 ± 0.08**	**0.12 ± 0.06**	**85.42 ± 0.20**	**0.57 ± 0.14**	**81.19 ± 0.22**	**1.77 ± 0.28**	**71.58 ± 0.27**	**0.24 ± 0.11**



(a) Forgetting in CIFAR-10

(b) Forgetting in CIFAR-100

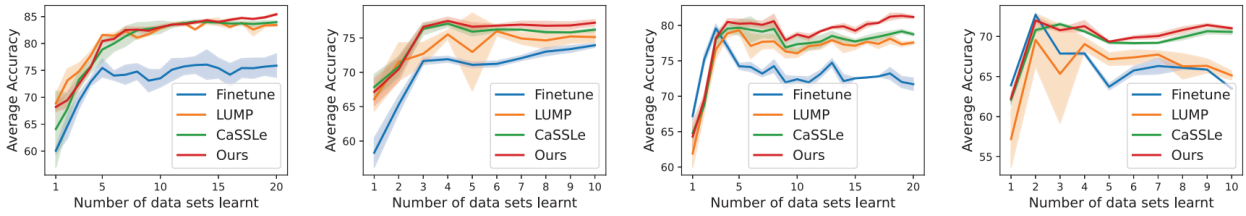
(c) Forgetting in Tiny-ImageNet

(d) Forgetting in DomainNet-real

Fig. 4. The forgetting matrix $\mathbf{F}$ of the models learned on four image data sets. $F_{i,j}$ evaluates the accuracy decrease on the old data set X^j after learning the

TABLE V
EXPERIMENTS ON DIFFERENT STORAGE METHODS AND WHETHER ADD NOISES. MEMORY SIZES ARE THE SAME FOR ALL THE METHODS, 256 FOR CIFAR-10 AND 640 FOR CIFAR-100 AND TINY-IMAGENET.

DataSet		No Replay (CaSSLe)	Random	K-means [80]	Min-Var [61]	Distant [79]	High Entropy
			Replay with $\mathcal{L}_{dis}(\cdot)$				
CIFAR-10	$Acc \uparrow$	92.28 ± 0.13	93.04 ± 0.11	93.08 ± 0.19	92.96 ± 0.15	93.03 ± 0.03	**93.17 ± 0.26**
	$Fgt \downarrow$	0.62 ± 0.05	**0.07 ± 0.05**	0.20 ± 0.09	0.19 ± 0.07	0.14 ± 0.05	0.08 ± 0.01
CIFAR-100	$Acc \uparrow$	83.67 ± 0.35	84.94 ± 0.68	84.71 ± 0.52	85.14 ± 0.23	84.87 ± 0.41	**85.23 ± 0.31**
	$Fgt \downarrow$	1.33 ± 0.15	0.89 ± 0.13	0.75 ± 0.20	**0.48 ± 0.10**	0.81 ± 0.09	0.73 ± 0.03
Tiny-ImageNet	$Acc \uparrow$	78.76 ± 0.25	79.50 ± 0.35	**80.36 ± 0.16**	79.83 ± 0.35	79.55 ± 0.17	80.27 ± 0.56
	$Fgt \downarrow$	2.48 ± 0.40	2.02 ± 0.20	1.45 ± 0.19	1.41 ± 0.11	1.77 ± 0.33	**1.31 ± 0.41**
			Replay with $\mathcal{L}_{rpl}(\cdot)$				
CIFAR-10	$Acc \uparrow$	92.28 ± 0.13	92.96 ± 0.08	92.90 ± 0.13	93.07 ± 0.19	93.05 ± 0.24	**93.14 ± 0.08**
	$Fgt \downarrow$	0.62 ± 0.05	0.06 ± 0.02	0.18 ± 0.02	0.20 ± 0.09	0.16 ± 0.09	**0.12 ± 0.06**
CIFAR-100	$Acc \uparrow$	83.67 ± 0.35	85.04 ± 0.39	85.22 ± 0.27	85.26 ± 0.38	84.95 ± 0.45	**85.42 ± 0.20**
	$Fgt \downarrow$	1.33 ± 0.15	0.70 ± 0.11	0.72 ± 0.19	0.71 ± 0.02	0.91 ± 0.11	**0.57 ± 0.14**
Tiny-ImageNet	$Acc \uparrow$	78.76 ± 0.25	79.81 ± 0.22	80.66 ± 0.35	79.89 ± 0.10	79.99 ± 0.59	**81.19 ± 0.22**
	$Fgt \downarrow$	2.48 ± 0.40	1.54 ± 0.19	1.79 ± 0.35	**1.51 ± 0.09**	1.70 ± 0.41	1.77 ± 0.28



