基于初始交互点注意力的交互图像分割

林铮　张钊　陈林卓　程明明　卢少平 *
计算机学院, 南开大学
http://mmcheng.net/fclick/

Abstract

在交互式图像分割任务中，用户最初单击一个点来分割目标对象的主体，然后在错误预测的区域上迭代地提供更多的点，以实现更为精确的分割。现有的方法将所有交互点同等对待，忽略了初始交互点和其余的交互点之间的区别。在本文中，我们展示了初始交互点对于提供目标对象的位置和主体信息的关键作用。为了更好地利用初始交互点，提出了深度神经网络框架：初始交互点注意力网络（FCA-Net）。在这个网络中，交互式分割结果可以得到改进，其优点如下：聚焦不变能力，位置指导能力，容错能力。同时本文提出了一种基于交互点的损失函数和一种结构完整性策略来提高分割效果。可视化的分割结果和在五个数据集的充分实验证明了初始交互点的重要性和我们的 FCA-Net 的优越性。

1. 介绍

交互式图像分割的目的是用最少的用户交互输入来分割出感兴趣的目标物体。它对于许多应用都有实际作用，如图像编辑 [7, 10, 28] 和医疗图像分析 [45]。近年来，随着数据驱动的深度学习技术的普及，在某些领域，对与像素级别注释的需求急剧增加，，如显著性物体检测 [4, 8, 13, 22]，语义分割 [33]，实例分割 [20, 34]，伪装物体检测 [14]，和图像/视频处理 [16, 27, 47]。迫在眉睫得需要高效的交互式分割技术以减轻标注成本。因此，越来越多的研究者正在这一领域进行广泛的探索。

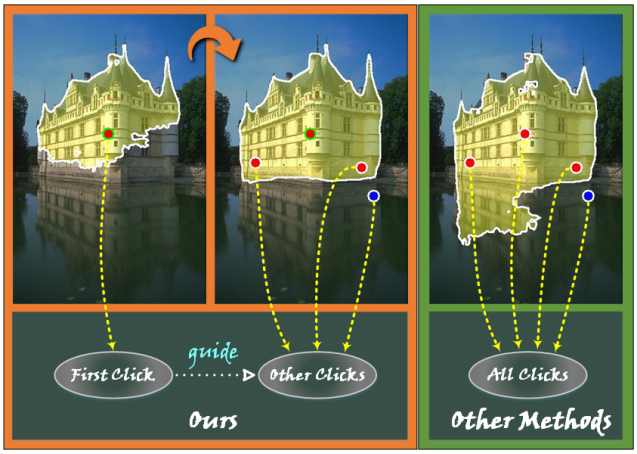


图 1. 初始交互点在我们的方法中的关键作用。我们利用初始交互点作为分割锚点来指导其他交互点进行精确的分割，而传统的基于点的交互式分割方法对所有点进行了不加区分的处理。

许多形式都交互方式都被实践探讨过，如包围盒 [9, 42]，涂鸦 [2, 5]，和点 [24, 29, 30, 36, 46]。

绘制包围盒作为交互方式是一种应用广泛、方便的方法。然而，在大多数情况下，用户通常需要对分割结果进行进一步的修正，该交互难以满足此种需求。因此，更实用的方法是基于交互点或涂鸦，通过迭代标记错误区域，进一步提高分割结果。与涂鸦相比，点对用户的负担更小，因为它不需要拖动的过程。基于点的方法的典型交互工作流程，见图. 1，如下所示：用户首先在目标对象上提供一个初始前景点。根据初始分割结果，用户进一步在图像上提供一个正样本点或一个负样本点，对分割结

*本文是 CVPR2020 论文的中译版。

果进行迭代细化，直到满足用户的要求。

许多传统的以及基于深度学习的方法已经在这个方向上探索了许久。对于大多数现有的工作，他们不加区别地使用所有的交互点来生成最终的预测结果。然而，我们观察到并非所有交互点都具有相同的分割效果。我们使用其中一种交互式分割方法 [29]收集了 2000 多幅真实人类交互的统计数据，如表. 1所示。我们发现初始点在交互分割中起着重要的作用。首先，初始交互点的性能改善非常明显，并且初始交互点通常靠近目标对象的中心。结合之前描述的工作流程，通过直观的观察分析可以得到，初始交互点十分重要，可以作为目标对象的位置指示和全局信息引导。从图. 1可以看出，只要点击一次，初始分割结果已经相当不错。相反，其他交互点的作用主要是在初始交互点的基础上实现更细致的分割。因此，初始交互点更有利于获取对象的整体信息，而其他点则侧重于细节修复。基于以上分析，我们推测特别处理初始交互点将有利于交互式分割。

在本文中，我们首先将这两种点区别对待。我们提出了一个初始交互点注意力网络（FCA-Net），在该网络中构造了一个简单的辅助分支来进一步验证。在我们的网络中，我们使用初始交互点作为侧边输入来监督全局分割。利用初始交互点作为锚点来进行交互式分割，可以更好地引导目标对象的位置和主体信息。预测结果将集中在初始交互点周围的区域，可以得到更好的结果。对于网络训练，我们提出了一种改进的损失函数，它考虑了用户提供的所有交互点，并将分割重心集中在交互点周围的这些区域。最后我们提出了一种新的后处理策略，可以有效地去除一些小的预测错误的区域，并保持分割对象的结构完整性。我们在 GrabCut [42]、Berkeley [38]、PASCAL VOC [12]、DAVIS [41]、MSCOCO [31]数据集上进行了全面的实验，取得了领先的性能。对比实验的结果和分析证明了我们提出的方法的独特性和有效性。

我们的贡献可以总结如下：

- ▷ 这是第一个展示初始交互点关键作用的工作。我们还提出了一个 FCA-Net 网络，它包含了一个简单而有效的模块来利用初始交互点的引导信息。
- ▷ 提出了一种考虑用户交互点的损失和一种结构完整性策略，有助于交互式分割任务的实现。
- ▷ 五个数据集上的实验结果证明了初始交互点的重要性和我们的 FCA-Net、交互点损失函数和结构完整性策略的有效性。

