Your Diffusion Model is Secretly a Zero-Shot Classifier

摘要

到目前为止，几乎所有扩散模型都都只关注采样，但扩散模型还可以提供条件概率估计，这对于图像生成以外的任务很有用。利用大规模文本到图像扩散模型的密度估计来执行零样本分类，而无需任何额外的训练。

我们的生成分类方法（我们称之为扩散分类器）在各种基准测试中取得了优异的结果，并且优于从扩散模型中提取知识的替代方法。

尽管在零样本识别任务上生成方法和判别方法之间仍然存在差距，但我们的基于扩散的方法比竞争的判别方法具有明显更强的多模态合成推理能力。

最后，我们使用扩散分类器从在 ImageNet 上训练的类条件扩散模型中提取标准分类器。我们的模型仅使用弱增强即可实现强大的分类性能，并对分布变化表现出更好的“有效鲁棒性”。总的来说，我们的结果是朝着在下游任务中使用生成模型而非判别模型迈出的一步。

介绍

要识别形状，首先要学会生成图像 [31]——在这篇开创性的论文中，Geoffrey Hinton 强调生成建模是训练人工神经网络执行图像识别等判别任务的关键策略。尽管生成模型解决了精确建模底层数据分布这一更具挑战性的任务，但它们可以创建更完整的世界表示，可用于各种下游通信任务。因此，在过去的十年中，人们提出了大量的隐式和显式生成建模方法 [26,42,46,21,77,70,79]。然而，这些作品的主要焦点是内容创作，而不是执行区分任务的能力。在本文中，我们在扩散模型（当前最先进的生成模型系列）的背景下重新审视这一经典的生成与判别式辩论。特别是，我们研究了扩散模型如何与图像分类任务中最先进的判别模型进行比较。

给定输入x和有限类c集，使用该模型来计算类条件似然$p_\theta(x|c)$。接着使用适当的先验分布p(c)，并应用贝叶斯定理，就可以获得预测的类别概率$p(c|x)$。对于使用辅助输入的条件扩散模型，例如类条件模型的类索引或文本到图像模型的提示，我们可以通过利用 ELBO 作为近似的类条件对数似然 log p(x | c).

实际上，通过贝叶斯定理获得扩散模型分类器包括重复添加噪声并计算每个类别的预期噪声重建损失（也称为 ε 预测损失）的蒙特卡罗估计。我们将这种方法称为扩散分类器。

通过与十一个不同基准上的多个基线进行比较，我们强调了我们提出的扩散分类器在零样本分类、组合推理和监督分类任务上的惊人有效性。通过利用稳定扩散[65]，扩散分类器实现了强大的零样本精度，并且优于从预训练扩散模型中提取知识的替代方法。我们的方法在具有挑战性的 Winoground 组合推理基准上也优于最强的对比方法 。最后，我们使用我们的方法通过 Diffusion Transformer (DiT)（一种 ImageNet 训练的类条件扩散模型）执行标准分类。我们的生成方法仅使用弱增强就在 ImageNet 上实现了 79.1% 的准确率，并且与在同一数据集上训练的竞争判别分类器相比，对分布变化表现出更好的鲁棒性。我们的结果表明，也许是时候重新审视生成分类方法了。

相关工作

生成模型用于判别任务

解决常见的分类或回归任务的机器学习算法通常在两种范式下进行：

（1）判别方法直接学习对基础任务的决策边界进行建模
（2）生成方法则学习对数据的分布进行建模，然后将基本任务理解为最大似然估计问题

常见的生成方法有 朴素贝叶斯、VAE、GAN、EBM（energy-baed method）

开创性的著作强调了对数据分布进行建模以更好地学习判别特征的想法。这些工作训练了深度信念网络将底层的图像数据建模为隐变量，随后用于图像识别任务。最近的生成建模任务还学习了全局和密集预测任务的有效表示，例如分类 分割 任务。此外，此类模型还证明了拥有更好的对抗鲁棒性和更好的校准能力。

上述大多数工作要么联合训练判别模型和生成模型，要么微调下游任务的生成表示。直接利用生成模型进行判别任务是一个相对较少研究的问题，在这项工作中，我们特别强调了直接使用最近的扩散模型作为图像分类器的功效。

扩散模型

扩散模型能够生成高保真和多样化的内容，例如图像 视频 语音通过各种输入方式（如文本）

扩散模型同时也和**EBMs**、**去噪分数匹配**以及**随机微分方程**密切相关

（Yang Song Score-based generative modeling through stochastic differential equations）

零样本图像分类器

迄今为止，分类器通常是在监督环境中进行训练的，其中训练集和测试集是固定且有限的。 CLIP 表明，利用大规模图像文本数据可以实现对各种新任务的零样本泛化

尽管它们被称为“零样本”，但仍不清楚评估样本是否位于其训练数据分布中。与上面的判别方法相反，我们建议从大规模生成模型中提取零样本分类器。

方法——基于扩散模型的分类

1.扩散模型预备知识

$$p_\theta(\mathbf{x}_0 \mid \mathbf{c}) = \int_{\mathbf{x}_{1:T}} p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{c})\, \mathrm{d}\mathbf{x}_{1:T} \quad (1)$$

这里x表示图像，c表示低维文本嵌入（用于文本到图像的合成）或类索引（用于类条件生成）

$$\log p_\theta(\mathbf{x}_0 \mid \mathbf{c}) \geq \mathbb{E}_q \left[\log \frac{p_\theta(\mathbf{x}_{0:T}, \mathbf{c})}{q(\mathbf{x}_{1:T} \mid \mathbf{x}_0)} \right] \tag{2}$$

直接最大化积分不好算，于是采用最小化对数似然ELBO

$$-\mathbb{E}_\epsilon \left[\sum_{t=2}^{T} w_t \|\epsilon - \epsilon_\theta(\mathbf{x}_t, \mathbf{c})\|^2 - \log p_\theta(\mathbf{x}_0 \mid \mathbf{x}_1, \mathbf{c}) \right] + C \tag{3}$$

