关于 45s 内将单张图像重建为 3D 网格论文的复现

摘要

单张图像的三维重建是一个重要但具有挑战性的任务，需要对自然世界有广泛的了解。许多现有方法通过在二维扩散模型的指导下优化神经辐射场来解决这个问题，但存在优化时间长、三维不一致性结果和几何性能差的问题。在本文复现的论文中，作者提出了一种新方法，它以任意物体的单张图像作为输入，并在一次前向传递中生成完整的 360 度三维纹理网格。给定一张单张图像，该方法首先使用一种视角条件的二维扩散模型（Zero123）为输入视角生成多视图图像，然后旨在将它们提升到三维空间。由于传统的重建方法在多视图预测上存在一致性问题，该方法基于基于 SDF 的通用神经表面重建方法构建了我们的三维重建模块，并提出了几种关键的训练策略，以实现 360 度网格的重建。在不需要昂贵的优化的情况下，我们的方法在比现有方法更短的时间内重建三维形状。此外，该方法更有利于几何性能，生成更一致的三维结果，并更贴近输入图像。

关键词：深度学习；单张图像三维重建；二维扩散模型；神经辐射场

1 引言

单图像三维重建是计算机视觉领域一个长期存在的问题，其任务是从单个 2D 图像中重建物体的三维模型。这一问题对于许多应用至关重要，包括机器人物体操控和导航、三维内容创建，以及增强现实/虚拟现实等领域 [1]。这个问题十分具有挑战性，因为它不仅需要重建可见部分，还需要通过虚构来重建不可见区域。因此，由于来自单个图像的信息不足，这个问题通常是不适定的，并且对应于多个合理的解决方案。另一方面，人类能够根据自身对三维世界的大量知识，熟练地推断看不见的三维内容。为了赋予智能体这种能力，一些现有方法 [2,3] 通过在三维形状数据集 [4] 上训练三维生成网络来利用类别特定的先验知识。然而，这些方法通常泛化性较差，无法推广到未见过的类别，并且它们的重建质量受制于公共三维数据集的有限大小。

在这项工作中，作者追求一个通用的解决方案，将任何对象的图像，无论其类别如何，转化为高质量的三维纹理网格。为了实现这一目标，作者提出了一种新颖的方法，可以有效地利用 2D 扩散模型学到的强先验知识进行三维重建。与常见的基于优化的范式不同，该方法利用 2D 先验模型进行三维建模。作者方法的核心是将 2D 扩散模型与基于代价体积的三维重建技术相结合，通过前向传递在不进行每场景优化的情况下实现从单个图像重建高质量的 360° 纹理网格。

与三维数据相比，2D 图像更容易获得和扩展。最近的 2D 生成模型（例如 Stable Diffusion [5]）以及视觉语言模型（例如 CLIP [6]）通过在互联网规模的图像数据集上进行预训练，

取得了显著的进展。由于它们学到了广泛的视觉概念，并具有关于我们三维世界的强先验知识，将它们与三维任务结合起来是自然而然的。因此，最近出现了一系列研究，例如 DreamField [7]、DreamFusion [8] 和 Magic3D [9]，利用 2D 扩散模型或视觉语言模型来辅助三维生成任务。它们的共同范式是使用可微分渲染、CLIP 模型或 2D 扩散模型的指导来执行每个形状的优化。尽管已经探索了许多其他 3D 表示方法，但在优化过程中神经场是最常用的表示方法。

尽管这些基于优化的方法在文本到三维 [7, 8] 和图像到三维任务 [10] 上取得了令人印象深刻的结果，但它们面临一些共同的问题：(a) 耗时长。每个形状的优化通常涉及成千上万次完整图像体积渲染和先验模型推断，导致每个形状通常需要数十分钟。(b) 内存密集。由于 2D 先验模型需要完整图像，当图像分辨率升高时，体积渲染可能会占用大量内存。(c) 三维上的不一致性。由于二维先验模型每次迭代只看到一个视图，并试图使每个视图看起来像输入图像，它们经常生成三维不一致的形状（例如，具有两个面或者 Janus 问题 [8, 10]）。(d) 几何质量差。许多方法在体积渲染中使用密度场作为表示。它们通常可以生成良好的 RGB 渲染，但提取高质量的网格往往较为困难。