No.	1	2	3	4	5	6	7	8	9	10
PI	.751	.076	.045	.027	.020	.017	.015	.015	.009	.010
CD	.769	.312	.243	.207	.201	.211	.189	.188	.178	.186

表 1. 用户交互统计数据. PI: 加入不同交互点后的性能提升。CD: 描述交互点靠近物体中心的程度（只统计前景交互点）。CD 越高代表交互点越靠近中心。具体计算细节在节. 3.5中提及。

2. 相关工作

在早期，大多数传统的交互式分割方法主要利用手工提取的特征。一些研究方法如 [39]非常关注边界性质。在 [5]之后，基于图模型的方法变得更加流行，即将交互式分割任务建模为图割优化问题，并用著名的最小割/最大流算法求解。其中基于图割的经典方法 GrabCut 是在 [42]中提出的。该算法以高斯混合模型为颜色模型，边界框为输入，简化了分割过程。Kimet al. [25]改进了 [17]中提出的随机游走算法。Kimet al. [26]还引入了一种新的高阶公式，并附加了软标签一致性约束。Gulshanet al. [18]和 Baiet al. [3]都将测地线距离应用于交互式图像分割的优化中。Baiet al. [2]提供了一种容错方法，允许用户进行一些错误的交互。这些基于低层特征的方法不能适应复杂多变场景下的目标分割。

神经网络具有感知复杂的全局特征和局部特征的能力。随着深度学习的普及，越来越多的研究尝试将神经网络应用于交互式分割。近年来，Xuet al. [46]首次提出了一个基于 CNN 的模型，并给出了一些有效的点采样策略。然后，Liewet al. [30]提出了一个基于正负交互点对来提取区域信息的 RIS-Net，用于局部细化。Songet al. [43]应用强化学习，使计算机产生更多潜在的交互点。Scunaet al. [1]利用循环

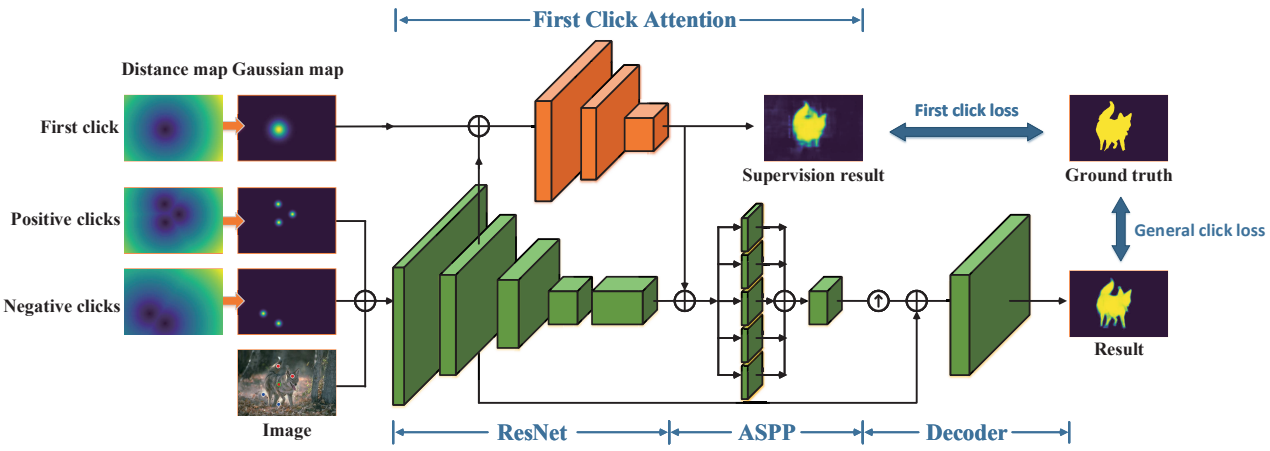


图 2. FCA-Net 的整体结果。绿色部分显示了基础网络，包括主干网络、空洞卷积池化金字塔模块和解码器模块。橙色部分显示初始交互点注意力模块。符号 “⊕” 和 “↑” 分别表示拼接和上采样操作。在节. 3.1中详细描述了细节。

神经网络对图像进行精确分割，得到由多个点组成的多边形。然后，Linget al. [32]最近用图卷积网络对上述基于多边形的方法进行了改进。Liet al. [29]利用神经网络提供和选择一个更精确分割目标来解决交互式分割中的歧义情况。Maninisetal. [37]为交互分割提供了一种新颖的交互方式。Mahadevan 等人提出了一种有效的迭代训练策略。Huet al. [23]提出了一种用于交互式分割的双流融合网络。Jang 和 Kim-cite 提出了一种反向传播的改进方案，以迫使每个交互点都有正确的分割结果。Majumder 和 Yao [36]利用交互点生成特殊的引导图，作为神经网络的输入，如超像素等。所有这些方法都有一个共同点：它们不加区别地对待神经网络中的所有交互点。但是，我们发现并提出了初始交互点的特殊性，并将其作为我们网络结构的一个特殊指导。

3. 方法

本节包括五个部分。节. 3.1介绍了我们提出的 FCA-Net 网络，它特殊处理初始交互点。在节. 3.2描述了所提出的基于交互点损失的计算过程，以帮助我们的交互式分割网络获得更好的性能。在节. 3.3阐述了用于后处理的结构完整性策略。在节. 3.4，我们通过一些例子的比较分析了采用初始交互点的好处。最后，我们将在节. 3.5中展示交互点模拟策略的实现细节和训练设置。

3.1. 网络结构

FCA-Net 的结构展示在图. 2。为了更好地说明初始交互点的有效性，我们没有对交互式分割网络的结构做太多的改变。取而代之的是，一个简单的附加模块称为初始交互点注意力模块被添加到基本的分割网络中。因此，FCA-Net 可以分为基本分割网络和初始交互点注意力模块。

基础分割网络 同 [24, 29, 30, 36, 46]，我们采用最常见的 FCN 结构的网络，它的结构与 DeepLab v3+ [6]相似。如图. 2所示，它包含三个模块：主干网络，空洞卷积池化金字塔模块（ASPP）和解码器模块。