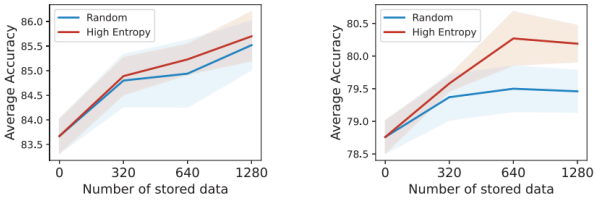
(a) CIFAR-100: 20 subsets (b) CIFAR-100: 10 subsets (c) Tiny-ImageNet: 20 subsets (d) Tiny-ImageNet: 10 subsets

Fig. 7. Experiments on learning CIFAR-100 and Tiny-ImageNet in two settings, 20 subsets of 5 classes and 10 subsets of 10 classes.

TABLE VI
EXPERIMENTS OF DIFFERENT CSSL LOSSES. AVERAGE ACCURACY *Acc* IS REPORTED.

Methods	SimSiam		BarlowTwins	
	CIFAR-100	Tiny-ImageNet	CIFAR-100	Tiny-ImageNet
Multitask	86.31 ± 0.38	85.09 ± 0.01	87.16 ± 0.52	83.01 ± 0.10
Finetune	75.51 ± 0.64	57.13 ± 10.31	71.97 ± 0.54	68.81 ± 0.29
LUMP	83.41 ± 0.14	77.58 ± 0.24	**83.14 ± 0.87**	75.02 ± 0.36
CaSSLe	83.67 ± 0.35	78.76 ± 0.25	79.60 ± 0.80	70.30 ± 1.44
Ours	**85.42 ± 0.20**	**81.19 ± 0.22**	80.66 ± 1.67	**75.59 ± 1.11**

1) Hyper-parameter Study: Here we discuss how the se-



(a) CIFAR-100 (b) Tiny-ImageNet

Fig. 8. Experiments on the effect of different amounts of stored data.

基础任务

准备采取 CIFAR-10 CIFAR-100 ImageNet 做对照

（此处的研究目标也要转化为与分类类别数量相关，来说明我们的方法在更多类别上的分类任务也是能work的？）

训练三个模型，用resnet做baseline，准确率在CIFAR-100上要达到80+

然后对其蒸馏出一个小模型，准确率要在60左右

（上述先在CIFAR-100上尝试）

然后训练基础的调度程序

再寻找一个开源的训练好的优秀模型，最好准确率要在90左右?

https://paperswithcode.com/sota/image-classification-on-cifar-100

对于阈值的讨论

我们训练过程中的调度程序要达到预测熵下降超过a之后才会选择调度，那么对于a的讨论是否也是必要进行的?

