Deep Residual Learning for Image Recognition

深度神经网络面对的两个问题：

1、网络越深，随着深度的增加，会出现梯度消失的现象，举个简单例子，假如网络中的权重参数都小于1，在多层之后，通过矩阵乘法这样的操作，会使梯度越来越小，难以训练

2、随着神经网络深度的增加，loss会饱和，甚至出现退化的现象，我个人将其理解为信息的丢失，网络深度太深，失去了原始图像的重要特征，所以导致了模型的效果变差

为什么要解决这个问题呢，直接找到最合适的层数，不再继续往后加难道不就可以了吗，但是大量实验表明，前期网络的深度对于模型表现效果的提升是显著的，作者希望通过小小的修正，让这种提升效果继续发挥它的作用。

对于第一个问题，也就是梯度消失的问题，前人已经给出了很好地解决方法，分别是初始归一化（网络权重的初始化）以及中间归一化层（BN层），初始归一化就是让权重在服从均匀或正态分布（与所在层数n有关的）中随机采样，初始化网络权重；BN一般插在全连接网络或卷积网络之后，激活函数之前，针对某一个小的batch，首先进行均值方差的归一化，然后乘上缩放系数、加上平移系数（这两个系数是可学习参数）。这两个操作保证了数值的稳定性，可以很好的解决梯度消失与爆炸问题。

那第二个问题该怎么解决呢

解决这个问题可以从两个方面入手，从哲学上看就是改革派与修正派。第一，彻底改变我的网络架构，设计一个不会饱和或退化的完美的神经网络，这样的工作需要很大的工作，如果实现，深度学习领域会迎来很大的变革，但就目前深度学习的发展状况来看，刚刚起步有起色，他还没有发展到该大修大改、变革的时候；第二，通过小小的对网络的修改，来修正这个问题，这看起来是一个可行的方法；

所以论文作者就采用修正的方法，来解决此问题。对于修正方法，需要考虑四方面的问题：

1、该方法能够很好地解决网络深度增加引起的准确率退化问题

2、该方法的普适性要很强，在各种网络上都很有帮助，而不是针对单一网络类型、单一任务的修改

3、该方法不会产生或很小增加模型复杂度、不会带来负面的效果

4、该模型要有一定的可解释性，而不是黑箱式的魔改

这篇文章对于前三个问题解决的都很好，第四个问题文章里没有很好的体现，但好在该模型改进比较简单，理解起来也感觉有一定的道理

首先介绍以下论文的主要思想，然后介绍论文是怎么验证他解决了如上四方面问题的：

模型其实很简单，就是在不同的层之间架上一个桥梁，也就是恒等映射，这样进入下一层的信息就分为两部分：从隐藏层里出来的潜在特征、之前缓存过的前面某一层的恒等特征，如果是相同的层恒等映射是好实现的，如果是规模不同的层，通过线性的伸缩后再加上去即可。

不进行实验就可以分析这个方法对于模型复杂度的影响：

可以从参数量以及计算量两个方面考虑：resnet这种方法没有引进新的需要学习的参数，其参数量与普通的网络是相同的；针对计算量：残差网络主要采用加法计算，其开销是非常小的。所以如果这个方法能够解决问题1、2的话，他就是一个很好地改进了，作者通过了大量对比实验来验证此方法的有效性以及普适性

主要是两个大的对比实验，一个是在ImageNet数据集上的分类实验、一个是在CIFAR-10数据集上的分类实验

ImageNet上的实验

作者首先进行了这个实验，验证了Resnet在ImageNet上很好地解决了网络深度引起的准确率饱和或退化问题，聚体步骤如下：

首先设计一个baseline：构建一个普通的网络，论文里使用的是VGG，将其称为plain network

然后在基线网络的基础上，加入残差连接，得到Resnet

根据层数的不同，plain-18、Resnet-18；plain-34、Resnet-34（根据残差添加方式的不同，分为ABC三类）；Resnet-50、Resnet-101、Resnet-152

通过对VGG-18、Resnet-18的实验效果，发现二者准确率吧差不多，但Resnet的收敛速度更快，所以，验证了Resnet可以提升优化速度、其也能对我们最开始提出的两个问题中的第一个有帮助；

通过几个34层网络结果的对比，plain-18/34训练误差和测试误差对比发现，深度增加确实会导致退化现象，Resnet的误差显著优于二者，且优于Resnet-18，说明其很好的解决了模型退化的问题

对于50、101、152，模型准确率都有提升，性能进一步提高，且表现效果均比当时的最佳模型效果好，进一步验证了他对深度网络性能的提高效果。

在CIFAR-10上的实验，

设计了如上的对比实验，具体过程不再叙述，与上述实验类似，结果也类似；

这个实验证明了残差网络的泛化能力，其不止在ImageNet上是有效的。

目标检测实验

为了进一步验证其普适性，论文还进行了在PASCAL和MS COCO上的目标检测实验，都取得了领先当时最好模型的效果，展现了该修正方法极好的普适性与泛化能力

点评

该文章通过对神经网络上一个小小的修改，对模型效果实现了极大的提升，并且没有增加模型复杂度，没有使用让人很难理解的方法，而且论文写的很有条理，很清晰的脉络：提出问题、前人工作、提出方法、解释方法、设计实验、验证有效性与泛化能力。让人读着心情舒畅。

但论文对此方法的可解释性所说甚少，仅通过实验验证其有效性，至于其为什么有效，着墨甚少。

Learning Transferable Visual Models From Natural Language Supervision

这篇论文的abstract、introduction、approach是看的论文，后面实验部分太丰富了，找了知乎上的论文解读看的。论文理解如下：

当时CV领域面对的一些问题

1、对于CV领域当时广泛应用的是通过大量精确的带有标签的数据数据集对模型进行训练，这需要耗费大量的人力以及时间，而且这样得到的模型迁移能力并不是很难好

2、NLP领域在语言表征等方面表现得很出色，但将图片与自然语言结合起来训练的模型在当时表现并不是很好。

作者希望通过NL supervison 将图片以及其自然带有的caption一起训练一个模型

这样做有什么好处呢：第一，数据量多，可以直接从网上整理这样的数据集。第二、迁移性好，数据丰富，学习到的视觉概念就越多，就像我们人一样，看到的东西越丰富、阅历越深，越能做出精确的判断，因为我们了解的概念充足。