C不是依赖于c的常数项，并且T=1000很大，$logp_\theta(x_0|x_1,c)$通常很小，因此删除。最后发现去掉w_t可以让效果更好，后面许多工作也这样做，并且设置$w_t = 1$

$$-\mathbb{E}_{t,\epsilon} \left[\|\epsilon - \epsilon_\theta(\mathbf{x}_t, \mathbf{c})\|^2 \right] + C \tag{4}$$

2.利用扩散模型进行分类

一般来说，使用条件生成模型的分类可以通过对模型预测 pθ(x | ci) 和标签 {ci} 上的先验 p(c) 使用贝叶斯定理来完成：

$$p_\theta(\mathbf{c}_i \mid \mathbf{x}) = \frac{p(\mathbf{c}_i)\, p_\theta(\mathbf{x} \mid \mathbf{c}_i)}{\sum_j p(\mathbf{c}_j)\, p_\theta(\mathbf{x} \mid \mathbf{c}_j)} \tag{5}$$

给定$p(c_i)$先验（1/N）会导致所有项相互抵消。直接计算很困难，因此使用ELBO代替：

$$p_\theta(\mathbf{c}_i \mid \mathbf{x}) = \frac{\exp\{-\mathbb{E}_{t,\epsilon}[\|\epsilon - \epsilon_\theta(\mathbf{x}_t, \mathbf{c}_i)\|^2]\}}{\sum_j \exp\{-\mathbb{E}_{t,\epsilon}[\|\epsilon - \epsilon_\theta(\mathbf{x}_t, \mathbf{c}_j)\|^2]\}} \tag{6}$$

利用蒙特卡洛方法对每对采样进行无偏估计并且计算：

$$\frac{1}{N} \sum_{i=1}^{N} \left\| \epsilon_i - \epsilon_\theta(\sqrt{\bar{\alpha}_{t_i}}\mathbf{x} + \sqrt{1 - \bar{\alpha}_{t_i}}\epsilon_i, \mathbf{c}_j) \right\|^2 \tag{7}$$

模型图如下：

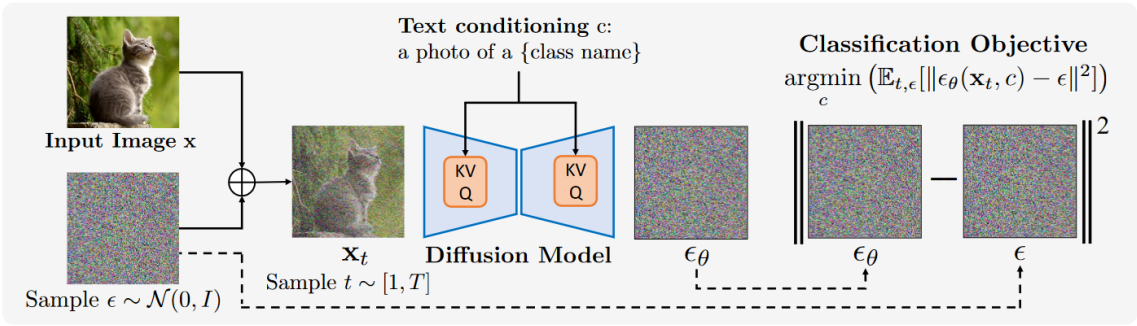


Figure 1. **Overview of our Diffusion Classifier approach:** Given an input image $\mathbf{x}$ and a set of possible conditioning inputs (e.g., text for Stable Diffusion or class index for DiT, an ImageNet class-conditional model), we use a diffusion model to choose the one that best fits this image. Diffusion Classifier is theoretically motivated through the variational view of diffusion models and uses the ELBO to approximate $\log p_\theta(\mathbf{x} \mid \mathbf{c})$. Diffusion Classifier chooses the conditioning $\mathbf{c}$ that best predicts the noise added to the input image. *Diffusion Classifier can be used to extract a zero-shot classifier from Stable Diffusion and a standard classifier from DiT without any additional training.*

算法如下:

