ARTICLE
https://doi.org/10.26599/AIR.2023.9150021
CAAI Artificial Intelligence Research

CamDiff: 基于扩散模型的伪装图像增强

罗雪婧 [†1], 王硕 [†12], 吴宗蔚[1], Christos Sakaridis[1], 程云[1], 范登平 [*1], Luc Van Gool[1]

ABSTRACT

伪装物体检测（Camouflaged Object Detection，COD）是一个新兴领域，旨在识别那些与其周围环境融为一体的物体。该领域具有广泛的实际应用。尽管近年来基于学习的模型在这个方向上取得了显著成效，但它们在鲁棒性方面仍存在局限性。具体来说，现有方法可能将本应显著的物体错误分类为伪装物体，尽管它们具有相反的特征。这一局限性可能源自于训练图像缺乏多样化的模式，导致模型在面对显著物体时的鲁棒性降低。为了解决多模式训练图像的稀缺问题，本研究受人工智能生成内容（AIGC）的启发，提出了一种新颖的方法 **CamDiff**。本文运用潜在扩散模型，在伪装场景中合成显著物体，同时借助对比语言-图像预训练（CLIP）模型的零样本图像分类能力，避免合成失败并确保合成物体与输入提示相符。因此，合成图像不仅保留了其原始的伪装标签，还融入了显著物体，使得伪装场景具有更丰富的特征。用户调研结果表明，本文合成的场景中的显著物体更能吸引用户的注意，因此，这样的样本对现有伪装物体检测模型来说构成了更大的挑战。本文的 CamDiff 可以在低成本下实现灵活的编辑和高效的大规模数据集生成。它显著增强了伪装物体检测基线模型的训练和测试阶段，使它们在不同领域具有鲁棒性。本文新生成的数据集和源代码已发布在 https://github.com/drlxj/CamDiff。

KEYWORDS

AI 生成内容，扩散模型，伪装物体检测，显著性物体检测

1 引言

伪装是自然物体通过生物适应进化而发展出的一种捕食和防御策略 [6]。从视觉上看，生物体改变其外观以适应周围环境，使其在第一眼看上去难以被发现。受此现象启发，近期伪装物体检测（COD）的研究领域 [7–9] 在计算机视觉界引起了极大关注 [10–12]。这一领域的研究具有广泛应用，包括医学图像诊断和分割 [13–15]、物种发现 [16] 以及裂纹检测 [17]。

尽管一些工作 [1, 5, 15] 直接将成熟的显著物体检测（SOD）技术扩展到 COD 任务上，但显著物体与伪装物体是两种对立的类别。显著性水平越高，伪装程度就越低，反之亦然 [18]。理想情况下，应该有一种方法当能同时检测显著物体和伪装物体，并注意模仿水平，但将这两种对立的模式错误地归为同一语义标签是不可接受的，因为这可能会妨碍各领域的操作效率。例如，在制造或质量控制过程中，将关键部件误认为伪装物体可能导致生产错误、延误或产品质量受损。在医疗领域，将显著的医疗状况（如明显的症状或异常）错误地分类为伪装物体可能导致误诊或延迟治疗，影响患者结果，阻碍医疗干预的有效性。因此，本文认为开发针对这两种不同物体类型的不同策略至关重要。SOD 模型基于全局和局部对比，而 COD 模型应避免这些高显著性区域。遗憾的是，本文的实验（章节. 4）显示，当前 COD 方法在显著物体与伪装物体同时存在于图像中时准确度下降。

正如图. 1所示，本文测试了多种最新（SOTA）的、仅用伪装样本训练的 COD 方法在面对显著物体时的鲁棒性。大多数 COD 方法却依然检测出显著物体。这些结果表明，当前的 COD 模型对于含有显著物体的场景还不够鲁棒。具体来说，PFNet [2] 和 ZoomNet [5] 等算法，仅检测更显著的物体（黄色球体），而忽略了不太显著的物体（绿色球体）。因此，本文推测现有的 COD 工作可能只学习到如何区分前景和背景，而不是伪装和显著性模式或提示。这突显了本文研究伪装模式的必要性，使 COD 模型更有效。

1 Department of Information Technology and Electrical Engineering, ETH Zurich, Zurich, 8092, Switzerland. † Co-first authors. * Corresponding author: Deng-Ping Fan (dengpfan@gmail.com).
2 School of Systems Science, Beijing Normal University, Beijing, 100875, China. 本文是 CamDiff: Camouflage Image Augmentation via Diffusion, CAAI AIR2023 的翻译版，由罗雪婧翻译，吴宗蔚和范登平校稿。

图 1 当前 COD 模型在处理显著物体的图像时的视觉结果。由于物体是显著的，所以对于 COD 任务，真实值（GT）应该是全黑的。然而，现有的 COD 方法，特别是 PFNet 和 ZoomNet，在处理显著物体方面的鲁棒性较差。

为了区分显著和伪装模式，一个直观的想法是通过对比学习训练网络，该方法在其他视觉任务中已显示出其有效性 [19–21]。正如 [22–24] 中所建议的那样，强大的数据增强可以显著支持对比学习，从而有效地建模特征表示。然而，在本文的设置中，由于常规伪装数据集中缺乏显著物体，生成对比训练的正负样本对是不可行的。此外，现有的 COD 数据集主要包含单个物体，使得对比学习的直接扩展不可行。同时，收集并标注包含伪装和显著物体的新数据集会消耗大量劳动。

在研究中，本文旨在提高未来 COD 模型对显著物体的鲁棒性。为实现这一目标，本文提出了通过利用最近的扩散模型 [25, 26] 作为数据增强形式来生成合成图像的方法。这一方法受到人工智能生成内容（AIGC）[27, 28] 和大规模生成模型成功的启发。虽然近期已有尝试使用扩散图像进行数据增强，但这些努力仅适用于更常见的场景，如日常室内场景 [29] 或城市景观 [30]，其中域差距较小。相比之下，本文对伪装场景更感兴趣，这些场景对预训练的扩散模型来说是罕见且具有挑战性的。这些差异使本文的任务在合成具有大域差距的多模式图像方面非常具有挑战性，据本文所知，该技术在伪装设置下尚未得到解决。此外，现有工作 [31] 依赖于额外的冻结权重深度网络来生成伪标签作为监督，限制了它们的性能和应用。这些局限性促使本文设计了一种新的框架，用于在伪装场景中生成真实的显著物体。本文的方法与同时期的扩散增强方法 [32, 33] 不同，主要在于（a）不可忽视的域差距和（b）保留的伪装标签。

为了解决当前问题，本文提出了一个基于扩散的对抗性生成框架，名为 **CamDiff**。具体来说，本文的方法包括一个生成器和一个鉴别器。生成器是一个固定权重的潜在扩散模型（LDM）[25]，它已经在大量类别上进行了训练，因此能够大规模地合成最显著的物体。对于鉴别器，本文采用了对比语言-图像预训练（CLIP）[34] 的通用方式。本文的鉴别器比较输入的图像提示和合成的物体，以确保语义一致性。为了保留原始的伪装标签，本文只在背景中添加生成的显著物体，即在真实标签之外。因此，CamDiff 有效地将问题转化为一种修复任务，无需额外的标注成本。通过这种方式，本文可以有效且轻松地实现定制编辑，从而从数据驱动的角度改进 COD 的发展。