在本文中，作者提出了一种新颖的方法，与常见的基于优化的范式不同，该方法利用 2D 先验模型进行三维建模。作者方法的核心是将 2D 扩散模型与基于代价体积的三维重建技术相结合，通过前向传递在不进行每场景优化的情况下实现从单个图像重建高质量的 360° 纹理网格。具体而言，作者利用最近的 2D 扩散模型 Zero123 [11]，该模型在 Stable Diffusion [5] 上进行了微调，以预测给定相机变换的输入图像的新视图。作者利用它生成输入单个图像的多视图预测，从而可以利用多视图三维重建技术获取三维网格。从合成的多视图预测进行重建涉及两个挑战：(a) 多视图预测内在的不完全一致性，这可能导致基于优化的方法（如 NeRF [12]）严重失败。(b) 输入图像的相机姿态是必需的但未知的。为了解决这些问题，作者使用基于代价体积的神经表面重建方法 SparseNeus [13] 构建了他们的重建模块，该方法是 MVSNeRF [14] 的一种变体。此外，作者引入了一系列关键的训练策略，使得能够从固有不一致的多视图预测中重建 360 度的网格。作者还提出了一个高程估计模块，该模块估计了 Zero123 规范坐标系中输入形状的高程，用于计算重建模块所需的相机姿态。

通过整合多视图合成、仰角估计和三维重建三个模块，作者的方法可以以前向传递的方式从单个图像中重建任何对象的三维网格。在无需昂贵的优化的情况下，作者的方法在更短的时间内（例如，仅需 45 秒）重建 3D 形状。由于使用了 SDF 表示，作者的方法更有利于生成更好的几何形状，并且由于相机条件的多视图预测，生成更一致的 3D 网格。

2 相关工作

2.1 具有二维先验的模型引导三维生成

最近，2D 生成模型（例如 Stable Diffusion [5]）以及视觉语言模型（例如 CLIP [6]）通过在互联网规模的图像数据集上进行预训练，学到了广泛的视觉概念。它们具有关于我们三维世界的强大先验知识，并激发了越来越多的研究，利用 2D 先验模型来辅助 3D 生成任务。以 DreamField [7]、DreamFusion [8] 和 Magic3D [9] 为例，一系列工作遵循每个形状优化的范式。它们通常优化 3D 表示（即 NeRF、网格、SMPL 人体模型），并利用可微渲染从各种视图生成 2D 图像。然后，这些图像被送入 CLIP 模型或 2D 扩散模型计算损失函数，用于引

导 3D 形状的优化。除了基于优化的 3D 形状生成之外，一些工作训练了一个 3D 生成模型，但利用了 CLIP 的嵌入空间 [15]，而一些工作则专注于使用 2D 模型先验为输入网格生成纹理或材质 [16, 17]。

2.2 三维物体数据集

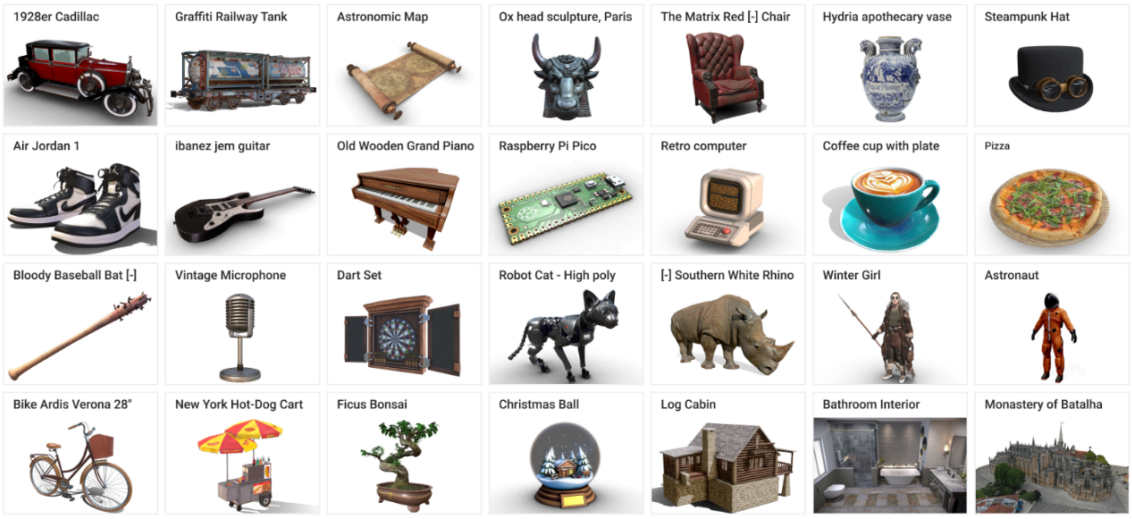

图 1. Objaverse 数据集部分物体

尽管在 3D 视觉领域存在相当大的兴趣和潜在应用，但高保真度的 3D 模型数据集仍然规模不足，且物体类别的多样性有限。为了填补这一空白，华盛顿大学提出了 Objaverse 1.0 [18]，这是一个包含超过 800,000 个（并不断增长）带有描述性字幕、标签和动画的 3D 模型的大型数据集，部分数据如图 1所示。与当今的 3D 存储库相比，Objaverse 在规模、类别数量和类别内实例的视觉多样性方面都有所提升。该方法的重建部分将使用该数据集进行训练。