多任务

对resnet做变种，为了使其特化其某方面的能力，预计采用训练集上的变化产生能力差异性

Superclass	Classes
aquatic mammals	beaver, dolphin, otter, seal, whale
fish	aquarium fish, flatfish, ray, shark, trout
flowers	orchids, poppies, roses, sunflowers, tulips
food containers	bottles, bowls, cans, cups, plates
fruit and vegetables	apples, mushrooms, oranges, pears, sweet peppers
household electrical devices	clock, computer keyboard, lamp, telephone, television
household furniture	bed, chair, couch, table, wardrobe
insects	bee, beetle, butterfly, caterpillar, cockroach
large carnivores	bear, leopard, lion, tiger, wolf
large man-made outdoor things	bridge, castle, house, road, skyscraper
large natural outdoor scenes	cloud, forest, mountain, plain, sea
large omnivores and herbivores	camel, cattle, chimpanzee, elephant, kangaroo
medium-sized mammals	fox, porcupine, possum, raccoon, skunk
non-insect invertebrates	crab, lobster, snail, spider, worm
people	baby, boy, girl, man, woman
reptiles	crocodile, dinosaur, lizard, snake, turtle
small mammals	hamster, mouse, rabbit, shrew, squirrel
trees	maple, oak, palm, pine, willow
vehicles 1	bicycle, bus, motorcycle, pickup truck, train
vehicles 2	lawn-mower, rocket, streetcar, tank, tractor

对于CIFAR-100的这些类别，我们采取三个superclass，将其的训练数据取满

但是使其分别在 flowers，trees，people这三个类的准确率要足够高

拟定具体方法是，对模型A，使其flowers这个super class上的训练样例要大于其他集的n倍（n待定）

多版本

可以对于前述cifar-100的多个版本增加几个版本，添加到输入空间看看其表现是上升还是下降?

多模态

物体检测？llm生成?

对于生成的token的熵也进行记录？采用GPT-2与GPT-3的结合？

真实场景

目前还不清楚如何实验

呈现方式

对于所有任务，要体现出每个任务的准确率，调度次数，执行时间

对于熵值bar的实验，可以以一个a不断变化之后，准确率与调度时间不断变化的图片（计算量会有些大？）

对于多任务情境，可以做一个ground truth与调度分类结果的热图，对于三个调整后的模型都要做，总之主要**体现三个模型在分类为什么的图像上被调度的次数最多**

计划一月内完成基础任务与多任务部分的实验

future work

对于多模态多版本上的模型调度的实验设计需要再好好想想

真实场景方面还没有应用

discussion

几个基础的调度策略

$$
\begin{aligned}
&\text{假设模型有}n\text{个，其各自的准确率为}acc_1, acc_2, \ldots, acc_n \\
&\text{对于}n=2\text{的情况，最坏策略准确率为}acc_{worst} = \max(0, acc_1 + acc_2 - 1) \\
&\text{对于}n\text{的情况，随机策略准确率为}acc_{random} = \frac{1}{n}\sum_{i=1}^{n} acc_i
\end{aligned}
\tag{3}
$$

两种可以证明我们的方法work的判据：

1. acc超越 $\max_{1\le i\le n} acc_i$
2. 添加一个表现极低的模型而总体不受影响

Past_250302

该部分是过去进展的汇报

Motivation

大模型更新迭代速度快，模型更新后，需要对所有待测样本都进行重新推理，消耗计算资源（如手机相册推理单张图片需要3s）

在先前研究中，过往的执行结果就被直接忽略了，但是我认为这一部分的信息有很多的价值。

Related Work

（1）权重/激活量化压缩。8/4-bit。

（2）Continuos Batching。在推理时，将多个请求整合为一批，在模型中并行处理，充分利用计算资源，提高处理效率。比如在文本生成任务中，同时处理多个用户输入文本，一次性生成多个输出。

（3）IO/FLOPs-Aware/稀疏化 Attention。

主要关注于**单次推理**的加速。

主动学习

这些论文主要利用熵作为**不确定性度量**来选择信息量大的样本，但并未直接量化熵的下降量：

- **熵作为不确定性指标**：大多数论文通过熵值高低选择最不确定的样本，但未计算选择这些样本后模型整体熵的变化量。或许是因为标注后的样本的熵就会是0 ，所以谈熵下降量没有意义。
- **熵最小化或正则化**：部分论文通过最小化预测熵或结合熵正则化来提升模型性能，但关注的是模型预测的置信度，而非熵下降的动态过程。
- **停止准则中的熵阈值**：使用熵低于阈值作为停止条件，但未将熵下降量作为主动学习的核心策略。

下述三篇论文分别对应上述三个方向

Title: "Active Learning with Uncertainty Sampling for Text Classification"

Conference/Journal: AAAI
Abstract: This paper explores the use of entropy-based uncertainty sampling in active learning for text classification tasks. The authors propose a method that selects the most informative samples based on the entropy of the predicted class probabilities. The approach is evaluated on several benchmark datasets, demonstrating significant improvements in classification accuracy with fewer labeled samples compared to passive learning.