本文的主要贡献总结如下：

- 本文提出了 CamDiff，它在保留原始标签的同时，可在伪装场景中叠加显著物体。这一框架便于在真实图像中整合和组合对比模式，而不会增加与学习和标记相关的额外成本。
- 本文进行实验，测试最先进的 COD 方法在 COD 测试集（即 Diff-COD）上的鲁棒性，这些测试集是使用 CamDiff 从原始 COD 测试集创建的。结果表明，当前的 COD 方法对显著物体的鲁棒性不足。
- 为了提高当前 COD 方法对显著物体的鲁棒性，本文使用 CamDiff 从原始 COD 训练集生成了一个名为 Diff-COD 的新训练集。实验结果表明，用这个新训练集训练现有的 COD 模型可以增强它们对显著物体的鲁棒性。

总的来说，本文的研究为伪装的概念提供了一个新的视角，新引入的伪装合成工具将成为推进这个迅速发展的领域的基础。

2 相关工作

2.1 扩散模型

扩散模型 [25, 26] 是一类生成模型，它们通过逐渐去除数据点中的噪声来从分布中生成样本。最近的研究 [35] 表明，扩散模型在高分辨率图像生成任务中超越了生成对抗网络（GANs）[36]，没有模式坍塌 [37] 和训练不稳定 [38] 的缺点，并在条件图像生成 [28] 方面取得了前所未有的结果。因此，它们已被应用于许多领域，如文本到图像和引导合成 [39, 40]、3D 形状生成 [41, 42]、分子预测 [43]、视频生成 [44] 和图像修复 [25]。

一些研究人员已经研究了扩散模型在图像修复方面的应用。例如，Meng 等人 [39] 发现扩散模型不仅可以填充图像的区域，而且可以根据图像的粗略草图进行条件填充。另一项研究 [45] 得出结论，当直接在图像修复任务上训练时，扩散模型可以平滑地填充图像区域，生成逼真的内容，而没有边缘伪影。

2.2 伪装物体检测

COD 旨在图像中检测隐藏的物体。许多研究（例如，SINet [1]、UGTR [46]、ZoomNet [5]）都集中于比较 COD

与 SOD 的区别，并得出结论，仅仅将 SOD 模型扩展应用于 COD 任务无法带来预期结果，因为目标物体具有不同的属性，即隐藏或显著。为了检测隐藏的图像，最近提出了许多方法。例如，一些方法利用多阶段策略解决伪装图像的隐藏问题。SINet [1] 是首个多阶段方法，用于定位和区分伪装物体。另一个多阶段方法是 SegMar [4]，它定位物体并逐步放大可能的物体区域以检测伪装物体。此外，多尺度特征聚合是另一个主要策略，已被许多方法采用，如 CubeNet [47]，通过引入 X 连接和注意力融合来整合低级和高级特征，以及 ZoomNet [5]，它在三个不同的尺度上处理输入图像，以全面探索候选物体和背景环境之间的微妙线索。COD 模型的详细综述超出了本文的范围；本文建议读者参考最近的顶级工作 [9,10]。

本文专注于分析端到端方法的鲁棒性。其他需要额外后处理的通用模型不在本文讨论范围内。例如，最近的 Segment Anything Model（SAM） [48] 在通用分割任务中表现出色；将这种方法扩展到 COD 任务需要额外的离线匹配，将 SAM 的所有候选掩码与目标 GT 框进行匹配，正如 [49] 所建议的那样。因此，本文的框架可能不会直接有益于原始的 SAM。然而，一个近期趋势是使用专门的提示在下游任务上微调大规模模型。在这种情况下，本文的框架有很大潜力提升提示感知的 SAM 变体。

2.3 伪装图像生成

虽然伪装图像生成受到的关注有限，但在这一领域仍有一些值得注意的作品。最早的方法之一是在 2010 年提出的，它依赖于人为设计的特征 [6]。Zhang 等人 [50] 最近提出了一种基于深度学习的方法来生成伪装图像。他们的方法采用迭代优化和注意力感知的伪装损失，以选择性地屏蔽前景物体的显著特征，同时一个显著性图确保这些特征仍然可识别。然而，缓慢的迭代优化过程限制了他们方法的实际应用。此外，背景图像到隐藏物体的风格转移通常会导致明显的外观不连续性，从而产生视觉上不自然的合成图像。为了克服这些限制，Li 等人 [51] 提出了一个位置无关的伪装图像生成网络。虽然这种方法在视觉质量上优于之前的方法 [50]，但在某些情况下可能无法保留所需的前景特征，或者使用显著性图使物体可识别。总的来说，现有方法都遵循相同的策略来产生伪装图像：它们使用两个图像分别代表前景图像和背景图像，然后尝试通过在合成图像中找到一个前景物体难以被检测到的位置，直接将前景图像与背景图像整合在一起。值得注意的是，这些方法中的大多数只合成新的 COD 图像，而不提供相关的掩码。因此，监督学习总是需要额外的标记。不同地，本文保留了伪装的真实掩码，并在训练有素的扩散模型的帮助下，将显著物体融入背景中，使本文能够从原始的 COD 掩码中受益，同时丰富场景中的更多模式。

算法 1 掩码生成:

将八个区域的索引按顺序放入列表 $candidates$ 中
打乱 $candidates$ 中的索引
for i in $candidates$ **do**
  if 区域 i 的面积高于 $RATIO_{MIN}$ **then**
    if 区域 i 的面积小于 $RATIO_{MAX}$ **then**
      选择从中心开始覆盖总区域面积为 $RATIO_{MASK}$ 的区域 $mask$
      break
    else
      选择从中心开始覆盖整个区域面积的 $RATIO_{MASK} \cdot RATIO_{MAX}$ 的区域 $mask$
      break
    end if
  else
    continue
  end if
end for
return $mask$

3 本文的 CamDiff

3.1 整体架构

为了评估现有 COD 方法在负样本（即含有显著物体的场景）上的有效性，本文建议在当前伪装数据集的基础上创建合成的显著物体。通常情况下，当使用 COD 数据集训练特定任务的模型时，它应该能够有效地检测伪装样本，同时具有鲁棒性，不检测合成的显著物体。因此，这种方法允许本文全面调查基于学习的 COD 方法是否能准确区分伪装和显著物体。

为了实现这一目标，本文提出了一个名为 CamDiff 的新生成网络，它是在现有的 COD 数据集之上构建的。由于这些数据集已经包含了伪装物体及其对应的伪装真值掩码，本文的目标是在背景中添加合成的显著物体。这样可以保持原始的伪装标签，同时利用它们，并引入具有对比特性的显著样本。

