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

Seungwoo Yoo[*1] Kunho Kim[*1] Vladimir G. Kim[2] Minhyuk Sung[1]
[1]KAIST [2]Adobe Research

In this supplementary material, we first illustrate the full algorithm (Sec. S1), followed by detailed descriptions of implementation of our main pipeline (Sec. S2), details of experiment setups (Sec. S3), and APAP-BENCH construction (Sec. S4). We also provide experiment results from 2D mesh deformation, including qualitative and quantitative analysis and user study (Sec. S5). Furthermore, we show the full list of qualitative results for 3D shape deformation (Sec. S6), as well as more complex 3D shape deformations achieved by leveraging classical deformation techniques (Sec. S7). Finally, we report human evaluation results on the plausibility of 3D deformations via user study (Sec. S8).

S1. APAP Algorithm

We present the full pseudo-code of the proposed algorithm in Alg. 1. As noted in the main paper, the proposed algorithm consists of two stage where we first fulfill geometric, handle constraints imposed by users and refine the intermediate results by distilling knowledge of visual plausibility from a pretrained text-to-image diffusion model.

S2. Implementation Details

We provide additional implementation details of Alg. 1. We used a modified version of the differentiable Poisson solver from [2], denoted by g in Alg. 1, and `nvdiffrast` [11] when implementing the differentiable renderer $\mathcal{R}$ in our pipeline. We render 2D/3D meshes at a resolution of 512×512.

When editing 2D meshes, we optimize $\mathcal{L}_h$ for $M = 300$ iterations in the `FirstStage` and jointly optimize $\mathcal{L}_h$ and $\mathcal{L}_{\text{SDS}}$ for $N = 700$ iterations in the `SecondStage`. For experiments involving the optimization of 3D meshes with increased geometric complexity, we use $M = 300$ and $N = 1000$ for each stage, respectively. We use ADAM [9] with a learning rate $\gamma = 1 \times 10^{-3}$ throughout the optimization. We use the Classifier-Free Guidance (CFG) scale of 100.0 and randomly sample $t \in [0.02, 0.98]$ when evaluating $\mathcal{L}_{\text{SDS}}$ following DreamFusion [13].

We use a script from *diffusers* [5] to finetune Stable Diffusion [15] with LoRA [7]. We employ `stabilityai/stable-diffusion-2-1-base` as our base model and augment its cross-attention layers in the U-Net with rank decomposition matrices of rank 16. For the task of 2D mesh editing, we train the injected parameters for 60 iterations, utilizing a rendering of a mesh as a training image. In the 3D shape deformation, where renderings from 4 canonical viewpoints (front, back, left, and right) are available, we finetune the model for 200 iterations. In both cases, we use the learning rate $\gamma = 5 \times 10^{-4}$.

S3. 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 [3] 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 [12]. We then extract foreground masks from the generated images using SAM [10] 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

Algorithm 1 As-Plausible-As-Possible

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

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

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.

S4. Details of APAP-BENCH

Image Generation. For evaluation purposes, we build APAP-BENCH 2D by generating 2 images of real-world objects for each of the 20 categories using Stable Diffusion-XL [12]. We segment the foreground objects from the generated images and run Delaunay triangulation to populate a collection of 2D meshes. When generating the images, we use the following template prompt "`a photo of [category name] in a white background`" for all categories to facilitate foreground object segmentation. Tab. S1 summarizes the list of categories. Note that the list includes both human-made and organic objects that can be easily found in the daily environment to test the generalization capability of a deformation technique to various object types.

Handle and Anchor Assignment. We manually assign two handle and anchor pairs to each mesh to imitate user instructions. Specifically, we choose vertices on the shape boundaries instead of internal vertices to induce deformations that alter object silhouettes. For instance, users would try to drag the bottom of a backpack downward to enlarge the shape, instead of dragging an interior point which may flip triangles, distorting the appearance. As an anchor, we use the vertex closest to the center of mass of each mesh.

In experiments using APAP-BENCH 3D and APAP-BENCH 2D, we note that utilization of neighboring vertices of the given handles and anchors during deformation helps retain smooth geometry near the handle. Therefore, we additionally sample vertices near the handles and anchors that lie in the sphere of radius $r = 0.01$ and denote the extended sets of handles

Human-Made	Organic
backpack	flying bird
bike	side view of cat
chair	side view of dog
high-heeled shoes	runway model
purse	sitting bird
side view of car	standing cheetah
sneakers	standing dragon
table	standing raccoon
airplane	standing sheep
	standing white duck
	starfish

Table S1. **Object categories of 2D meshes in APAP-BENCH 2D.** APAP-BENCH 2D includes 2D triangle meshes depicting various objects, including both human-made and organic objects.

and anchors *region handles* and *region anchors*, respectively. We use region anchors and a single handle for 3D experiments and region anchors and region handles for 2D cases. Note that we use the same sets of handles and anchors when deforming shapes with our baselines for fair comparisons.