Title: "Active Learning with Entropy Minimization"

Conference/Journal: KDD
Abstract: This paper introduces an active learning framework that minimizes the entropy of the model's predictions over the unlabeled data. The authors argue that minimizing entropy leads to more confident predictions and, consequently, more effective sample selection. The method is tested on several real-world datasets, demonstrating its effectiveness in reducing the labeling effort while maintaining high accuracy.

Title: "Entropy-Based Active Learning for Reinforcement Learning"

Conference/Journal: CoRL
Abstract: This paper explores the use of entropy-based active learning in reinforcement learning, where the entropy of the policy's action distribution is used to guide the selection of informative states for exploration. The authors show that their approach leads to more efficient learning and better policy performance in complex environments.

Bayesian Active Learning for Classification and Preference Learning

Learning with Bounded Instance- and Label-Dependent Label Noise

Methodology

由于笔记本算力限制，采用CIFAR-10数据集，使用两个不同表现的模型，后将其称为模型A与模型B。实际采用中称其为teacher 与 student

- 两个模型的表现

```
Accuracy of the A network on the 10000 test images: 72.63%
Accuracy of the B network on the 10000 test images: 67.17%

A模型的推理时间: 0.014664 秒
B模型的推理时间: 0.000200 秒
```

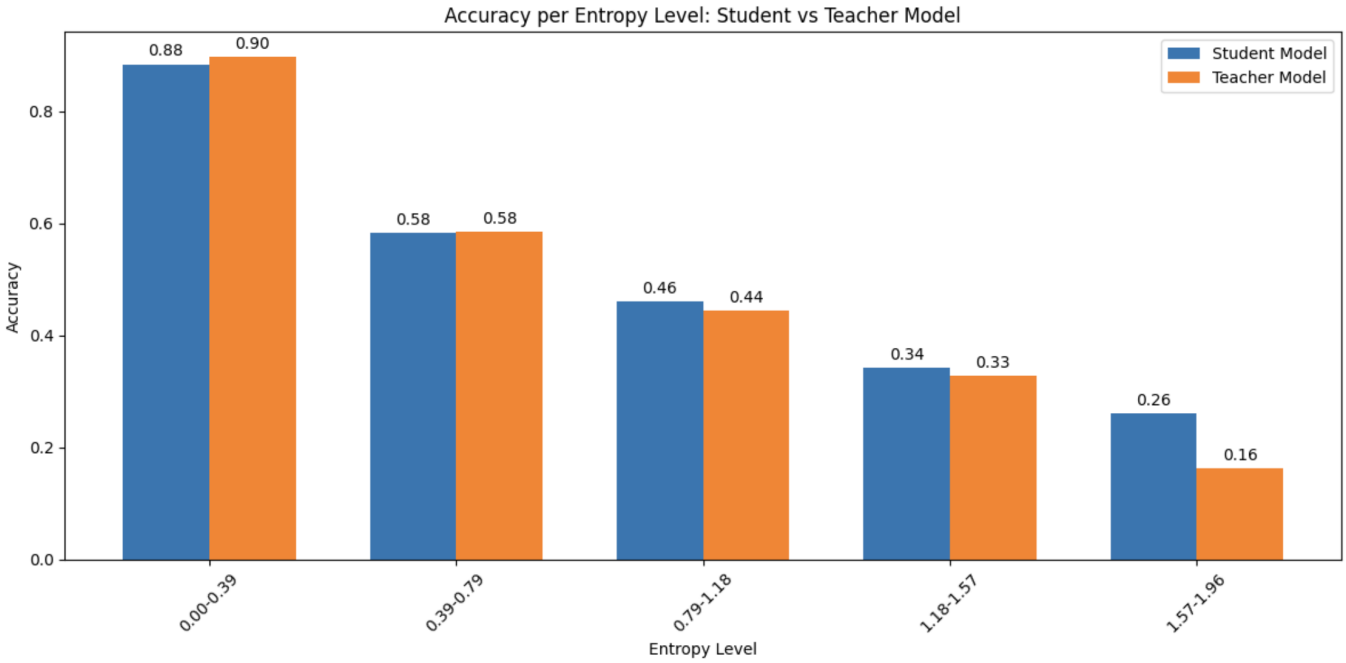
可以看到，两个模型准确率有差别，同样在推理速度上也有显著差别，那么我们实际上就需要做一个“准确率”与“推理时间“的权衡

