

Reference-Based 3D-Aware Image Editing with Triplanes 论文复现及改进

成员：陈建豪 2200094817、李大韩 2200094819

简介

研究背景与问题

近年来，生成对抗网络在高保真图像生成与真实图像编辑任务中取得了显著进展。典型方法通常先通过 GAN inversion 将输入图像映射到生成器的潜空间，再在潜空间中进行属性操作以实现图像编辑。随着 EG3D 等 3D-aware GAN 的提出，模型能够从单张图像中重建具有三维一致性的几何与外观表示，从而支持多视角一致的图像渲染与编辑。

然而，在3D-aware 场景下，reference-based local editing 仍缺乏统一且成熟的系统框架：一方面，许多 3D-aware 编辑方法并不支持从参考图像中拷贝局部属性；另一方面，现有的 reference-based 2D 编辑方法通常缺乏 3D 一致性，在视角变化时容易产生漂移、几何不一致或明显伪影。

核心观察

论文提出一个关键观察：在 EG3D 的 triplane 表征中，不同输入图像（即使相机位姿不同）都可以被投影到同一个规范的 triplane 空间。因此，在 triplane 域中进行局部特征的迁移与融合，更容易实现跨视角的一致对齐。

相比直接在 2D 图像空间进行像素级拷贝操作（需要严格的空间对齐），triplane 表征天然地引入了三维结构约束，使得局部属性的复制与粘贴在语义和几何层面上更加稳定。

方法概览

论文将 reference-based 3D-aware 编辑组织为一个端到端的集成框架，主要包含以下几个核心模块：

- Encoding：将源图像与参考图像分别编码到生成器的 latent/triplane 表示（得到 T_{src}, T_{refT_src} , $T_{ref}T_{src}, T_{ref}$ 等）。针对融合任务，论文进一步引入并微调了专用编码器 $E^*E^*E^*$ ，以提升颜色一致性与局部细节（如眼镜框）表现，并减少背景信息泄漏。
- Automatic localization：利用 2D 语义分割获得目标部位的 ROI mask，并通过 masked residual gradients 将该定位信息反传到 triplane 空间，从而得到对应部位的 triplane 掩码。

- Spatial disentanglement: 基于 triplane 掩码，在空间维度上区分“需要从参考图像拷贝的区域”和“需要保留源图像的区域”，实现 triplane 域中的局部属性迁移。
- Fusion learning / implicit fusion: 直接在 triplane 中进行硬拼接（hard stitching）会导致边界处颜色或纹理不连续。为此，论文利用编码器 $E * E^* E^*$ 对初步融合结果进行隐式融合，从而获得更自然、无缝的边界过渡和更一致的外观表现。

总体而言，该方法旨在在尽可能保持源图像身份与非编辑区域不变的前提下，将参考图像的指定局部属性高保真地迁移到编辑结果中，并同时保证多视角的一致性。

改进动机与原因分析

尽管论文提出了一个完整且有效的 reference-based 3D-aware 图像编辑框架，并在多个任务上取得了较好的实验效果，但从方法设计与实际应用角度来看，该方法仍存在一定局限性。

对自动定位与掩码质量的依赖

原方法在 triplane 域进行局部属性编辑之前，依赖于自动定位模块（automatic localization）来确定需要编辑的空间区域。该过程通常基于 2D 语义分割结果构建 ROI mask，并通过 masked residual gradients 将定位信息映射到 triplane 空间。因此，后续所有编辑操作的有效性在很大程度上取决于前序定位结果的准确性。

当语义分割模型未能正确识别目标属性，或在某些样本中对应类别缺失时，生成的 ROI mask 可能为空或不准确，从而导致后续 triplane 编辑过程无法生效。该问题在属性边界模糊或样本分布较为复杂的情况下尤为明显。

原方法的局限性分析

在 triplane 空间进行局部特征迁移时，原方法仍需要依赖掩码来区分“需要从参考图像拷贝的区域”和“需要保留源图像的区域”。然而，这类掩码在本质上是离散的二值信号，其取值通常为 0 或 1。