Baselines. We compare our method (**APAP**) and As-Rigid-As-Possible (ARAP) [17] since it is one of the widely used mesh deformation techniques that permits shape manipulation via direct vertex displacement. Throughout the experiments, we use the implementation in `libigl` [8] with default parameters.

Evaluation Metrics. In 2D experiments, we conduct quantitative analysis based on k-NN GIQA score [6] 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.

S5. 2D Mesh Editing

Qualitative Evaluation. We present qualitative results using the baselines and our method in Fig. S1. 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. S1, ARAP [17] 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 [16], 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. S2, DragDiffusion [16] 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. S2 summarizes k-NN GIQA scores measured on the outputs from ARAP [17] (the first row) and **APAP** (the sixth row) using APAP-BENCH 2D. As shown, **APAP** demonstrates superior performance over ARAP [17]. This again verifies the observations from qualitative evaluation where ARAP [17] introduces distortions that harm visual plausibility. As in qualitative evaluation, we also report the k-NN GIQA score of DragDiffusion [16], 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 [14] 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 [17] and **APAP**.

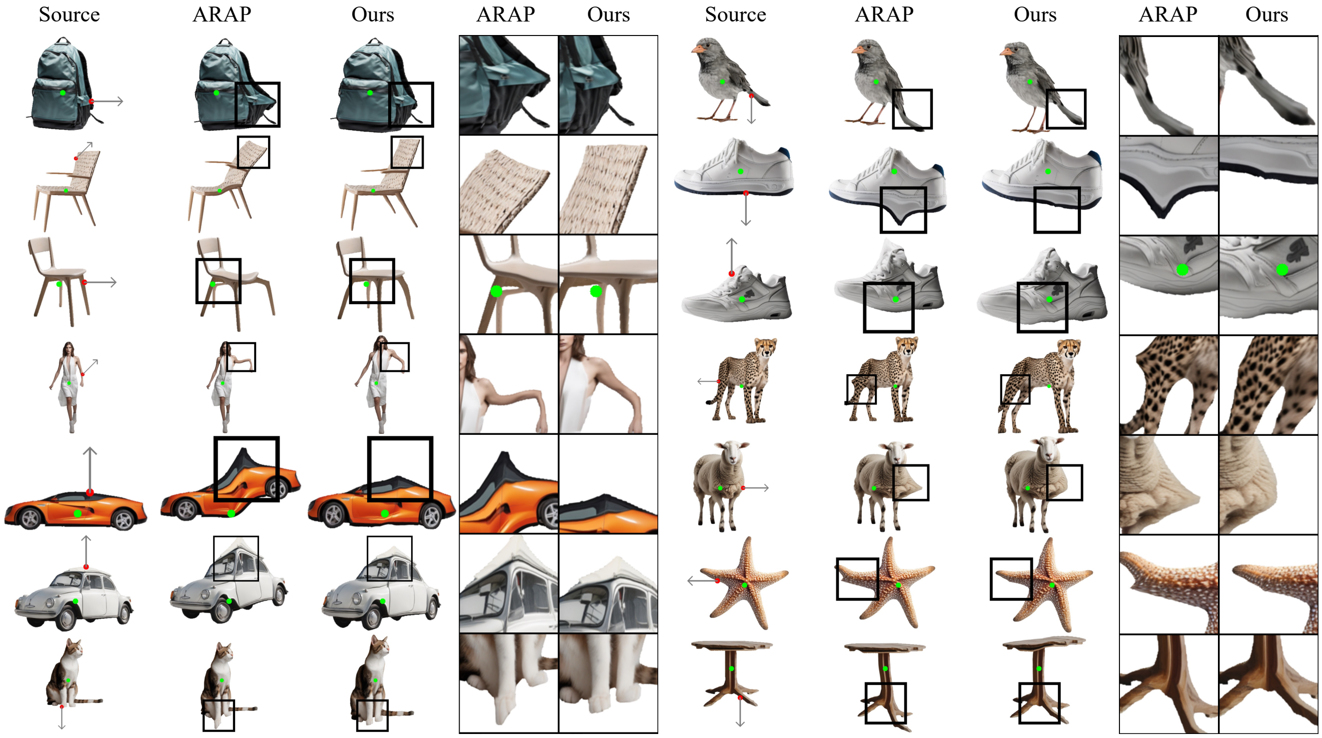


Figure S1. Qualitative results from 2D mesh deformation. 2D meshes are edited using ARAP [17] and the proposed method following the edit instruction consisting of a handle (*red*), a target direction (*gray*), and an anchor (*green*). We showcase the rendered images of the edited meshes, as well as a zoom-in view highlighting local details.