图. 2展示了本文提出方法的整体架构。本文从一个 COD 数据集开始，该数据集为本文提供了源图片及其对应的 GT。使用 GT，本文识别出具有最小覆盖区域的边界框，以防止 CamDiff 改变伪装图像。接下来，本文通过网格线将源图片分成九个区域，并使用边界框保留放置伪装物体的区域。只有八个区域可用于输入到 CamDiff。本文随机选择其中一个区域并从源图片中剪切出来，覆盖中心的特定比例（例如，在本文的实验中默认设置为 75%）的总面积。然后将被掩码的图像输入到生成网络中，CamDiff 在掩码区域内生成一个显著物体。最后，本文将选定的区域放回源图片的原始位置。通过这种方式，本文不仅可以保留伪装物体的 GT 标签，还可以添加具有对比性的合成显著样本。

为了生成显著物体，本文提出了一个基于 GAN 架构的生成框架。具体来说，本文使用广受认可的 LDM 作为

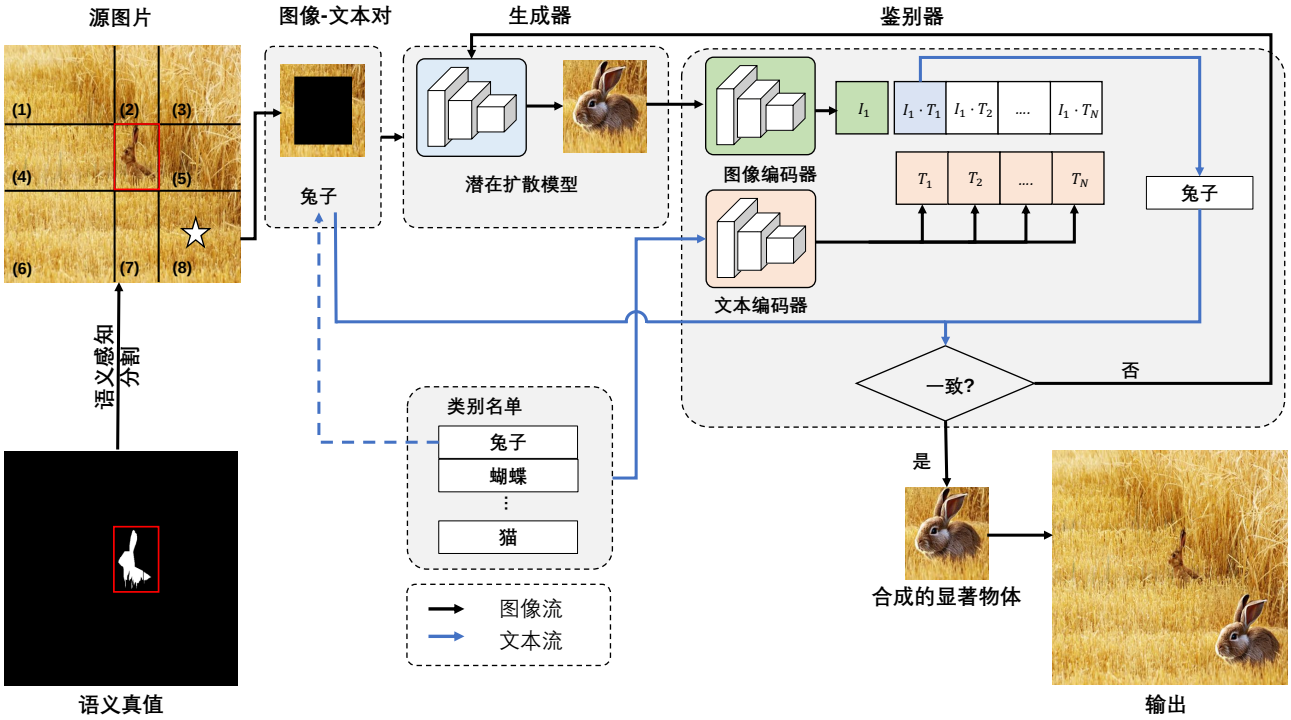

图 2 本文的模型包括一个生成器和一个判别器。该模型的输入是一对被掩码的图像和文本提示。只有在判别器判断合成对象与文本输入一致后，合成图像才能被输出并放回到源图片中。源图片中的白色星星意味着区域（8）被选为掩码区域。

生成器，CLIP 作为鉴别器。如图. 2所示，本文框架的输入是一个具有先前掩码区域的图像，以及描述目标物体的文本提示。然后将这个掩码区域和文本提示输入到生成器中。基于提示，LDM 模块在掩码区域上生成目标物体。填充的区域随后发送到鉴别器，以确定它是否与输入提示匹配。如果不匹配，生成器调整种子以生成新的显著物体。目标是训练生成网络，只在鉴别器预测与输入提示匹配的概率高时产生验证过的图像。

本文的框架将图像生成任务转化为修复任务，因此需要一个掩码来覆盖选定区域。掩码生成过程在算法 1 中进行了解释。

表 1 虚参数设置.

参数	值
$RATIO_{MIN}$	6.25%
$RATIO_{MAX}$	25%
$RATIO_{MASK}$	75%

掩码的设计是为了覆盖选定区域的一定比例，以避免在将合成物体与源图片融合时产生瑕疵。掩码区域与区域面积的比率被设置为一个常数，$RATIO_{MASK}$。选定区域的大小对于修复任务至关重要，因为它会影响生成的显著物体的质量。如果区域太小，LDM 可能会填充背景而非物体；而如果区域太大，则显著物体可能会比隐藏物体大很多，误导 COD 方法。因此，本文为源图片的总面积与区域面积之比设置了一个上限（$RATIO_{MAX}$）和下限（$RATIO_{MIN}$）。这些参数的值列在表. 1中。

3.2 潜在扩散模型

本文使用预先在大规模数据集上训练的 LDM [25] 作为生成器的基础模型。LDM 是一个两阶段的方法，包括一个自编码模型来学习图像的潜在表示和一个去噪扩散概率模型（DDPM）[26]。在第一阶段，自编码模型被训练来学习一个与图像空间在感知上等价的空间。编码器 $\mathcal{E}$ 将给定图像 $x \in \mathbb{R}^{H\times W\times 3}$ 编码为潜在表示 $z \in \mathbb{R}^{H\times W\times C}$，使得 $z = \mathcal{E}(x)$，同时解码器 $\mathcal{D}$ 从潜在表示重构估计图像 $\tilde{x}$，使得 $\tilde{x} = \mathcal{D}(\tilde{z})$ 且 $\tilde{x} \approx x$。在第二阶段，DDPM 被训练为基于随机高斯噪声输入 z_t 在预训练的潜在空间内生成潜在表示。LDM 的神经网络主干 $\epsilon_\theta(z_t, t)$ 实现为一个时间条件的 UNet，DDPM 在潜在空间上训练的目标简化为：

$$L_{DM} := \mathbb{E}_{\mathcal{E}(x),\epsilon\sim\mathcal{N}(0,1),t}\left[\|\epsilon - \epsilon_\theta(z_t, t)\|_2^2\right], \quad (1)$$