由于 triplane 特征本身是连续的高维表示，直接使用二值掩码进行特征拼接（hard stitching）会在掩码边界处引入突变，进而可能导致颜色、纹理或光照上的不连续现象。这种边界不连续在最终渲染图像中往往表现为可见的接缝（seam），尤其在肤色、头发等对连续性较为敏感的区域更为明显。

复现中观察到的具体问题

论文通过引入并微调专用编码器 $E * E^* E^*$ 来实现隐式融合（implicit fusion），以缓解直接拼接带来的边界问题。该策略在实践中确实能够显著改善融合质量，但其效果主要依赖于编码器的学习能力。

一方面，隐式融合过程缺乏显式的边界建模机制，其行为较难解释和控制；另一方面，当掩码本身较为粗糙或边界位置存在偏差时，编码器并不总能完全消除由硬拼接引入的伪影。因此，融合质量在一定程度上仍受限于掩码的离散性与初始拼接方式。

综上所述，原方法在整体框架与性能上已具备较高完成度，但其在 triplane 融合阶段仍存在对离散掩码和隐式融合的依赖。这为后续在不改变网络结构的前提下，通过引入显式的掩码平滑机制来改善边界过渡提供了空间，也构成了本文改进方法的主要动机。

改进方法

改进方法一：基于Gaussian Blur的掩码平滑

改进动机

在上一节中分析可知，原方法在 triplane 融合阶段仍然依赖于离散的二值掩码来区分参考区域与保留区域。尽管论文通过引入隐式融合编码器 $E * E^* E^*$ 在一定程度上缓解了硬拼接带来的边界问题，但当掩码边界较为粗糙或定位存在偏差时，融合结果仍可能出现可见的边界不连续现象。

基于上述观察，我们希望在不改变原有网络结构和训练流程的前提下，对融合阶段进行轻量级改进，使得 triplane 特征在边界区域能够更加平滑地过渡。为此，本文尝试在掩码层面引入简单而有效的平滑机制，以显式建模边界区域的连续性。

方法原理

原方法中的掩码通常为二值形式，其取值仅为 0 或 1，用于指示某一 triplane 位置是否来自参考图像。当该掩码直接参与 triplane 特征融合时，会在掩码边界处引入特征突变。

为缓解这一问题，本文引入 Gaussian blur 对掩码进行平滑处理，其核心思想如下：

- 保留原有的二值掩码作为逻辑判断与定位结果，不对其本身进行修改；
- 在融合阶段，基于该二值掩码生成一个连续的 soft mask；
- 通过 Gaussian blur 操作，将原本离散的边界扩展为一个连续过渡区域，使掩码取值在 0~1 之间平滑变化；
- 使用该 soft mask 对源 triplane 与参考 triplane 进行加权融合，从而在边界处形成更加自然的特征过渡。

通过上述方式，掩码不再作为“硬选择信号”，而是转变为一组连续的融合权重，有助于减轻边界区域的拼接伪影。

实现细节

在具体实现中，本文在不影响原有掩码逻辑的前提下，对融合阶段进行了如下调整：

- 原始二值掩码 M_{bin} 仍用于表示目标区域的空间位置；
- 仅在 triplane 融合阶段，引入平滑后的掩码 M_{soft} ；
- 使用 Gaussian blur 对 M_{bin} 进行平滑，得到 $M_{soft} \in [0, 1]$ ；
- triplane 的融合方式由原先的硬拼接改为加权融合：

$$T = (1 - M_{soft}) \cdot T_{dst} + M_{soft} \cdot T_{src}$$

其中，Gaussian blur 的核大小与迭代次数作为超参数进行控制，不引入额外的可学习参数。该改进属于纯工程层面的增强，计算开销较小，且可以灵活嵌入原有 pipeline 中。