我们采用 ResNet101 [21]作为特征提取网络。我们将后四层的特征定义为 $\{F_1, F_2, F_3, F_4\}$. 为了捕捉交互分割中的多尺度物体，我们同样在 ResNet101 的最后一层采用了空洞卷积而不是采用步长 2。因此，主干网络的输出步长为 16。主干网络的输入是 RGB 彩色图像与两个基于交互点的高斯点图的拼接。高斯点图是通过欧式距离计算而来，如图. 2所示。我们实验中的高斯点半径设置为 10。

如 图. 2中的空洞卷积模块所示，输入是拼接的特征 $(F_4 \oplus F_{FCA})$，其中 $\oplus$ 代表拼接操作，F_{FCA} 代

表初始交互点注意力模块的输出。拼接后的特征被输入进 4 个尺寸为 1、6、12、18 的空洞卷积层，以及一个全局池化层。接着这 5 路的输出的特征拼接后属于到一个额外的卷积层。如图. 2中所示的解码器模块，它将底层特征 F_1 和空洞卷积模块的输出特征作为输入并使用卷积层来回的最终的预测结果。为了对预测结果进行监督，我们设计了一个基于交互点的损失函数去替换传统的二值交叉熵损失函数。我们将这个称作全局交互点损失，具体细节在 节. 3.2中有详细的描述。

初始交互点注意力模块： 为了更好地利用初始交互点信息，我们在基础分割网络上设计了一个简单的模块。它使用底层特征 F_1 和在初始点周围的高斯点图 M_f 作为输入。这些拼接的特征 $(\mathrm{F}_1 \oplus \mathrm{M}_f)$ 被输入进 6 个 3×3 的卷积层。在第一和第四层我们使用步长为 2 来降低分辨率。前三层的通道数是 256，后三层的通道数是 512。因此，输出的特征 $\mathrm{F}_{\mathrm{FCA}}$ 拥有 512 通道，它将在 ASPP 前被融合进基础分割网络。除此之外，我们使用一个初始交互点损失来监督 $\mathrm{F}_{\mathrm{FCA}}$，它会重点关注初始交互点周围的像素，具体细节在 节. 3.2中有详细的描述。

为了更好地说明初始交互点注意力的效果，在图. 3，我们分别用 FCA（c-d）和不加 FCA（b）对模型的预测图进行可视化。注意的是，在这三个测试（b-d）中，这些正点的坐标完全一致。如图. 3（b）所示，在没有 FCA 的情况下，这两个前景交互点具有相同的重要性。通过引入 FCA（c-d），模型的注意力转移。在测试（c）和（d）中，用户标记前景交互点的顺序不同。我们可以看到无论它在哪里，第一次点击会引起更多的关注，作为分割的锚点，其余点则对细节修复起辅助作用。与将所有交互点同等处理相比，FCA 的引入使模型更符合节. 1中讨论的用户实际交互行为。

3.2. 交互点损失

为了在下文中更好地进行说明，我们在这里定义了一些符号和操作。所有的像素被表示作 $\mathcal{G}$。我们使用 $\mathcal{G}_p$ 和 $\mathcal{G}_n$ 去表示根据真值标注图得到的前景像素点和背景像素点。$\mathcal{A}$ 表示所有交互点。$\mathcal{A}_p$ 和 $\mathcal{A}_n$ 分别表示所有的前景和背景交互点。我们使用 $d(p_1,p_2)$ 来表示点 p_1 和点 p_2 的欧氏距离。我们使用 $\phi(p,\mathcal{S})$ 来表示从点 p 到另外一个区域 $\mathcal{S}$ 的最短距离，它的定义如下：

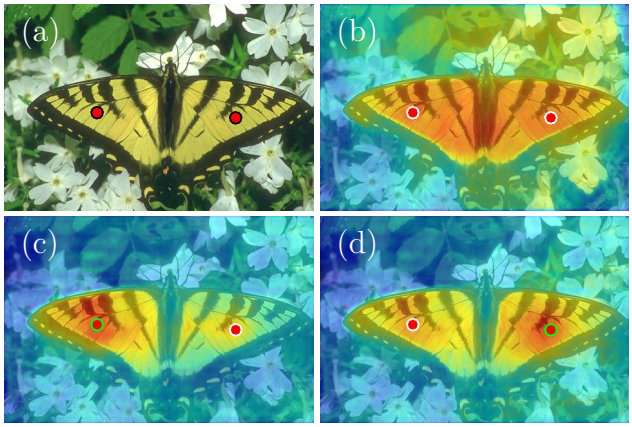


图 3. 初始交互点注意力的可视化。(b) 是有 FCA 模块的预测图；(c) 和 (d) 是初始交互点模块作用在不同位置的预测图。

$$\phi\left(p,\mathcal{S}\right)=\min_{\forall p_s\in\mathcal{S}} d\left(p,p_s\right). \tag{1}$$

在二值分割任务中，通常采用二值交叉熵(BCE)作为损失函数来监督神经网络。该损失函数有利于关注全局分割质量。而对于交互式分割任务，我们更希望看到用户交互同样能够起到监督作用。最好在这些交互点处及其周围能得到更准确的结果。因此，我们设计了一个基于用户交互点的损失函数来帮助我们的 FCA-Net 获得更好的性能。

该交互点损失可以被视作一种特殊的二值交叉熵损失。传统的二值交叉熵损失函数可以表示如下：

$$\ell\left(p\right)=-\left(y_p\log\left(x_p\right)+\left(1-y_p\right)\log\left(1-x_p\right)\right), \tag{2}$$

其中 x_p 代表点 p 在预测图中的概率值，而 y_p 代表点 p 在真值图中的标签（0 或 1）。

首先，我们定义了一个函数 ψ 来表示点 p 和一个交互点集 $\mathcal{S}$ (如 $\mathcal{A}_p$ 和 $\mathcal{A}_n$) 的距离，其计算如下：

$$\psi\left(p,\mathcal{S}\right)=1-\frac{\min\left(\phi\left(p,\mathcal{S}\right),\tau\right)}{\tau}, \tag{3}$$

其中 τ 是每个交互点的影响范围。

对于监督最后预测图的损失函数，我们提出了一个考虑了所有交互点的全局交互点损失 ($\mathcal{L}_g$)，它的计算如下：

$$\mathcal{L}_g = \frac{1}{N} \sum_{p \in \mathcal{G}} \left(\widehat{w}_p \cdot \ell \left(p \right) \right). \tag{4}$$