Algorithm 2 `Diffusion Classifier (Adaptive)`

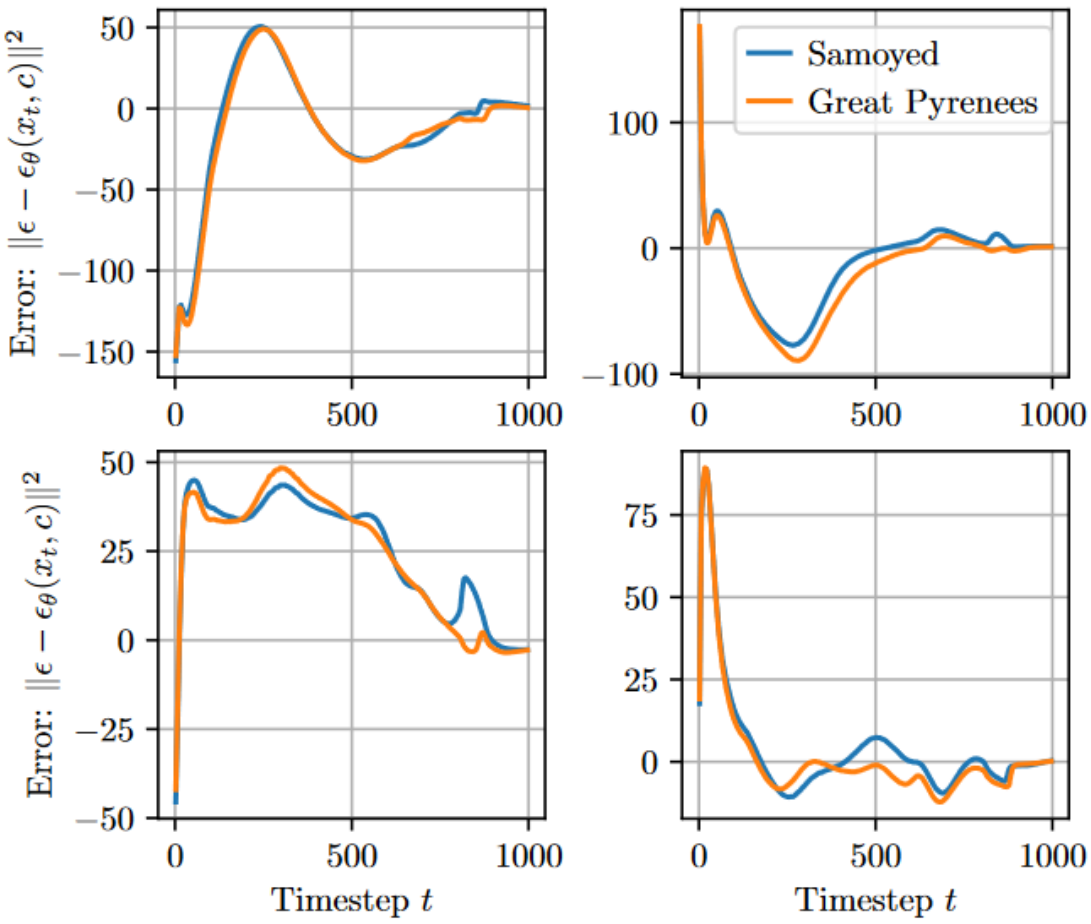
Input: test image $\mathbf{x}$, conditioning inputs $\mathcal{C} = \{\mathbf{c}_i\}_{i=1}^n$ (e.g., text embeddings or class indices), number of stages N_{stages}, list `KeepList` of number of $\mathbf{c}_i$ to keep after each stage, list `TrialList` of number of trials done by each stage
Initialize $\texttt{Errors}[\mathbf{c}_i] = \text{list}()$ for each $\mathbf{c}_i$
Initialize `PrevTrials` = 0 // How many times we've tried each remaining element of $\mathcal{C}$ so far
for stage i $= 1, \ldots, N_{\text{stages}}$ **do**
  for trial $j = 1, \ldots, \texttt{TrialList}[i] - \texttt{PrevTrials}$ **do**
    Sample $t \sim [1, 1000]$
    Sample $\epsilon \sim \mathcal{N}(0, I)$
    $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x} + \sqrt{1 - \bar{\alpha}_t}\epsilon$
    for conditioning $\mathbf{c}_k \in \mathcal{C}$ **do**
      $\texttt{Errors}[\mathbf{c}_k].\text{append}(\|\epsilon - \epsilon_\theta(\mathbf{x}_t, \mathbf{c}_k)\|^2)$
    end for
  end for
  $\mathcal{C} \leftarrow \underset{\substack{\mathcal{S} \subset \mathcal{C}; \\ |\mathcal{S}| = \texttt{KeepList}[i]}}{\arg\min} \sum_{\mathbf{c}_k \in \mathcal{S}} \text{mean}(\texttt{Errors}[\mathbf{c}_k])$ // Keep top $\texttt{KeepList}[i]$ conditionings $\mathbf{c}_k$ with the lowest errors
  `PrevTrials` = `TrialList[i]`
end for
return $\underset{\mathbf{c}_i \in \mathcal{C}}{\arg\min}\ \text{mean}(\texttt{Errors}[\mathbf{c}_i])$

3.通过差异测试减少方差

为了准确估计每个类的式子（4）似乎需要大量的样本，事实上，即使使用上千个样本进行蒙特卡洛估计也不够精确，无法可靠的区分类别。然后，对于一个分类需要的是预测误差之间的相对差异而不是绝对差异，因此可以重写为：

$$\frac{1}{\sum_j \exp\left\{\mathbb{E}_{t,\epsilon}[\|\epsilon - \epsilon_\theta(\mathbf{x}_t, \mathbf{c}_i)\|^2 - \|\epsilon - \epsilon_\theta(\mathbf{x}_t, \mathbf{c}_j)\|^2]\right\}} \tag{8}$$

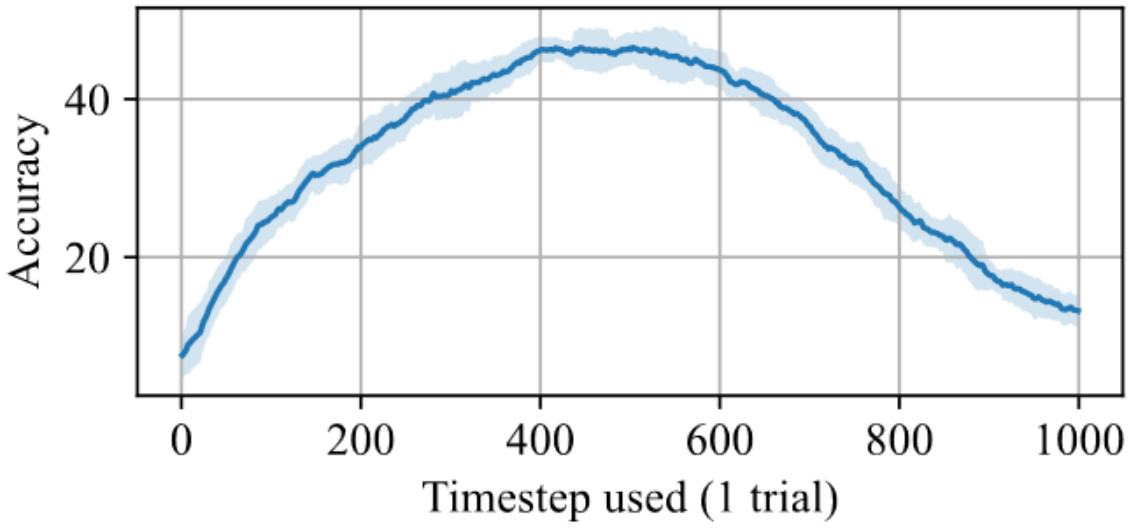
图2:



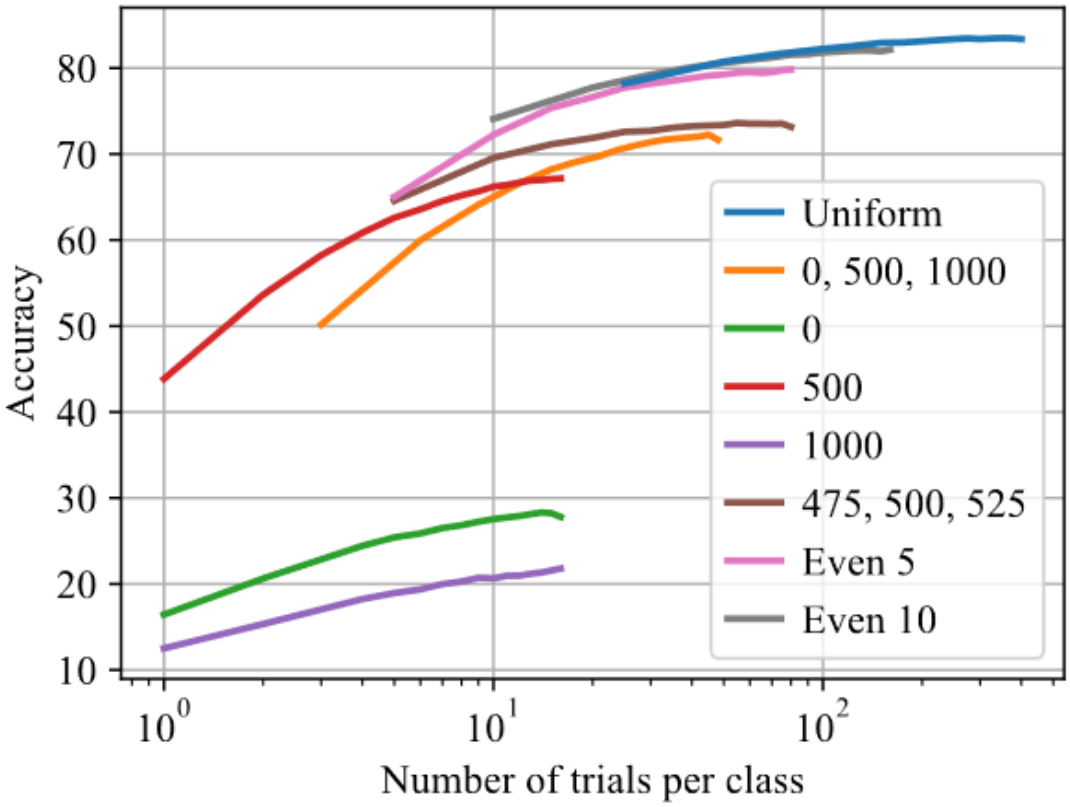
我们展示了大比利牛斯犬图像和两个提示（“萨摩耶”和“大比利牛斯”）的 ε 预测误差。每个子图对应于一个 εi，在每个 t ∈ {1, 2, ..., 1000} 处评估误差。 4 个图中每个时间步长的误差均值均值为零，且越低越好。不同 ε 之间的 ε 预测误差的方差很大，但提示之间的误差差的方差要小得多。

4.实际考虑

（1）时间步长的影响



扩散分类器是一种用于估计 pθ(ci | x) 的理论原理方法，它使用时间步 t 上的均匀分布来估计 ε 预测误差。在这里，我们检查 t 上的替代分布是否会产生更准确的结果。图 3 显示了每个类别仅使用单个时间步长评估时的 Pets 准确性。也许直观上来说，使用中间时间步长 (t ≈ 500) 时准确度最高。这就引出了一个问题：**我们能否通过对中间时间步长进行过采样和对低或高时间步长进行欠采样来提高准确性**?



不同时间步采样策略的零样本缩放曲线。我们评估了多种策略来选择评估 ε 预测误差的时间步长。每个策略名称表示它使用的时间步长，例如，“0”仅使用第一个时间步长，“0,500,1000”仅使用第一个、中间和最后一个时间步长，“Even 10”使用 10 个均匀间隔的时间步长。随着试验次数的增加，我们在选定的时间步分配更多的 ε 评估。从一组有限的时间步长中重复采样的策略（例如“475、500、525”）在试验中的扩展性很差。统一使用全范围 [1, 1000] 范围内的时间步长效果最佳。

（2）高效分类

我们发现更简单的方法也同样有效。我们将评估分为一系列阶段，在每个阶段中，我们对每个剩余的 ci 进行多次尝试，然后删除平均误差最高的阶段。这使我们能够有效地消除几乎肯定不是最终输出的类，并将更多计算分配给合理的类。例如，在 Pets 数据集上，我们的 Nstages = 2。我们在第一阶段对每个类尝试 25 次，然后剪枝到平均误差最小的 5 个类。最后，在第二阶段，我们对剩余 5 个类别中的每一个类别额外尝试 225 次。在算法 2 中，我们将其写为 KeepList = (5, 1) 和 TrialList = (25, 250)。采用这种评估策略，在 RTX 3090 GPU 上对一张宠物图像进行分类需要 18 秒。由于我们的工作重点是理解扩散模型的功能，而不是开发完全实用的推理算法，因此我们没有显着调整评估策略。有关适应性评估的更多详细信息请参见附录 A。

进一步减少推理时间可能是未来工作的宝贵途径。当有很多类时，推理仍然不切实际。即使使用我们的自适应策略，在 512×512 分辨率下使用稳定扩散对具有 1000 个类别的单个 ImageNet 图像进行分类也需要大约 1000 秒。表 7 显示了每个数据集的推理时间，我们在第 7 节中讨论了有前景的加速方法。

实验细节

1.零样本分类

扩散分类器设置

扩散分类器设置：零样本扩散分类器利用稳定扩散 2.0 [65]，这是一种在 LAION5B [68] 的过滤子集上训练的文本到图像潜在扩散模型。此外，我们不使用平方 l2 范数来计算 ε 预测误差，而是将 l1 和 l2 选择保留为每个数据集的推理超参数。

基准线

我们使用两个强判别性零样本模型提供结果：(a) CLIP ResNet-50 [60] 和 (b) OpenCLIP ViT-H/14 [11]。我们提供这些仅供参考，因为这些模型是在不同的数据集上训练的，其架构与我们的非常不同，因此不能进行同类比较。我们进一步将我们的方法与从扩散模型中提取类标签的两种替代方法进行比较：(c) 合成 SD 数据：我们在使用稳定扩散生成的合成数据上训练 ResNet-50 分类器（以类名称作为提示），(d) SD特征：该基线不是零样本分类器，因为它需要真实世界图像和类名的标记数据集。受 Label-DDPM [3] 的启发，我们提取稳定扩散特征（中间层 U-Net 特征，时间步长 t = 100，分辨率为 [8 × 8 × 1024]），然后在提取的特征上拟合 ResNet50 分类器，相应的真实标签。