实验效果与分析



gaussian blur



based

在保持源图像、参考图像与编辑属性一致的条件下，我们分别对原方法（不使用掩码平滑）与引入 Gaussian blur 掩码平滑后的方法进行了对比实验。

实验结果表明，在 eyes、hair 等对边界连续性较为敏感的属性编辑中，引入 Gaussian blur 后，融合区域的过渡更加自然，边界处的颜色和纹理不连续现象得到明显缓解。相比之下，原方法在部分样本中仍可观察到轻微的接缝或突变。

需要指出的是，当自动定位阶段未能获得有效 ROI（例如某些样本中 glasses 类别在语义分割结果中缺失）时，无论是否引入掩码平滑，编辑效果均较为有限。这一现象也验证了本改进方法对定位模块的依赖性。

改进方法二：眉毛编辑实现

改进动机

在原始的 Triplane Editing 实现中，虽然代码的属性通道字典中已经定义了眉毛对应的语义分割通道（通道 2 和 3，分别对应左眉和右眉），但实际的编辑逻辑并未实现。这一缺失使得用户无法对眉毛这一重要的面部特征进行编辑操作。

然而，眉毛编辑的实现面临着若干技术挑战。一方面，眉毛与眼睛的距离非常近，在进行眉毛编辑时容易产生区域冲突。如果 mask 区域设置不当，可能会意外地修改眼睛区域，或者眼睛编辑会影响到眉毛，这就需要我们在设计融合策略时充分考虑这种空间上的邻近关系。另一方面，BiSeNet 语义分割网络对于小尺度特征的分割精度相对有限。眉毛作为细长且面积较小的特征，其分割结果可能存在边界不够精确、局部缺失等问题，这进一步增加了实现高质量眉毛编辑的难度。

方法原理

在参数设计方面，通过对对比分析眼睛、嘴巴等其他部位的参数配置，我们发现这些部位普遍使用较大的参数值以适应其较大的面积和较宽的过渡区域。考虑到眉毛的面积约为眼睛的三分之一到二分之一，我们相应地缩小了这些参数。较小的模糊核尺寸能够在平滑 mask 边界的同时保留眉毛的细节特征，避免过度模糊导致眉毛形状失真。较小的形态学核尺寸确保腐蚀和膨胀操作不会过度扩张或收缩眉毛区域，从而精确控制编辑范围。较小的标准差则使得高斯模糊更加温和，在确保边界平滑的同时最大限度地保持眉毛的原始形态。

在融合策略方面，我们采用了与眼睛和嘴巴相同的三区域融合模式。这种策略将编辑区域划分为三个部分：核心区域、过渡区域和保留区域。核心区域对应腐蚀后的眉毛 mask，在这一区域内，我们直接使用源图像的 triplane 特征，确保眉毛的核心形状和纹理被完整移植。过渡区域是膨胀 mask 与腐蚀 mask 的差集，在这一区域内，我们使用隐式融合生成的 triplane 特征，通过编码器的重新编码来确保眉毛边界与周围皮肤的自然过渡。保留区域则是膨胀 mask 之外的所有区域，在这一区域内我们保持目标图像的原始 triplane 特征不变，从而保护眼睛、额头等其他面部特征不受影响。这种三区域设计既保证了眉毛特征的准确移植，又确保了编辑边界的自然性，有效避免了生硬的拼接痕迹。

实现细节

眉毛编辑的核心代码实现如下。首先，我们调用 [forward_manual_fusion] 方法进行手动融合，该方法接收源图像和目标图像的 W 编码及 triplane，以及语义通道信息，返回手动融合的 triplane 和源、目标的 triplane mask。

接下来，我们使用 [create_dilated_eroded_tp_masks] 方法对源 mask 进行形态学处理，生成腐蚀 mask 和膨胀 mask。这一步骤中，我们传入精细化参数。基于这些 mask，我们定义了三区域融合所需的四个 mask 变量：`M_src` 设为腐蚀 mask，代表眉毛核心区域；`M_imp_mid` 设为膨胀 mask 与腐蚀 mask 的差值，代表过渡区域；`M_dst` 设为 1 减去膨胀 mask，代表保留区域；`M_man` 设为 0，表示不使用手动融合区域（这与眼睛、嘴巴的策略一致，而与眼镜的策略不同）。