测试数据集除了上述提及的 Objaverse [18] 外，还包括谷歌提出的 GSO [19] 数据集，交互式的 3D 模拟已经在机器人领域和计算机视觉领域取得了突破，但为了深度学习需要模拟广泛多样的环境，需要大量逼真的 3D 物体模型。为了解决这个需求，谷歌推出了 Google Scanned Objects，这是一个开源数据集，包含一千多个 3D 扫描的家居物品模型，以 Creative Commons 许可发布；这些模型经过预处理，可用于 Ignition Gazebo 和 Bullet 仿真平台，并且也容易迁移到其他物理仿真环境。

2.3 单张图像三维重建

在 CLIP 和大规模 2D 扩散模型出现之前，人们通常通过学习来自 3D 合成数据 [4] 或真实扫描的 3D 先验知识。最近，越来越多的工作都在学习如何从单个图像生成 3D 隐式场。如前所述，一些最近的工作利用 2D 扩散模型进行每个形状的优化，从而允许进行文本到三维的任务 [7–9]，因为扩散模型通常以文本为条件。为了实现从单个图像生成 3D 模型，一些工作 [10, 16] 利用文本反演 [20]，找到与输入图像最匹配的文本嵌入，然后将其输入到扩散模型中。NeuralLift-360 [21] 添加了一个 CLIP 损失，以强制渲染图像与输入图像之间的相似性。3DFuse [22] 通过使用 LoRA 层 [23] 和稀疏深度注入器对 Stable Diffusion 模型进行微调，以

确保更大的 3D 一致性。最近的一个工作 Zero123 [11] 对 Stable Diffusion [5] 模型进行微调，以基于相对相机姿态生成输入图像的新视图。除了这些方法之外，OpenAI 训练了一个 3D 本地扩散模型 Point-E [24]，该模型使用数百万个内部 3D 模型生成点云。最近，他们发布了另一个模型 Shap-E [25]，该模型经过训练，可以生成用于产生纹理网格或神经辐射场的隐式函数参数。

2.4 生成式神经重建

传统的类似 NeRF 的方法 [12] 使用神经网络表示单个场景，并需要每个场景进行优化。然而，一些方法的目标是学习跨场景的先验知识，并推广到新颖的场景。这些方法通常将几个源视图作为输入，并利用 2D 网络提取 2D 特征。然后，像素特征被投影到 3D 空间，并在其上应用基于 NeRF 的渲染流程。通过这种方式，它们可以在单个前向传递中给定几个源视图生成一个 3D 隐式场。除了密度场表示，一些方法如 SparseNeuS [13] 和 VolRecon [26] 则利用 SDF 表示进行几何重建。

3 本文方法

3.1 本文方法概述

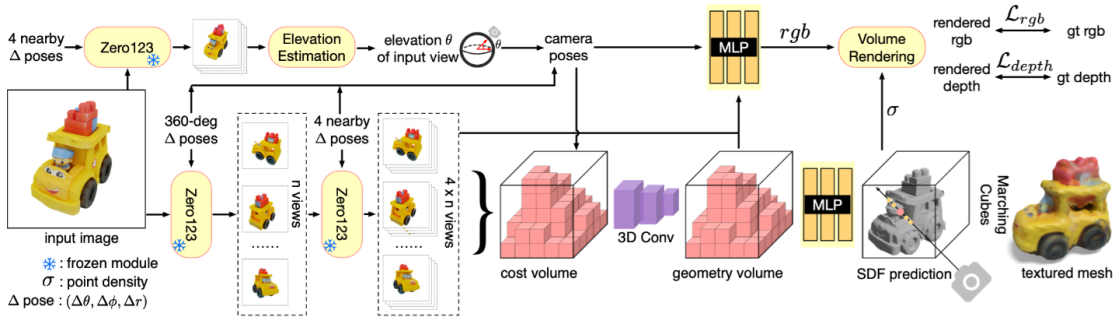

图 2. 模型方法 pipeline 示意图

该模型的 pipeline 如图 2所示，主要包括两个部分：

多视角合成：使用一个视角条件的 2D 扩散模型，Zero123 [11]，以两阶段方式生成多视角图像。Zero123 的输入包括一张单一图像和一个相对的相机变换，该变换由相对球面坐标 $(\Delta\theta, \Delta\phi, \Delta r)$ 参数化。

