As-Plausible-As-Possible: Plausibility-Aware Mesh Deformation Using 2D Diffusion Priors

Anonymous CVPR submission

Paper ID 15

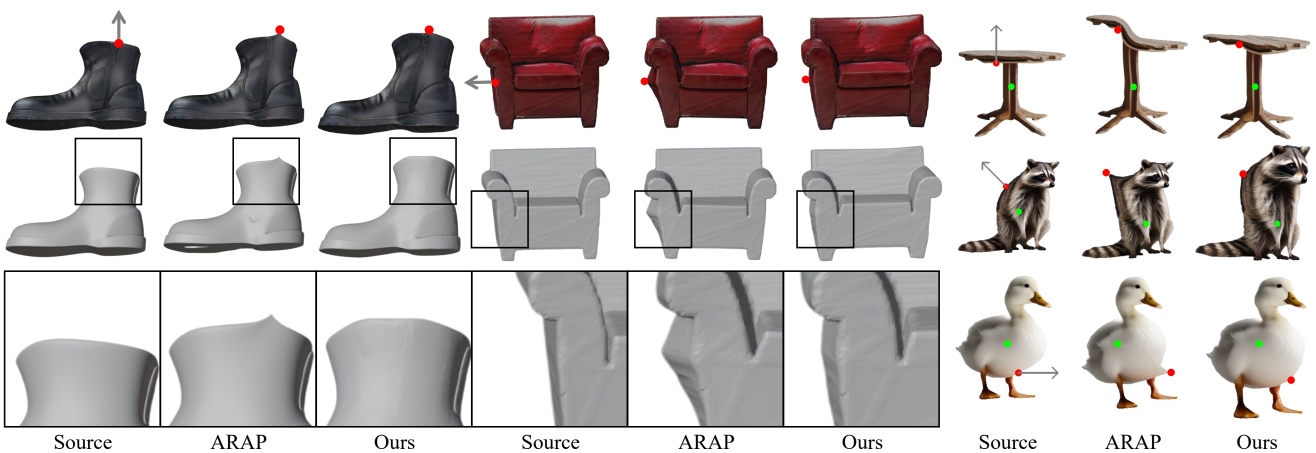

Figure 1. APAP, our novel shape deformation method, enables plausibility-aware mesh deformation and preservation of fine details of the original mesh offering an interface that alters geometry by directly displacing a handle (*red*) along a direction (*gray*). The improvement achieved by leveraging a diffusion prior is illustrated by the smooth geometry near the handle in the armchair example (the middle column).

Abstract

We present As-Plausible-as-Possible (APAP) mesh deformation technique that leverages 2D diffusion priors to preserve the plausibility of a mesh under user-controlled deformation. Our framework uses per-face Jacobians to represent mesh deformations, where mesh vertex coordinates are computed via a differentiable Poisson Solve. The deformed mesh is rendered, and the resulting 2D image is used in the Score Distillation Sampling (SDS) process, which enables extracting meaningful plausibility priors from a pretrained 2D diffusion model. To better preserve the identity of the edited mesh, we fine-tune our 2D diffusion model with LoRA. Gradients extracted by SDS and a user-prescribed handle displacement are then backpropagated to the per-face Jacobians, and we use iterative gradient descent to compute the final deformation that balances between the user edit and the output plausibility. We evaluate our method with 2D and 3D meshes and demonstrate qualitative and quantitative improvements when using plausibility priors over geometry-preservation or distortion-minimization priors used by previous techniques.

1. Introduction

For 2D and 3D content, mesh is the most prevalent representation, thanks to its efficiency in storage, simplicity in rendering and also compatibility in common graphics pipelines, versatility in diverse applications such as design, physical simulation, and 3D printing, and flexibility in terms of decomposing geometry and appearance information, with widespread adoption in the industry.

For the creation of 2D and 3D meshes, recent breakthroughs in generative models [29, 35, 39, 46, 47, 49, 53, 56] have demonstrated significant advances. These breakthroughs enable users to easily generate content from a text prompt [35, 39, 47, 53, 56], or from photos [41, 47]. However, visual content creation typically involves numerous editing processes, deforming the content to satisfy users' desires through interactions such as mouse clicks and drags. Facilitating such interactive editing has remained relatively underexplored in the context of recent generative techniques.

Mesh deformation is a subject that has been researched for decades in computer graphics. Over time, researchers have established well-defined methodologies, characterizing mesh deformation as an optimization problem that aims to preserve specific geometric properties, such as the Mesh Laplacian [32, 33, 51], local rigidity [16, 50], and mesh surface Jacobians [2, 11], while satisfying given constraints. To facilitate user interaction, these methodologies have been extended to introduce specific user-interactive deformation handles, such as keypoints [18, 26, 55], cage mesh [21, 23, 24, 31, 57, 62], and skeleton [4, 60, 61], with the blending functions defined based on the preservation of geometric properties.

Despite the widespread use of classical mesh deformation methods, they often fail to meet users' needs because they do not incorporate the perceptual plausibility of the outputs. For example, as illustrated in Fig. 1, when a user intends to drag a point on the top of a table image, the classical deformation technique may introduce unnatural bending instead of lifting the tabletop. This limitation arises because deformation techniques solely based on geometric properties do not incorporate such semantic and perceptual priors, resulting in the mesh editing process becoming more tedious and time-consuming.

Recent learning-based mesh deformation techniques [2, 21, 26, 34, 52, 60, 62] have attempted to address this problem in a data-driven way. However, they are also limited by relying on the existence of certain variations in the training data. Even recent large-scale 3D datasets [6–8, 59] have not reached the scale that covers all possible visual content users might intend to create.

To this end, we introduce our novel mesh deformation framework, dubbed APAP (As-Plausible-As-Possible), which exploits 2D image priors from a diffusion model pretrained on an Internet-scale image dataset to enhance the plausibility of deformed 2D and 3D meshes while preserving the geometric priors of the given shape. Recently, score distillation sampling (SDS) [39] has demonstrated great success in generating plausible 2D and 3D content, such as NeRF [22, 27, 65] and vector images [17, 20], using the distilled 2D image priors from a diffusion model. We incorporate these diffusion-model-based 2D priors into the optimization-based deformation framework, achieving the best synergy between geometry-based optimization and distilled-prior-based optimization.