3.3 条件 LDM

为了控制图像合成，条件性 LDM 通过输入 y，诸如文本、语义图或其他图像到图像翻译任务，实现了一个条件性去噪自编码器 $\epsilon_\theta(z_t, y, t)$ [25]。本文提出的模型利用这种能力通过文本输入来控制图像合成。为了将 DDPM 转变为更灵活的条件性图像生成器，它们的基础 UNet 主干被增加了交叉注意力机制。来自 CLIP ViT-L/14 编码器的嵌入序列 $\tau_\theta(y) \in \mathcal{R}^{M\times d_\tau}$ 通过一个实现为如下的交叉注意力层与潜在特征图融合：

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}} \cdot V\right), \quad (2)$$

其中 $Q = W_Q^{(i)} \cdot \varphi_i(z_t), K = W_K^{(i)} \cdot \tau_\theta(y), V = W_V^{(i)} \cdot \tau_\theta(y)$，且 $\varphi_i(z_t)$ 是实现 ϵ_θ 的 UNet 的一个中间表示。$W_Q^{(i)}$、$W_K^{(i)}$ 和 $W_V^{(i)}$ 是可学习的投影矩阵。条件性 LDM 的目标从方程 1 转换为：

$$L_{CDM} := \mathbb{E}_{\mathcal{E}(x),y,\epsilon\sim\mathcal{N}(0,1),t}\left[\|\epsilon - \epsilon_\theta(z_t, t, \tau_\theta(y))\|_2^2\right], \quad (3)$$

3.4 CLIP 用于零次图像分类

为了提高基于文本输入生成物体的质量，需要使用一个判别器来评估生成物体与文本提示的一致性。然而，由于文本提示可以是任意类别，只能识别固定一组物体类别的传统分类器不适用于此任务。因此，CLIP 模型为此任务提供了更好的选择。

CLIP 模型包括一个图像编码器和一个文本编码器。图像编码器可以采用各种计算机视觉架构，包括五种不同大小的 ResNets 和三种视觉自注意力模型架构。与此同时，文本编码器是一个仅解码的自注意力模型，它使用掩蔽自注意力确保自注意力模型对序列中每个词元的表示仅依赖于它之前出现的词元。这种方法防止任何词元向后查看以更好地知其表示。两个编码器都经过预训练，以便在向量空间中对齐类似的文本和图像。这是通过获取图像-文本对并在向量空间中将它们的输出向量推得更近，同时分离非成对的向量来实现的。CLIP 模型在互联网上公开可用的 4 亿个文本-图像对的大型数据集上接受训练。由于 CLIP 模型的图像和文本编码器已经在多样化、未过滤和嘈杂的数据上进行了训练，在本文的应用中，将冻结 CLIP 参数以利用其泛化能力，使本文的方法能以零样本的方式进行。

在本文的模型中，通过图像编码器对带有合成物体的图像进行编码，而文本编码器则对所有类别的列表进行编码。如图. 2所示，由图像编码器输出的嵌入通过点积操作与文本编码器生成的每个类别的嵌入相结合。在所有类别中，生成的输出向量的最高值代表与图像最一致的类别的嵌入。如果最高值的类别与用户给出的语言提示一致，那么合成的图像就可以被放回原始图像并随后输出。

4 实验

4.1 实验设置

数据集。为了合成用于 COD 任务的多模式图像，本文选择了四个广泛使用的 COD 数据集：CAMO [52]、CHAM [53]、COD10K [1] 和 NC4K [54]。

值得注意的是，COD10K 数据集提供了以文件名形式的语义标签。因此，本文直接使用该标签作为文本提示。一些提示显示在图. 2中，其中列出了类别。然而，其他三个数据集并没有直接提供类别列表。由于它们包含常见的动物物种，如鸟类、猫、狗等，本文随机选择了一个来自 COD10K 标签列表的文本提示。

表 2 生成数据集与原始 COD 和 SOD 数据集的比较。类型 “orig.” 表示原始数据集，而类型 “new” 表示基于相应 COD 数据集合成的合成数据集。

	数据集	类型	Inception 分数 ↑
SOD	DUTSE-TE	orig.	71.63
	ECSSD	orig.	24.40
	XPIE (Salient)	orig.	96.79
	XPIE (Not Salient)	orig.	13.96
COD	CAMO	orig.	6.61
		new	9.90
	CHAM	orig.	4.38
		new	5.98
	COD10K	orig.	7.00
		new	14.85
	NC4K	orig.	7.00
		new	12.87

基准模型。 为了评估现有 COD 方法对显著物体和伪装物体的鲁棒性，本文选择了四种具有代表性和经典的 COD 方法作为本文的基准模型：SINet [1]、PFNet [2]、C2FNet [3] 和 ZoomNet [5]。值得注意的是，自从本文提交论文以来，出现了几种新的最先进模型，包括 FSPNet [10] 和 EVP 模型 [55]。然而，本文旨在探索检测伪装模式的新机制，因此全面测试所有模型超出了本文的范围。

评估指标。 为了评估合成图像的质量，本文采用了 inception 分数 [56]。在图像生成模型的背景下，更高的 inception 分数意味着生成的图像质量更好、更多样化。对于 COD 模型，本文使用了 4 个黄金评估指标：平均绝对误差（M）、最大 F 值（F_m）、S 值（S_m）和最大 E 值（E_m）。

实现细节。 本文在 Pytorch 框架中实现了 CamDiff，掩码生成相关的超参数在表. 1中有说明。整个学习过程

在 2080Ti GPU 上执行。本文遵循了传统的训练-测试划分 [1,5,7,47]，使用了来自 COD10K 和 CAMO 的 4,040 张训练集图像。

在这些训练样本中，本文用合成的多模式图像替换了 3,717 张图像。原始测试样本由来自 CAMO、CHAM、COD10K 和 NC4K 的 6,473 张图像组成。为了形成自己的 Diff-COD 测试集，本文用自己生成的图像替换了 5,395 张图像。虽然本文不能完全替换伪装数据集，因为某些图像包含特定物体，扩散模型可能无法使用预训练的权重很好地生成，但本文的成功率仍然很高。具体来说，超过 92% 的训练图像和 83% 的测试图像可以用额外的显著模式进行修改。这种高成功率证实了本文生成框架的有效性。需要注意的是，本文将图像和掩码调整为 512×512 以满足 LDM 的要求。

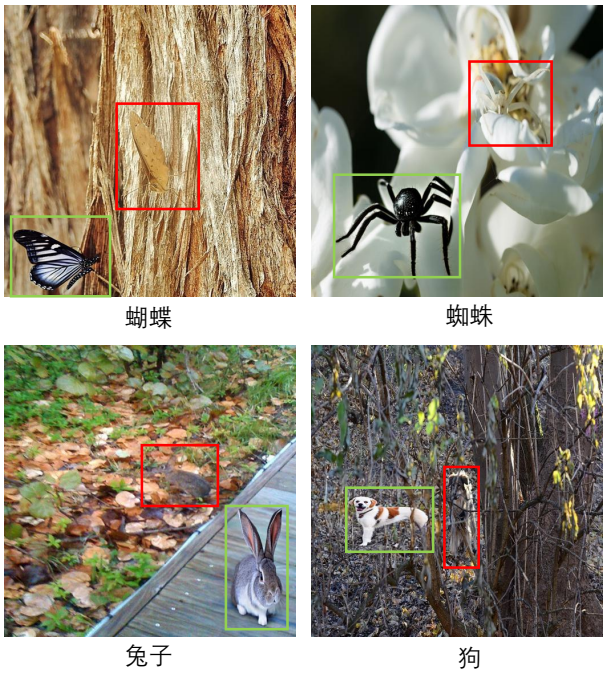

图 3 在用户研究中，本文的方案将在绿色框内呈现合成的对象，而图像中的原始对象则被封装在红色框内。研究结果表明，用户更有可能在绿色框中画圈，也就是说合成的对象比图像中的原始对象更突出且容易被发现。