3D 重建：将多视角姿势图像输入一个基于 SDF 的可推广神经表面重建模块，以进行 360° 网格重建。

3.2 Zero123: 视图条件的二维扩散模型

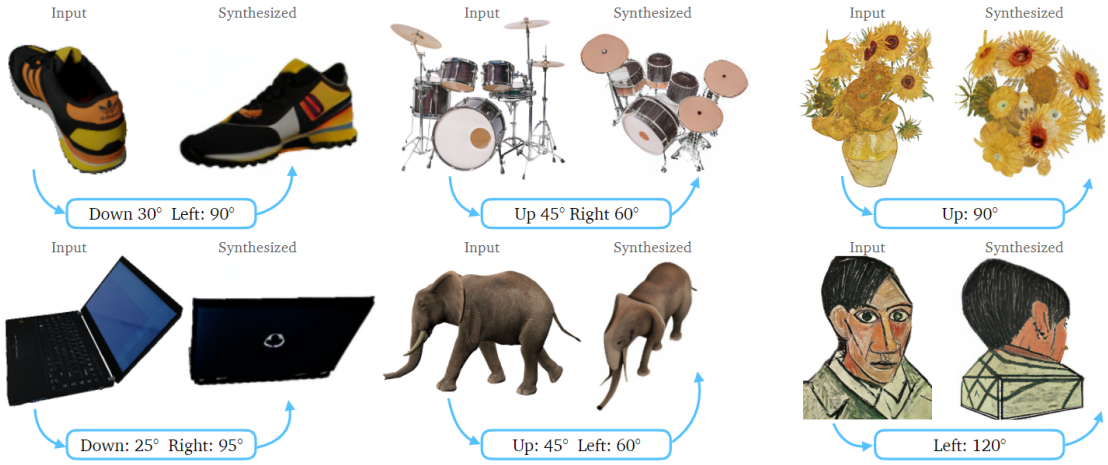

图 3. Zero-1-to-3 部分示例

虽然最初的扩散模型主要专注于文本到图像的任务，但最近的工作 [23,27] 表明，对预训练模型进行微调使我们能够向扩散模型添加各种条件控制，并根据特定条件生成图像。最近的工作 Zero123 [11] 具有相似的思路，旨在为 Stable Diffusion 模型 [5] 添加视点条件控制。具体而言，给定一个物体的单个 RGB 图像和一个相对的相机变换，Zero123 的目标是控制扩散模型以在这个转换后的相机视图下合成一个新的图像。为了实现这一目标，Zero123 通过对其相对相机变换的配对图像进行微调，这些图像是从大规模 3D 数据集 [18] 中合成的。在创建微调数据集时，Zero123 假设物体位于坐标系的原点，并使用球面相机，即相机位于球面的表面，并始终朝向原点。对于两个相机姿态 (θ_1, ϕ_1, r_1) 和 (θ_2, ϕ_2, r_2)，其中 θ_i, ϕ_i 和 r_i 分别表示极角、方位角和半径，它们的相对相机变换被参数化为 $(\theta_2 - \theta_1, \phi_2 - \phi_1, r_2 - r_1)$。它们的目标是学习一个模型 f，使得 $f(x_1, \theta_2 - \theta_1, \phi_2 - \phi_1, r_2 - r_1)$ 在感知上类似于 x_2，其中 x_1 和 x_2 是从不同视角拍摄的物体的两幅图像，部分例子如图3所示

3.3 基于不完美多视图预测的神经表面重建

如 2所示，我们的重建模块以 m 个姿势不同的源图像作为输入。该模块首先使用一个 2D 特征网络提取 m 个 2D 特征图。接下来，模块构建一个 3D 代价体积，其内容首先通过将每个 3D 体素投影到 m 个 2D 特征平面，然后获取跨 m 个投影的 2D 位置的特征方差来计算。然后，使用稀疏 3D CNN 处理代价体积，以获取几何体积，该几何体积对输入形状底层几何进行编码。为了预测任意 3D 点的 SDF，MLP 网络将 3D 坐标和其对应的来自几何编码体积的插值特征作为输入。为了预测 3D 点的颜色，另一个 MLP 网络接受在投影位置的 2D 特征、几何体积中的插值特征以及查询光线相对于源图像的视线方向的输入。网络预测每个源视图的混合权重，3D 点的颜色被预测为其投影颜色的加权和。最后，对于 RGB 和深度渲染，两个 MLP 网络之上应用了基于 SDF 的渲染技术 [28]。