最后，我们设置 `w_sr = w_dst`，表示最终渲染时使用目标图像的 W 编码，这确保了编辑后图像的整体风格与目标图像保持一致。

实验效果与分析



眉毛编辑

实现眉毛编辑功能后，我们发现，目标图像的眉毛清晰地显示在编辑结果中，且眉毛的形状、粗细和位置都得到了准确的编辑。

多属性同时编辑

改进动机

在原始的 Triplane Editing 实现中，系统仅支持单属性编辑，即每次只能对一个面部特征（如眼睛或嘴巴）进行编辑操作。这种设计在处理单一编辑需求时表现良好，但在实际应用场景中存在明显的局限性。当用户希望同时修改多个面部特征时，必须进行多次独立的编辑操作，每次操作都需要重新加载图

像、编码、编辑和渲染，这不仅耗费大量时间，还可能因为多次操作的累积误差导致最终结果不够理想。此外，多次独立编辑还可能导致不同属性之间缺乏协调性，因为每次编辑都是在前一次编辑的基础上进行的，而不是在统一的框架下进行整体优化。因此，实现多属性同时编辑功能不仅能够显著提高编辑效率，还能够确保多个属性之间的协调一致性，为用户提供更加流畅和高质量的编辑体验。

实现多属性同时编辑面临着三个核心技术挑战。首先是 Mask 冲突问题。不同面部属性的语义分割区域可能存在重叠或邻接关系，例如眉毛和眼睛距离很近，鼻子和嘴巴也有部分重叠区域。当我们尝试同时编辑这些属性时，它们各自的 mask 区域可能会产生冲突，导致某些像素点被多个 mask 同时覆盖。如果不妥善处理这种冲突，可能会导致编辑结果出现不连续、不自然的现象，甚至可能完全破坏某些属性的编辑效果。其次是编辑顺序问题。如果采用顺序编辑的策略，即依次对每个属性进行编辑，那么后续的编辑操作可能会累积前面编辑产生的误差。每一次编辑都会对 triplane 进行修改，而这些修改可能包含一定的噪声或不精确性。当这些误差在多次编辑中累积时，最终结果可能会偏离预期，甚至出现明显的质量下降。最后是整体一致性问题。多个属性的编辑需要在整体上保持协调，确保编辑后的人脸看起来自然、真实。如果各个属性的编辑相互独立，缺乏全局的协调机制，可能会导致编辑后的人脸出现风格不统一、特征不协调等问题，影响最终的视觉效果。

方法原理

针对上述挑战，我们设计并实现了 Parallel (并行) 融合策略。该策略的核心思想是将多属性编辑从顺序执行转变为并行处理，通过智能的 mask 合并机制和一次性融合操作，同时解决 mask 冲突、误差累积和一致性问题。

Parallel 策略的算法流程可以分为三个主要步骤。第一步是并行计算所有属性的 mask。对于用户指定的每个待编辑属性（如眼睛、眉毛、嘴巴），我们分别调用 [reference_tp_edit] 方法中的 mask 计算逻辑，但不执行实际的 triplane 融合操作。这一步骤的关键在于，所有的 mask 计算都是基于原始的源 triplane 和目标 triplane，而不是基于中间编辑结果，从而避免了误差的累积。

第二步是智能合并所有 mask。由于不同属性的 mask 可能存在重叠区域，我们需要设计一种合理的合并策略来处理这种冲突。我们采用的方法是对所有属性的同类 mask 进行逐像素的最大值操作。这种策略的优势在于，它能够确保每个属性的核心区域都被保留，同时在重叠区域选择最强的信号。对于过渡区域、保留区域和手动融合区域，我们采用相同的最大值合并策略。合并完成后，我们还需要对 mask 进行归一化处理，确保所有 mask 的总和在每个像素位置都等于 1。这一归一化步骤保证了融合操作的数学正确性，避免了某些区域被过度强调或忽略。