4.2 合成图像的质量

Inception 分数。 为了证明 CamDiff 能够生成显著物体而非隐藏物体，本文选择 inception 分数作为评估指标，并在 SOD 数据集 [57–59]、COD 数据集 [1,52–54] 以及本文生成的包含多模式图像的数据集上进行评估。表. 2显示原始 SOD 数据集的 inception 分数高于原始 COD 数据集，这符合本文的预期。Inception 分数背后的理念是，一个合成得好的图像应该包含易于识别的物体，以供现成的识别系统使用。识别系统更有可能检测到显著物体而非伪装物体。因此，含有多模式的图像往往比含有伪装模式的图像具有更高的 inception 分数。

通过比较修改前后的 inception 分数，可以轻松评估本文框架的有效性。将 COD 数据集中的图像替换为多模式图像表明，所有 COD 数据集的 inception 分数都有所提高。这表明本文成功地在原始 COD 数据集之上融合了显著的图案。

用户调研。 本文还进行了用户调研，以评估合成图像的质量。其目标是根据提示（例如，在图. 3中的“蝴蝶”）从图像中找到目标对象。

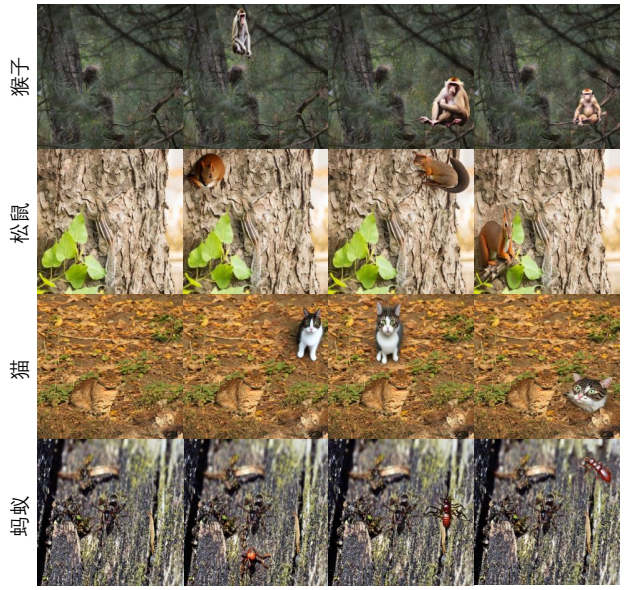

图 4 这是来自 CamDiff 的多个类别的合成图像示例。每张图片都被扩展，生成另外三张图片，生成图片展示了外观各异的同类型物体。

参与者得到本文合成图像的一个子集，被要求根据相应的标签圈出他们首先观测到的物体。用户选择的显著物体被认为是最突出的，因为它吸引了最多的人类注意力。

本文用户研究结果显示，超过 10 位参与者中，平均 98%的用户选择了合成的物体，即显著的物体。这表明合成的物体比图像中的原始物体更显著，更容易被检测到。

总的来说，增加的 inception 分数和用户研究的积极结果支持本文的说法，即本文的模型生成的是显著的物体而不是合成图像中隐藏的物体。此外，本文的模型还展示了其生成单一物体类型的样式和姿态多样性的强大能力。图. 4提供了各种类别的合成图像的例子，每张图片都可以扩展生成同一类别的三张不同图片。

表 3 这是预训练的 COD 模型在 Diff-COD 测试数据集和 COD 数据集上的定量结果。↑ (↓) 为数值越高（越低）越好。

Freezed		SINet [1]	PFNet [2]	C2FNet [3]	ZoomNet [5]
CAMO	$M\downarrow$	.099	.085	.079	.066
	$F_m\uparrow$	.762	793	.802	.832
	$S_m\uparrow$	.751	.782	.796	.819
	$E_m\uparrow$	.790	.845	.856	.881
Diff-CAMO	$M\downarrow$	.130	.122	.116	.136
	$F_m\uparrow$	.581	.626	.632	.557
	$S_m\uparrow$	.651	.686	.700	.664
	$E_m\uparrow$	.768	.792	.802	.790
CHAM	$M\downarrow$	.044	.033	.032	.023
	$F_m\uparrow$	.845	.859	.871	.883
	$S_m\uparrow$	.868	.882	.888	.900
	$E_m\uparrow$	.908	.927	.936	.944
Diff-CHAM	$M\downarrow$	.065	.065	.061	.088
	$F_m\uparrow$	.700	.795	.726	.596
	$S_m\uparrow$	.787	.708	.798	.726
	$E_m\uparrow$	.869	.865	.869	.850
COD10K	$M\downarrow$	.051	.040	.036	.029
	$F_m\uparrow$	.708	.747	.764	.799
	$S_m\uparrow$	.771	.800	.813	.836
	$E_m\uparrow$	.832	.880	.894	.887
Diff-COD10K	$M\downarrow$	.057	.054	.052	.064
	$F_m\uparrow$	.620	.644	.656	.585
	$S_m\uparrow$	.727	.751	.757	.729
	$E_m\uparrow$	.826	.832	.839	.841
NC4K	$M\downarrow$	.058	.053	.049	.044
	$F_m\uparrow$	.804	.820	.831	.845
	$S_m\uparrow$	.808	.829	.838	.851
	$E_m\uparrow$	.873	.891	.898	.896
Diff-NC4K	$M\downarrow$	.090	.084	.080	.076
	$F_m\uparrow$	.640	.664	.666	.631
	$S_m\uparrow$	.719	.744	.746	.739
	$E_m\uparrow$	.821	.830	.834	.841

4.3 定量比较

本文引入定量实验，并在其模型生成的合成样本上评估最新的 COD 方法。表. 3显示了预训练模型在原始和生成的测试样本上的结果；表. 4比较了在原始 COD 图像和本文生成的训练样本上训练的测试结果；表. 5体现了在 SOD 数据集测试的鲁棒性分析。

预训练权重。本文创建了一个新的 Diff-COD 数据集，以评估现有 COD 方法在包含显著和伪装物体的图像上的有效性。这个数据集包括了这两种类型的图像，本文在 Diff-COD 训练集上训练了 4 个最新的 COD 模型（SINet [1]，PFNet [2]，C2FNet [3] 和 ZoomNet [5]）。

接下来，本文评估了它们在 Diff-COD 测试集上的表现。需要注意的是，预训练的 LDM 模块只能输出分辨率为 512×512 的图像。这个分辨率适用于大多数以分辨率小于 352×352 训练的现有方法。