尽管原始的 SparseNeuS [13] 论文仅展示了正面视图的重建，但作者通过在训练过程中选择特定的源视图并添加深度监督，将其扩展为在单次前向传递中重建 360 度网格。具体而言，重建模型在冻结 Zero123 的情况下，通过训练一个 3D 对象数据集进行训练。作者按照

Zero123 的规范化训练形状的方式，并使用球面相机模型。对于每个形状，首先从均匀放置在球体上的 n 个相机位姿中渲染出 n 个真值 RGB 和深度图像。对于 n 个视角中的每一个，作者使用 Zero123 预测四个附近的视角。在训练过程中，我们将所有 $4*n$ 个预测与真值位姿一起馈送到重建模块，并随机选择 n 个真值 RGB 图像视角的其中一个作为目标视角。作者将这种视角选择策略称为两阶段源视角选择。通过真值 RGB 和深度值监督训练，模块可以学会处理来自 Zero123 的不一致预测并重建一致的 360 度网格。

3.4 损失函数设置

重建模块的损失函数如下：

$$\mathcal{L} = \mathcal{L}_{rgb} + \lambda_0 \mathcal{L}_{depth} + \lambda_1 \mathcal{L}_{eikonal} + \lambda_2 \mathcal{L}_{sparsity} \tag{1}$$

其中，$\mathcal{L}_{rgb}$ 表示渲染图像的颜色与真值颜色之间的 $l1$ 损失，通过累积权重之和加权；$\mathcal{L}_{depth}$ 对应于渲染图像的深度与真值深度之间的 $l1$ 损失；$\mathcal{L}_{eikonal}$ 和 $\mathcal{L}_{sparsity}$ 分别代表 SparseNeuS [13] 方法中程函损失和稀疏损失。

根据以往经验，将权重设置为 $\lambda_0 = 1, \lambda_1 = 0.1$ 和 $\lambda_2 = 0.02$。对于 λ_2，作者采用了 SparseNeuS [13] 中的线性预热策略。为了训练方法中的重建模块，作者使用 Objaverse [18] 数据集中的 LVIS 子集，该数据集包含 1,156 个类别的 46,000 个 3D 模型。重建模块使用两个 A10 GPU 进行 300,000 次迭代训练，训练过程大约持续 6 天。需要注意的是，作者认为他们的重建模块并不严重依赖大规模的训练数据，因为它主要利用局部对应关系来推断几何形状，这相对容易学习和推广。

4 复现细节

4.1 与已有开源代码对比

```
    cam_vis = CameraVisualizer(vis_output)

    # Define the function to be called when any of the btn_retry buttons are clicked
    def on_retry_button_click(*btn_retrys):
        any_checked = any([btn_retry for btn_retry in btn_retrys])
        print('any_checked:', any_checked, [btn_retry for btn_retry in btn_retrys])
        if any_checked:
            return (gr.update(visible=True), gr.update(visible=True))
        else:
            return (gr.update(), gr.update())
    # make regen_btn visible when any of the btn_retry is checked
    for btn_retry in btn_retrys:
        # Add the event handlers to the btn_retry buttons
        btn_retry.change(fn=on_retry_button_click, inputs=[*btn_retrys], outputs=[regen_view_btn, regen_mesh_btn], queue=False)


    run_btn.click(fn=partial(update_guide, _SAM), outputs=[guide_text], queue=False
                  ).success(fn=partial(preprocess_run, predictor, models),
                            inputs=[image_block, preprocess_chk, *bbox_sliders],
                            outputs=[sam_block]
                  ).success(fn=partial(update_guide, _GEN_1), outputs=[guide_text], queue=False
                  ).success(fn=partial(stage1_run, models, device, cam_vis),
                            inputs=[tmp_dir, sam_block, scale_slider, steps_slider],
                            outputs=[elev_output, vis_output, *views]
                  ).success(fn=partial(update_guide, _GEN_2), outputs=[guide_text], queue=False
                  ).success(fn=partial(stage2_run, models, device),
                            inputs=[tmp_dir, elev_output, scale_slider, glb_chk],
                            outputs=[mesh_output]
                  ).success(fn=partial(update_guide, _DONE), outputs=[guide_text], queue=False)

```

图 4. 可视化界面部分核心代码

本文复现了 One-2-3-45 [29] 中的单张图片作为输入的三维重建算法，本文的代码已开源，在复现过程中，主要方法部分引用了源码。除此之外，我还增加了一个可交互的可视化界面，该界面允许用户直接输入自定义的图片到模型中与模型进行直接交互，让模型对任意图片进行在线重建。同时，我们还提供了在线可视化功能，用户可以实时查看最终重建得到的网格模型。部分核心代码如图 4所示。