第三步是一次性融合所有属性。基于合并和归一化后的 mask，我们执行最终的 triplane 融合操作。融合公式为：

$$\mathcal{T}_{\text{final}} = \mathcal{T}_{\text{src}} \odot M_{\text{src}}^{\text{combined}} + \mathcal{T}_{\text{imp}} \odot M_{\text{imp}}^{\text{combined}} + \mathcal{T}_{\text{dst}} \odot M_{\text{dst}}^{\text{combined}} + \mathcal{T}_{\text{man}} \odot M_{\text{man}}^{\text{combined}}$$

其中， \mathcal{T}_{imp} 是通过隐式融合生成的 triplane， \mathcal{T}_{man} 是通过手动融合生成的 triplane。这一步骤只执行一次融合操作，因此不会产生误差累积，同时由于所有属性的 mask 都已经被合并，融合结果能够自然地

处理属性之间的边界和过渡，确保整体的协调性。

Parallel 策略相比传统的顺序编辑方法具有多方面的优势。首先，它有效避免了误差累积问题。由于所有的 mask 计算都基于原始的 triplane，而不是基于中间编辑结果，每个属性的编辑都是独立且准确的，不会受到其他属性编辑的干扰。其次，它通过智能的 mask 合并机制优雅地处理了 mask 重叠问题。最大值操作确保了在重叠区域选择最强的编辑信号，而归一化操作则保证了融合的数学一致性。第三，Parallel 策略在计算效率上也具有优势。虽然需要为每个属性计算 mask，但由于只执行一次隐式融合和最终融合，总体计算时间反而少于多次独立编辑。最后，也是最重要的，Parallel 策略能够产生更好的编辑效果。由于所有属性在统一的框架下进行融合，编辑结果在整体上更加协调自然，避免了多次独立编辑可能导致的不一致性问题。

Sequential 策略的问题

在实现 Parallel 策略之前，我们也尝试了一种更直观的 Sequential (顺序) 融合策略。该策略的思路是依次对每个属性进行编辑，每次编辑都在前一次编辑的结果基础上进行。然而，实验结果表明，Sequential 策略存在严重的缺陷，几乎无法产生有效的多属性编辑效果。

Sequential 策略的核心问题在于其"无记忆"特性导致的编辑覆盖现象。具体而言，当我们按顺序编辑多个属性时，第一个属性的编辑可能是成功的，例如成功地将源图像的眼睛移植到目标图像上。然而，当我们进行第二个属性（如眉毛）的编辑时，问题就出现了。在眉毛编辑过程中，算法会计算眉毛的核心区域 mask (M_{src}) 和保留区域 mask (M_{dst})。保留区域 mask 定义为 $M_{dst} = 1 - M_{dilated}$ ，即除了眉毛区域之外的所有区域。在最终融合时，这个保留区域会从原始的目标 triplane (\mathcal{T}_{dst}) 中提取特征。问题在于，原始的目标 triplane 并不包含第一步编辑后的眼睛特征，因此这一步操作实际上会用原始目标的眼睛特征覆盖第一步移植的眼睛特征，导致第一步的编辑效果完全丧失。

为了更形象地理解这一问题，我们可以用一个比喻来说明。假设我们要将小明的眼睛和眉毛移植到小红的脸上。第一步，我们成功地将小明的眼睛移植到小红脸上，此时小红的脸具有小明的眼睛和小红自己的其他特征。第二步，我们尝试移植小明的眉毛。然而，在计算保留区域时，算法会从原始的小红脸（不包含小明眼睛）中提取非眉毛区域的特征，这就包括了眼睛区域。因此，第二步的融合操作实际上是：小明的眉毛 + 小红的眼睛（原始）+ 小红的其他特征。这样一来，第一步移植的小明眼睛就被小红的原始眼睛覆盖了，最终结果几乎等同于只进行了眉毛编辑，眼睛的编辑效果完全消失。