N 是像素个数。公式. (4) 中的权重可以被表示如下：

$$\widehat{w}_p = \begin{cases} \alpha + \psi \left(p, \mathcal{A}_p \right) \left(\beta - \alpha \right), & y_p = 1 \\ \alpha + \psi \left(p, \mathcal{A}_n \right) \left(\beta - \alpha \right), & y_p = 0 \end{cases}, \tag{5}$$

其中 α 和 β 被用来调整损失的范围。

对于监督 FCA 模块输出的损失函数，我们使用了一个初始交互点损失 ($\mathcal{L}_f$)，它会集中关注初始交互点周围的区域。它的计算如下：

$$\mathcal{L}_f = \frac{1}{N} \sum_{p \in \mathcal{G}} \left(\widetilde{w}_p \cdot \ell \left(p \right) \right). \tag{6}$$

公式. (6) 其中的权重可以表示如下：

$$\widetilde{w}_p = \alpha + \psi \left(p, \{ a_f \} \right) \left(\beta - \alpha \right) y_p, \tag{7}$$

其中 a_f 代表 $\mathcal{A}_p$ 中的初始交互点。

在我们的实验中，我们把 τ 设置成 100, α 设置成 0.8, β 设置成 2.0。

3.3. 结构完整性策略

通过实验，我们发现神经网络的预测很可能包含一些错误分割的分散区域。在大多数情况下，在交互式分割任务中，人们更希望得到保持结构完整性的对象结果。因此，我们提出了一种基于交互点的保持分割结构完整性的简单策略。

通常，我们以 0.5 作为阈值，从神经网络的输出中得到最终的二值化预测。让 $\mathcal{P}$ 表示这些预测为前景的点，我们将根据交互点对这些预测区域进行后处理，得到新的 $\mathcal{P}'$，计算如下：

$$\mathcal{P}' = \{ p \in \mathcal{P} | \exists_{a \in \mathcal{A}_p} \sigma(p, a) = 1 \}, \tag{8}$$

其中当存在一条点 p_1 到点 p_2 的八连通路径时，$\sigma(p_1, p_2) = 1$。该结构完整性策略可以在大多数情况下起到一定效果。它的效果可以在表. 2看到.

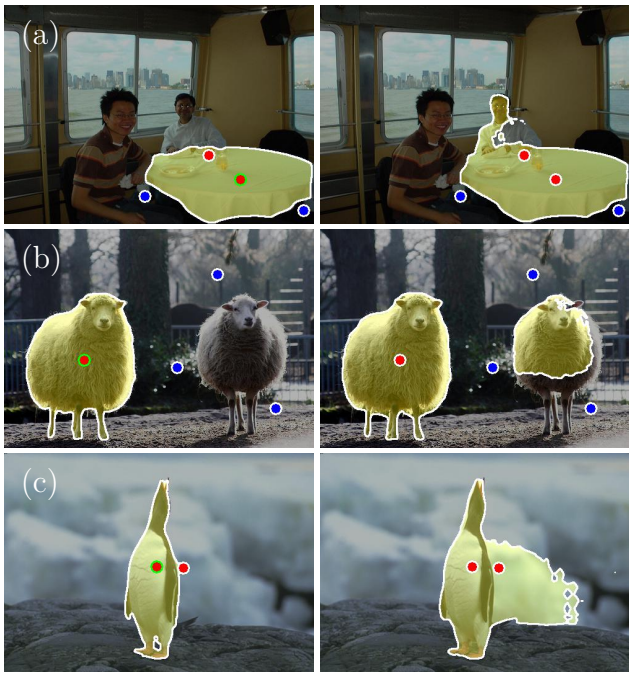


图 4. 初始交互点注意里的好处。左右列分别显示了是否带有 FCA 模块的预测结果。

3.4. 优点分析

初始交互点注意力真的能提高分割结果吗？在本节中，我们将通过比较图. 4中的一些可视化结果来说明加初始交互点的一些好处。

聚焦不变能力： 我们知道，在大多数方法中，所有的正负交互点都是同等处理的。它们将所有交互点作为输入，以生成最终结果。初始交互点外的其他点经常被用来修复局部细节，并且可能靠近目标对象的边界。如果神经网络将这些点同等对待，往往会导致错误的分割。例如，在 图. 4中，我们分割白色的桌布的桌子，初始交互点靠近桌子的中心，另一个正点用于修复桌子边缘附近的分割错误。如果没有初始交互点的引导，神经网络会错误地分割出图像中的人，因为它对每个交互点都是平等对待的的。在我们初始交互点的帮助下，错误的分割将会减少。

位置指导能力： 显然，初始交互点指导了目标对象的位置。如果场景中有多个物体，那么利用初始交互点可以减少局部区域的错误分割。例如，在图. 4 (b) 中，我们要分割左边的羊。我们点击右羊周围

的三个负点。由于没有对全局位置信息的准确理解，神经网络可能会误认为在这些负点包围的区域中有一个目标对象，这可能会导致一些错误，例如对右边羊的错误预测。有了初始交互点的帮助，预测就会集中在初始交互点的位置上，以得到更好的结果。

容错能力： 在交互式分割过程中，不可避免地会出现一些点击错误，特别是在目标边缘或背景与前景相似的区域。例如，在图. 4 (c)，我们要分割企鹅。右侧靠近目标对象边界的正点意外地落入背景区域。我们可以看到，如果我们不使用初始交互点注意力，这可能会导致严重的分割错误，如图. 4 (c) 的右侧所示。而在初始交互点的引导下，这些误差影响将大大减小。

3.5. 实现细节

在本节中，我们将展示有关训练的一些细节。这些分割数据集不存在用户交互的标注，我们如大多数论文中所做的一样，采取一些策略来模拟各种生成的交互点，包括全局交互点和初始交互点。同时，我们还将介绍我们的训练设置。