数据集

由于计算限制，我们对 ImageNet 的 2000 个测试图像进行了评估。我们还在 Winoground 基准上评估零样本组合推理能力

2.监督学习分类

扩散分类器设置

我们在DiT之上构建扩散分类器，这是一种仅在 ImageNet-1k上训练的类条件潜在扩散模型。我们使用分辨率为 2562 和 5122 的 DiT-XL/2，并对每个图像的每个类别进行 250 次评估。

基准线

我们与以下在 ImageNet1k 上使用交叉熵损失训练的判别模型进行比较：ResNet-18、ResNet-34、ResNet-50 和 ResNet101，以及 ViT-L/32、ViT-L/16 和ViT-B/16。

数据集

我们评估模型在 ImageNet [17] 上的分布内精度以及对 ImageNetV2 [64]、ImageNet-A [30] 和 ObjectNet [4] 的分布外泛化。 ObjectNet 准确度是在与 ImageNet 共享的 113 个类上计算的。由于计算限制，我们在 ImageNet 的 10,000 张验证图像上评估扩散分类器的准确性。我们在相同的 10,000 个图像子集上计算基线的 ImageNet 准确度。

实验结果

1. 扩散分类器与 CLIP 等最先进的零样本分类器相比如何？
2. 我们的方法与扩散模型分类的替代方法相比如何？
3. 我们的方法在组合推理任务上表现如何？
4. 我们的方法与在同一数据集上训练的判别模型相比效果如何？
5. 与各种分布变化的判别分类器相比，我们的模型有多稳健？

1.零样本分类结果

	Zero-shot?	Food	CIFAR10	Aircraft	Pets	Flowers	STL10	ImageNet	ObjectNet
Synthetic SD Data	✓	12.6	35.3	9.4	31.3	22.1	38.0	18.9	5.2
SD Features	✗	73.0	84.0	**35.2**	75.9	**70.0**	87.2	56.6	10.2
Diffusion Classifier (ours)	✓	**77.7**	**88.5**	26.4	**87.3**	66.3	**95.4**	**61.4**	**43.4**
CLIP ResNet-50	✓	81.1	75.6	19.3	85.4	65.9	94.3	58.2	40.0
OpenCLIP ViT-H/14	✓	92.7	97.3	42.3	94.6	79.9	98.3	76.8	69.2

Table 1. **Zero-shot classification performance**. Our zero-shot Diffusion Classifier method (which utilizes Stable Diffusion) significantly outperforms the zero-shot diffusion model baseline that trains a classifier on synthetic SD data. Diffusion Classifier also generally outperforms the baseline trained on Stable Diffusion features, despite "SD Features" using the entire training set to train a classifier. Finally, although making a fair comparison is difficult due to different training datasets, our generative approach surprisingly outperforms CLIP ResNet-50 and is competitive with OpenCLIP ViT-H. We report average accuracy or mean-per-class accuracy in accordance with [44].

扩散分类器的性能显着优于合成 SD 数据基线（一种从扩散模型中提取信息的替代零样本方法）。这可能是因为在综合生成的数据上训练的模型学会了依赖于不会转移到真实数据的特征。令人惊讶的是，我们的方法通常也优于 SD 特征基线，这是一个使用每个数据集的整个标记训练集以监督方式训练的分类器。相比之下，我们的方法是零样本的，不需要额外的训练或标签。最后，虽然由于训练数据集的差异很难进行公平的比较，但我们的方法优于 CLIP ResNet-50，并且与 OpenCLIP ViT-H 具有竞争力。

Dataset	Resolution	Aesthetic	SFW	A + S	R + A + S
Food	61.5	90.5	99.9	90.5	56.3
CIFAR10	0.0	3.4	90.3	3.2	0.0
Aircraft	98.6	95.7	100.0	95.6	94.4
Pets	1.1	89.1	100.0	89.1	0.9
Flowers	0.0	82.4	100.0	82.4	0.0
STL10	0.0	31.6	93.1	30.6	0.0
ImageNet	4.5	84.1	98.0	82.5	3.4
ObjectNet	98.8	20.5	98.8	20.3	20.2

Table 2. **How in-distribution is each test set for Stable Diffusion?** We show the percentage of each test set that would remain after the Stable Diffusion 2.0 data filtering process. The first three columns show the percentage of images that pass resolution ($\geq 512^2$), aesthetic (≥ 4.5), and safe-for-work (≤ 0.1) thresholds, respectively. The last two columns show the proportion of images that pass multiple filters, and the last column (R + A + S) corresponds to the actual filtering criteria used to train SD 2.0.

这是生成方法性能的重大进步，并且有明显的改进途径。首先，我们没有执行手动提示调整，只是使用 CLIP 作者使用的提示。将提示调整为稳定扩散训练分布应该可以提高其识别能力。其次，我们怀疑稳定扩散分类的准确性可以通过更广泛的训练分布来提高。 Stable Diffusion 在 LAION-5B [68] 的一个子集上进行训练，该子集经过积极过滤，以去除低分辨率、潜在的 NSFW 或不美观的图像。这降低了它看到我们许多数据集的相关数据的可能性。表 2 中最右一列显示，在应用所有三个过滤器后，CIFAR10、Pets、Flowers、STL10、ImageNet 和 ObjectNet 中仅保留 0-3% 的测试图像。因此，许多零样本测试集完全不符合稳定扩散的分布。如果在较少策划的训练集上训练稳定扩散，扩散分类器的性能可能会显着提高。

提高组合推理能力

大型文本到图像的扩散模型能够生成具有令人印象深刻的组合概括的样本。在本节中，我们将测试这种生成能力是否可以转化为改进的构图推理。

Winoground Benchmark

Figure 5. **Example visualizations of Winoground swap types.** Each category corresponds to a different type of linguistic swap in the caption. Object swaps noun phrases, Relation swaps verbs, adjectives, or adverbs, and Both can swap entities of both kinds.