4.2 实验环境搭建

想要运行该模型，硬件上需要一张显存 $\geq$ 18GB 的 NVIDIA 显卡，操作系统要求为 Ubuntu，因此配置要求如表 1。

操作系统	Ubuntu
GPU	an NVIDIA GPU with memory $\geq$ 18GB

表 1. 配置要求

在复现过程中，我所用的服务器操作系统为 Ubuntu 20.04，CPU 为 Intel(R) Xeon(R) Gold 5120 CPU @ 2.20GHz，且配备有具有 24G 显存的 Quadro RTX 6000 显卡，具体配置如表 2。

操作系统	Ubuntu 20.04
CPU	Intel(R) Xeon(R) Gold 5120 CPU @ 2.20GHz
GPU	Quadro RTX 6000 24GB
编译语言	Python
DL 开发平台	Pytorch

表 2. 复现过程所用服务器配置

项目环境的搭建分为如下五步：

- Step 1: 安装 Debian 包。

```
sudo apt update && sudo apt install git-lfs libsparsehash-dev build-essential
```

- Step 2: 创建并激活 conda 环境。

```
conda create -n One2345 python=3.10
conda activate One2345
```

- Step 3: 将 github 仓库克隆到本地。

```
# 确保已经安装了 git-lfs.
git lfs install
git clone https://github.com/One-2-3-45/One-2-3-45
cd One-2-3-45
```

- Step 4: 安装项目依赖。

```
# Ensure that the installed CUDA version matches the torch's CUDA version.
# Example: CUDA 11.8 installation
wget https://developer.download.nvidia.com/compute/cuda/11.8.0/local_installers/
    cuda_11.8.0_520.61.05_linux.run
sudo sh cuda_11.8.0_520.61.05_linux.run
export PATH="/usr/local/cuda-11.8/bin:$PATH"
export LD_LIBRARY_PATH="/usr/local/cuda-11.8/lib64:
    $LD_LIBRARY_PATH"
# Install PyTorch 2.0.1
pip install --no-cache-dir torch==2.0.1 torchvision torchaudio --index-url
    https://download.pytorch.org/whl/cu118
# Install dependencies
pip install -r requirements.txt
# Install inplace_abn and torchsparse
export TORCH_CUDA_ARCH_LIST="7.0;7.2;8.0;8.6+PTX" # CUDA
    architectures. Modify according to your hardware.
export IABN_FORCE_CUDA=1
pip install inplace_abn
FORCE_CUDA=1 pip install --no-cache-dir git+https://github.com/mit-han-
    lab/torchsparse.git@v1.4.0
```