To achieve this optimal synergy between geometric and perceptual priors within a unified framework, we introduce an alternative optimization approach. At each step, we first update the Jacobian of each mesh face using the SDS loss and user-provided constraints. Subsequently, the mesh vertex positions are recalculated by solving Poisson's equation with the updated face Jacobians. The direct application of the 2D diffusion prior via SDS, however, tends to compromise the identity of the given objects—an essential aspect in deformation. We thus enhance the identity awareness of the diffusion prior by finetuning it with the provided source image. The model is integrated into our two-stage pipeline that initiates deformation without the perceptual prior (SDS) and refines it with SDS and the given constraints afterward to create deformations that adhere to user-defined editing instructions while remaining visually plausible.

In experiments, we examine **APAP** using APAP-BENCH consisting of 3D and 2D triangular meshes and editing instructions. The proposed method produces plausible deformations of 3D meshes compared to its baseline [50] based exclusively on a geometric prior. Evaluation in the task of 2D mesh editing further verifies the effectiveness of **APAP** as illustrated by the highest k-NN GIQA score [12] in quantitative analysis, and the higher preference over the baseline in a user study.

2. Related Work

2.1. Geometric Mesh Deformation

Mesh deformation has been one of the central problems in geometry processing and is thus addressed by a wide range of techniques. Cage-based methods [23, 24, 31, 57] let users alter meshes by manipulating cages enclosing them, calculating a point inside as a weighted sum of cage vertices. Skeleton-based approaches [4, 58, 60, 61] offer animation control by mapping surface points to underlying joints and bones, ideal for animating human/animal-like figures. Unlike the previous techniques that require the manual cage or skeleton construction, biharmonic coordinates-based methods [18, 55] automate establishing mappings from control points to vertices by formulating optimization problems. Other types of works instead allow users to manipulate shapes via direct vertex displacement while imposing constraints on local surface geometry, including rigidity [16, 50] and Laplacian smoothness [32, 33, 51]. Such hand-crafted deformation priors often lack consideration of visual plausibility, necessitating careful control point placement and iterative manual refinement to achieve satisfactory results.

2.2. Data-Driven Mesh Deformation

Data-driven approaches to mesh deformation [2, 21, 26, 34, 52, 60, 62] learn from shape collections, utilizing neural networks to infer parameters for classical deformation techniques, such as cage vertex coordinates and displacements [62], keypoints [21, 26, 55], subspaces of keypoint arrangements [34], differential coordinates [2], etc. However, these methods assume the availability of large-scale category-specific shape collection [21, 26, 55, 60, 62] or require dense correspondences between them [2, 52], limiting their applicability to new, out-of-sample shapes. We instead propose to directly mine deformation priors from pretrained diffusion models. Leveraging a generic (category-agnostic) image generative model trained on an Internet-scale image dataset, we devise a method that easily generalizes to novel 2D and 3D shapes while lifting the requirement for shape collections.

2.3. Pretrained 2D Priors for Shape Manipulation

Image analysis [40] and generation [3, 30, 43, 63] techniques can serve as effective visual priors for image editing tasks [5, 14, 48, 54, 64]. In addition, recent work [10, 44] and their adaption [9], enable personalized image generation and editing by learning a text embedding [10] or fine-tuning additional parameters, such as LoRA [15] to preserve and replicate the identities of given exemplars during editing. One interesting work is DragDiffusion [48], akin to DragGAN [37], which introduces a drag-based user interface for image editing through the manipulation of latent representations. However, it is not extendable to the deformation of parametric images, such as 2D meshes, and also 3D shapes. Another interesting line of works [11, 25, 36] extends the idea further to manipulate shapes by propagating image-based gradients to the underlying shape representations. They maximize CLIP [40] similarity between the renderings and text prompts to either add geometric textures [36], jointly update both vertices and texture [25], or deform a shape parameterized by per-triangle Jacobians [11]. In contrast to such text-driven editing techniques, we build on Score Distillation Sampling (SDS) [39] to enable direct manipulation of shapes via handle displacement, ensuring visual plausibility. While the technique is prevalent in various problems ranging from text-to-3D [35, 39, 47, 53, 56], image editing [13] and neural field editing [65], it has not been adopted for shape deformation.

3. Method

We present **APAP**, a novel handle-based mesh deformation framework capable of producing visually plausible deformations of either 2D or 3D triangular meshes. To achieve this goal, we integrate powerful 2D diffusion priors into a learnable Jacobian field representation of shapes.

We emphasize that leveraging 2D priors, such as latent diffusion models (LDMs) [43] trained on large-scale datasets [45], for shape deformation poses challenges that require meticulous design choices. The following sections will delve into the details of shape representation (Sec. 3.1) and diffusion prior (Sec. 3.2), offering a rationale for the design decisions underpinning our framework (Sec. 3.3).

3.1. Representing Shapes as Jacobian Fields

Let $\mathcal{M}_0 = (\mathbf{V}_0, \mathbf{F}_0)$ denote a source mesh to be deformed, represented by vertices $\mathbf{V}_0 \in \mathbb{R}^{V \times 3}$ and faces $\mathbf{F}_0 \in \mathbb{R}^{F \times 3}$. Users are allowed to select a set of vertices used as movable handles designated by an indicator matrix $\mathbf{K}_h \in \{0, 1\}^{V_h \times V}$. We also require users to select a set of anchors, represented as another indicator matrix $\mathbf{K}_a \in \{0, 1\}^{V_a \times V}$, to avoid trivial solutions (i.e., global translations). Then, the handle and anchor vertices become $\mathbf{V}_h = \mathbf{K}_h \mathbf{V}_0$ and $\mathbf{V}_a = \mathbf{K}_a \mathbf{V}_0$.

Our framework also expects a set of vectors $\mathbf{D}_h \in \mathbb{R}^{V_h \times 3}$ that indicate the directions along which the handles will be displaced. Furthermore, we let $\mathbf{T}_h = \mathbf{V}_h + \mathbf{D}_h$ and $\mathbf{T}_a = \mathbf{V}_a$ denote the target positions of the user-specified handles and anchors, respectively.

In this work, we employ a Jacobian field $\mathbf{J}_0 = \{\mathbf{J}_{0,f} | f \in \mathbf{F}_0\}$, a dual representation of $\mathcal{M}_0$, defined as a set of per-face Jacobians $\mathbf{J}_{0,f} \in \mathbb{R}^{3 \times 3}$ where

$$\mathbf{J}_{0,f} = \boldsymbol{\nabla}_f \mathbf{V}_0, \tag{1}$$

and $\boldsymbol{\nabla}_f$ is the gradient operator of triangle f.