然而，目前最先进的方法，ZoomNet [5]，需要一个主分辨率为 384×384 和一个与主分辨率相比比例为 1.5 的更高分辨率（576×576），这大于 LDM 模型的容量。为了确保公平比较，本文使用主要比例为 288×288 重新训练了 ZoomNet。为了确保评估的公平性，本文在原始训练集和本文的新训练集上，使用相同的主分辨率 288×288 训练了 ZoomNet。

表. 3比较了每个模型在 Diff-COD 和原始 COD 数据集上使用其预训练权重的性能。结果表明，所有 COD 方法在 Diff-COD 数据集上的表现都明显下降。这是因为这些方法检测到了额外生成的显著物体，将它们认为是伪装物体，这表明它们对显著性的鲁棒性不足。因此，本文可以得出结论，Diff-COD 测试集是一个更具挑战性的基准，可以用作鲁棒性分析的额外工具。

在本文生成的数据集上训练。如前所述，本文的框架能够生成包含显著和伪装物体的新训练样本。通过仅使用伪装监督在本文的 Diff-COD 数据集上训练，网络应该学会区分这两个对立的概念，并对显著性更具鲁棒性。

表. 4展示了使用在原始 COD 训练集训练的预训练 COD 模型与在本文的 Diff-COD 训练集上重新训练的 COD 模型的结果。显然，在 Diff-COD 训练集上训练的模型在 Diff-COD 测试集上的表现比其对应的模型更好。

为了进一步确认本文的方法在增强 COD 模型对显著性的鲁棒性方面的有效性，本文在传统的显著性数据集上进行了实验，包括 DUTS-TE [57]、ECSSD [58] 和 XPIE [59]。如表. 5所示，当模型使用本文的 Diff-COD 数据集进行训练时，它们在显著性基准测试上的性能下降了。这个结果是可以预期的，因为在 SOD 数据集上的性能较差表明新训练的模型确实学会了伪装模式，但没有学会显著模式。因此，这些模型更能抵抗显著物体的影响。

4.4 定性比较

图. 5展示了在多模式图像上进行训练对 COD 模型性能的影响。此图分为三种情况，分别展示了不同伪装物体（鱼、蟹和青蛙）的结果。在每种情况下，虚线左侧显示了 COD 数据集的原始图像、合成的多模式图像和语义真值。虚线右侧第一行显示了四个预训练模型（SINet、PFNet、C2FNet 和 ZoomNet）在原始 COD 数据集上的测试结果。图示的第二行呈现了使用与第一行相同的权重在合成图像上测试的模型的结果。其中大多数模型检测到了显著物体，这是不希望被看到的，并且检测伪装物体的准确性下降。例如，与第一行中的掩码相比，SINet 失去了一些部分，ZoomNet 则忽略了伪装物体。这些结果表明 COD 方法对显著性缺乏鲁棒性。图示的第三行呈现了在本文的

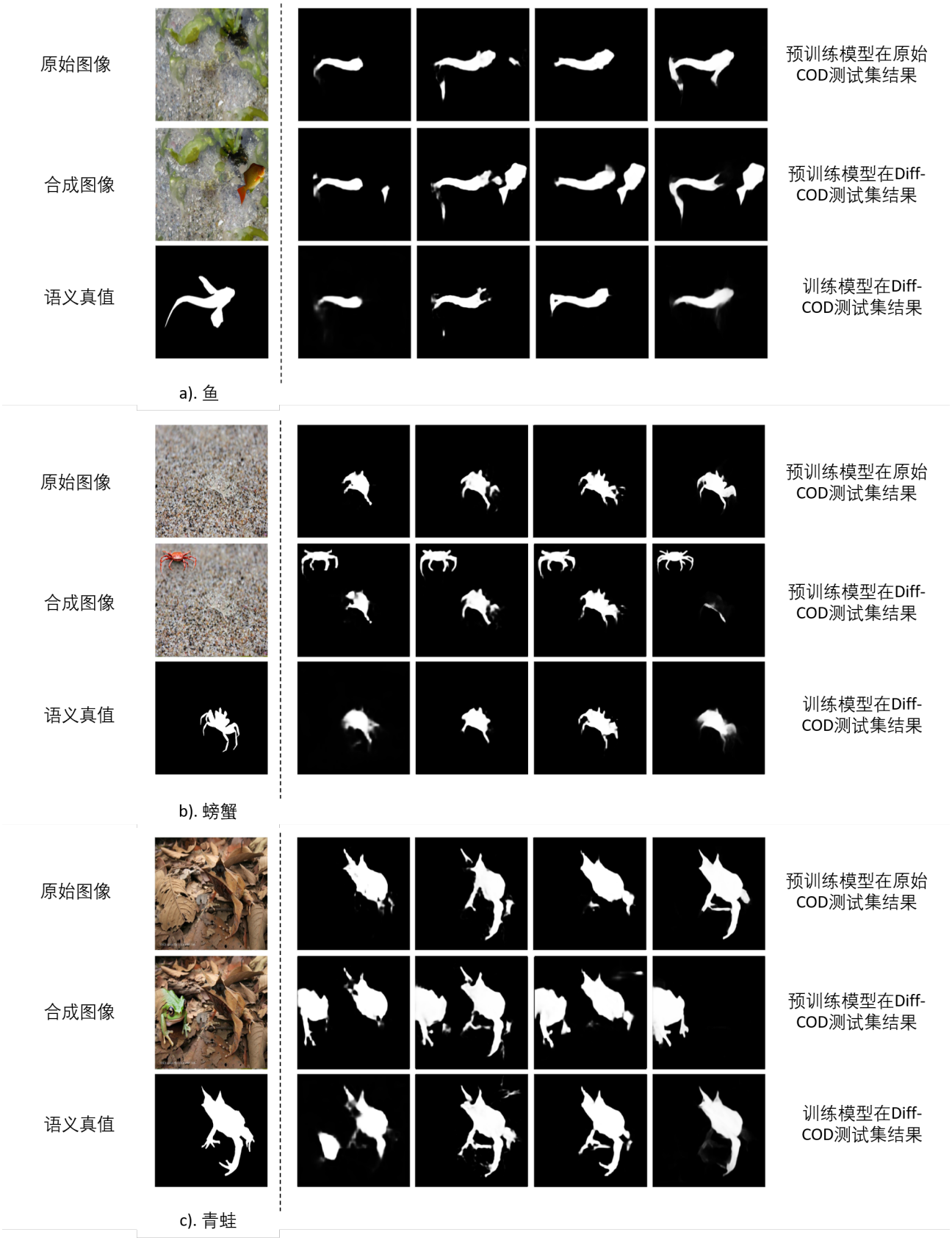

图 5 定性比较。本文对三种情况进行了定性比较：鱼、螃蟹和青蛙。本文通过比较前两行的结果，分析了将显著对象添加到伪装图像对预训练的 SINet、PFNet、C2FNet 和 ZoomNet 的影响。此外，本文通过将训练模型与预训练模型进行比较，评估了在 Diff-COD 测试集的表现。