这一问题的根本原因在于 triplane 编辑算法的设计假设。在单属性编辑中，算法假设除了编辑区域之外的所有区域都应该保持目标图像的原始特征。这一假设在单属性编辑中是合理的，但在多属性顺序编辑中则会导致后续编辑覆盖前面编辑的问题。从算法的角度来看，Sequential 策略本质上缺乏对已编辑区域的"记忆"机制，每次编辑都将目标 triplane 视为完全未经编辑的原始状态，因此无法保留之前的编辑结果。

实验效果



Eyes + Brows Sequential



Eyes + Brows Parallel



Eyes + Brows + Mouth Parallel

为了验证这一分析，我们进行了对比实验。我们选择了眼睛和眉毛的组合作为测试案例，分别使用 Sequential 和 Parallel 策略进行编辑。实验结果清晰地展示了两种策略的差异。使用 Sequential 策略时，最终的编辑结果几乎与只进行眼睛编辑的结果相同，眉毛的编辑效果完全消失，验证了我们关于编辑覆盖问题的分析。

相比之下，使用 Parallel 策略时，眼睛和眉毛的编辑效果都得到了很好的保留，最终结果同时具有源图像的眼睛和眉毛特征，且两者之间的过渡自然流畅。这一对比实验不仅验证了 Parallel 策略的有效性，也深刻揭示了 Sequential 策略的根本缺陷，为我们最终选择 Parallel 作为多属性编辑的标准策略提供了充分的实验依据。

基于上述分析和实验结果，我们在最终的实现中完全移除了 Sequential 策略，只保留了 Parallel 策略作为多属性编辑的唯一方法。这一决策不仅简化了代码结构，也为用户提供了更加可靠和高质量的编辑体验。

讨论

成功之处

首先，在基于 Gaussian Blur 的掩码平滑改进中，我们成功地通过简单而有效的工程手段缓解了原方法在 triplane 融合阶段的边界不连续问题。该改进无需修改网络结构或重新训练模型，仅通过对掩码进行平滑处理即可显著改善融合区域的过渡效果。实验结果表明，在 eyes、hair 等对边界连续性较为敏感的属性编辑中，引入 Gaussian blur 后，融合区域的颜色和纹理过渡更加自然，有效减少了可见的接缝现象。这种轻量级的改进方式具有良好的实用价值，可以灵活嵌入原有 pipeline 中，为其他基于掩码融合的方法提供了有益的参考。

其次，在眉毛编辑功能的实现方面，我们成功填补了原始代码中的功能空白。通过参数配置和三区域融合策略，眉毛编辑能够准确地移植眉毛的形状、粗细和纹理特征，同时确保编辑边界的自然过渡。我们通过系统的参数对比分析，为眉毛的面部特征选择了合适的编辑参数，使得眉毛编辑能够与眼睛、嘴巴等其他部位的编辑保持一致的质量水平。实验结果表明，眉毛编辑在多视角渲染中保持了良好的 3D 一致性，证明了我们的实现方案在技术上是可靠和有效的。

第三，在多属性同时编辑功能方面，我们设计并实现的 Parallel 融合策略在效率和质量上都展现出了显著优势。通过并行计算所有属性的 mask 并进行智能合并，Parallel 策略有效解决了 mask 冲突、误差累积和整体一致性等核心技术挑战。与 Sequential 策略的对比实验清晰地展示了 Parallel 策略的优越性：Sequential 策略因其“无记忆”特性导致后续编辑覆盖前面编辑的问题，几乎无法产生有效的多属性编辑效果；而 Parallel 策略则能够同时保留多个属性的编辑结果，且整体效果自然协调。这一成果不仅显著提高了编辑效率，使得用户可以在一次操作中完成多个属性的编辑，也为用户提供了更加流畅和高质量的编辑体验。