提出了一个新的任务Winoground，用于测量视觉-语言的组合推理，要求模型必须正确配对两个图片和对应的两个caption，问题在于两个caption的单词构成完全相同，但顺序不同。因此模型不仅要很好地编码文本和图像（即对每个模态中存在的组合结构敏感），而且还必须能够跨两种模态合成信息。

每个示例都通过两个标题之间的语言交换类型（对象、关系和两者）进行标记：

1. 对象：重新排序元素，例如通常指代现实世界对象/主题的名词短语。
2. 2. 关系：重新排序动词、形容词、介词和/或修饰对象的副词等元素。
3. 3.两者：前两种类型的组合。

评价方式包含三个方面：

text score：对于图像和标题对，固定图像，替换标题如果能够把两个图像-标题对正确配对就计分。

$$f(C_0, I_0, C_1, I_1) = \begin{cases} 1 & \text{if } s(C_0, I_0) > s(C_1, I_0) \\ & \quad \text{and } s(C_1, I_1) > s(C_0, I_1) \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

image score：与上面一致，固定标题替换图像。

$$g(C_0, I_0, C_1, I_1) = \begin{cases} 1 & \text{if } s(C_0, I_0) > s(C_0, I_1) \\ & \quad \text{and } s(C_1, I_1) > s(C_1, I_0) \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

最终指标：f+g

$$h(C_0, I_0, C_1, I_1) = \begin{cases} 1 & \text{if } f(C_0, I_0, C_1, I_1) \\ & \quad \text{and } g(C_0, I_0, C_1, I_1) \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

结果:

Model	Text	Image	Group
MTurk Human	**89.50**	**88.50**	**85.50**
Random Chance	25.00	25.00	16.67
VinVL	**37.75**	17.75	14.50
UNITER$_{large}$	**38.00**	14.00	10.50
UNITER$_{base}$	**32.25**	13.25	10.00
ViLLA$_{large}$	**37.00**	13.25	11.00
ViLLA$_{base}$	**30.00**	12.00	8.00
VisualBERT$_{base}$	15.50	2.50	1.50
ViLT (ViT-B/32)	**34.75**	14.00	9.25
LXMERT	19.25	7.00	4.00
ViLBERT$_{base}$	23.75	7.25	4.75
UniT$_{ITMfinetuned}$	19.50	6.25	4.00
FLAVA$_{ITM}$	**32.25**	20.50	14.25
FLAVA$_{Contrastive}$	**25.25**	13.50	9.00
CLIP (ViT-B/32)	**30.75**	10.50	8.00
VSE++$_{COCO}$ (ResNet)	22.75	8.00	4.00
VSE++$_{COCO}$ (VGG)	18.75	5.50	3.50
VSE++$_{Flickr30k}$ (ResNet)	20.00	5.00	2.75
VSE++$_{Flickr30k}$ (VGG)	19.75	6.25	4.50
VSRN$_{COCO}$	17.50	7.00	3.75
VSRN$_{Flickr30k}$	20.00	5.00	3.50

Table 3. Results on the Winoground dataset across the text, image and group score metrics. Results above random chance in **bold**.

获得高文本分数极具挑战性。人类（通过 Mechanical Turk）在此基准测试中达到了 89.5% 的准确率，但即使是最好的模型也勉强高于偶然性。模型只有理解每种模态的组成结构才能发挥作用。人们发现 CLIP 在这个基准测试中表现不佳，因为它的嵌入往往更像是一个“概念袋”，无法将主语与属性或动词绑定

Model	Object	Relation	Both	Average
Random Chance	25.0	25.0	25.0	25.0
CLIP ViT-L/14	27.0	25.8	57.7	28.2
OpenCLIP ViT-H/14	39.0	26.6	57.7	33.0
Diffusion Classifier (ours)	**46.1**	**29.2**	**80.8**	**38.5**

Table 3. **Compositional reasoning results on Winoground**. Diffusion Classifier obtains signficantly better text score (Eq. 9) than the contrastive baselines for all three swap categories.

表 3 将扩散分类器与两个强对比基线进行了比较：OpenCLIP ViT-H/14（其文本嵌入启用了稳定扩散条件）和 CLIP ViT-L/14。扩散分类器明显优于 Winoground 上的两种判别方法。我们的方法在所有三种交换类型上都更强，即使是具有挑战性的“关系”交换，其中对比基线的效果并不比随机猜测更好。这表明扩散分类器的生成方法表现出更好的组合推理能力。由于 Stable Diffusion 使用与 OpenCLIP ViT-H/14 相同的文本编码器，因此这种改进来自概念与图像更好的跨模式绑定。总的来说，我们发现令人惊讶的是，仅考虑样本生成进行训练的稳定扩散可以重新用于如此强大的分类器和推理器，而无需任何额外的训练。

监督分类结果

我们将利用 ImageNet 训练的 DiT-XL/2 模型 [58] 的扩散分类器与在 ImageNet 上训练的 ViTs 和 ResNets 进行比较。此设置特别有趣，因为它可以在同一数据集上训练的模型之间进行公平比较。表 4 显示扩散分类器的性能优于 ResNet-101 和 ViTL/32。扩散分类器在分辨率 256 和 512 下的 ImageNet 准确率分别为 77.5% 和 79.1%。据我们所知，我们是第一个证明训练学习 $p_\theta(x \mid c)$ 的生成模型可以实现与高度竞争的判别方法相当的 ImageNet 分类精度。

Method	ID	OOD		
	IN	IN-V2	IN-A	ObjectNet
ResNet-18	70.3	57.3	1.1	27.2
ResNet-34	73.8	61.0	1.9	31.6
ResNet-50	76.7	63.2	0.0	36.4
ResNet-101	77.7	65.5	4.7	39.1
ViT-L/32	77.9	64.4	11.9	32.1
ViT-L/16	80.4	67.5	16.7	36.8
ViT-B/16	**81.2**	**69.6**	20.8	**39.9**
Diffusion Classifier 256^2	77.5	64.6	20.0	32.1
Diffusion Classifier 512^2	79.1	66.7	**30.2**	33.9

Table 4. **Standard classification on ImageNet.** We compare Diffusion Classifier (using DiT-XL/2 at 256^2 and 512^2 resolutions) to discriminative models trained on ImageNet. We highlight cells where Diffusion Classifier does better. All models (generative and discriminative) have only been trained on ImageNet.