全局交互点模拟： 对于大多数交互点，我们使用与 [46] 相似的策略。正负交互点数量分别在 [1, 10] 和 [0, 10] 之间。

对于正交互点，它们来自前景，至少远离物体边界 P_1 像素并且它们之间至少互相远离 P_2 像素。我们定义 $\mathcal{A}^*$ 为先前已经标记过的交互点，一个新的正交互点来自一个候选点集 $\mathcal{C}_p$，其可以表示如下：

$$\mathcal{C}_p = \{p \in \mathcal{G}_p | \phi(p, \mathcal{G}_n) > P_1, \phi(p, \mathcal{A}^*) > P_2\}. \quad (9)$$

对于负交互点，它们来自背景，至少远离物体边界 $N_1 \sim N_2$ 像素并且它们之间至少互相远离 N_3 像素。一个新的负交互点来自一个候选点集 $\mathcal{C}_n$，其可以表示如下：

$$\mathcal{C}_n = \{p \in \mathcal{G}_n | \phi(p, \mathcal{G}_p) \in (N_1, N_2), \phi(p, \mathcal{A}^*) > N_3\}. \quad (10)$$

在我们的实验中，我们选择 P_1 来自集合 $\{5, 10, 15, 20\}$，P_2 来自集合 $\{7, 10, 20\}$，N_1 来自集合 $\{15, 40, 60\}$，N_2 来自集合 $\{80\}$，N_3 来自集合 $\{10, 15, 25\}$.

初始交互点模拟： 初始交互点总是来自目标物体内部，它通常靠近物体中心。因此我们使用 $\mathcal{E}(p)$ (在表. 1中被称作 CD) 来表示点 p 距离物体中心的程度，它的计算如下：

$$\mathcal{E}(p) = \frac{\phi(p, \mathcal{G}_n)}{\max_{\forall p_0 \in \mathcal{G}_p} \phi(p_0, \mathcal{G}_n)}. \quad (11)$$

$\mathcal{E}(p)$ 越接近 1 代表初始交互点越靠近物体中心。在我们的实验中，我们选择裁剪图像中 $\mathcal{E}(p)$ 为 1 的点作为初始交互点，并且将它的高斯半径设置成一般交互点的的三倍。

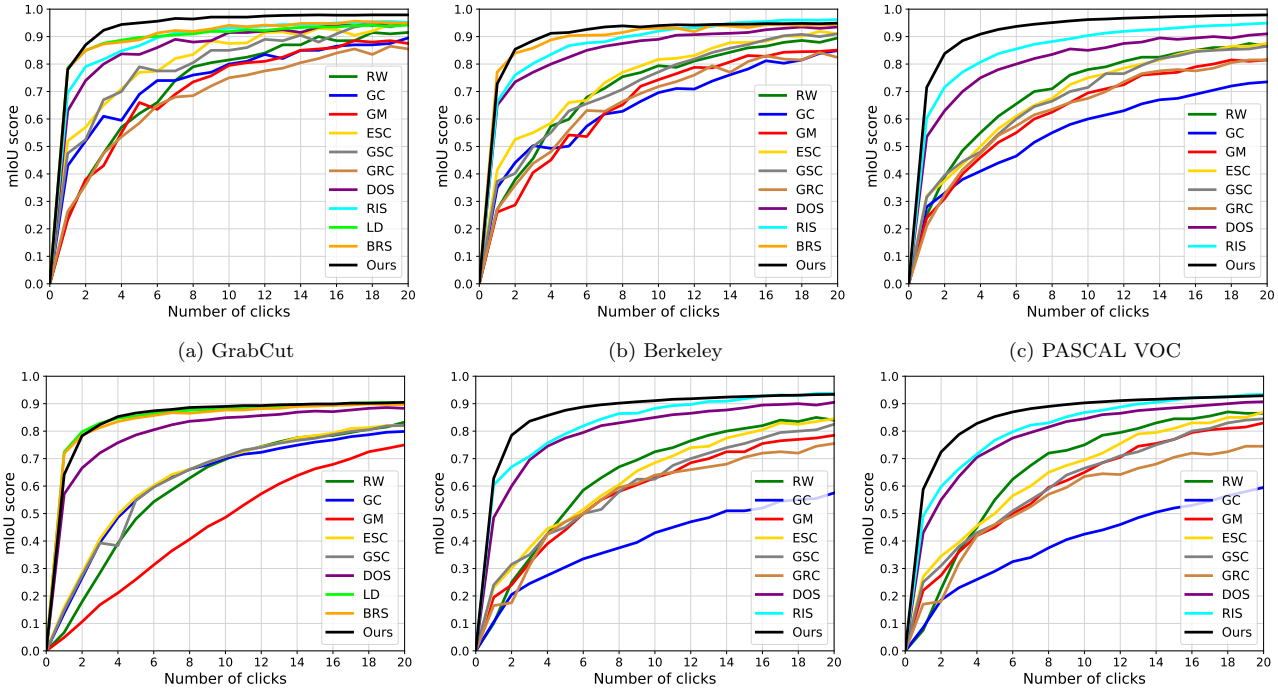
训练设置： 我们使用扩展数据集 (PASCAL VOC [12]+SBD [19]) 的除去了 PASCAL VOC 验证集的 10582 个训练图像对 FCA-Net 进行训练，。实际上，我们可以得到 25832 个实例级的图像和相应的真值图用于训练。输入图像的使其较小的一侧固定为 512 像素，并按等比例调整大小。然后，我们随机裁剪 512×512 像素，保证裁剪后的图像至少包含对象的一部分。我们采用相同的迭代训练策略 [35, 36]进行交互点模拟。我们用在 ImageNet [11]上预训练的 ResNet101 为主干网络。我们将批大小设置为 8。我们设定 ResNet 的初始学习率为 0.007，其他部分为 0.07，并采用 0.9 动量的随机梯度下降进行优化。最后采用多项式学习率衰减法对 30 个周期进行学习，对另外尾部 3 个周期采用恒定学习速率。所有的实验都是用 PyTorch [40]框架实现的，并在单个 NVIDIA Titan XP GPU 上运行。

4. 实验部分

4.1. 评测细节

数据集： 我们采用了以下广泛使用的数据集进行评测：

- GrabCut [42]: 该数据集包含 50 幅图像，在大多数交互式分割方法中使用。大多数图像的前景和背景有明显的差异。



(a) GrabCut (b) Berkeley (c) PASCAL VOC

(d) DAVIS (e) MSCOCO (seen) (f) MSCOCO (unseen)