Diff-COD 数据集上进行训练，然后在合成图像上进行测试的模型的结果。与第二行相比，模型对显著性的鲁棒性显著提高。然而，与第一行相比，ZoomNet 失去了伪装物体的一些部分。本文认为这可能是由于在训练集中添加了噪声，使拟合变得更加困难，但本文计划在未来的工作中评估原因。

总的来说，从图. 5可以得出结论，显著物体的存在会损害 COD 模型在检测伪装物体方面的性能。然而，将 COD 模型在多模式图像上进行训练可以增加其对显著物体影响的鲁棒性。

表 4 Diff-COD 测试数据集的定量结果。"Pre." 表示模型加载了官方发布的预训练权重。"Tr." 表示模型加载了在本文合成的训练集上训练的权重。

Dataset		SINet [1]		PFNet [2]		C2FNet [3]		ZoomNet [5]	
		Pre.	Tr.	Pre.	Tr.	Pre.	Tr.	Pre.	Tr.
Diff-CAMO	$M\downarrow$	.130	.094	.122	.087	.116	.078	.136	.092
	$F_m\uparrow$	.581	.769	.626	.787	.632	.800	.557	.758
	$S_m\uparrow$	.651	.753	.686	.773	.700	.789	.664	.773
	$E_m\uparrow$	.768	.802	.792	.828	.802	.848	.790	.803
Diff-CHAM	$M\downarrow$	.065	.036	.065	.033	.061	.030	.088	.058
	$F_m\uparrow$	.700	.864	.795	.858	.726	.870	.596	.764
	$S_m\uparrow$	.787	.884	.708	.880	.798	.888	.726	.816
	$E_m\uparrow$	.869	.931	.865	.933	.869	.949	.850	.845
Diff-COD10K	$M\downarrow$	.057	.047	.054	.041	.052	.038	.064	.053
	$F_m\uparrow$	.620	.708	.644	.735	.656	.748	.585	.691
	$S_m\uparrow$	.727	.773	.751	.794	.757	.801	.729	.770
	$E_m\uparrow$	.826	.849	.832	.874	.839	.887	.841	.805
Diff-NC4K	$M\downarrow$	.090	.060	.084	.052	.080	.047	.076	.069
	$F_m\uparrow$	.640	.807	.664	.821	.666	.834	.631	.789
	$S_m\uparrow$	.719	.811	.744	.830	.746	.840	.739	.814
	$E_m\uparrow$	.821	.866	.830	.894	.834	.905	.841	.847

表 5 原始 SOD 测试数据集的定量结果。"Pre." 表示模型加载了论文提供的预训练权重，而"Tr." 表示模型加载了在本文的合成训练集上训练的权重。

Dataset		SINet [1]		PFNet [2]		C2FNet [3]		ZoomNet [5]	
		Pre.	Tr.	Pre.	Tr.	Pre.	Tr.	Pre.	Tr.
DUTS-TE	$M\downarrow$	.065	.082	.064	.079	.065	.069	.080	.083
	$F_m\uparrow$	.820	.760	.808	.748	.807	.780	.715	.718
	$S_m\uparrow$	.806	.741	.806	.751	.802	.777	.772	.768
	$E_m\uparrow$	.846	.757	.845	.778	.832	.812	.840	.842
ECSSD	$M\downarrow$	.106	.135	.105	.130	.116	.115	.129	.134
	$F_m\uparrow$	.844	.784	.822	.762	.802	.790	.744	.751
	$S_m\uparrow$	.766	.692	.766	.703	.748	.734	.722	.715
	$E_m\uparrow$	.786	.688	.784	.702	.750	.740	.834	.841
XPIE-SAL	$M\downarrow$	.090	.119	.093	.115	.099	.101	.115	.123
	$F_m\uparrow$	.822	.763	.804	.739	.786	.762	.720	.703
	$S_m\uparrow$	.770	.691	.762	697	.749	.728	.723	.705
	$E_m\uparrow$	.805	.697	.792	.709	.768	.749	.820	.815

5 总结

总的来说，本文设计了 CamDiff 模型，这是一个生成显著对象并保留伪装场景中原始标签的框架，使它能够更轻松地在逼真图像中收集与组合对比模式，而无需涉及与学习和标记相关的额外成本。通过在 Diff-COD 测试集上进行实验，本文证明了当前的 COD 方法在负样本（例如，包含显著对象的场景）上缺乏鲁棒性。为了解决这一局限性，本文使用 CamDiff 创建了一个新颖的 Diff-COD 训练集。通过生成同时包含显著和伪装对象的多模式图像，CamDiff 为训练 COD 模型提供了更具挑战性和代表性的数据集，从而在现实世界的场景中改善了 COD 模型的性能，因为伪装对象可能由于显著对象的存在而更难以检测。通过这种方式，本文希望未来的 COD 方法能在区分显著对象和伪装对象的性能上得到提高。本文的实验结果表明，在这一数据集上训练现有的 COD 模型可以提高它们对显著对象的鲁棒性。总体而言，本文的工作为伪装领域提供了新的视角，并有助于这一新兴领域的发展。

未来工作。本文的目标是扩展本文的框架，考虑包含多个对象的原始图像，并为它们的生成留出空间。此外，虽然本文在实验中仅实施了多模式图像作为数据增强方法，但本文计划使用其他数据增强方法来评估结果，以更全面地分析多模式图像对这些模型性能和鲁棒性的影响。

参考文献

[1] Deng-Ping Fan, Ge-Peng Ji, Guolei Sun, Ming-Ming Cheng, Jianbing Shen, and Ling Shao. Camouflaged object detection. In *CVPR*, 2020.

[2] Haiyang Mei, Ge-Peng Ji, Ziqi Wei, Xin Yang, Xiaopeng Wei, and Deng-Ping Fan. Camouflaged object segmentation with distraction mining. In *CVPR*, 2021.

[3] Yujia Sun, Geng Chen, Tao Zhou, Yi Zhang, and Nian Liu. Context-aware cross-level fusion network for camouflaged object detection. In *IJCAI*, 2021.

[4] Qi Jia, Shuilian Yao, Yu Liu, Xin Fan, Risheng Liu, and Zhongxuan Luo. Segment, magnify and reiterate: Detecting camouflaged objects the hard way. In *CVPR*, 2022.

[5] Youwei Pang, Xiaoqi Zhao, Tian-Zhu Xiang, Lihe Zhang, and Huchuan Lu. Zoom in and out: A mixed-scale triplet network for camouflaged object detection. In *CVPR*, 2022.

[6] Hung-Kuo Chu, Wei-Hsin Hsu, Niloy J Mitra, Daniel Cohen-Or, Tien-Tsin Wong, and Tong-Yee Lee. Camouflage images. *ACM TOG*, 29(4), 2010.

[7] Deng-Ping Fan, Ge-Peng Ji, Ming-Ming Cheng, and Ling Shao. Concealed object detection. *IEEE TPAMI*, 44(10):6024–6042, 2022.

[8] Ruozhen He, Qihua Dong, Jiaying Lin, and Rynson WH Lau. Weakly-supervised camouflaged object detection with scribble annotations. In *AAAI*, 2023.