更好的分布外泛化(better out-of-distribution generalization)

我们发现，令人惊讶的是，扩散分类器在 ImageNet-A 上比所有基线都具有更强的分布外 (OOD) 性能。事实上，我们的方法显示出与判别方法相比有质的不同且更好的 OOD 泛化行为。之前的工作 [74] 评估了数百个判别模型，发现它们的分布内 (ID) 和 OOD 精度之间存在紧密的线性关系——对于给定的 ID 精度，没有模型的 OOD 效果比线性关系预测的更好。对于仅在 ImageNet-1k（无额外数据）上训练的模型，从对抗性训练到有针对性的增强到不同架构的各种方法都无法实现比预测更好的 OOD 准确性。我们将这些判别模型的 ID ImageNet 准确度（二次采样到与 ImageNet-A 重叠的类）和 ImageNet-A 上的 OOD 准确度之间的关系显示为图 6 中的蓝点（“标准训练”）。OOD 准确度进行了描述通过分段线性拟合，ResNet-50 模型的 ImageNet 精度存在问题，该模型用于识别组成 ImageNet-A 的硬图像。没有任何判别模型显示出有意义的“有效鲁棒性”，即模型的实际 OOD 精度与线性拟合预测的 OOD 精度之间的差距 [74]。

然而，与这数百个判别模型相比，扩散分类器在 ImageNet-A 上实现的 OOD 精度比预测的要高得多。图 6 显示扩散分类器远高于线性拟合，并实现了 15-25% 的有效鲁棒性。据我们所知，这是第一种在训练期间无需使用任何额外数据即可实现显着有效鲁棒性的方法。我们的发现有一些警告。扩散分类器并未表现出对 ImageNetV2 或 ObjectNet 分布变化的有效鲁棒性的提高，尽管这些变化的性质可能与 ImageNetA 不同。扩散分类器在 ImageNet-A 上可能会做得更好，因为它的预测可能与用于查找 ImageNet-A 困难示例的（判别性）ResNet-50 相关性较低。尽管如此，ImageNet-A 有效鲁棒性的显着提高是令人兴奋的，这表明生成分类器是实现更好的分布偏移鲁棒性的有前途的方法。

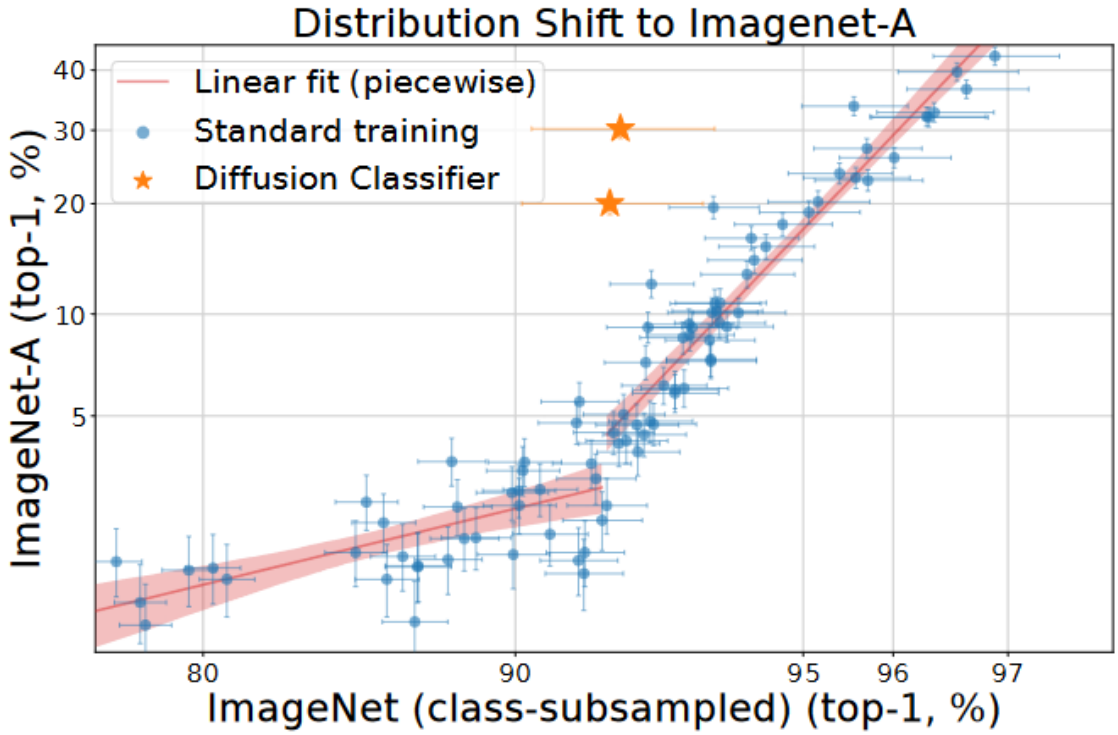


Figure 6. **Diffusion Classifier exhibits effective robustness without using extra labeled data**. Compared to discriminative models trained on the same amount of labeled data ("standard training"), Diffusion Classifier achieves much higher ImageNet-A accuracy than predicted by its ImageNet accuracy. Diffusion Classifier points correspond to DiT-XL/2 at resolution 256^2 and 512^2. Points are shown with 99.5% Clopper-Pearson confidence intervals. The red lines show the linear relationship between ID and OOD accuracy for discriminative models, with a "break" at the accuracy of the model used to create ImageNet-A. The axes were adjusted using logit scaling, since accuracies fall within $[0, 100]$.