Conversely, we compute a set of *deformed* vertices $\mathbf{V}^*$ from a given Jacobian field $\mathbf{J}$ by solving a Poisson's equation

$$\mathbf{V}^* = \arg\min_{\mathbf{V}} \|\mathbf{L}\mathbf{V} - \boldsymbol{\nabla}^T \mathcal{A} \mathbf{J}\|^2, \tag{2}$$

where $\boldsymbol{\nabla}$ is a stack of per-face gradient operators, $\mathcal{A} \in \mathbb{R}^{3F \times 3F}$ is the mass matrix and $\mathbf{L} \in \mathbb{R}^{V \times V}$ is the cotangent Laplacian of $\mathcal{M}_0$, respectively. Since $\mathbf{L}$ is rank-deficient, the solution of Eqn. 2 cannot be uniquely determined unless we impose constraints. We thus consider a constrained optimization problem

$$\mathbf{V}^* = \arg\min_{\mathbf{V}} \|\mathbf{L}\mathbf{V} - \boldsymbol{\nabla}^T \mathcal{A} \mathbf{J}\|^2 + \lambda \|\mathbf{K}_a \mathbf{V} - \mathbf{T}_a\|^2, \tag{3}$$

where $\lambda \in \mathbb{R}^+$ is a weight for the constraint term. Note that we solve Eqn. 3 with the user-specified anchors as constraints to determine $\mathbf{V}^*$.

Taking the derivative with respect to $\mathbf{V}$, the problem in Eqn. 3 turns into a system of equations

$$\left(\mathbf{L}^T \mathbf{L} + \lambda \mathbf{K}_a^T \mathbf{K}_a\right) \mathbf{V} = \mathbf{L}^T \boldsymbol{\nabla}^T \mathcal{A} \mathbf{J} + \lambda \mathbf{K}_a^T \mathbf{T}_a, \tag{4}$$

which can be efficiently solved using a differentiable solver [2] implementing Cholesky decomposition.

We let g denote a functional representing the aforementioned differentiable solver for notational convenience, $\mathbf{V}^* = g(\mathbf{J}, \mathbf{K}_a, \mathbf{T}_a)$. Since g is differentiable, we can deform $\mathcal{M}_0$ by propagating upstream gradients from various loss functions to the underlying parameterization $\mathbf{J}$. For instance, one may impose a *soft* constraint on the locations of selected handles during optimization with the objective of the form:

$$\mathcal{L}_h = \|\mathbf{K}_h \mathbf{V}^* - \mathbf{T}_h\|^2. \tag{5}$$

We will discuss how such a soft constraint can be blended into our framework in Sec. 3.3. Next, we describe how to incorporate a pretrained diffusion model as a prior for visual plausibility.

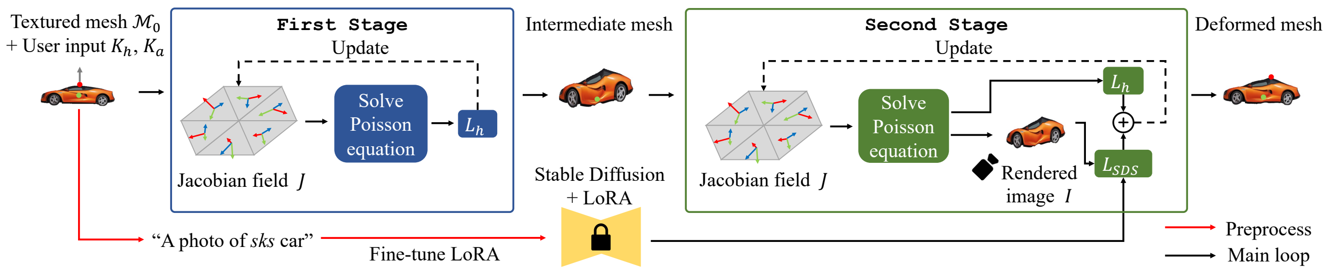

Figure 2. The overview of APAP. APAP parameterizes a triangular mesh as a per-face Jacobian field that can be updated via gradient-descent. Given a textured mesh and user inputs specifying the handle(s) and anchor(s), our framework initializes a Jacobian field as a trainable parameter. During the first stage, the Jacobian field is updated via iterative optimization of $\mathcal{L}_h$, a soft constraint that initially deforms the shape according to the user's instruction. In the following stage, the mesh is rendered using a differentiable renderer $\mathcal{R}$ and the rendered image is provided as an input to a diffusion prior finetuned with LoRA [15] that computes the SDS loss $\mathcal{L}_{\text{SDS}}$. The joint optimization of $\mathcal{L}_h$ and $\mathcal{L}_{\text{SDS}}$ improves the visual plausibility of the mesh while conforming to the given edit instruction.

3.2. Score Distillation for Shape Deformation

While traditional mesh deformation techniques make variations that match the given *geometric* constraints, their lack of consideration on *visual plausibility* results in unrealistic shapes. Motivated by recent success in text-to-3D literature, we harness a powerful 2D diffusion prior [43] in our framework as a critic that directs deformation by scoring the realism of the current shape.

Specifically, we distill its prior knowledge via Score Distillation Sampling (SDS) [39]. Let $\mathbf{J}$ denote the current Jacobian field and $\mathbf{V}^*$ be the set of vertices computed from $\mathbf{J}$ following the procedure described in Sec. 3.1.

We render $\mathcal{M}^* = (\mathbf{V}^*, \mathbf{F})$ from a viewpoint defined by camera extrinsic parameters $\mathbf{C}$ using a differentiable renderer $\mathcal{R}$, producing an image $\mathcal{I} = \mathcal{R}(\mathcal{M}^*, \mathbf{C})$. The diffusion prior $\hat{\epsilon}_\phi$ then rates the realism of $\mathcal{I}$, producing a gradient

$$\nabla_{\mathbf{J}} \mathcal{L}_{\text{SDS}}(\phi, \mathcal{I}) = \mathbb{E}_{t,\epsilon}\left[w(t)\left(\hat{\epsilon}_\phi(\mathbf{z}_t; y, t) - \epsilon\right) \frac{\partial \mathcal{I}}{\partial \mathbf{J}} \right], \quad (6)$$

where $t \sim \mathcal{U}(0,1)$, $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and $\mathbf{z}_t$ is a noisy latent embedding of $\mathcal{I}$. The propagated gradient alters the geometry of $\mathcal{M}$ by modifying $\mathbf{J}$.