那实现这一工作，现在作者就需要解决如下几个问题：

数据来源问题：之前也有类似图片+文本的数据集如YFCC100M,但是这些数据集质量不好，有的caption甚至只有一些摄像机、日期这些信息

模型结构问题：之前也有人做过类似的工作，但表现很差，作者需要发现问题出现在哪里，并作出合适的修改

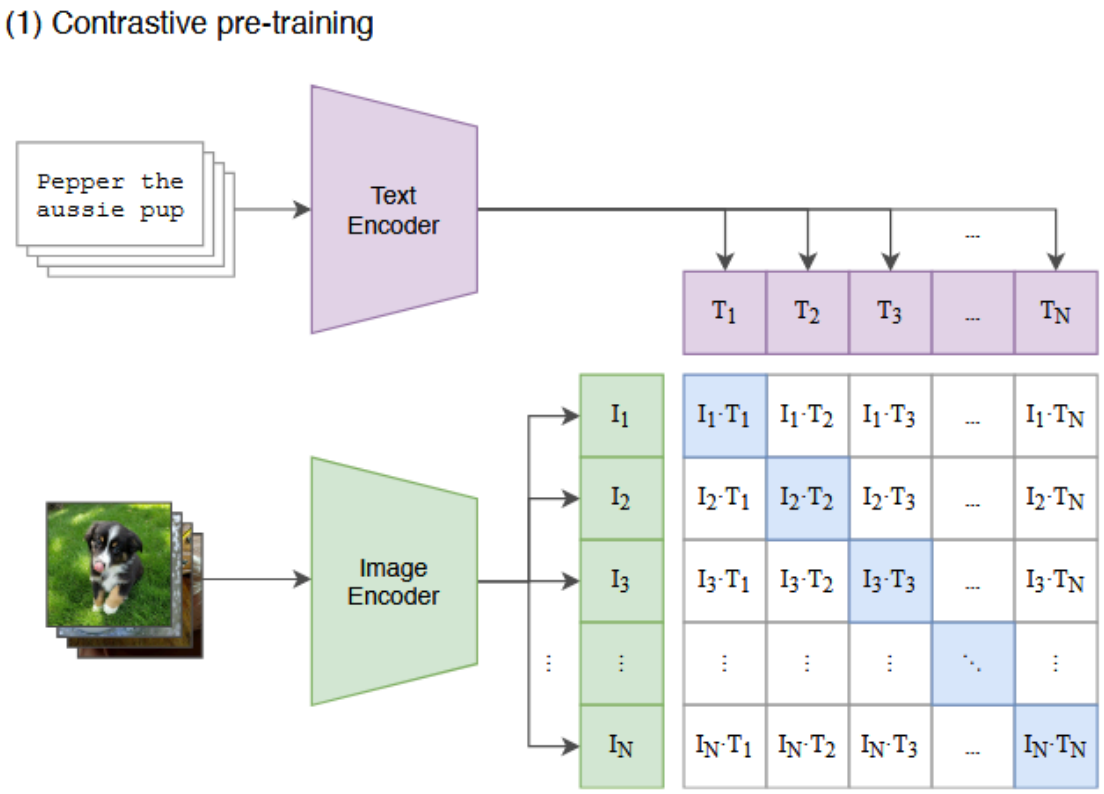
针对这两个问题，作者做了以下的工作：

1、we constructed a new dataset of 400 million (image, text) pairs collected form a variety of publicly available sources on the Internet.他们新搞了一组数据集

2、作者发现了之前工作表现差的一个关键点：Both approaches also use static softmax classifiers to perform prediction and lack a mechanism for dynamic outputs. This severely curtails their flexibility and limits their "zero-shot" capabilities，之前的人用的都是静态的softmax输出，并没有动态的调整，这样严格的限制了模型的灵活性

3、针对2发现的问题设计了下面的预训练　对比学习模型结构

Given a batch of N (image, text) pairs, CLIP is trained to predict which of the N × N possible (image, text) pairings across a batch actually occurred. To do this, CLIP learns a multi-modal embedding space by jointly training an image encoder and text encoder to maximize the cosine similarity of the image and text embeddings of the N real pairs in the batch while minimizing the cosine similarity of the embeddings of the N 2 − N incorrect pairings



除了这些，作者还对模型进行了一些小修小改，这些小修小改真的是震撼到我了，我感觉作者的工程技巧应该很强，肯定进行了大量的尝试以及论文的阅读，这是我最喜欢的一点，我把他们修改的点罗列在了下面：

论文主要设计了两个架构：Resnet+transformer的图像-文本编码、Vit+transformer的图像文本编码。作者既对编码器进行了修改，又对编码器外其他细节进行了修改，我就分这两个大点来说吧

一、编码器的修改

作者并没有直接套用原始的Resnet、Vit，而是对二者进行了修改

- Resnet修改

采用了 **ResNet-D** 的改进（来自论文《Bag of Tricks for Image Classification with Convolutional Neural Networks》）

作者没有像传统做法那样只增加深度或宽度，而是采用了类似EfficientNet的复合缩放策略。具体来说，当增加计算预算时，会均等地增加网络的深度（层数）、宽度（通道数）和输入图像的分辨率。例如，从ResNet-50缩放到ResNet-101时，不仅深度增加，宽度和分辨率也会按比例增加。

注意力池化：在卷积骨干网络输出的特征图上，模型不再简单地求所有位置的平均值。而是将特征图视为一个序列的空间特征向量。然后引入一个Q，与这些空间特征向量进行注意力计算。这个注意力操作会为每个空间位置生成一个权重，最终的特征表示是所有这些空间特征的加权和。也就是把平均池化改成了注意力池化

- VIT修改：

在将图像块嵌入和位置嵌入相加之后、送入Transformer编码器之前，额外添加了一个LayerNorm层。这个修改好像在后来的应用中大家都采用了

二、其他的小修小改

1. 训练与初始化

We train CLIP from scratch without initializing the image encoder with ImageNet weights or the text encoder with pre-trained weights.

无论是ResNet还是ViT图像编码器，权重都是随机初始化的，而不是用预训练在ImageNet上的权重来初始化。文本编码器也是如此。这证明了CLIP方法从零开始、仅通过大规模噪声对比学习就能学习到强大表征的能力。

2. 投影头

We do not use the non-linear projection… We instead use only a linear projection…

在之前的对比学习模型中（如SimCLR），通常会在编码器之后使用一个非线性的投影头（例如，一个MLP，包含线性层+ReLU+线性层）将表示映射到对比学习空间，最后再计算损失。CLIP发现，对于图像-文本对比任务，一个简单的线性投影层就足够了。因此，在两个架构中，图像编码器和文本编码器的输出后，都只有一个线性层来映射到最终的共享嵌入空间。

3. 文本转换函数

“We also remove the text transformation function tu…”

之前的某些工作会从一个长文本中随机采样多个句子来构建正样本对。但CLIP的数据集（如WebImageText）中很多文本本身就是单个句子。因此，CLIP简化了流程，直接使用给定的（图像，文本）对，不再进行复杂的文本采样。这属于数据预处理层面的修改，不直接影响模型架构。

4. 图像数据增强

“A random square crop from resized images is the only data augmentation used during training.”