图 5. FCA-Net 和其它 10 个方法在 5 个数据集 6 个子集上的交互点 vs. 平均交并比 (NoC-mIoU) 曲线。

- Berkeley [38]: . 该数据集包含 96 幅图像上的 100 个对象。由于前景和背景的相似性，在这个数据集中有些图像很难进行分割。

- PASCAL VOC [12]: 我们使用这个数据集中的验证集进行评测，该数据集包含 1449 个图像和 3427 个实例。因此，我们使用这些实例级对象掩码进行验证。这些对象与用于训练的数据在语义上是一致的。

- MSCOCO [31]: 该数据集包含 80 个类别的对象。我们将此数据集分为 MSCOCO（seen）和 MSCOCO（unseen），并按照 [30, 46]中的方法为每个类别抽取 10 张图像进行评估。

- DAVIS [41]: 该数据集用于视频对象分割。它包含 50 个视频，它们的对象掩码是高质量的。我们采用和 [24]同样的 10% 的帧作为评估。

评测指标： 同 [23, 24, 29, 30, 35, 36, 46]一样, 我们使用平均交并比（mIoU）作为评测指标。我们还使用一个机器用户来模拟评测中的交互点击。首先，初始交互点无疑是目标对象的一个前景点，我们将得到一个基于初始交互点的预测图。然后在最大误差区域的中心取下一模拟交互点。我们绘制了 mIoU 和点击次数的曲线，以比较每种方法在固定交互下的性能。我们采用平均交互点数（mNoC）作为评估指标，它反映了在数据集的每个样本上获得特定 IoU 阈值的平均交互效果。对于每个数据集，IoU 阈值的选择是不同的每个样本的默认最大交互次数限制为 20 次。以上设置与之前的工作一致。

推理时间： 我们在 Intel i7-8700K 3.70GHz CPU 和单个 NVIDIA Titan XP GPU 上测试推理时间。每次点击 512 ×512 的图像大约需要 0.07 秒，这个速度足够满足实时交互的需要。

4.2. 性能对比

我们将我们的结果与其它现存的方法进行了比较，包括 graph cut (GC) [5], growcut (GRC) [44], random walk (RW) [17], geodesic matting (GM) [3], Euclidean star convexity (ESC) [18], geodesic

Method	GrabCut @90%	Berkeley @90%	PASCAL VOC @85%	DAVIS @90%	MSCOCO (seen)@85%	MSCOCO (unseen)@85%
GC [5]$_{ICCV01}$	11.10	14.33	15.06	17.41	18.67	17.80
GRC [44]$_{POG05}$	16.74	18.25	14.56	N/A	17.40	17.34
RW [17]$_{PAMI06}$	12.30	14.02	11.37	18.31	13.91	11.53
GM [3]$_{IJCV09}$	12.44	15.96	14.75	19.50	17.32	14.86
ESC [18]$_{CVPR10}$	8.52	12.11	11.79	17.70	13.90	11.63
GSC [18]$_{CVPR10}$	8.38	12.57	11.73	17.52	14.37	12.45
DOS [46]$_{CVPR16}$	6.04	8.65	6.88	12.58	8.31	7.82
RIS [30]$_{ICCV17}$	5.00	6.03	5.12	N/A	5.98	6.44
LD [29]$_{CVPR18}$	4.79	N/A	N/A	9.57	N/A	N/A
BRS [24]$_{CVPR19}$	3.60	5.08	N/A	8.24	N/A	N/A
CMG [36]$_{CVPR19}$	3.58	5.60	3.62	N/A	5.40	6.10
FCA-Net	2.24	4.23	2.98	8.05	4.49	5.54
FCA-Net (SIS)	2.14	4.19	2.96	7.90	4.45	5.33
FCA-Net*	2.16	3.92	2.79	7.64	4.34	5.36
FCA-Net* (SIS)	2.08	3.92	2.69	7.57	4.08	5.01

表 2. 个数据集 6 个子集上的平均交互点数（mNoC）对比。SIS 表示使用了结构完整性策略进行后处理。FCA-Net* 表示该模型使用 Res2Net [15]作为主干网络。

star convexity (GSC) [18], deep object selection (DOS) [46], regional image segmentation (RIS) [30], latent diversity based segmentation (LD) [29], back-propagating refinement scheme (BRS) [24], and content-aware multi-level guidance (CMG) [36]. 部分数据来自 [24, 29, 30, 46]。

图. 5 说明了每个方法在不同交互点下的 mIoU。这些曲线的绘制是原始的，没有使用结构完整性策略。我们可以看出，在大多数情况下，我们的方法在第一点后的曲线优于其他方法，这符合我们的期望。以初始交互点作为主体引导，神经网络的预测会包含较少的错误区域。因此，FCA-Net 可以产生更准确的结果。

表. 2显示五个数据集六部分的的 mNoC 指标。我们的 FCA-Net 在五个数据集中达到了最好的水平。在采用结构完整性策略对结果进行后处理后，性能将进一步提高。我们在网络架构上并没有做太多的改变，只是设置了一个简单的初始交互点注意力模块。然而，效果的提高是显著的，这也间接反映了初始交互点的独特性。

#	FCANet	PASCAL	Berkeley
1	BS	4.21	5.74
2	BS + FCA	3.66	5.22
3	BS + FCA + CL	3.33	4.94
4	BS + FCA + CL + Iter	2.98	4.23
5	BS2 + FCA + CL + Iter	2.79	3.92

表 3. 方法的消融实验。BS: 基础网络; BS2: 使用 Res2Net 的基础网络; FCA: 初始交互点注意力模块; CL: 交互点损失; Iter: 迭代训练。