To increase the instance-awareness of the diffusion model, we follow recent work [44, 48] on personalized image editing and finetune the model using LoRA [15]. In particular, we first render $\mathcal{M}$ from n different viewpoints to obtain a set $\mathcal{I} = \{\mathcal{I}_1, \ldots, \mathcal{I}_n\}$ of training images and inject additional parameters to the model, resulting in an expanded set of network parameters ϕ'. The parameters are then optimized with a denoising loss [43]

$$\mathcal{L} = \mathbb{E}_{t,\epsilon,\mathbf{z}}\left[\|\hat{\epsilon}_{\phi'}(\mathbf{z}_t; y, t) - \epsilon\|^2 \right], \quad (7)$$

where $\mathbf{z}_t$ denotes a latent of a training image perturbed with noise at timestep t.

The finetuned diffusion prior, together with a learnable Jacobian field representation of the source mesh $\mathcal{M}_0$, comprises the proposed framework described in the following section.

3.3. As-Plausible-As-Possible (APAP)

APAP tackles the problem of plausibility-aware shape deformation by harmonizing the best of both worlds: a learnable shape representation founded on classical geometry processing, robust to noisy gradients, and a powerful 2D diffusion prior finetuned with the image(s) of the source mesh for better instance-awareness.

We provide an overview of the proposed pipeline in Fig. 2 and the algorithm in Alg. 1. We will delve into details in the following. Provided with a textured mesh $\mathcal{M}_0$, handles $\mathbf{K}_h$, anchors $\mathbf{K}_a$, as well as their target positions $\mathbf{T}_h$ and $\mathbf{T}_a$ as inputs, **APAP** yields a plausible deformation $\mathcal{M}$ of $\mathcal{M}_0$ that conforms to the given handle-target constraints. Before deforming $\mathcal{M}_0$, we render $\mathcal{M}_0$ from a single view in the case of 2D meshes and four canonical views (i.e., front, back, left, and right) for 3D meshes and use the images to finetune Stable Diffusion [43] by optimizing LoRA [15] parameters injected to the model (the *red* line in Fig. 2). Simultaneously, **APAP** computes the Jacobian field $\mathbf{J}_0$ of the input mesh $\mathcal{M}_0$ and initializes it as a trainable parameter $\mathbf{J}$.

APAP deforms the input mesh through two stages. In the `FirstStage`, it first deforms the input mesh according to instructions from users without taking visual plausibility into account. The subsequent `SecondStage` integrates a 2D diffusion prior into the optimization loop, simultaneously enforcing user constraints and visual plausibility.

At every iteration of the `FirstStage` illustrated as the *blue* box in Fig. 2, we compute the vertex positions $\mathbf{V}^*$ corresponding to the current Jacobian field $\mathbf{J}$ by solving Eqn. 3 using the anchors specified by $\mathbf{K}_a$ as hard constraints. Then, we compute the soft constraint $\mathcal{L}_h$ defined as Eqn. 5 that drags a set of handle vertices $\mathbf{K}_h\mathbf{V}^*$ toward the corresponding targets $\mathbf{T}_h$. The interleaving of differentiable Poisson solve and optimization of $\mathcal{L}_h$ via gradient-descent

Algorithm 1 As-Plausible-As-Possible

Parameters: g, $\mathcal{R}$, ϕ, γ, M, N
Inputs: $\mathcal{M}_0 = (\mathbf{V}_0, \mathbf{F}_0)$, $\mathbf{K}_a$, $\mathbf{K}_h$, $\mathbf{T}_a$, $\mathbf{T}_h$, $\{\mathbf{C}_i\}_{i=1}^n$
Output: $\mathcal{M}$

procedure FIRSTSTAGE($\mathbf{J}$, $\mathbf{K}_a$, $\mathbf{K}_h$, $\mathbf{T}_a$, $\mathbf{T}_h$, g)
  for $i = 1, 2, \ldots, M$ **do**
    $\mathbf{V}^* \leftarrow g(\mathbf{J}, \mathbf{K}_a, \mathbf{T}_a)$ ▷ Solving Eqn. 4
    $\mathbf{J} \leftarrow \mathbf{J} - \gamma \nabla_{\mathbf{J}} \mathcal{L}_h(\mathbf{V}^*, \mathbf{K}_h, \mathbf{T}_h)$
  end for
  return J
end procedure
procedure SECONDSTAGE($\mathbf{J}$, $\mathbf{F}_0$, $\mathbf{K}_a$, $\mathbf{K}_h$, $\mathbf{T}_a$, $\mathbf{T}_h$, g, ϕ, $\{\mathbf{C}_i\}$)
  for $i = 1, 2, \ldots, N$ **do**
    $\mathbf{V}^* \leftarrow g(\mathbf{J}, \mathbf{K}_a, \mathbf{T}_a)$ ▷ Solving Eqn. 4
    $\mathcal{M}^* \leftarrow (\mathbf{V}^*, \mathbf{F}_0)$
    $\mathbf{C} \sim \mathcal{U}(\{\mathbf{C}_i\})$ ▷ Viewpoint Sampling
    $\mathcal{I} \leftarrow \mathcal{R}(\mathcal{M}^*, \mathbf{C})$ ▷ Rendering
    $\mathbf{J} \leftarrow \mathbf{J} - \gamma \nabla_{\mathbf{J}} (\mathcal{L}_{\text{SDS}}(\phi, \mathcal{I}) + \mathcal{L}_h(\mathbf{V}^*, \mathbf{K}_h, \mathbf{T}_h))$
  end for
  return J
end procedure

$\phi \leftarrow$ LoRA(ϕ, $\mathcal{M}_0$, $\mathcal{R}$, $\{\mathbf{C}_i\}$)
$\mathbf{J} \leftarrow \{\mathbf{J}_{0,f} | f \in \mathbf{F}_0\}$
$\mathbf{J} \leftarrow$ FIRSTSTAGE($\mathbf{J}$, $\mathbf{K}_a$, $\mathbf{K}_h$, $\mathbf{T}_a$, $\mathbf{T}_h$, g)
$\mathbf{J} \leftarrow$ SECONDSTAGE($\mathbf{J}$, $\mathbf{F}_0$, $\mathbf{K}_a$, $\mathbf{K}_h$, $\mathbf{T}_a$, $\mathbf{T}_h$, g, ϕ, $\{\mathbf{C}_i\}$)
$\mathbf{V} \leftarrow g(\mathbf{J}, \mathbf{K}_a, \mathbf{T}_a)$
$\mathcal{M} \leftarrow (\mathbf{V}, \mathbf{F}_0)$
return $\mathcal{M}$

is repeated for M iterations. This progressively updates $\mathbf{J}$, treated as a learnable black box in our framework, deforming $\mathcal{M}_0$. Consequently, the edited mesh $\mathcal{M}^* = (\mathbf{J}, \mathbf{F}_0)$ follows user constraints at the cost of the degraded plausibility, mitigated in the following stage through the incorporation of a diffusion prior.