数据增强策略非常简洁。在训练时，对图像只使用了随机方形裁剪这一种增强方式。这比当时许多CV模型使用的复杂增强组合（颜色抖动、旋转、剪切等）要简单得多，降低了工程复杂性。

5. 温度参数τ

“the temperature parameter... is directly optimized during training as a log-parameterized multiplicative scalar...”

温度参数τ用于缩放对比损失函数中的logits。CLIP没有将其设为需要精心调整的超参数，而是将其视为一个可学习的模型参数。为了防止其梯度更新过大，对其取对数（`log(τ)`）并进行优化。

6. 分布式计算优化

“The calculation of embedding similarities was also sharded with individual GPUs computing only the subset of the pairwise similarities necessary for their local batch of embeddings.”

这应该是计算上的优化，我不太了解哈哈。AI说：在大规模分布式训练中，计算所有图像和文本嵌入之间的巨大相似度矩阵非常耗时。通过“分片”计算，每个GPU只计算本地批次所需的那部分相似度，然后再通过集合通信进行汇总，这极大地提高了训练效率。这项优化对ResNet和ViT两种架构的训练都至关重要。

基于以上修改，训练了他们想要的模型

以上问题都解决了，作者就开始通过实验来展现模型的能力

这篇论文的实验是非常丰富的，让我意识到，有时做深度学习，并不一定需要天马星空的想象力，一步一步的设计完整的、完美的实验，对架构进行适当的修改，这些才是一个深度学习的工程师最应该具备的能力。

主要介绍论文的几个实验，不全部介绍了，主要参考知乎 ：鱼子酱【CLIP系列Paper解读】CLIP: Learning Transferable Visual Models From Natural Language Supervision

1 Zero-shot CLIP v.s. Linear Probe on ResNet50

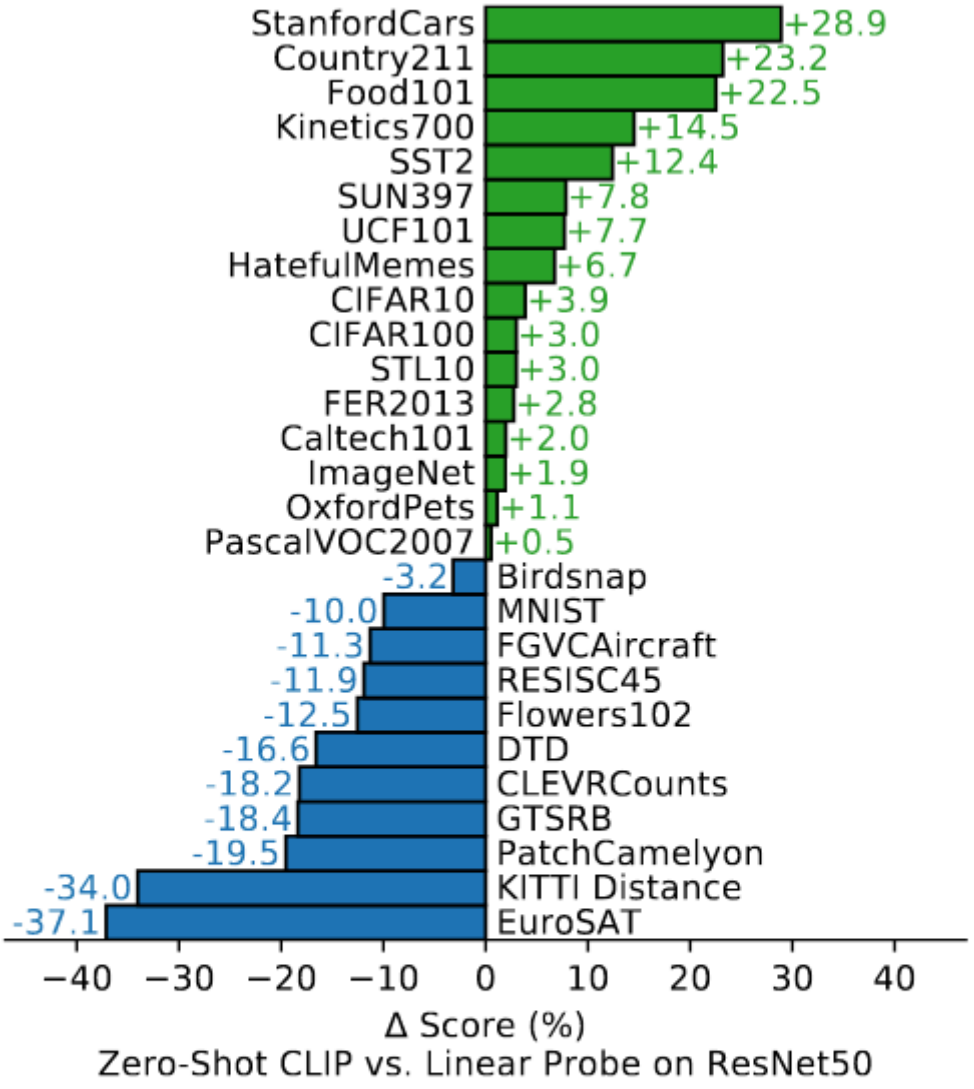


***Figure 5.* Zero-shot CLIP is competitive with a fully supervised baseline.** Across a 27 dataset eval suite, a zero-shot CLIP classifier outperforms a fully supervised linear classifier fitted on ResNet-50 features on 16 datasets, including ImageNet.

论文在27个数据集上做zero shot 分类，并与传统的SOTA linear probe on Resnet50进行结果比较，模型在其中的16个数据集上战胜了Linear probe on Resnet50,能力还是很不错的，毕竟是零样本预测

2 Prompt engineering and ensembling

作者做了一些prompt修改，作者默认prompt模板是："A photo of a {label}."，但作者发现这样的模板还是有点粗糙，可以考虑加一些context比如"A photo of a {label}, a type of pet."。对于不同类型任务，作者做了一些手动的、特定的prompt工程。

从另一个角度，一张图的text描述其实有很多种的，只要text的核心语义和image相同就行，那么我们还可以做一些ensemble，比如ensemble一下"A photo of a big {label}."和"A photo of a small {label}."。