扩散分类器在不使用额外标记数据的情况下表现出有效的鲁棒性。与在相同数量的标记数据上训练的判别模型（"标准训练"）相比，扩散分类器实现的 ImageNet-A 精度比其 ImageNet 精度预测的要高得多。扩散分类器点对应于分辨率 256 和 512 处的 DiT-XL/2。点以 99.5% Clopper-Pearson 置信区间显示。红线显示了判别模型的 ID 和 OOD 准确度之间的线性关系，其中用于创建 ImageNet-A 的模型的准确度出现"中断"。由于精度在 [0, 100] 范围内，所以使用 logit 缩放来调整轴。

稳定训练和无过拟合

Diffusion Classifier 的 ImageNet 准确性尤其令人印象深刻，因为 DiT 仅通过随机水平翻转进行训练，这与使用 RandomResizedCrop、Mixup 、RandAugment 和其他技巧来避免过度拟合的典型分类器不同。使用更高级的增强训练 DiT 应该会进一步提高其准确性。此外，DiT 训练是稳定的，学习率固定，除了权重衰减之外没有正则化 [58]。这与 ViT 训练形成鲜明对比，ViT 训练不稳定并且经常出现 NaN，尤其是对于大型模型 [28]。这些结果表明，生成目标 $\log p_\theta(x \mid c)$ 可能是一种有前途的方法，可以将训练扩展到更大的模型，而不会过度拟合或不稳定

结论和讨论

我们通过利用扩散模型作为条件密度估计器来研究扩散模型的零样本和标准分类能力。通过对每个类的学习条件 ELBO 进行简单、无偏的蒙特卡罗估计，我们提取了扩散分类器——一种无需任何额外训练即可将任何条件扩散模型转变为分类器的强大方法。我们发现该分类器缩小了在零样本和标准分类方面与最先进的判别方法的差距，并且在**多模态组合推理方面**显着优于已有方法。扩散分类器还对**分布变化表现出更好的"有效鲁棒性"**。加速推理虽然推理时间目前是一个实际瓶颈，但有几种明确的方法可以加速扩散分类器。从默认的 512 × 512（标清）降低分辨率将带来显着的加速。 256 × 256 的推理速度至少快 4 倍，128 × 128 的推理速度将快 16 倍以上。另一种选择是使用弱判别模型来快速消除明显不正确的类。附录 B 表明，这将同时提高准确性并减少推理时间。基于梯度的搜索可以通过扩散模型反向传播来求解 arg maxc log p(x | c)，这可以消除运行时对类数量的依赖。新的架构可以设计为仅在网络末端使用类条件 c，从而实现跨类的中间激活的重用。最后，请注意，错误预测过程很容易并行化。未来有了足够的扩展或更好的 GPU，所有扩散分类器步骤都可以与单个前向传递的延迟并行完成。

扩散模型设计决策的作用

由于我们不改变扩散分类器的基本扩散模型，因此扩散训练期间所做的选择会影响分类器。例如，Stable Diffusion根据 OpenCLIP 的文本嵌入来调节图像生成。然而，OpenCLIP 中的语言模型比 T5-XXL 等开放式大型语言模型要弱得多，因为它仅针对图像标题对中可用的文本数据进行训练，而图像标题对是互联网上总文本数据的很小的子集。因此，我们认为在 T5-XXL 嵌入之上训练的扩散模型（例如 Imagen ）应该显示更好的零样本分类结果，但这些模型不是开源的，无法进行实证验证。其他设计选择，例如是否在潜在空间（例如稳定扩散）或像素空间（例如 DALLE 2）中执行扩散，也可能影响分类器的对抗鲁棒性，并为未来的工作提供有趣的途径。

总之，虽然生成模型以前未能达到分类的判别模型，但当今生成模型的进步速度意味着它们正在迅速迎头赶上。我们强大的分类、多模态组合推理和稳健性结果代表了朝着这个方向迈出的令人鼓舞的一步。

扩散分类算法

尽管扩散分类器可以直接按照算法 1 中描述的过程工作，但我们对加速推理感兴趣，如第 4.2 节中所述。算法 2 显示了高效的扩散分类器过程，该过程自适应地选择要继续评估的类别。表 6 显示了用于每个零样本数据集的评估策略。我们根据每个数据集中的类数量精心挑选策略。通过更多的评估，可能会进一步提高准确性。

Algorithm 2 `Diffusion Classifier (Adaptive)`

Input: test image $\mathbf{x}$, conditioning inputs $\mathcal{C} = \{\mathbf{c}_i\}_{i=1}^n$ (e.g., text embeddings or class indices), number of stages N_{stages}, list `KeepList` of number of $\mathbf{c}_i$ to keep after each stage, list `TrialList` of number of trials done by each stage
Initialize $\texttt{Errors}[\mathbf{c}_i] = \text{list}()$ for each $\mathbf{c}_i$
Initialize `PrevTrials` = 0 // How many times we've tried each remaining element of $\mathcal{C}$ so far
for stage i $= 1, \ldots, N_{\text{stages}}$ **do**
  for trial $j = 1, \ldots, \texttt{TrialList}[i] - \texttt{PrevTrials}$ **do**
    Sample $t \sim [1, 1000]$
    Sample $\epsilon \sim \mathcal{N}(0, I)$
    $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x} + \sqrt{1 - \bar{\alpha}_t}\epsilon$
    for conditioning $\mathbf{c}_k \in \mathcal{C}$ **do**
      $\texttt{Errors}[\mathbf{c}_k].\text{append}(\|\epsilon - \epsilon_\theta(\mathbf{x}_t, \mathbf{c}_k)\|^2)$
    end for
  end for
  $\mathcal{C} \leftarrow \underset{\substack{\mathcal{S} \subset \mathcal{C}; \\ |\mathcal{S}| = \texttt{KeepList}[i]}}{\arg\min} \sum_{\mathbf{c}_k \in \mathcal{S}} \text{mean}(\texttt{Errors}[\mathbf{c}_k])$ // Keep top $\texttt{KeepList}[i]$ conditionings $\mathbf{c}_k$ with the lowest errors
  `PrevTrials` = `TrialList`[i]
end for
return $\underset{\mathbf{c}_i \in \mathcal{C}}{\arg\min}\ \text{mean}(\texttt{Errors}[\mathbf{c}_i])$

Dataset	Prompts kept per stage	Evaluations per stage	Avg. evaluations per class	Total evaluations
Food101	20 10 5 1	20 50 100 500	50.7	5120
CIFAR10	5 1	50 500	275	2750
Aircraft	20 10 5 1	20 50 100 500	51	5100
Pets	5 1	25 250	51	1890
Flowers102	20 10 5 1	20 50 100 500	50.4	5140
STL10	5 1	100 500	300	3000
ImageNet	500 50 10 1	50 100 500 1000	100	100000
ObjectNet	25 10 5 1	50 100 500 1000	118.6	13400

Table 6. Adaptive evaluation strategy for each zero-shot dataset.