The result of `FirstStage` then serves as an initialization for the `SecondStage`, illustrated as the *green* box in Fig. 2 guided by plausibility constraint $\mathcal{L}_{\text{SDS}}$. Unlike the `FirstStage` where the update of $\mathbf{J}$ was purely driven by the geometric constraint $\mathcal{L}_h$, we aim to steer the optimization based on the visual plausibility of the current mesh $\mathcal{M}^*$. To achieve this, we render $\mathcal{M}^*$ using a differentiable renderer $\mathcal{R}$ using the same viewpoint(s) from which the training image(s) for finetuning was rendered. When deforming 3D meshes, we randomly sample one viewpoint at each iteration. The rendered image $\mathcal{I}$ is used to evaluate $\mathcal{L}_{\text{SDS}}$ which is optimized jointly with $\mathcal{L}_h$ for N iterations. The combination of geometric and plausibility constraints improves the visual plausibility of the output while encouraging it to conform to the given constraints.

We note that the iterative approach in the `FirstStage` leads to better results than alternative update strategies such as deforming the source mesh $\mathcal{M}_0$ by minimizing ARAP energy [50] or, solving Eqn. 3 using both $\mathbf{K}_h$ and $\mathbf{K}_a$ as hard constraints. In our experiments (Sec. 4), we show that both methods produce distortions that cannot be corrected by the diffusion prior in the subsequent stage. Specifically, directly solving Eqn. 3 using all available constraints only yields the least squares solution $\mathbf{V}^*$ without updating the underlying Jacobians $\mathbf{J}$, resulting in the aforementioned distortions.

4. Experiments

We evaluate **APAP** in downstream applications involving manipulation of 3D and 2D meshes.

4.1. Experiment Setup

Benchmark. To evaluate the plausibility of a mesh deformation we propose a novel benchmark APAP-BENCH of textured 3D and 2D triangular meshes spanning both human-made and organic objects annotated with handle vertices and their editing directions, and anchor vertices. The set of 3D meshes, APAP-BENCH 3D, is constructed using meshes from ShapeNet [6] and *Genie* [1]. The meshes are normalized to fit in a unit cube. Each mesh is manually annotated with editing instructions, including a set of anchors, handles, and corresponding targets to simulate editing scenarios. APAP-BENCH offers another subset called APAP-BENCH 2D, a collection of 80 textured, planar meshes of various objects, to facilitate quantitative analysis and user study described later in this section. To create APAP-BENCH 2D, we first generate 2 images of real-world objects for each of the 20 categories using Stable Diffusion-XL [38]. We then extract foreground masks from the generated images using SAM [28] and sample pixels that lie on the boundary and interior. The sampled pixels are used for Delaunay triangulation, constrained with the edges along the main contour of the masks, that produces 2D triangular meshes with texture. We assign two handle and anchor pairs to each mesh that imitate user instructions. For evaluation purposes, we populate the reference set by sampling $1,000$ images for each object category using Stable Diffusion-XL. The generated images are used to evaluate a perceptual metric to assess the plausibility of 2D mesh editing results as described in Sec. 4.3.

Baselines. We compare our method (**APAP**) and As-Rigid-As-Possible (ARAP) [50] since it is one of the widely used mesh deformation techniques that permits shape manipulation via direct vertex displacement. Throughout the

Figure 3. Qualitative results from 3D shape deformation. We visualize the source shapes and their deformations made using ARAP [50] and ours by following the instructions each of which specifies a handle (*red*), an edit direction denoted with an arrow (*gray*), and an anchor (*green*). We showcase the rendered images captured from two different viewpoints, as well as one zoom-in view highlighting local details.

experiments, we use the implementation in `libigl` [19] with default parameters.

Evaluation Metrics. In 2D experiments, we conduct quantitative analysis based on k-NN GIQA score [12] as an evaluation metric to assess the plausibility of instance-specific editing results. The metric quantifies the perceptual proximity between the edited image and its k nearest neighbors in the reference set included in APAP-BENCH 2D. As our objective is to make plausible variations of 2D meshes via deformation, an edited object should remain perceptually similar to other objects in the same category. We use $k = 12$ throughout the experiments.

4.2. 3D Shape Deformation

Qualitative Results. We showcase examples of 3D shape deformation where each deformation is specified by a han-

Figure 4. Failure cases of DragDiffusion. DragDiffusion [48] can easily compromise the identity of edited instances as it manipulates their latents without an explicit parameterization, the identity of instances can be broken during editing.

Methods	k-NN GIQA ($\times 10^{-2}$) ↑
ARAP [50]	4.753
DragDiffusion [48]	4.545
Ours ($\mathcal{L}_h$ Only)	4.797
Ours (ARAP Init.)	4.740
Ours (Poisson Init.)	4.316
Ours	**4.887**

Table 1. **Quantitative analysis for 2D mesh editing.** APAP outperforms its baselines in quantitative evaluation using k-NN GIQA [12].

Methods	Preference (%) ↑
ARAP [50]	40.83
Ours	**59.17**

Table 2. **User study preference for 2D image editing.** In a user study targeting users on Amazon Mechanical Turk (MTurk), the results produced using ours were preferred over the outputs from the baseline.

dle (*red*), an edit direction (*gray*), and an anchor (*green*). As shown in Fig. 3, **APAP** is capable of manipulating 3D shapes to improve visual plausibility which is not achievable by solely relying on geometric prior such as ARAP [50]. For instance, given a user input that drags a handle on one blade of an axe (the first row) along an arrow, **APAP** simultaneously expands both blades of the axe whereas ARAP [50] produces distortions near the head. Similar examples that demonstrate symmetry-awareness of **APAP** can be found in other cases such as a car (the second row), and an owl (the sixth row) where a user lifts only one side of the shape upward and the symmetry is recovered by **APAP** which cannot be achieved by ARAP [50]. Also, note that **APAP** is capable of making a smooth articulation at the leg of the wolf (the fourth row) by adjusting the overall posture in comparison to ARAP which creates an excess bending.

4.3. 2D Mesh Editing

Qualitative Evaluation. We present qualitative results using the baselines and our method in Fig. 5. Each row shows two different results obtained by editing an image based on a handle moved from the original position (*red*) along a direction indicated by an arrow (*gray*) while fixing an anchor (*green*), similar to the 3D experiments discussed in the previous section.