3 Few-shot CLIP v.s. SOTA (ImageNet) SSL methods

看图说话

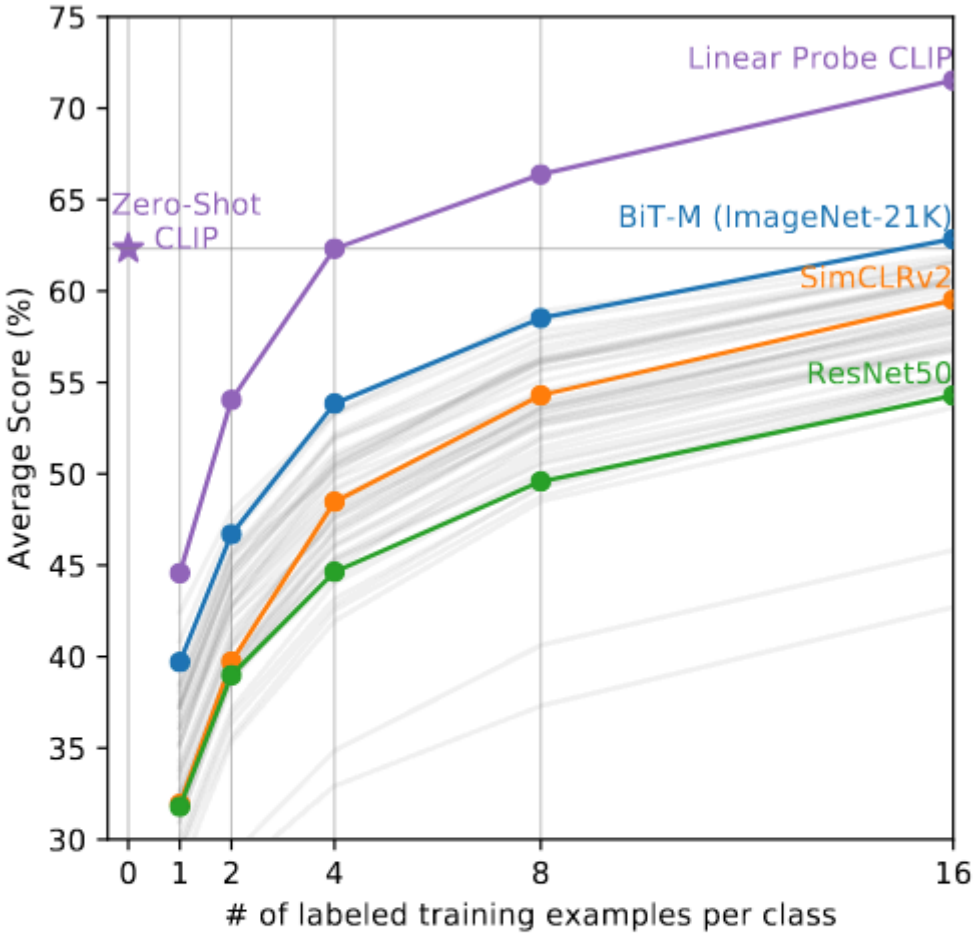


Figure 6. **Zero-shot CLIP outperforms few-shot linear probes.** Zero-shot CLIP matches the average performance of a 4-shot linear classifier trained on the same feature space and nearly matches the best results of a 16-shot linear classifier across publicly available models. For both BiT-M and SimCLRv2, the best performing model is highlighted. Light gray lines are other models in the eval suite. The 20 datasets with at least 16 examples per class were used in this analysis.

4 Linear probe CLIP 的表现

看图

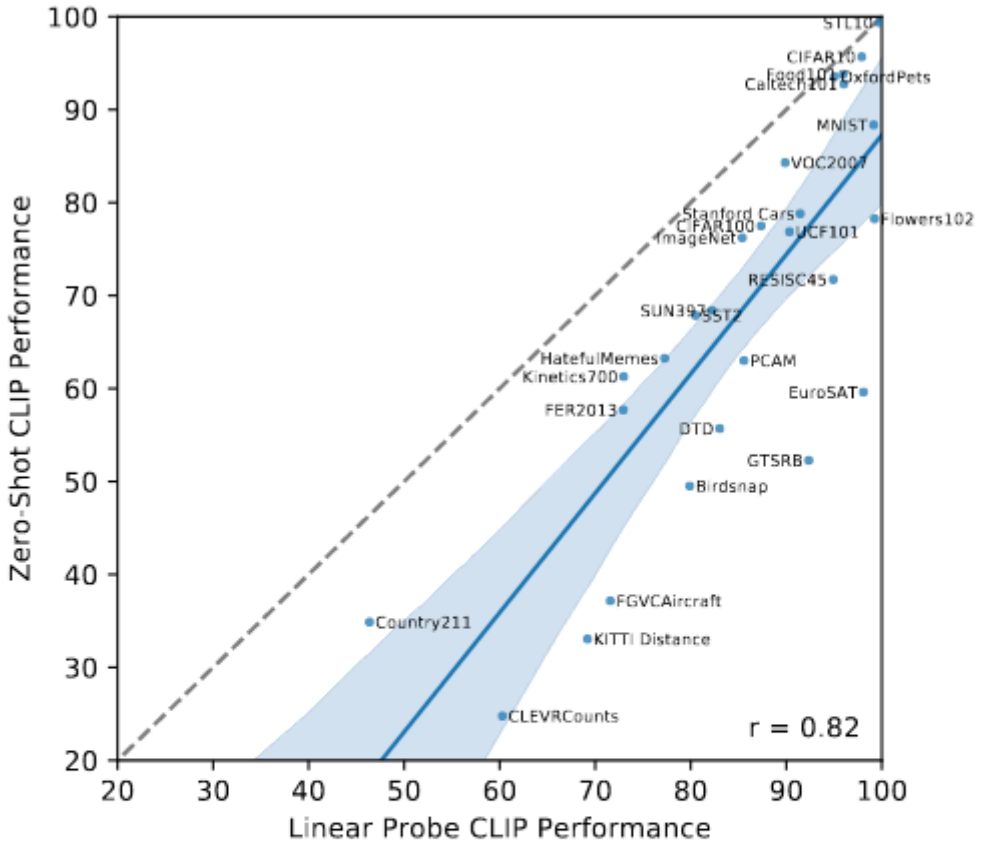


Figure 8. **Zero-shot performance is correlated with linear probe performance but still mostly sub-optimal.** Comparing zero-shot and linear probe performance across datasets shows a strong correlation with zero-shot performance mostly shifted 10 to 25 points lower. On only 5 datasets does zero-shot performance approach linear probe performance ($\leq$3 point difference).