- Step 5: 下载模型检查点。

```
python download_ckpt.py
```

4.3 界面分析与使用说明

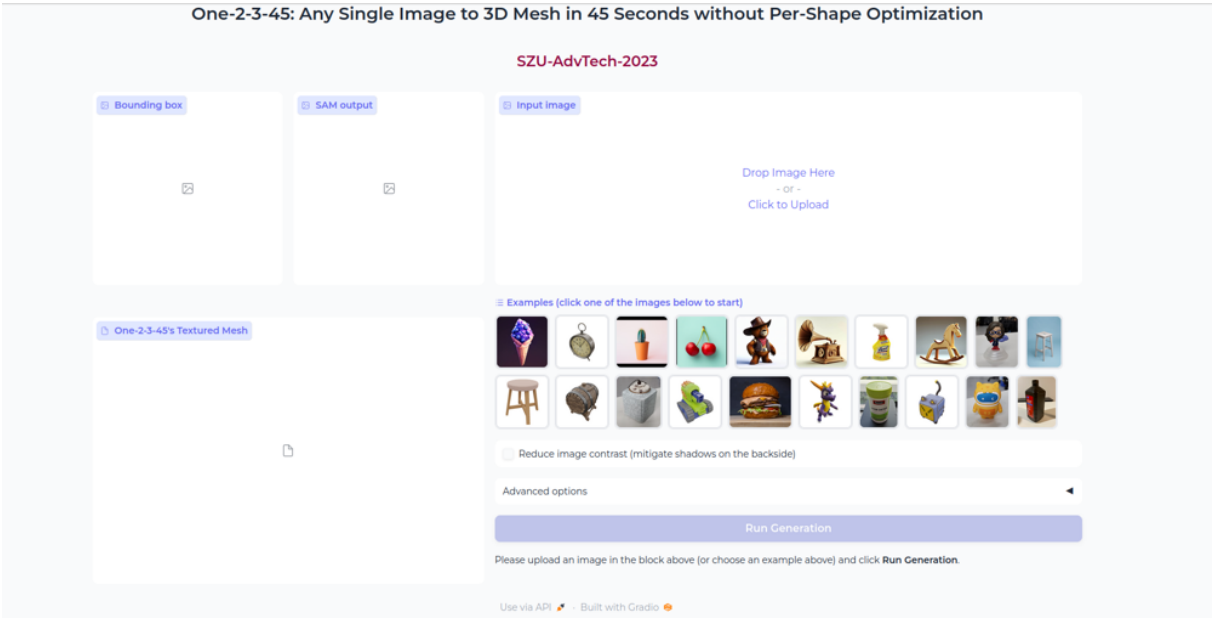

图 5. 可视化界面

本文在复现了算法的基础上，增加了一个可以让用户直接与模型进行交互的可视化界面，如图5所示。用户可以选择右边中间例子中的图片，选中的图片会在右上角的输入图片框中进行显示，模型会分析图片中主要物体的碰撞盒，并将结果存放在左上角的碰撞盒框中，然后图片会输入 SAM 中进行分割，将主要物体分割出来，展示在 SAM 输出框中，并用于后续的重建。SAM 分割完成后，用户可以点击 **Run Generation** 按钮开始重建，重建后的结果会存放在左下角的输出框内。其中一个例子如图6所示。

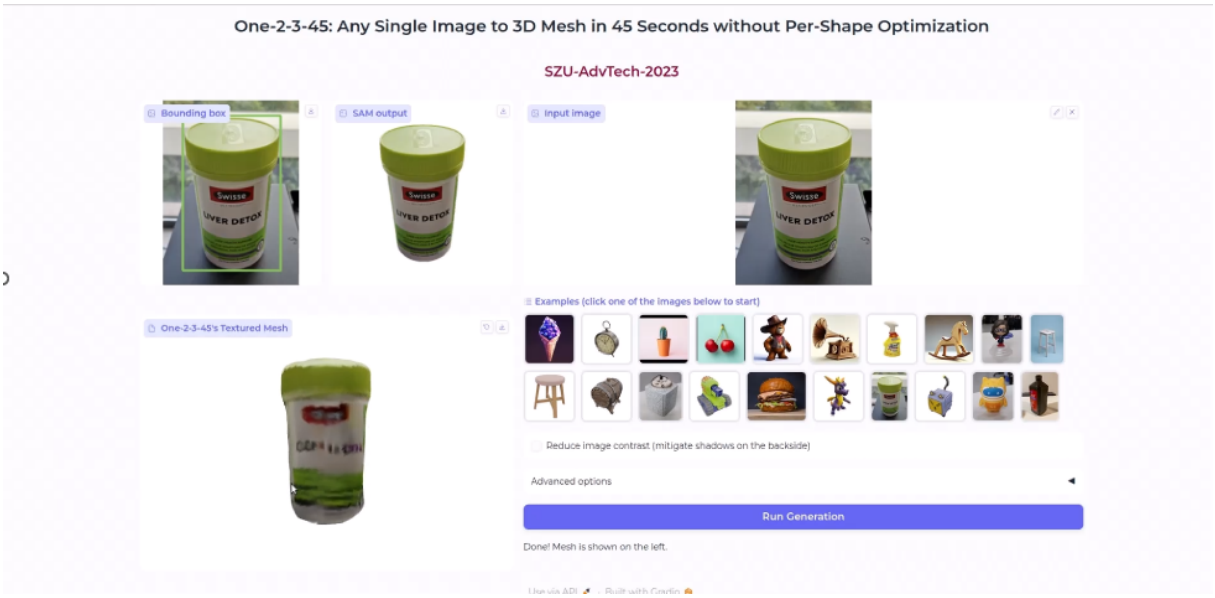

图 6. 可视化界面 demo 展示

5 实验结果分析

图 7. 定性实验结果展示

在图7中展示了方法的定性示例，说明了它在处理合成图像和真实图像方面的有效性。我们还将 One-2-3-45 与现有的零样本单图像 3D 重建方法进行了比较，包括 Point-E [24]、Shap-E [25]、Zero123 [11]。其中，Point-E 和 Shap-E 是由 OpenAI 发布的两个 3D 本地扩散模型，它们在几百万个 3D 数据上进行训练。Zero123 是基于优化的方法，利用了 Stable Diffusion [5] 的先验知识。

	先验	F-Score			渲染时间
		GSO	Obj.	平均	
Point-E [24]	内部三维数据集	81.2	80.8	81.0	∼80s
Shape-E [25]		83.1	81.5	82.3	∼30s
Zero123 [11]	2D 扩散模型	75.1	69.9	72.5	∼15mins
本文方法 [29]		84.0	83.1	83.5	∼45s