As shown in Fig. 5, ARAP [50] enforces local rigidity and often results in implausible deformations. For example, it does not account for the mechanics of the human body and introduces an unrealistic articulation of a human arm (the fourth row). In addition, it twists the body of a sports car (the fifth row). Both of them originate from the lack of understanding of the appearance of objects. **APAP** alleviates this issue by incorporating a visual prior into shape deformation producing a bending near the elbow and preserving the smooth silhouette of the car, respectively.

While **APAP** is designed for meshes not images, we provide an additional qualitative comparison against DragDiffusion [48], an image editing technique that operates in pixel space, to demonstrate the effectiveness of mesh-based parameterization in applications where identity preservation is crucial. As shown in Fig. 4, DragDiffusion [48] may corrupt the identity of the instances depicted in input images during the encoding and decoding procedure. **APAP**, on the other hand, makes plausible variations of the given objects while maintaining their originality, benefiting from an explicit mesh representation it is grounded.

Quantitative Evaluation. Tab. 1 summarizes k-NN GIQA scores measured on the outputs from ARAP [50] (the first row) and **APAP** (the sixth row) using APAP-Bench 2D. As shown, **APAP** demonstrates superior performance over ARAP [50]. This again verifies the observations from qualitative evaluation where ARAP [50] introduces distortions that harm visual plausibility. As in qualitative evaluation, we also report the k-NN GIQA score of DragDiffusion [48], degraded due to artifacts caused during direct manipulation of latents.

User Study. We further conduct a user study for a more precise perceptual analysis. We follow Ritchie [42] and recruit participants on Amazon Mechanical Turk (MTurk). Each participant is provided with a set of 20 randomly sampled images of the source meshes paired with editing results of ARAP [50] and **APAP**. To check whether the response from a participant is reliable we present 5 vigilance tests and collect 102 responses from the participants who passed

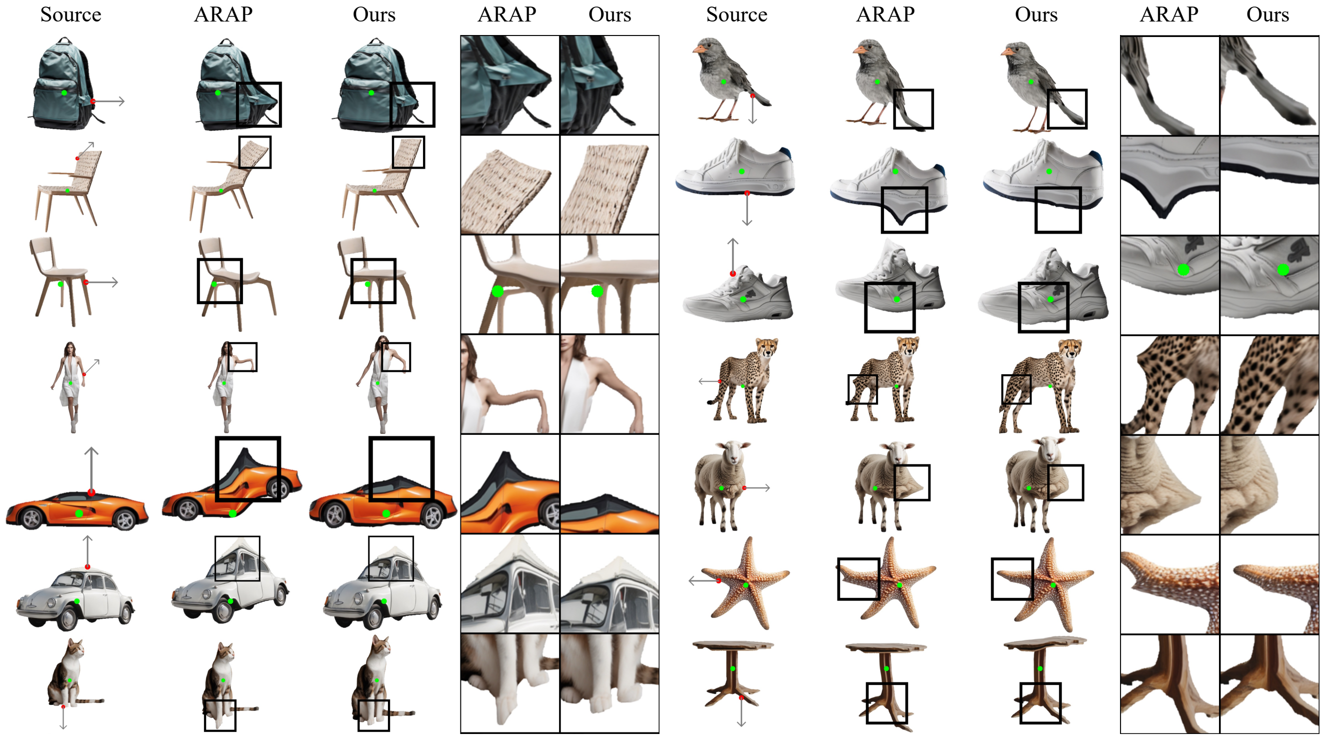

Figure 5. Qualitative results from 2D mesh deformation. 2D meshes are edited using ARAP [50] and the proposed method following the edit instruction consisting of a handle (*red*), a target direction (*gray*), and an anchor (*green*). We showcase the rendered images of the edited meshes, as well as a zoom-in view highlighting local details.

the vigilance test.

We instructed participants to select the most anticipated outcome when the displayed source image is edited by the dragging operation visualized as an arrow. We have provided detailed settings and examples of the user study environment and statistical methods in the **supplementary material**. Tab. 2 shows a higher preference of the participants on our method over ARAP [50] implying that our method produces more visually plausible deformations by utilizing a visual prior.

Ablation Study. Tab. 1 summarizes the impact of different initialization strategies in the first stage on k-NN GIQA score. As reported in the third row of the table, optimizing $\mathcal{L}_h$ that aims to exclusively satisfy geometric constraints leads to unnatural distortions. We provide a qualitative comparison in the the **supplementary material**.

While designing the algorithm illustrated in Alg. 1, we considered other options for `FirstStage`. Instead of optimizing $\mathcal{L}_h$ to initially deform a shape, we used a shape produced by ARAP [50] or by solving a Poisson's equation constrained not only on anchor positions but also on handles at their target positions reached by following the given edit directions. We report k-NN GIQA scores of the alternatives in the fourth and fifth row of Tab. 1, respectively. Both initialization strategies degrade the plausibility of results due to large distortions introduced by either solely enforcing local rigidity or, finding least square solutions without updating Jacobians. This poses a challenge to the diffusion prior, making it struggle to induce meaningful update directions when provided with renderings with noticeable distortions, which can be found in qualitative analysis in the **supplementary material**.