5 Robustness to Natural Distribution Shift

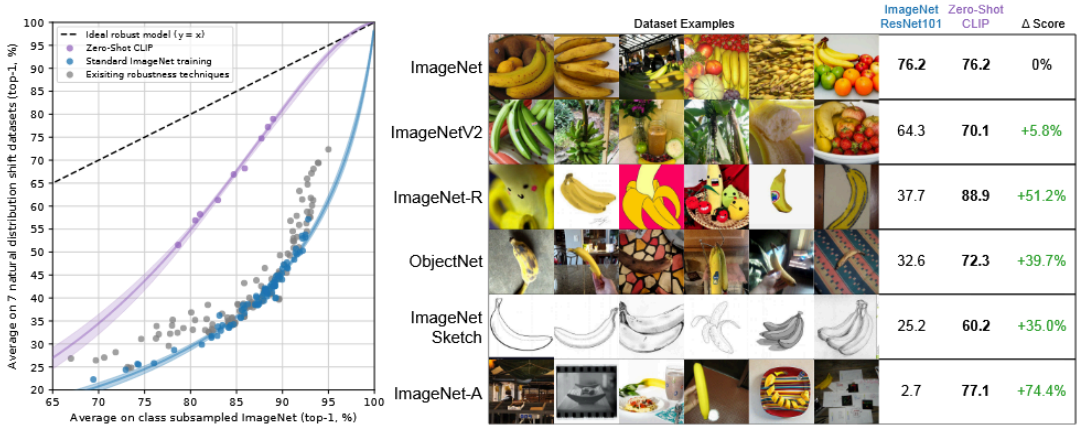


Figure 13. **Zero-shot CLIP is much more robust to distribution shift than standard ImageNet models.** (Left) An ideal robust model (dashed line) performs equally well on the ImageNet distribution and on other natural image distributions. Zero-shot CLIP models shrink this "robustness gap" by up to 75%. Linear fits on logit transformed values are shown with bootstrap estimated 95% confidence intervals. (Right) Visualizing distribution shift for bananas, a class shared across 5 of the 7 natural distribution shift datasets. The performance of the best zero-shot CLIP model, ViT-L/14@336px, is compared with a model that has the same performance on the ImageNet validation set, ResNet-101.

针对不同风格的图片，zero-shot CLIP 体现出了和好的鲁棒性

点评：CLIP的出现是有划时代意义的，CLIP证明，当模型在海量、噪声较大的互联网数据（图像-文本对）上进行训练时，能够自发地学习到强大的视觉概念。没有使用任何人工标注的标签，而是将自然语言作为监督信号的丰富来源。这种方法，极大地释放了可利用数据的规模。

但CLIP是还只是基于海量数据训练出来的统计学模型，并没有真正的生成能力，其不能生成新的东西

CLIP功能比较少，只有图文对比、图文匹配、类别判断这些功能，

BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation

这篇论文的结构非常的清晰，写的很清爽，主要结构就是先提出VLP领域的前景以及当前面临的问题，然后直截了当的提出他们的解决方法，随后在第三部分对方法进行了介绍，然后在4、5、6通过实验验证方法的合理性与模型的有效性，最后总结。

首先说一下当时VLP领域面对的问题，作者把他总结成了两个方面：

1、模型结构方面：过去的VLP模型功能太过单一，大多数方法仅采用 基于编码器的模型或者基于解码器的模型，二者功能各有优缺点，我的理解是基于编码器的模型，比如上一篇CLIP模型，在图文匹配、图文检索、图像分类等标签式的任务比较擅长，但对于生成式的任务比如对图像生成标题、图像描述等表现不好，基于解码器的模型对于生成任务表现较好但是对于编码器擅长的那些任务表现得还是很差

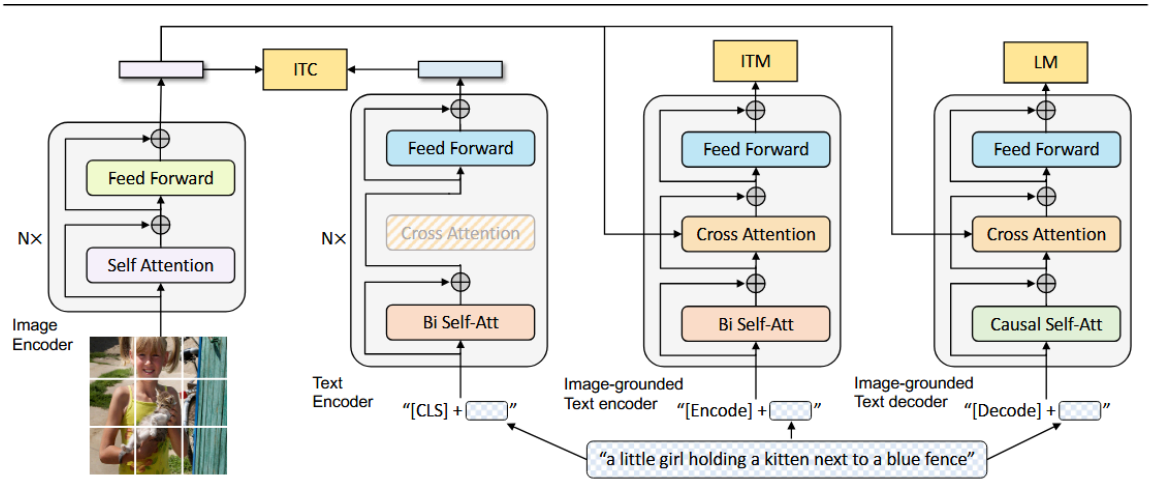
2、数据方面：现有的基于 图像-文本对的模型比如CLIP的数据是通过网上收集起来的，其中含有大量的噪声、偏执的信息：比如男女对立、种族歧视等信息（这是我自己的想法），用这样的信息去训练模型虽然能取得还不错的效果，但作者在文章中说明用这样的数据训练模型不是最好的选择。

基于以上两个问题，论文分别给出了解决方案：

1、对于模型结构方面的问题，作者建立了MED的模型架构：Multimodal mixture of Encoder-Decoder (MED)

2、对于数据方面的问题：作者提出了标注与过滤图像的修正策略，他们基于1提出的模型，设计了一个修正图像的预训练模型。

下面主要通过论文里的两个图来具体的介绍这两部分



第一个图主要是模型MED的具体架构，第二个图是他们CapFit的具体结构

MED：

主要分为三个模块：ITC（图文对比）、ITM（图文匹配）、LM（语言生成），我们来分别介绍

1、ITC，这个模块其实就是类似于CLIP论文里的图文对比学习模型：首先图像通过Vit编码生成对应的潜在特征，文本依次通过 Bi-SA、FFN得到对应的文本潜在特征，训练模型让是一对的图像文本有较高的相似度、非一对的图像文本有较低的相似度。这个模型对图文检索等任务有较大的帮助，他是在一个batch上最大化匹配的图文相似度、最小化不匹配的图文相似度，这正是检索所作的事情

2、ITM，这个模块的主要任务是判断输入的文本与图像是否是匹配的，与ITC不同，ITC是直接对图像和文本的潜在特征进行匹配，计算相似度，而ITM是将图像和文本的潜在特征再混合同时输入到Cross-attention、FFN，训练一个二分类任务，这样的任务更加精细。为后面的过滤器提供支持