4.3. 消融实验

为了进一步验证我们的贡献，我们对 PASCAL-VOC 和 Berkeley 的验证集进行了消融实验。我们

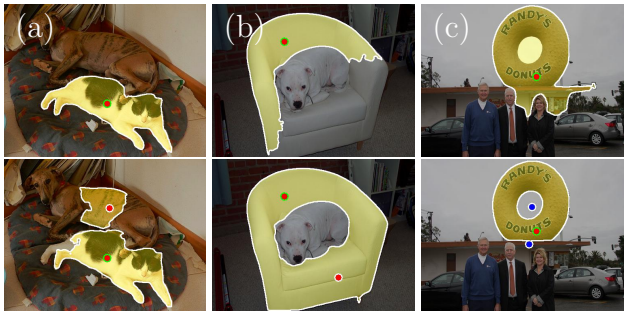


图 6. FCA-Net 网络可能存在的局限性。带有绿色圆环的点表示初始交互点。

以基础网络为基线（No.1），并逐步添加上本文中提到的策略（No.2-5）。平均交互点数（mNoC）的消融结果如表. 3所示。与基线比较，加入 FCA 模块后，mNoC 降低了 0.55 和 0.52，性能得到了显著提高。这一改进符合我们的期望，通过引入 FCA，可以更有效地利用第一次点击的引导信息。比较 3 号和 2 号实验，我们看到，本文提出的交互点损失带来了改善效果。我们也采用相同的迭代训练策略 [35, 36]，在一定程度上提高了最终模型的效果。由于所出的 FCA-Net 只是一个简单的实现来探索初始交互点的关键作用，我们没有对广泛使用的框架进行过多的修改；因此，在实践中，我们可以通过更换更有效的主干网络或更复杂的设计来获得更好的结果。例如，在第 5 号实验中，我们用 Res2Net [15]代替 ResNet 作为主干网络，进一步提高了性能。最后，我们使用所提出的结构完整性策略对结果进行后处理，并在表. 2中展示它的效果。

4.4. 局限性分析

在本节中，我们讨论了 FCA-Net 在某些特殊情况下可能存在的局限性。如图. 6 (a) 所示，由于初始交互点提供了很强的位置先验证，我们的 FCA-Net 不擅长同时分割图像中的多个实例。不过，在实际应用中，通过初始交互点为每个实例对象添加标注，可以减轻此限制。在图. 6（b-c）中，我们展示了两个有趣的场景，其中，由于结构或遮挡，用户可能无法单击这些实例的中心。在这种情况下，初始交互点定位可能偏离中心。有时会导致分割结果的不理想，用户需要添加更多的点来进行修复。

5. 结论

在本文中，我们探讨并论证了初始交互点对于交互式分割的重要性。我们提出了一个 FCA-Net，它在基本分割网络上增加了一个简单的模块，将更多的注意力转移到初始交互点上。除此之外，我们还提出了一种有效的基于交互点的损失函数和一种结构完整性策略来提升性能。5 个数据集上的最好性能表明了初始交互点的重要性和我们方法的优越性。

参考文献

[1] David Acuna, Huan Ling, Amlan Kar, and Sanja Fidler. Efficient interactive annotation of segmentation datasets with polygon-rnn++. In IEEE Conf. Comput. Vis. Pattern Recog., pages 859–868, 2018. 2

[2] Junjie Bai and Xiaodong Wu. Error-tolerant scribbles based interactive image segmentation. In IEEE Conf. Comput. Vis. Pattern Recog., pages 392–399, 2014. 1, 2

[3] Xue Bai and Guillermo Sapiro. Geodesic matting: A framework for fast interactive image and video segmentation and matting. Int. J. Comput. Vis., 82(2):113–132, 2009. 2, 7, 8

[4] Ali Borji, Ming-Ming Cheng, Qibin Hou, Huaizu Jiang, and Jia Li. Salient object detection: A survey. Computational Visual Media, pages 1–34, 2014. 1

[5] Yuri Y Boykov and M-P Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in nd images. In Int. Conf. Comput. Vis., volume 1, pages 105–112. IEEE, 2001. 1, 2, 7, 8

[6] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In Eur. Conf. Comput. Vis., pages 801–818, 2018. 3

[7] Ming-Ming Cheng, Qi-Bin Hou, Song-Hai Zhang, and Paul L. Rosin. Intelligent visual media processing: When graphics meets vision. Journal of Computer Science and Technology, 32(1):110–121, 2017. 1

[8] Ming-Ming Cheng, Niloy J. Mitra, Xiaolei Huang, Philip H. S. Torr, and Shi-Min Hu. Global contrast based salient region detection. IEEE TPAMI, 37(3):569–582, 2015. 1

[9] Ming-Ming Cheng, Victor A Prisacariu, Shuai Zheng, Philip H. S. Torr, and Carsten Rother. Densecut: Densely connected crfs for realtime grabcut. Comput. Graph. Forum, 34(7):193–201, 2015. 1

[10] Ming-Ming Cheng, Fang-Lue Zhang, Niloy J. Mitra, Xiaolei Huang, and Shi-Min Hu. Repfinder: Finding approximately repeated scene elements for image editing. ACM Trans. Graph., 29(4):83:1–8, 2010. 1

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In IEEE Conf. Comput. Vis. Pattern Recog., pages 248–255. Ieee, 2009. 6

[12] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. Int. J. Comput. Vis., 88(2):303–338, 2010. 2, 6, 7

[13] Deng-Ping Fan, Ming-Ming Cheng, Jiang-Jiang Liu, Shang-Hua Gao, Qibin Hou, and Ali Borji. Salient objects in clutter: Bringing salient object detection to the foreground. In Eur. Conf. Comput. Vis., pages 186–202, 2018. 1

[14] Deng-Ping Fan, Ge-Peng Ji, Guolei Sun, Ming-Ming Cheng, Jianbing Shen, and Ling Shao. Camouflaged object detection. In IEEE Conf. Comput. Vis. Pattern Recog., 2020. 1

[15] Shang-Hua Gao, Ming-Ming Cheng, Kai Zhao, Xin-Yu Zhang, Ming-Hsuan Yang, and Philip Torr. Res2net: A new multi-scale backbone architecture. IEEE Trans. Pattern Anal. Mach. Intell., 2020. 8, 9

[16] Shiming Ge, Xin Jin, Qiting Ye, Zhao Luo, and Qiang Li. Image editing by object-aware optimal boundary searching and mixed-domain composition. Computational Visual Media, 4(1):71–82, 2018. 1

[17] Leo Grady. Random walks for image segmentation. IEEE Trans. Pattern Anal. Mach. Intell., 28(11):1768–1783, 2006. 2, 7, 8