5. Conclusion

We presented **APAP**, a novel deformation framework that tackles the problem of plausibility-aware shape deformation while offering intuitive controls over a wide range of shapes represented as triangular meshes. To this end, we carefully orchestrate two core components, a learnable Jacobian-based parameterization that originates from geometry processing and powerful 2D priors acquired by text-to-image diffusion models trained on Internet-scale datasets. We assessed the performance of the proposed method against an existing geometric-prior-based deformation technique and also thoroughly investigated the significance of our design choices through experiments.

References

[1] Luma AI. Genie. 5

[2] Noam Aigerman, Kunal Gupta, Vladimir G Kim, Siddhartha Chaudhuri, Jun Saito, and Thibault Groueix. Neural Jacobian Fields: Learning Intrinsic Mappings of Arbitrary Meshes. *ACM TOG*, 2022. 2, 3

[3] Omer Bar-Tal, Lior Yariv, Yaron Lipman, and Tali Dekel. MultiDiffusion: Fusing Diffusion Paths for Controlled Image Generation. In *ICML*, 2023. 3

[4] Ilya Baran and Jovan Popović. Automatic Rigging and Animation of 3D Characters. *ACM TOG*, 2007. 2

[5] Mingdeng Cao, Xintao Wang, Zhongang Qi, Ying Shan, Xiaohu Qie, and Yinqiang Zheng. MasaCtrl: Tuning-Free Mutual Self-Attention Control for Consistent Image Synthesis and Editing. In *ICCV*, 2023. 3

[6] 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. 2, 5

[7] Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, Eli VanderBilt, Aniruddha Kembhavi, Carl Vondrick, Georgia Gkioxari, Kiana Ehsani, Ludwig Schmidt, and Ali Farhadi. Objaverse-XL: A Universe of 10M+ 3D Objects. In *NeurIPS*, 2023.

[8] 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 *CVPR*, 2023. 2

[9] Hugging Face. DreamBooth fine-tuning with LoRA. 3

[10] 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. In *ICLR*, 2023. 3

[11] William Gao, Noam Aigerman, Groueix Thibault, Vladimir Kim, and Rana Hanocka. TextDeformer: Geometry Manipulation using Text Guidance. *ACM TOG*, 2023. 2, 3

[12] Shuyang Gu, Jianmin Bao, Dong Chen, and Fang Wen. GIQA: Generated Image Quality Assessment. In *ECCV*, 2020. 2, 6, 7

[13] Amir Hertz, Kfir Aberman, and Daniel Cohen-Or. Delta Denoising Score. In *ICCV*, 2023. 3

[14] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-Prompt Image Editing with Cross-Attention Control. In *ICLR*, 2023. 3

[15] Yelong Hu, Edward J.and Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models. In *ICLR*, 2022. 3, 4

[16] Takeo Igarashi, Tomer Moscovich, and John F. Hughes. As-Rigid-as-Possible Shape Manipulation. *ACM TOG*, 2005. 2

[17] Shir Iluz, Yael Vinker, Amir Hertz, Daniel Berio, Daniel Cohen-Or, and Ariel Shamir. Word-As-Image for Semantic Typography. *ACM TOG*, 2023. 2

[18] Alec Jacobson, Ilya Baran, Jovan Popović, and Olga Sorkine. Bounded Biharmonic Weights for Real-Time Deformation. *ACM TOG*, 2011. 2

[19] Alec Jacobson, Daniele Panozzo, et al. libigl: A simple C++ geometry processing library, 2018. 6

[20] Ajay Jain, Amber Xie, and Pieter Abbeel. VectorFusion: Text-to-SVG by Abstracting Pixel-Based Diffusion Models. In *CVPR*, 2023. 2

[21] Tomas Jakab, Richard Tucker, Ameesh Makadia, Jiajun Wu, Noah Snavely, and Angjoo Kanazawa. KeypointDeformer: Unsupervised 3D Keypoint Discovery for Shape Control. In *CVPR*, 2020. 2

[22] Jong Chul Ye Jangho Park, Gihyun Kwon. ED-NeRF: Efficient Text-Guided Editing of 3D Scene using Latent Space NeRF. *arXiv*, 2023. 2

[23] Pushkar Joshi, Mark Meyer, Tony DeRose, Brian Green, and Tom Sanocki. Harmonic Coordinates for Character Articulation. *ACM TOG*, 2007. 2

[24] Tao Ju, Scott Schaefer, and Joe Warren. Mean Value Coordinates for Closed Triangular Meshes. *ACM TOG*, 2005. 2

[25] Nasir Mohammad Khalid, Tianhao Xie, Eugene Belilovsky, and Popa Tiberiu. CLIP-Mesh: Generating Textured Meshes from Text Using Pretrained Image-Text Models. *SIGGRAPH ASIA*, 2022. 3

[26] Kunho Kim, Mikaela Angelina Uy, Despoina Paschalidou, Alec Jacobson, Leonidas J. Guibas, and Minhyuk Sung. OptCtrlPoints: Finding the Optimal Control Points for Biharmonic 3D Shape Deformation. *Computer Graphics Forum*, 2023. 2

[27] Subin Kim, Kyungmin Lee, June Suk Choi, Jongheon Jeong, Kihyuk Sohn, and Jinwoo Shin. Collaborative Score Distillation for Consistent Visual Synthesis. In *NeurIPS*, 2023. 2

[28] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment Anything. In *ICCV*, 2023. 5

[29] Juil Koo, Seungwoo Yoo, Minh Hieu Nguyen, and Minhyuk Sung. SALAD: Part-Level Latent Diffusion for 3D Shape Generation and Manipulation. In *ICCV*, 2023. 1

[30] Yuseung Lee, Kunho Kim, Hyunjin Kim, and Minhyuk Sung. SyncDiffusion: Coherent Montage via Synchronized Joint Diffusions. In *NeurIPS*, 2023. 3

[31] Yaron Lipman, David Levin, and Daniel Cohen-Or. Green Coordinates. *ACM TOG*, 2008. 2

[32] Yaron Lipman, Olga Sorkine, Daniel Cohen-Or, David Levin, Christian Rössl, and Hans-Peter Seidel. Differential Coordinates for Interactive Mesh Editing. In *Proceedings of Shape Modeling International*, pages 181–190, 2004. 2

[33] Yaron Lipman, Olga Sorkine, David Levin, and Daniel Cohen-Or. Linear Rotation-Invariant Coordinates for Meshes. *ACM TOG*, 2005. 2