3、LM：LM是解码模块，主要是通过transformer生成对图像的解释

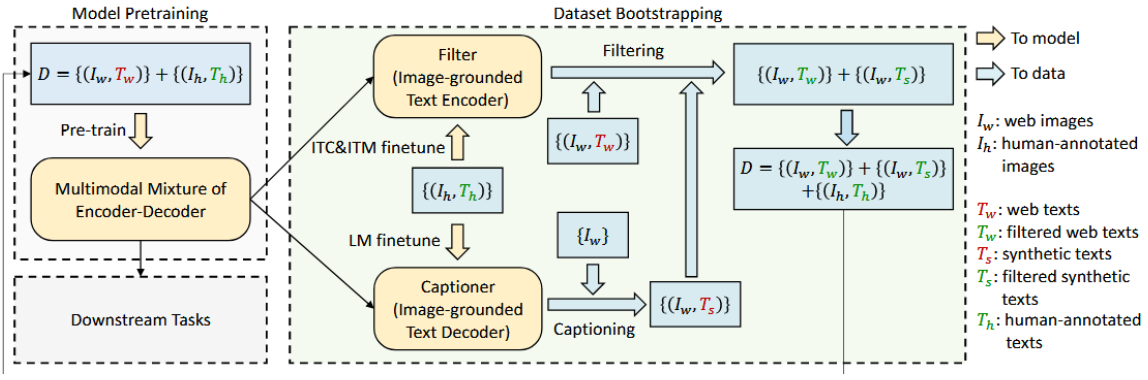
MED中的一些小巧思：

1、In order to find more informative negatives, we adopt the hard negative mining strategy by Li et al. (2021a), where negatives pairs with higher contrastive similarity in a batch are more likely to be selected to compute the loss.

这是ITM中的一个小的技巧，ITM是判断文本与图像是否是相匹配的，在训练的时候，如果文本与图像差距过大，模型并不容易学到有用的loss，所以他们使用了hard mining strategy,让相似度高但不匹配的图像与文本更有可能的被选作计算loss的对象，比如我们有一个小狗的图片，有两个文本：“这是一只狼“，”这是一座房子“，hard mining strategy就更有可能选择图片与第一个文本计算loss

2、In order to perform efficient pre-training while leveraging multi-task learning, the text encoder and text decoder share all parameters except for the SA layers.

这是训练时的一个设计，主要用来提高训练效率，在模型训练过程中，除了SA层的参数是单独的，其余层的参数都是共享的，作者也在后续实验中验证了这个方法的有效性，但是并没有很explainable的解释，文中给的解释是the differences between the encoding and decoding tasks are best captured by the SA layers.但具体为什么并没有涉及。



CapFit：

这个结构主要是用来优化训练集数据的

主要分为两个块

1、Captioner：它主要是fintuned上文的LM模块，利用它给搜集到的图像加标题，也就是输入I_w,输出T_s,最后我们得到了这样的数据$(I_w, T_w), (I_w, T_s), (I_h, T_h)$

这三个T分别是：网络文本、生成文本、人工标注文本

2、Filter：然后将上述前两个数据丢进Filter，Filter主要是fintuned前面MED的ITM模块，他会判断(I_w, T_w)以及(I_w, T_s)是否是匹配的，如果匹配就可以用于后续训练，如果不匹配，丢掉

下面就是BLIP里设计的实验：

我觉得他主要有两方面的实验：第一是验证他们小巧思的有效性、第二是通过在多个任务上与SOTA对比，验证模型的能力

首先说小巧思的验证：

Effect of CapFilt

Parameter Sharing and Decoupling

上面两个实验就是简单的控制变量法的对比试验，不具体说了，通过对比得到了这些方法对模型的表现确实有提升的结论

Additional Ablation Study:这是我在读这篇论文中新学到的一个概念，机器翻译成消融实验，我的理解是当我们修改模型、引入一个模块时，会带来一些附加的影响，我们需要判断到底是这个附加影响对模型有了提升，还是我们想引入的影响对模型有了提升，以这个模型为例，Capfit模块对图像生成了一些标题，这在一定程度上相当于data augmentation，丰富了训练的数据，所以到底是训练数据增多对模型的表现有了提高，还是Capfit对不好的数据进行了过滤、并且生成了好的数据对模型有了提升呢，作者于是设计了消融实验来验证：

CapFilt	#Texts	Retrieval-FT (COCO)		Retrieval-ZS (Flickr)		Caption-FT (COCO)		Caption-ZS (NoCaps)	
		TR@1	IR@1	TR@1	IR@1	B@4	CIDEr	CIDEr	SPICE
No	15.3M	78.4	60.7	93.9	82.1	38.0	127.8	102.2	13.9
No	24.7M	78.3	60.5	93.7	82.2	37.9	127.7	102.1	14.0
Yes	24.7M	80.6	63.1	94.8	84.9	38.6	129.7	105.1	14.4

设计了三个对象：一个是text含量少，没有进行CapFit的对照组，一个是通过复制数据，得到的text含量多，没有CapFit的实验组，一个是通过Capfit得到的texts含量多的实验组；然后在多个任务上测试性能

通过1、2对比发现，单一的增加数据，不注重质量并不会提升模型的训练效果，有些表现又很小的提升，有些甚至下降了，产生了过拟合的现象

通过2、3对比发现，CapFit确实发挥了除增加数据量之外的作用

然后是在多种任务上，体现MED的多任务能力

Image-Text Retrieval

Image Captioning

Visual Question Answering (VQA)

Visual Dialog (VisDial)

Zero-shot Transfer to Video-Language Tasks

这些任务还是比较丰富的，实验主要是对预训练的模型进行微调让其适应特定任务，然后与其他方法进行对比，来体现模型的能力

点评：这篇论文提出的MED模型让单一模型具有了更广泛的能力，而不只是一个单一的针对特定任务的模型，这其实也是大势所趋，但是这篇论文并没有特别创新的点，就像是对之前人们所做任务的一个拼接。

Denoising Diffusion Probabilistic Models

由于我对图像生成比较感兴趣，所以我选了一篇图像生成领域的论文，一片很短小的论文但是是一篇对图像生成领域有划时代意义的论文。

生成模型主要是解决一个mapping problem：

> **The Transport Mapping Problem** *Given empirical observations of two distributions* $X_0 \sim \pi_0, X_1 \sim \pi_1$ *on* $\mathbb{R}^d$*, find a transport map* $T\colon \mathbb{R}^d \to \mathbb{R}^d$ *(hopefully nice or optimal in certain sense), such that* $Z_1 \coloneqq T(Z_0) \sim \pi_1$ *when* $Z_0 \sim \pi_0$*, that is,* (Z_0, Z_1) *is a coupling (a.k.a transport plan) of* π_0 *and* π_1*.*

给一个原始空间和一个目标空间，我们通过学习一个传输方程吗，来实现从简单分布到目标分布的任务

在这之前，图像生成领域主要的方法都依赖于VAE以及GAN等编码解码模型，这样的模型面临这一些问题

这些模型都是一次性生成图像，然后对比生成图像以及原始图像，训练模型，让模型生成的图像与原始想要的图像尽可能的像，但是这会有一些问题，直接生成会导致模型生成的图像具有模糊以及单一性：