熵的引入

我们的总体思路是，创建一个调度程序，能够通过过去的执行结果判断我是否要用更复杂的模型更新该程序来获取更高的准确率。

进一步地，我们只希望更新小模型最不确定的结果，或者是大模型提升效果显著的那些样例。

于是我们做实验得到下面的图，我们可以发现，无论是对大模型还是小模型，其结果分布的熵增加，都会导致准确率的下降，那么这里我的想法就是，由于对于单个样本讨论准确率没有意义，因为它只能是错或者对两种状态，而熵可以更好地体现其不确定度，所以我在此处将最大化准确率的任务 转化为 最小化熵的任务。（此处应该有相关理论研究，需要进一步理论证明准确率与熵的关系）



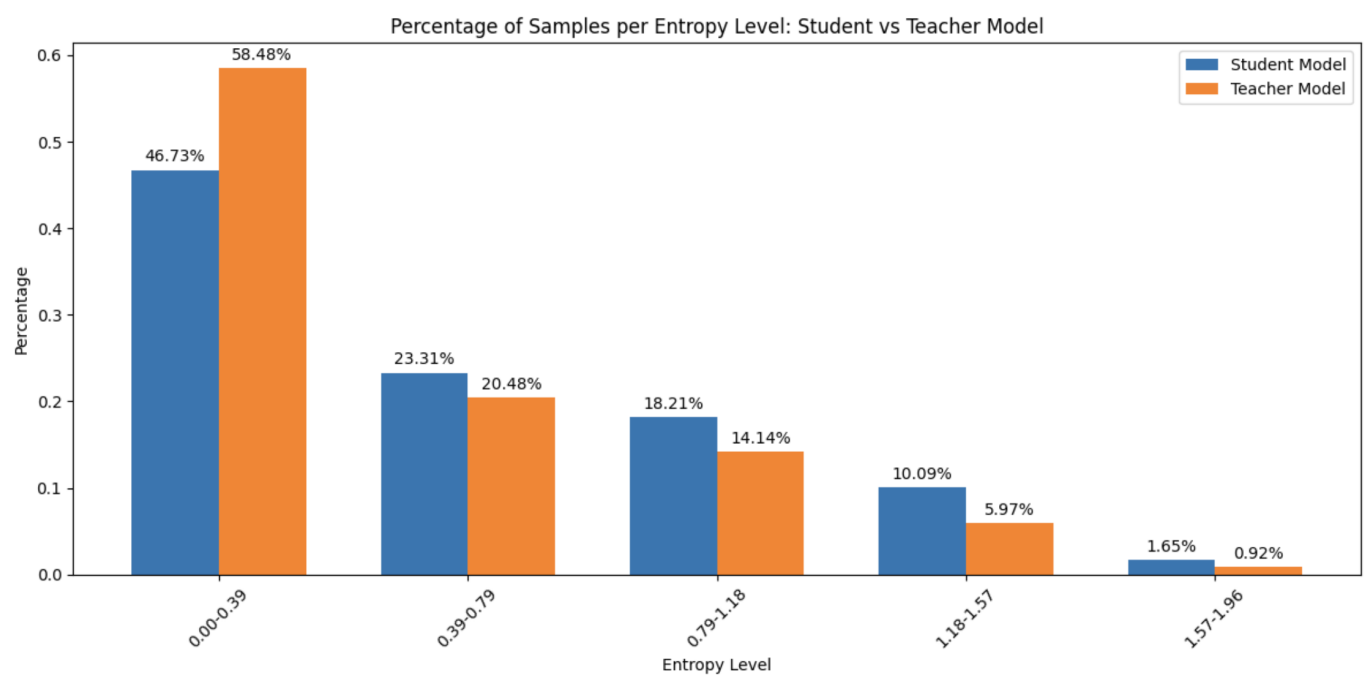
调度方法初探

为了最小化熵，我对10000个样本下学生模型执行结果与教师模型执行做了分析。

我对熵从0到最大值分为了五个档。

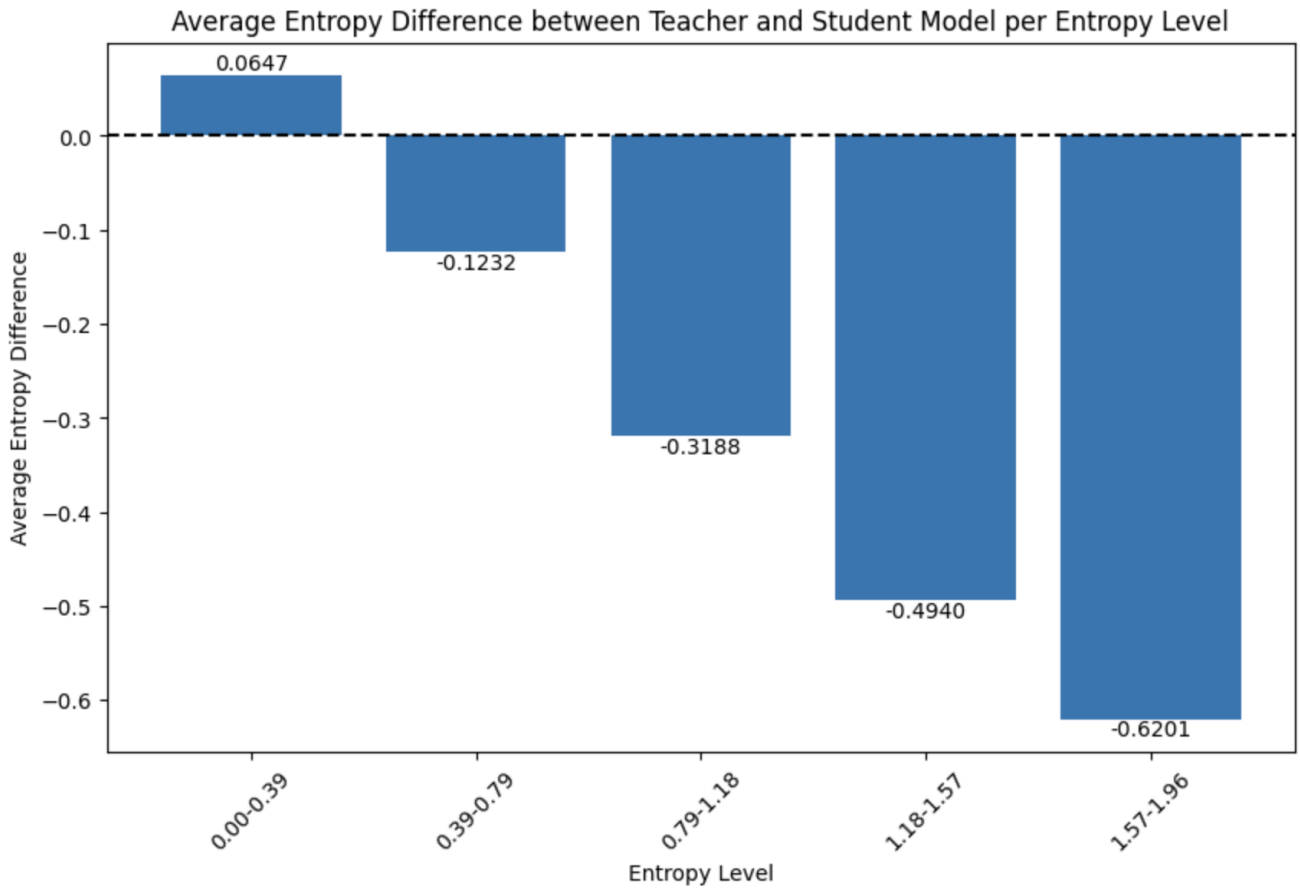
图一：教师模型与学生模型不同熵等级的占比分析

结论：教师模型之所以准确率高，与其低熵样本占比高有关系。