不足之处

尽管本项目取得了一定成果，但在实际应用中仍存在一些局限性需要在未来工作中加以改进。

基于 Gaussian Blur 的掩码平滑改进虽然能够有效缓解边界不连续问题，但其效果在很大程度上依赖于前序自动定位模块的准确性。当语义分割模型未能正确识别目标属性，或在某些样本中对应类别缺失时，生成的 ROI mask 可能为空或不准确，此时无论是否引入掩码平滑，编辑效果均较为有限。例如，在我们的实验中，当某些样本的 glasses 类别在语义分割结果中缺失时，即使使用了掩码平滑，也无法产生有效的眼镜编辑效果。这一现象表明，掩码平滑作为一种后处理手段，无法从根本上解决定位不准确的问题，其改进效果受限于上游模块的质量。

眉毛编辑在实际应用中面临的一个主要问题是几何对齐的缺失。当源图像和目标图像的眉毛在位置、大小或角度上存在显著差异时，直接的特征移植可能会导致不自然的结果。具体而言，由于当前的编辑方法主要基于语义分割 mask 在 triplane 空间进行特征融合，并未考虑眉毛的几何属性（如位置、长度、弯曲度等），因此当源眉毛和目标眉毛的几何特征差异较大时，移植后的眉毛可能会出现位置不匹配或大小不协调的问题。理想的解决方案是在移植前进行几何对齐，根据眉毛与眼睛的相对位置关系、眉毛的长度和弯曲度等几何属性，自适应地调整源眉毛的位置、大小和形状，使其与目标图像的面部结构相匹配。这可以通过 facial landmarks 检测来获取眉毛和眼睛的关键点位置，然后基于这些关键点进行仿射变换或更复杂的几何变换。然而，这种几何对齐机制的引入需要额外的算法设计和实现工作，且可能增加计算复杂度，因此在当前的实现中尚未包含。

在多属性同时编辑方面，我们观察到一个明显的现象：随着同时编辑的属性数量增加，编辑质量呈现下降趋势。虽然理论上 Parallel 策略可以支持任意数量的属性同时编辑，但在实际应用中，当编辑的属性数量超过 3-4 个时，最终结果的质量和协调性会受到明显影响。这一问题的主要原因在于 mask 合并的复杂度随属性数量增加而上升。当多个属性的 mask 存在重叠或邻接关系时，我们采用的最大值合并和归一化策略虽然能够在数学上保证融合的一致性，但可能导致某些区域的 mask 值过小或过大。例如，

当同时编辑眼睛、眉毛、鼻子和嘴巴四个属性时，这些属性的 mask 在面部中心区域可能存在多处重叠，经过最大值合并后，某些过渡区域的 mask 值可能被过度强调，而另一些区域的 mask 值可能被过度削弱，从而影响融合的准确性和平滑性。

同时，目前的多属性同时编辑，我们暂时实现了相同源图像的不同部位编辑。在未来，我们可以使用不同源图像的部位进行编辑，得到更融合有趣的结果。

总结

本项目成功地扩展了 Triplane Editing 的功能，在原有单属性编辑框架的基础上实现了三项重要改进。首先，我们通过引入 Gaussian Blur 掩码平滑机制，有效缓解了原方法在 triplane 融合阶段的边界不连续问题。其次，我们实现了眉毛编辑功能，填补了原始代码中虽有定义但未实现的功能空白。最后，我们设计并实现了多属性同时编辑功能，使用户能够一次性编辑多个面部特征，显著提高了编辑效率。我们提出的 Parallel 融合策略通过并行计算所有属性的 mask 并进行智能合并，有效解决了 mask 冲突、误差累积和整体一致性等核心技术挑战。

总之，本项目通过实现眉毛编辑和多属性同时编辑功能，成功扩展了 Triplane Editing 的能力，为用户提供了更加灵活和高效的编辑工具。通过系统的实验验证和深入的理论分析，我们不仅证明了所提方法的有效性，也为未来的研究提供了有价值的参考和启发。我们相信，这些工作将为 3D 人脸编辑领域的发展做出积极的贡献。