模糊性：VAE是训练一个变分下界函数来实现图像的生成，这样会导致模型的输出是一个平均化的、最保险的结果，因为越清晰、越尖锐的图像，越容易导致重建图与原始图之间的ELOB（描述相似性的一个函数，主要是基于信息论的知识设计出来的，好像是越大越不相似）更大

单一性：GAN训练出的模型会面临单一性的问题，这个模型主要通过生成器与判别器的对抗，在对抗中提升双方的能力（其实就是一个minmax问题），这会导致什么结果呢，生成器可能发现只要完美生成某几张图就能骗过判别器，于是它就会“偷懒”，只反复生成这几张图，导致多样性不足。

于是在2020年，DDPM横空出世

先说其解决的问题：

首先是模糊性的问题：这个模型的学习过程是一个马尔可夫过程，不像VAE和GAN（一步生成），DDPM是一步一步的重建图像，一步一步的学习这个传输方程T，这就类似于把方程拆解成多个简单函数的复合，学习过程更加简单，不会出现不稳定的问题，学习到的细节也更多。

然后是单一性的问题：这个模型的训练对象是噪声，噪声具有很大的随机性，通过这样的方法训练出的生成模型具有极大的多样性。

另外DDPM的可解释性更强，其涉及随机微分方程、信息论、概率论等既前沿又经典的数学知识，让后续的优化有了更准确地方向，而不是不可解释的魔改。

下面我就主要介绍其中的数学原理以及实验：

打Latex太麻烦了，所以我直接手写了：

我们要得到目标分布 $P_\theta(x_0)$，

目标 $P_\theta(x_0) := \int P_\theta(x_{0:T})\, dx_{1:T}$ （以后 x_0 就代表目标，x_T 就代表简单分布）

$P_\theta(x_{0:T})$ 是指从 $x_T \to x_0$ 的可能的正确路径

这个积分实际上就是由联合分布求边际分布

那联合分布怎么求呢（从 $x_T \to x_0$）

$P_\theta(x_{0:T}) = P(x_T) \prod_{t=1}^{T} P_\theta(x_{t-1}|x_t)$ 可以看作一个马尔可夫链

这里 $P_\theta(x_{t-1}|x_t) := \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t))$

往后好像难以进行下去了，这时候就到了论文最关键的部分了。（因为这个积分是难以求出的）

作者引入了前向过程（扩散过程）以及反向过程（去噪过程）

然后希望让反向过程学习前向过程的逆过程

上面说的实际上是去躁，即从简单分布一直到目标分布

前向过程是从目标分布一点一点加入噪声，这实际上也是一个马尔可夫过程：

$$q(x_{1:T}|x_0) := \prod_{t=1}^{T} q(x_t|x_{t-1}) \qquad q(x_t|x_{t-1}) := \mathcal{N}(x_t; \sqrt{1-\beta_t}\, x_{t-1}, \beta_t I)$$

这样之后，我们可以最大化 $P_\theta(x_0)$ 的对数似然，然后引入变分上界来得到要优化的 Loss

$$\mathbb{E}[-\log P_\theta(x_0)] \le \mathbb{E}_q\left[-\log \frac{P_\theta(x_{0:T})}{q(x_{1:T}|x_0)}\right] = \mathbb{E}_q\left[-\log(x_T) - \sum_{t\ge 1} \log \frac{P_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1})}\right] =: L$$

作者为了进一步得到类似标红那一部分的学习过程，进一步化简 L.

他们提出可以不一步一步的加噪，可以由 x_0 直接加到 x_t

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}\, x_0, (1-\bar{\alpha}_t) I)$$

这里 $\alpha_t = 1 - \beta_t$ $\bar{\alpha}_t := \prod_{s=1}^{t} \alpha_s$，$\beta_t$ 是上文一步步加噪的参数（论文中把它固定为超参数）

这样可以优化 L 为：

$$\mathbb{E}_q\left[D_{KL}(q(x_T|x_0) \| P(x_T)) + \sum_{t>1} D_{KL}(q(x_{t-1}|x_t, x_0) \| P_\theta(x_{t-1}|x_t)) - \log_{P_\theta}(x_0|x_1)\right]$$

L_T L_{t-1} L_0

L_T 与 L_0 都是已知项，现只剩 L_{t-1} 这一项还没研究清楚

L_{t-1} 看它的形式实际上就是衡量了从 $(t \to t-1)$ 的反向过程与从 $(t-1 \to t)$ 的正向过程的逆过程之间的相似度.

所以训练时只需优化 L_{t-1} 即可. 现在的目标是将 L_{t-1} 进行能化为一个神经网络可训练的目标.

我们先看 L_{t-1} 中的 $q(x_{t-1} | x_t, x_0)$ 项.

因 $q(x_{1:T} | x_0)$ $q(x_t | x_{t-1})$ 都是已知的，且该过程是马尔可夫的

由贝叶斯定理知

$$q(x_{t-1} | x_t, x_0) = \frac{q(x_t | x_{t-1}, x_0)\, q(x_{t-1} | x_0)}{q(x_t | x_0)}$$

$$q(x_t | x_{t-1}, x_0) = q(x_t | x_{t-1})$$

$$q(x_{t-1} | x_t, x_0) \propto q(x_t | x_{t-1})\, q(x_{t-1} | x_0)$$

反正通过化简、配方可以知

$$q(x_{t-1} | x_t, x_0) = \mathcal{N}(x_{t-1}; \tilde{\mu}_t(x_t, x_0), \tilde{\beta}_t I)$$

这里 $\tilde{\mu}_t(x_t, x_0) := \frac{\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_t} x_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t} x_t$

$$\tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t} \beta_t$$

现在 L_{t-1} 是这样的

$$D_{KL}\left(\mathcal{N}(x_{t-1};\tilde{\mu}_t(x_t,x_0),\tilde{\beta}_t I)\,\|\,\mathcal{N}(x_{t-1};\mu_\theta(x_t,t),\Sigma_\theta(x_t,t))\right)$$

作者发现将方差均固定就可以得到很好的效果

故他们设成了一样的 $\sigma_t^2 I$.

此时, $L_{t-1}=D_{KL}\left(\mathcal{N}(\tilde{\mu}_t,\sigma_t I)\|\mathcal{N}(\mu_\theta,\sigma_t^2 I)\right)=\mathbb{E}_{x\sim\mathcal{N}(\tilde{\mu}_t,\sigma_t^2 I)}\left[\frac{1}{2\sigma_t^2}\|\tilde{\mu}_t(x_t,x_0)-\mu_\theta(x_t,t)\|^2\right]+C$

故

$$\tilde{L}_{t-1}=L_{t-1}-C=\mathbb{E}_q\left[\frac{1}{2\sigma_t^2}\|\tilde{\mu}_t(x_t,x_0)-\mu_\theta(x_t,t)\|^2\right]$$

我们把 KL 散度化成了 MSE，已经很简了.