图二：在学生模型样本的不同熵等级下，执行教师模型后的熵平均下降值

结论：对于熵高的样本，理所当然地，执行教师模型的提升空间会更大，这也告诉我，单纯对于熵样本的分类的方法无法得出优秀的调度程序。



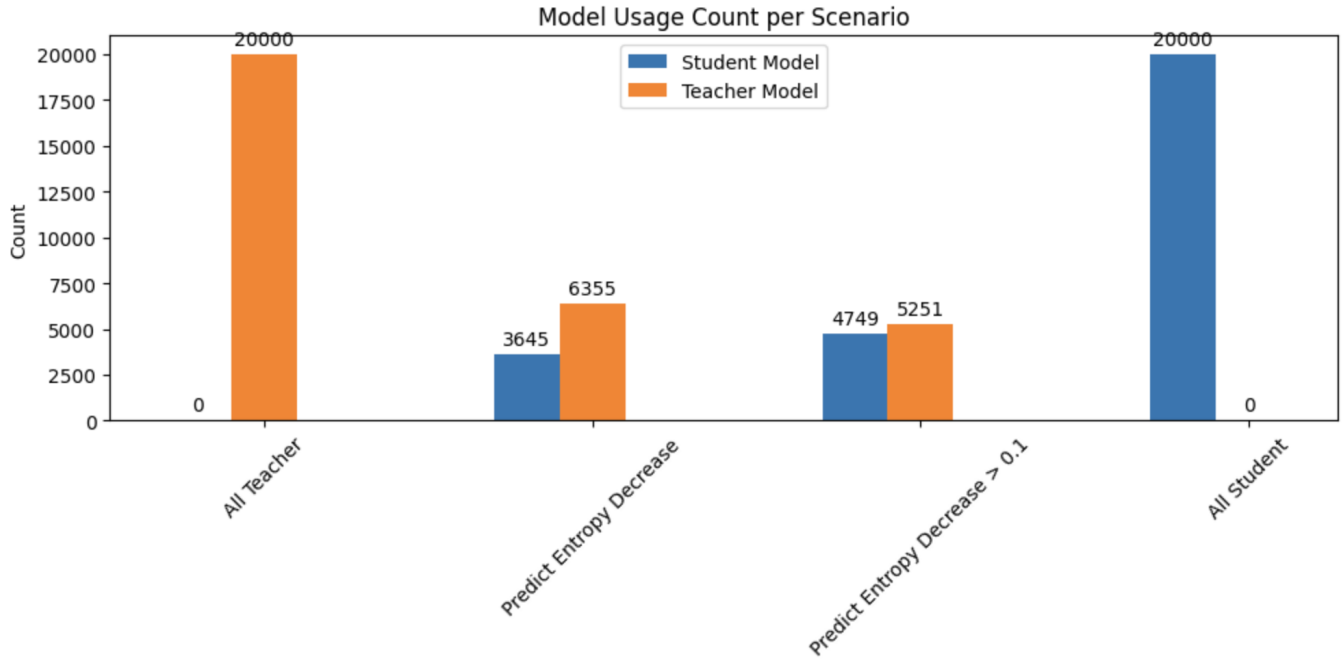
调度程序

由此启发，我训练了一个神经网络，input是过去学生模型的执行结果，output是预测教师模型执行后，熵会下降多少。此处仅采用了一个二层全连接网络，进行了初步的实验。