[9] Xiaobin Hu, Shuo Wang, Xuebin Qin, Hang Dai, Wenqi Ren, Ying Tai, Chengjie Wang, and Ling Shao. High-resolution iterative feedback network for camouflaged object detection. In *AAAI*, 2023.

[10] Zhou Huang, Hang Dai, Tian-Zhu Xiang, Shuo Wang, Huai-Xin Chen, Jie Qin, and Huan Xiong. Feature shrinkage pyramid for camouflaged object detection with transformers. In *CVPR*, 2023.

[11] Zongwei Wu, Danda Pani Paudel, Deng-Ping Fan, Jingjing Wang, Shuo Wang, Cédric Demonceaux, Radu Timofte, and Luc Van Gool. Source-free depth for object pop-out. *arXiv preprint arXiv:2212.05370*, 2023.

[12] Yijie Zhong, Bo Li, Lv Tang, Senyun Kuang, Shuang Wu, and Shouhong Ding. Detecting camouflaged object in frequency domain. In *CVPR*, 2022.

[13] Hemin Ali Qadir, Younghak Shin, Johannes Solhusvik, Jacob Bergsland, Lars Aabakken, and Ilangko Balasingham. Polyp detection and segmentation using mask r-cnn: Does a deeper feature extractor cnn always perform better? In *ISMICT*, 2019.

[14] Ferhat Ucar and Deniz Korkmaz. Covidiagnosis-net: Deep bayes-squeezenet based diagnosis of the coronavirus disease 2019 (covid-19) from x-ray images. *Medical hypotheses*, 140:109761, 2020.

[15] Bo Dong, Wenhai Wang, Deng-Ping Fan, Jinpeng Li, Huazhu Fu, and Ling Shao. Polyp-pvt: Polyp segmentation with pyramid vision transformers. *CAAI AIR*, 2023.

[16] Ricardo Pérez-de la Fuente, Xavier Delclòs, Enrique Peñalver, Mariela Speranza, Jacek Wierzchos, Carmen Ascaso, and Michael S Engel. Early evolution and ecology of camouflage in insects. *PNAS*, 109(52):21414–21419, 2012.

[17] Fen Fang, Liyuan Li, Ying Gu, Hongyuan Zhu, and Joo-Hwee Lim. A novel hybrid approach for crack detection. *PR*, 107:107474, 2020.

[18] Aixuan Li, Jing Zhang, Yunqiu Lv, Bowen Liu, Tong Zhang, and Yuchao Dai. Uncertainty-aware joint salient object and camouflaged object detection. In *CVPR*, 2021.

[19] Bo Dai and Dahua Lin. Contrastive learning for image captioning. In *NeurIPS*, 2017.

[20] Yuhao Zhang, Hang Jiang, Yasuhide Miura, Christopher D Manning, and Curtis P Langlotz. Contrastive learning of medical visual representations from paired images and text. In *MLHC*, 2022.

[21] Minguk Kang and Jaesik Park. Contragan: Contrastive learning for conditional image generation. In *NeurIPS*, 2020.

[22] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020.

[23] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning? In *NeurIPS*, 2020.

[24] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In *NeurIPS*, 2020.

[25] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

[26] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.

[27] Yihan Cao, Siyu Li, Yixin Liu, Zhiling Yan, Yutong Dai, Philip S Yu, and Lichao Sun. A comprehensive survey of ai-generated content (aigc): A history of generative ai from gan to chatgpt. *arXiv preprint arXiv:2303.04226*, 2023.

[28] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

[29] Siyuan Huang, Zan Wang, Puhao Li, Baoxiong Jia, Tengyu Liu, Yixin Zhu, Wei Liang, and Song-Chun Zhu. Diffusion-based generation, optimization, and planning in 3d scenes. *arXiv preprint arXiv:2301.06015*, 2023.

[30] Chieh Hubert Lin, Hsin-Ying Lee, Willi Menapace, Menglei Chai, Aliaksandr Siarohin, Ming-Hsuan Yang, and Sergey Tulyakov. Infinicity: Infinite-scale city synthesis. *arXiv preprint arXiv:2301.09637*, 2023.

[31] Dat Huynh, Jason Kuen, Zhe Lin, Jiuxiang Gu, and Ehsan Elhamifar. Open-vocabulary instance segmentation via robust cross-modal pseudo-labeling. In *CVPR*, 2022.

[32] Yasser Benigmim, Subhankar Roy, Slim Essid, Vicky Kalogeiton, and Stéphane Lathuilière. One-shot unsupervised domain adaptation with personalized diffusion models. *arXiv preprint arXiv:2303.18080*, 2023.

[33] Mingyuan Zhang, Xinying Guo, Liang Pan, Zhongang Cai, Fangzhou Hong, Huirong Li, Lei Yang, and Ziwei Liu. Remodiffuse: Retrieval-augmented motion diffusion model. *arXiv preprint arXiv:2304.01116*, 2023.

[34] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

[35] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In *NeurIPS*, 2021.

[36] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Commun. ACM*, 63(11):139–144, 2020.

[37] Charlie Nash, Jacob Menick, Sander Dieleman, and Peter W Battaglia. Generating images with sparse representations. In *ICML*, 2021.

[38] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *ICLR*, 2018.

[39] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. In *ICLR*, 2021.

[40] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *ICLR*, 2022.

[41] Linqi Zhou, Yilun Du, and Jiajun Wu. 3d shape generation and completion through point-voxel diffusion. In *ICCV*, 2021.

[42] Shitong Luo and Wei Hu. Diffusion probabilistic models for 3d point cloud generation. In *CVPR*, 2021.

[43] Brian L Trippe, Jason Yim, Doug Tischer, Tamara Broderick, David Baker, Regina Barzilay, and Tommi Jaakkola. Diffusion probabilistic modeling of protein backbones in 3d for the motif-scaffolding problem. *ICLR*, 2023.

[44] Ruihan Yang, Prakhar Srivastava, and Stephan Mandt. Diffusion probabilistic modeling for video generation. *arXiv preprint arXiv:2203.09481*, 2022.

[45] Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. In *SIGGRAPH*, 2022.

[46] Fan Yang, Qiang Zhai, Xin Li, Rui Huang, Ao Luo, Hong Cheng, and Deng-Ping Fan. Uncertainty-guided transformer reasoning for camouflaged object detection. In *ICCV*, 2021.

[47] Mingchen Zhuge, Xiankai Lu, Yiyou Guo, Zhihua Cai, and Shuhan Chen. Cubenet: X-shape connection for camouflaged object detection. *PR*, 127:108644, 2022.

[48] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023.

[49] Ge-Peng Ji, Deng-Ping Fan, Peng Xu, Ming-Ming Cheng, Bowen Zhou, and Luc Van Gool. Sam struggles in concealed scenes–empirical study on "segment anything". *arXiv preprint arXiv:2304.06022*, 2023.

[50] Qing Zhang, Gelin Yin, Yongwei Nie, and Wei-Shi Zheng. Deep camouflage images. In *AAAI*, 2020.

[51] Yangyang Li, Wei Zhai, Yang Cao, and Zheng-jun Zha. Location-free camouflage generation network. *IEEE TMM*, 2022.