[18] Varun Gulshan, Carsten Rother, Antonio Criminisi, Andrew Blake, and Andrew Zisserman. Geodesic star convexity for interactive image segmentation. In IEEE Conf. Comput. Vis. Pattern Recog., pages 3129–3136. IEEE, 2010. 2, 7, 8

[19] Bharath Hariharan, Pablo Arbeláez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In Int. Conf. Comput. Vis., pages 991–998. IEEE, 2011. 6

[20] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In Int. Conf. Comput. Vis., pages 2961–2969, 2017. 1

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In IEEE Conf. Comput. Vis. Pattern Recog., pages 770–778, 2016. 3

[22] Qibin Hou, Ming-Ming Cheng, Xiaowei Hu, Ali Borji, Zhuowen Tu, and Philip Torr. Deeply supervised salient object detection with short connections. IEEE

Trans. Pattern Anal. Mach. Intell., 41(4):815–828, 2019. 1

[23] Yang Hu, Andrea Soltoggio, Russell Lock, and Steve Carter. A fully convolutional two-stream fusion network for interactive image segmentation. Neural Networks, 109:31–42, 2019. 3, 7

[24] Won-Dong Jang and Chang-Su Kim. Interactive image segmentation via backpropagating refinement scheme. In IEEE Conf. Comput. Vis. Pattern Recog., pages 5297–5306, 2019. 1, 3, 7, 8

[25] Tae Hoon Kim, Kyoung Mu Lee, and Sang Uk Lee. Generative image segmentation using random walks with restart. In Eur. Conf. Comput. Vis., pages 264–275. Springer, 2008. 2

[26] Tae Hoon Kim, Kyoung Mu Lee, and Sang Uk Lee. Nonparametric higher-order learning for interactive segmentation. In IEEE Conf. Comput. Vis. Pattern Recog., pages 3201–3208. IEEE, 2010. 2

[27] Thuc Trinh Le, Andrés Almansa, Yann Gousseau, and Simon Masnou. Object removal from complex videos using a few annotations. Computational Visual Media, 5(3):267–291, 2019. 1

[28] Yin Li, Jian Sun, Chi-Keung Tang, and Heung-Yeung Shum. Lazy snapping. ACM Trans. Graph., 23(3):303–308, 2004. 1

[29] Zhuwen Li, Qifeng Chen, and Vladlen Koltun. Interactive image segmentation with latent diversity. In IEEE Conf. Comput. Vis. Pattern Recog., pages 577–585, 2018. 1, 2, 3, 7, 8

[30] JunHao Liew, Yunchao Wei, Wei Xiong, Sim-Heng Ong, and Jiashi Feng. Regional interactive image segmentation networks. In Int. Conf. Comput. Vis., pages 2746–2754. IEEE, 2017. 1, 2, 3, 7, 8

[31] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In Eur. Conf. Comput. Vis., pages 740–755. Springer, 2014. 2, 7

[32] Huan Ling, Jun Gao, Amlan Kar, Wenzheng Chen, and Sanja Fidler. Fast interactive object annotation with curve-gcn. In IEEE Conf. Comput. Vis. Pattern Recog., pages 5257–5266, 2019. 3

[33] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In IEEE Conf. Comput. Vis. Pattern Recog., pages 3431–3440, 2015. 1

[34] Yifan Lu, Jiaming Lu, Songhai Zhang, and Peter Hall. Traffic signal detection and classification in street views using an attention model. Computational Visual Media, 4(3):253–266, 2018. 1

[35] Sabarinath Mahadevan, Paul Voigtlaender, and Bastian Leibe. Iteratively trained interactive segmentation. In Brit. Mach. Vis. Conf., 2018. 6, 7, 9

[36] Soumajit Majumder and Angela Yao. Content-aware multi-level guidance for interactive instance segmentation. In IEEE Conf. Comput. Vis. Pattern Recog., pages 11602–11611, 2019. 1, 3, 6, 7, 8, 9

[37] Kevis-Kokitsi Maninis, Sergi Caelles, Jordi Pont-Tuset, and Luc Van Gool. Deep extreme cut: From extreme points to object segmentation. In IEEE Conf. Comput. Vis. Pattern Recog., pages 616–625, 2018. 3

[38] Kevin McGuinness and Noel E O ' connor. A comparative evaluation of interactive segmentation algorithms. Pattern Recognition, 43(2):434–444, 2010. 2, 7

[39] Eric N Mortensen and William A Barrett. Intelligent scissors for image composition. In ACM SIGGRAPH, pages 191–198, 1995. 2

[40] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In NeurIPS, pages 8024–8035, 2019. 6

[41] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In IEEE Conf. Comput. Vis. Pattern Recog., pages 724–732, 2016. 2, 7

[42] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. ACM Trans. Graph., 23(3):309–314, 2004. 1, 2, 6

[43] Gwangmo Song, Heesoo Myeong, and Kyoung Mu Lee. Seednet: Automatic seed generation with deep reinforcement learning for robust interactive segmentation. In IEEE Conf. Comput. Vis. Pattern Recog., pages 1760–1768, 2018. 2

[44] Vladimir Vezhnevets and Vadim Konouchine. Growcut: Interactive multi-label nd image segmentation by cellular automata. proc. of Graphicon, 1(4):150–156, 2005. 7, 8

[45] Guotai Wang, Maria A Zuluaga, Wenqi Li, Rosalind Pratt, Premal A Patel, Michael Aertsen, Tom Doel, Anna L David, Jan Deprest, Sébastien Ourselin, et al. Deepigeos: a deep interactive geodesic framework for medical image segmentation. IEEE Trans. Pattern Anal. Mach. Intell., 41(7):1559–1572, 2018. 1

[46] Ning Xu, Brian Price, Scott Cohen, Jimei Yang, and Thomas S Huang. Deep interactive object selection. In IEEE Conf. Comput. Vis. Pattern Recog., pages 373–381, 2016. 1, 2, 3, 6, 7, 8

[47] Shuyang Zhang, Runze Liang, and Miao Wang. ShadowGAN: Shadow synthesis for virtual objects with conditional adversarial networks. Computational Visual Media, 5(1):105–115, 2019. 1