数据集搭建：采用CIFAR-10数据集中，train样本的学生模型分类分布作为输入，该样本执行教师模型后的熵下降量作为输出

评估

最后，我做了四组对照：



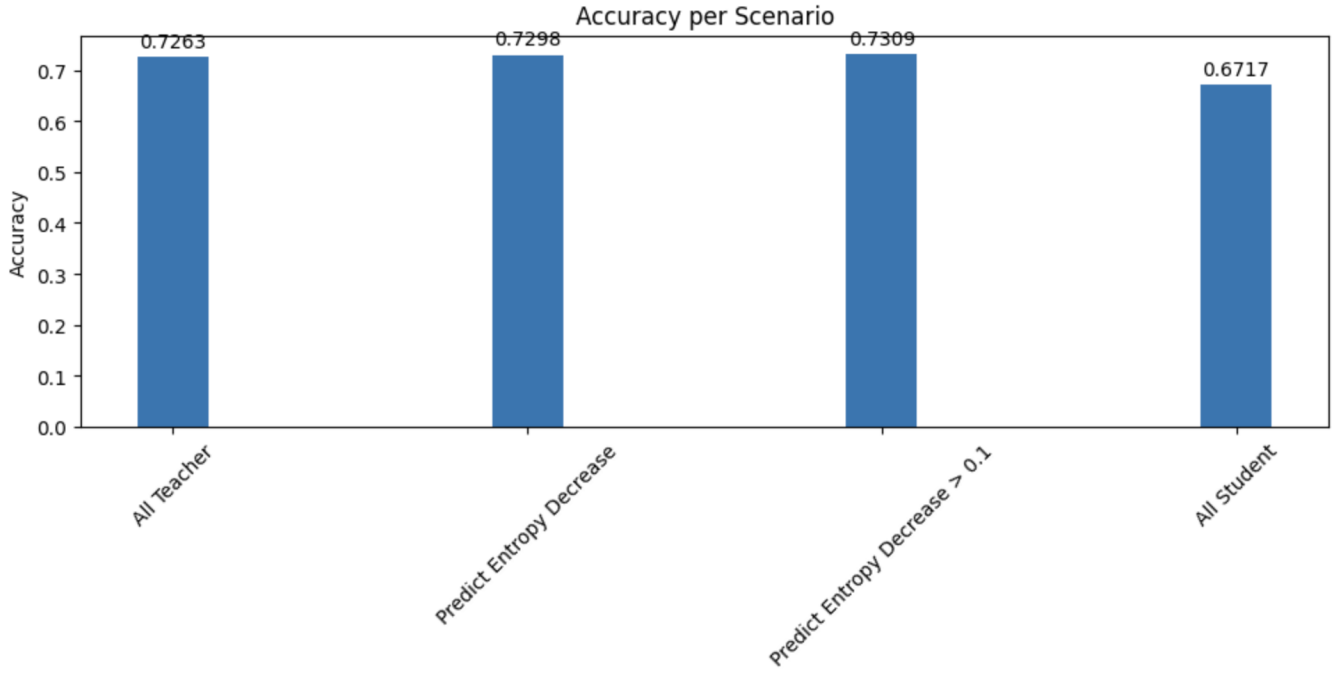
第一组是全使用teacher模型，

第二组是通过预测程序判断熵会下降的样本执行教师模型，其余执行学生模型

第三组是预测熵会下降至少0.1的执行教师模型，其余执行学生模型

第四组是全使用student模型

```
All Teacher:
  Student model output count: 0 (0.00%)
  Teacher model output count: 20000 (100.00%)
  Accuracy: 72.63%
Predict Entropy Decrease:
  Student model output count: 3645 (36.45%)
  Teacher model output count: 6355 (63.55%)
  Accuracy: 72.98%
Predict Entropy Decrease > 0.1:
  Student model output count: 4749 (47.49%)
  Teacher model output count: 5251 (52.51%)
  Accuracy: 73.09%
All Student:
  Student model output count: 20000 (100.00%)
  Teacher model output count: 0 (0.00%)
  Accuracy: 67.17%
```



results

第二组有64%的样本执行了教师模型，节省了接近36%的推理时间，准确率不减反增，而第三组，仅有52%的样本执行了教师模型，节省了接近一半的推理时间，准确率上升更多。