现在神经网络目标为预测 $\tilde{\mu}_t$

上面 $\tilde{\mu}_t(x_t,x_0)=\frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}x_0+\frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}x_t$

但实际训练时，我们输入的只有 x_t，并无 x_0。

这里就到了论文的一大亮点，噪声的引入.

x_t 由 x_0 与 $\varepsilon \sim \mathcal{N}(0, I)$ 生成:

$$x_t = \sqrt{\bar{\alpha}_t}\, x_0 + \sqrt{1-\bar{\alpha}_t}\, \varepsilon$$

反解出 $x_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}\left(x_t - \sqrt{1-\bar{\alpha}_t}\,\varepsilon\right)$

代入 $\tilde{\mu}_t$ 并化简得

$$\tilde{\mu}_t(x_t, \varepsilon) = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\varepsilon\right) \quad ①$$

上面是实际，
我们的预测值也应有这样的形式.

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\varepsilon_\theta(x_t, t)\right) ②$$

而 $L_{t-1} \propto \|\tilde{\mu}_t - \mu_\theta\|^2 = \|① - ②\|^2$

$$= \frac{(1-\alpha_t)^2}{\alpha_t(1-\bar{\alpha}_t)}\|\varepsilon - \varepsilon_\theta(x_t, t)\|^2 \quad !!!$$

终于我们得到了优化目标.

$$L_{simple} = \mathbb{E}_{t, x_0, \varepsilon}\left[\|\varepsilon - \varepsilon_\theta(x_t, t)\|^2\right]$$

$$= \mathbb{E}_{t, x_0, \varepsilon}\left[\|\varepsilon - \varepsilon_\theta(\sqrt{\bar{\alpha}_t}\, x_0 + \sqrt{1-\bar{\alpha}_t}\,\varepsilon, t)\|^2\right]$$

以上就是 DDPM 的 Loss 推导 ☒

有了训练目标之后我们就要看他怎么训练的，这里只给出前向传播即可

我把论文中写的算法截下来：

Algorithm 1 Training

1: **repeat**
2: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
3: $t \sim \text{Uniform}(\{1, \dots, T\})$
4: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
5: Take gradient descent step on
$$\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2$$
6: **until** converged

Algorithm 2 Sampling

1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \dots, 1$ **do**
3: $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
4: $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}$
5: **end for**
6: **return** $\mathbf{x}_0$

算法1是训练过程，算法2是训练后的采样过程，我们只说算法1；

准备数据：从数据集中随机抽取一批真实图像x_0

生成训练样本：

1、为每张图像随机采样一个时间步t

2、为每张图像随机生成一个与图像尺寸相同的噪声ϵ

3、通过公式计算$x_t = \sqrt{(\overline{\alpha}_t)}x_0 + \sqrt{(1-\overline{\alpha}_t)}\epsilon$，计算出对应的噪声图像$x_t$

网络预测：将噪声图像x_t和对应的时间输入到网络中（论文里用的U-net），但是有一个问题，我们不可能把一个整数t直接输入到网络，模型很难理解一个整数的时间，所以借鉴Transformer中的position embedding思想，对时间进行编码；这样之后模型就通过前向传播，生成一个与x_T相同大小的噪声$\epsilon_\theta(x_t, t)$

计算损失：然后按照我们推到的公式计算损失函数$L = ||\epsilon - \epsilon_\theta(x_t, t)||^2$

以上就是前向传播，优化过程我们就不具体的讲了，下面简单说一下该论文进行的实验

数据集：

CIFAR-10：10个类别、尺寸为32×32的彩色图像数据集，在生成模型中经常用来评估模型性能

CelebA-HQ：高清名人头像数据集，尺寸为256×256，用于测试模型在高分辨率图像上的生成能力

LSUN：大规模场景理解数据集，论文中使用了卧室、教堂等类别，尺寸为256×256

评估指标：

IS：生成图像的质量和多样性，越高越好

FID：衡量生成图像与原始图像之间的距离，越低越好

NLL：直接计算模型对数据概率分布的拟合程度，分数越低越好

实验结果

在此之前，对抗生成网络在图像生成质量上一直处于领先，但DDPM的表现对GAN实现了超越

Table 1: CIFAR10 results. NLL measured in bits/dim.

Model	IS	FID	NLL Test (Train)
Conditional			
EBM [11]	8.30	37.9	
JEM [17]	8.76	38.4	
BigGAN [3]	9.22	14.73	
StyleGAN2 + ADA (v1) [29]	**10.06**	**2.67**	
Unconditional			
Diffusion (original) [53]			≤ 5.40
Gated PixelCNN [59]	4.60	65.93	3.03 (2.90)
Sparse Transformer [7]			**2.80**
PixelIQN [43]	5.29	49.46	
EBM [11]	6.78	38.2	
NCSNv2 [56]		31.75	
NCSN [55]	8.87 ± 0.12	25.32	
SNGAN [39]	8.22 ± 0.05	21.7	
SNGAN-DDLS [4]	9.09 ± 0.10	15.42	
StyleGAN2 + ADA (v1) [29]	$\mathbf{9.74} \pm 0.05$	3.26	
Ours (L, fixed isotropic $\mathbf{\Sigma}$)	7.67 ± 0.13	13.51	≤ 3.70 (3.69)
Ours (L_{simple})	9.46 ± 0.11	**3.17**	≤ 3.75 (3.72)

Table 2: Unconditional CIFAR10 reverse process parameterization and training objective ablation. Blank entries were unstable to train and generated poor samples with out-of-range scores.

Objective	IS	FID
$\tilde{\boldsymbol{\mu}}$ prediction (baseline)		
L, learned diagonal $\mathbf{\Sigma}$	7.28 ± 0.10	23.69
L, fixed isotropic $\mathbf{\Sigma}$	8.06 ± 0.09	13.22
$\|\tilde{\boldsymbol{\mu}} - \tilde{\boldsymbol{\mu}}_\theta\|^2$	–	–
$\boldsymbol{\epsilon}$ prediction (ours)		
L, learned diagonal $\mathbf{\Sigma}$	–	–
L, fixed isotropic $\mathbf{\Sigma}$	7.67 ± 0.13	13.51
$\|\tilde{\boldsymbol{\epsilon}} - \boldsymbol{\epsilon}_\theta\|^2$ (L_{simple})	$\mathbf{9.46 \pm 0.11}$	**3.17**

点评：DDPM是一篇创新性十足的论文，为生成模型创造了新的范式，其在表现、可解释性等方面都超越了前面的模型，但是还是存在一些问题，比如采样速度较慢，如果时间步很密集，每一次前向传播每一张图都要经历一次U-net网络；DDPM是无条件的学习，无法控制模型具体生成什么，生成什么完全取决于训练的数据集，后续这个问题被解决了，引入了condition diffusion（如论文Stable Diffusion，我还没看），实现了模型的文生图能力。