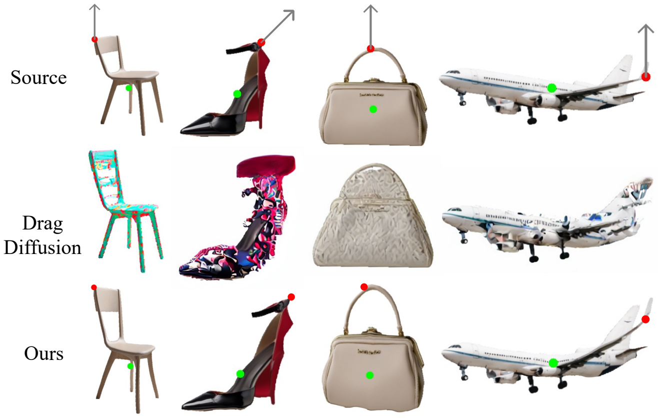


Figure S2. Failure cases of DragDiffusion. DragDiffusion [16] can easily compromise the identity of edited instances as it manipulates their latents without an explicit parameterization, the identity of instances can be broken during editing.

We instructed participants to select the most anticipated outcome when the displayed source image is edited by the dragging operation visualized as an arrow with the question: `"A visual designer wants to modify the object by clicking on a red point and dragging it in the direction of the arrow. Please choose a result that best satisfies the designer's edit, while retaining the characteristics and plausibility of the object."` To check whether the response from a participant is reliable we present 5 vigilance tests and collect 102 responses from the participants who passed the vigilance test. After collecting responses from the participants, we computed the preference statistics collected from 102 user study participants who passed the vigilance tests.

Fig. S3 (left) shows an example of a questionnaire provided to the participants. For vigilance tests, we included an editing

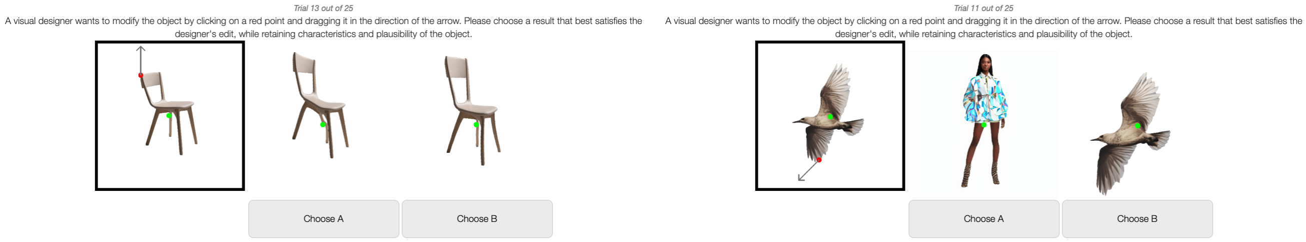
Methods	k-NN GIQA ($\times 10^{-2}$) ↑
ARAP [17]	4.753
DragDiffusion [16]	4.545
Ours ($\mathcal{L}_h$ Only)	4.797
Ours (ARAP Init.)	4.740
Ours (Poisson Init.)	4.316
Ours	**4.887**

Table S2. **Quantitative analysis for 2D mesh editing.** APAP outperforms its baselines in quantitative evaluation using k-NN GIQA [6].

Methods	Preference (%) ↑
ARAP [17]	40.83
Ours	**59.17**

Table S3. **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.

result from DragDiffusion [16] depicting an object irrelevant to the source image in each question. The participants were asked to answer the same question. We illustrate an example questionnaire of a vigilance test in Fig. S3 (right).



Trial 13 out of 25

A visual designer wants to modify the object by clicking on a red point and dragging it in the direction of the arrow. Please choose a result that best satisfies the designer's edit, while retaining characteristics and plausibility of the object.

Trial 11 out of 25

A visual designer wants to modify the object by clicking on a red point and dragging it in the direction of the arrow. Please choose a result that best satisfies the designer's edit, while retaining characteristics and plausibility of the object.

Figure S3. Examples of questionnaires displayed during the user study (2D mesh editing). During the user study, we asked the participants to evaluate 20 different result pairs from ARAP [17] and ours as shown on the left. To check whether a participant is focusing on the user study, we included 5 items for the vigilance test. As shown on the right, a question for the vigilance test includes an image of an object that is not related to the source image.

Tab. S3 shows a higher preference of the participants on our method over ARAP [17] implying that our method produces more visually plausible deformations.