表 3. 在 Objaverse [18] 和 GSO [19] 数据集上测试的定量指标

如表3所示，本文方法在 F-Score 方面优于其他基准方法。关于运行时间，本文的方法在

基于优化的方法上表现出明显的优势，并与 3D 本地扩散模型（如 Point-E [24] 和 Shap-E [25]）持平。具体而言，3D 重建模块在大约 5 秒内就能完成对 3D 网格的重建，其余的时间主要用于 Zero123 的预测。

6 总结与展望

One-2-3-45 是一种很新颖的方法，可以从单张图像中重建出任意物体的高质量 360 度网格模型。与现有的零样本方法相比，其结果展现出更优越的几何形状、更强的三维一致性以及对输入图像的显著粘合性。值得注意的是，该方法可以在单次前向传递中重建网格，无需耗时的优化过程，从而大大减少了处理时间。本次复现方法部分主要引用源码，但增加了一个可供用户与模型进行直接交互的在线可交互界面，后续研究可以考虑将此方法扩展到文本到 3D 的任务。

参考文献

[1] Tomas Lozano-Perez, Leslie Pack Kaelbling, and Han-Pang Chiu. Automatic class-specific 3d reconstruction from a single image. 2009.

[2] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII 14*, pages 628–644. Springer, 2016.

[3] Rohit Girdhar, David F Fouhey, Mikel Rodriguez, and Abhinav Gupta. Learning a predictable and generative vector representation for objects. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VI 14*, pages 484–499. Springer, 2016.

[4] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[5] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[6] 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 *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[7] Ajay Jain, Ben Mildenhall, Jonathan T Barron, Pieter Abbeel, and Ben Poole. Zero-shot text-guided object generation with dream fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 867–876, 2022.

[8] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022.

[9] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 300–309, 2023.

[10] Luke Melas-Kyriazi, Iro Laina, Christian Rupprecht, and Andrea Vedaldi. Realfusion: 360deg reconstruction of any object from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8446–8455, 2023.

[11] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9298–9309, 2023.

[12] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

[13] Xiaoxiao Long, Cheng Lin, Peng Wang, Taku Komura, and Wenping Wang. Sparseneus: Fast generalizable neural surface reconstruction from sparse views. In *European Conference on Computer Vision*, pages 210–227. Springer, 2022.

[14] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14124–14133, 2021.

[15] Yen-Chi Cheng, Hsin-Ying Lee, Sergey Tulyakov, Alexander G Schwing, and Liang-Yan Gui. Sdfusion: Multimodal 3d shape completion, reconstruction, and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4456–4465, 2023.

[16] Gal Metzer, Elad Richardson, Or Patashnik, Raja Giryes, and Daniel Cohen-Or. Latent-nerf for shape-guided generation of 3d shapes and textures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12663–12673, 2023.

[17] Oscar Michel, Roi Bar-On, Richard Liu, Sagie Benaim, and Rana Hanocka. Text2mesh: Text-driven neural stylization for meshes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13492–13502, 2022.

[18] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13142–13153, 2023.

[19] Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B McHugh, and Vincent Vanhoucke. Google scanned objects: A high-quality dataset of 3d scanned household items. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2553–2560. IEEE, 2022.

[20] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv preprint arXiv:2208.01618*, 2022.

[21] Dejia Xu, Yifan Jiang, Peihao Wang, Zhiwen Fan, Yi Wang, and Zhangyang Wang. Neurallift-360: Lifting an in-the-wild 2d photo to a 3d object with 360deg views. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4479–4489, 2023.

[22] Junyoung Seo, Wooseok Jang, Min-Seop Kwak, Jaehoon Ko, Hyeonsu Kim, Junho Kim, Jin-Hwa Kim, Jiyoung Lee, and Seungryong Kim. Let 2d diffusion model know 3d-consistency for robust text-to-3d generation. *arXiv preprint arXiv:2303.07937*, 2023.

[23] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[24] Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. Point-e: A system for generating 3d point clouds from complex prompts. *arXiv preprint arXiv:2212.08751*, 2022.

[25] Heewoo Jun and Alex Nichol. Shap-e: Generating conditional 3d implicit functions. *arXiv preprint arXiv:2305.02463*, 2023.

[26] Yufan Ren, Tong Zhang, Marc Pollefeys, Sabine Süsstrunk, and Fangjinhua Wang. Volrecon: Volume rendering of signed ray distance functions for generalizable multi-view reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16685–16695, 2023.

[27] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023.

[28] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689*, 2021.

[29] Minghua Liu, Chao Xu, Haian Jin, Linghao Chen, Zexiang Xu, Hao Su, et al. One-2-3-45: Any single image to 3d mesh in 45 seconds without per-shape optimization. *arXiv preprint arXiv:2306.16928*, 2023.