[34] Minghua Liu, Minhyuk Sung, Radomir Mech, and Hao Su. DeepMetaHandles: Learning Deformation Meta-Handles of 3D Meshes with Biharmonic Coordinates. In *CVPR*, 2021. 2

[35] Yuan Liu, Cheng Lin, Zijiao Zeng, Xiaoxiao Long, Lingjie Liu, Taku Komura, and Wenping Wang. SyncDreamer: Learning to Generate Multiview-consistent Images from a Single-view Image. *arXiv*, 2023. 1, 3

[36] Oscar Michel, Roi Bar-On, Richard Liu, Sagie Benaim, and Rana Hanocka. Text2Mesh: Text-Driven Neural Stylization for Meshes. In *CVPR*, 2022. 3

[37] Xingang Pan, Ayush Tewari, Thomas Leimkühler, Lingjie

Liu, Abhimitra Meka, and Christian Theobalt. Drag Your GAN: Interactive Point-based Manipulation on the Generative Image Manifold. *ACM TOG*, 2023. 3

[38] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: Improving Latent Diffusion Models for High-Resolution Image Synthesis. *arXiv*, 2023. 5

[39] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. DreamFusion: Text-to-3D using 2D Diffusion. In *ICLR*, 2023. 1, 2, 3, 4

[40] 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. 3

[41] Amit Raj, Srinivas Kaza, Ben Poole, Michael Niemeyer, Ben Mildenhall, Nataniel Ruiz, Shiran Zada, Kfir Aberman, Michael Rubenstein, Jonathan Barron, Yuanzhen Li, and Varun Jampani. DreamBooth3D: Subject-Driven Text-to-3D Generation. In *ICCV*, 2023. 1

[42] Daniel Ritchie. Rudimentary framework for running two-alternative forced choice (2afc) perceptual studies on mechanical turk. 7

[43] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-Resolution Image Synthesis with Latent Diffusion Models. In *CVPR*, 2022. 3, 4

[44] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. DreamBooth: Fine Tuning Text-to-image Diffusion Models for Subject-Driven Generation. In *CVPR*, 2023. 3, 4

[45] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. LAION-5B: An open large-scale dataset for training next generation image-text models. In *NeurIPS*, 2022. 3

[46] Tianchang Shen, Jun Gao, Kangxue Yin, Ming-Yu Liu, and Sanja Fidler. Deep Marching Tetrahedra: a Hybrid Representation for High-Resolution 3D Shape Synthesis. In *NeurIPS*, 2021. 1

[47] Yichun Shi, Peng Wang, Jianglong Ye, Long Mai, Kejie Li, and Xiao Yang. MVDream: Multi-view Diffusion for 3D Generation. *arXiv*, 2023. 1, 3

[48] Yujun Shi, Chuhui Xue, Jiachun Pan, Wenqing Zhang, Vincent YF Tan, and Song Bai. DragDiffusion: Harnessing Diffusion Models for Interactive Point-based Image Editing. *arXiv*, 2023. 3, 4, 7

[49] J. Ryan Shue, Eric Ryan Chan, Ryan Po, Zachary Ankner, Jiajun Wu, and Gordon Wetzstein. 3D Neural Field Generation using Triplane Diffusion. In *CVPR*, 2023. 1

[50] Olga Sorkine and Marc Alexa. As-Rigid-As-Possible Surface Modeling. *Proceedings of EUROGRAPHICS/ACM SIGGRAPH Symposium on Geometry Processing*, pages 109–116, 2007. 2, 5, 6, 7, 8

[51] Olga Sorkine, Daniel Cohen-Or, Yaron Lipman, Marc Alexa, Christian Rössl, and Hans-Peter Seidel. Laplacian Surface Editing. *Proceedings of the EUROGRAPHICS/ACM SIGGRAPH Symposium on Geometry Processing*, pages 179–188, 2004. 2

[52] Jiapeng Tang, Markhasin Lev, Wang Bi, Thies Justus, and Matthias Nießner. Neural Shape Deformation Priors. In *NeurIPS*, 2022. 2

[53] Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. DreamGaussian: Generative Gaussian Splatting for Efficient 3D Content Creation. *arXiv*, 2023. 1, 3

[54] Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-Play Diffusion Features for Text-Driven Image-to-Image Translation. In *CVPR*, 2023. 3

[55] Yu Wang, Alec Jacobson, Jernej Barbič, and Ladislav Kavan. Linear Subspace Design for Real-Time Shape Deformation. *ACM TOG*, 2015. 2

[56] Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. In *NeurIPS*, 2023. 1, 3

[57] Ofir Weber, Mirela Ben-Chen, and Craig Gotsman. Complex barycentric coordinates with applications to planar shape deformation. *Computer Graphics Forum*, 2009. 2

[58] Ofir Weber, Olga Sorkine, Yaron Lipman, and Craig Gotsman. Context-Aware Skeletal Shape Deformation. *Computer Graphics Forum*, 2007. 2

[59] Tong Wu, Jiarui Zhang, Xiao Fu, Yuxin Wang, Jiawei Ren, Liang Pan, Wayne Wu, Lei Yang, Jiaqi Wang, Chen Qian, Dahua Lin, and Ziwei Liu. OmniObject3D: Large-Vocabulary 3D Object Dataset for Realistic Perception, Reconstruction and Generation. In *CVPR*, 2023. 2

[60] Zhan Xu, Yang Zhou, Evangelos Kalogerakis, Chris Landreth, and Karan Singh. RigNet: Neural Rigging for Articulated Characters. *ACM TOG*, 2020. 2

[61] Zhan Xu, Yang Zhou, Li Yi, and Evangelos Kalogerakis. Morig: Motion-Aware Rigging of Character Meshes from Point Clouds. In *SIGGRAPH ASIA*, 2022. 2

[62] Wang Yifan, Noam Aigerman, Vladimir G. Kim, Chaudhuri Siddhartha, and Olga Sorkine. Neural Cages for Detail-Preserving 3D Deformations. In *CVPR*, 2020. 2

[63] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding Conditional Control to Text-to-Image Diffusion Models. In *ICCV*, 2023. 3

[64] Yuechen Zhang, Jinbo Xing, Eric Lo, and Jiaya Jia. Real-World Image Variation by Aligning Diffusion Inversion Chain. In *NeurIPS*, 2023. 3

[65] Jingyu Zhuang, Chen Wang, Lingjie Liu, Liang Lin, and Guanbin Li. DreamEditor: Text-Driven 3D Scene Editing with Neural Fields. *ACM TOG*, 2023. 2, 3