Ablation Study. Tab. S2 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 Fig. S4. The presented results are obtained by (1) optimizing only $\mathcal{L}_h$, (2) $\mathcal{L}_h$ and $\mathcal{L}_{\text{SDS}}$ without LoRA finetuning, (3) skipping the `FirstStage`, (4) using ARAP initialization, (5) using Poisson initialization, and (6) Ours. As shown in Fig. S4, optimizing only $\mathcal{L}_h$ (the second column) either distorts texture (the fifth row) or inflates or shrinks other parts of the given shape (the seventh and twelfth row). This demonstrates the necessity of a visual prior during deformation. Also, we observe the cases where skipping the `FirstStage` (the fourth column) does not lead to intended deformation as our diffusion prior is reluctant to modify shapes from their original states (the first, second, and fifth row). On the other hand, deformations initialized with the meshes produced by ARAP [17] (the fifth column) or Poisson solve (the sixth column) suffer from distortions that could not be resolved by optimizing $\mathcal{L}_{\text{SDS}}$ in the `SecondStage`.

Figure S4. Ablation study for 2D mesh editing. We examine the impact of each design choice on deformation outputs, including the use of diffusion prior (the second column), LoRA finetuning (the third column), two-stage pipeline (the fourth column), and initialization strategies during the `FirstStage` (the fifth and sixth column).

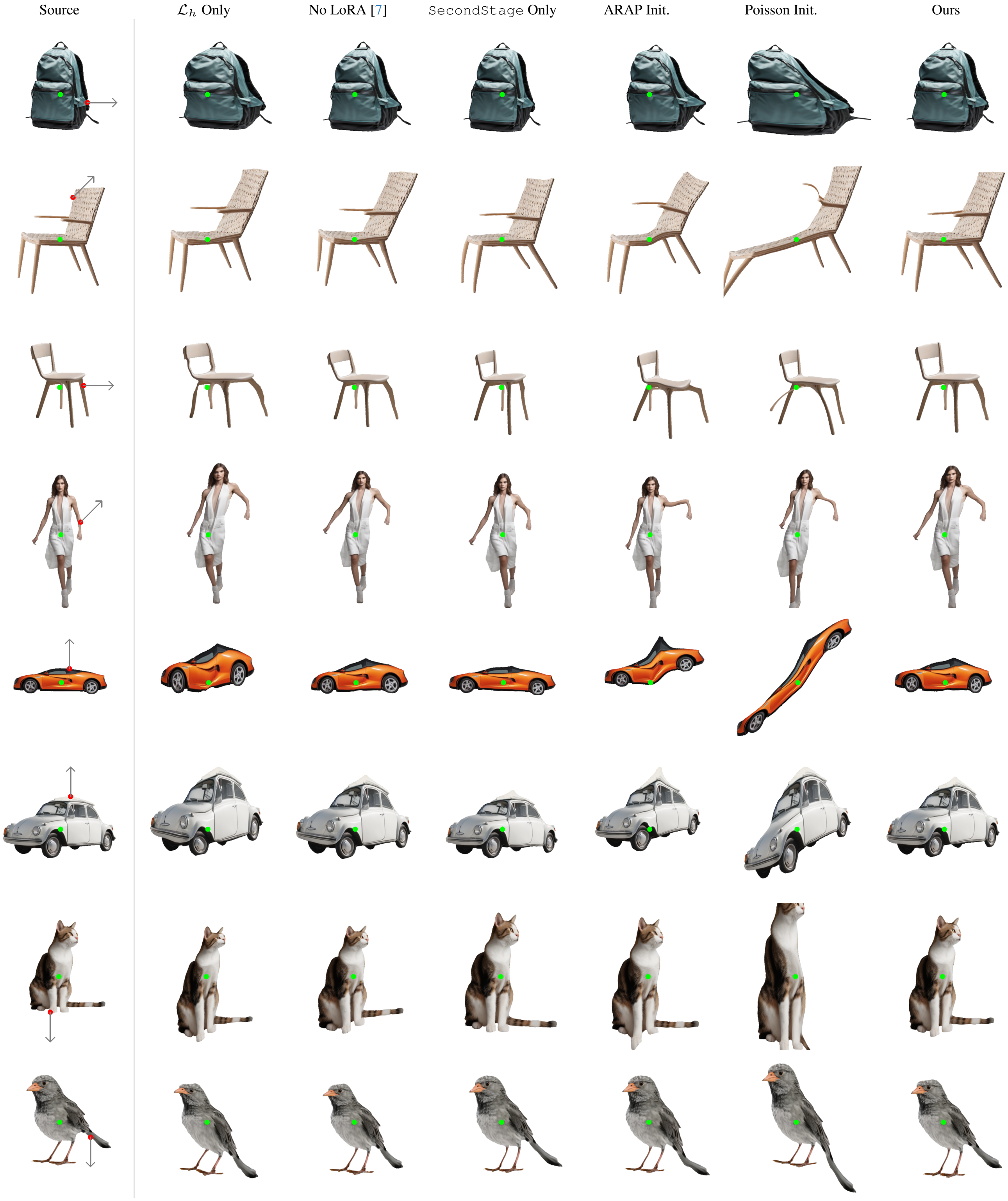
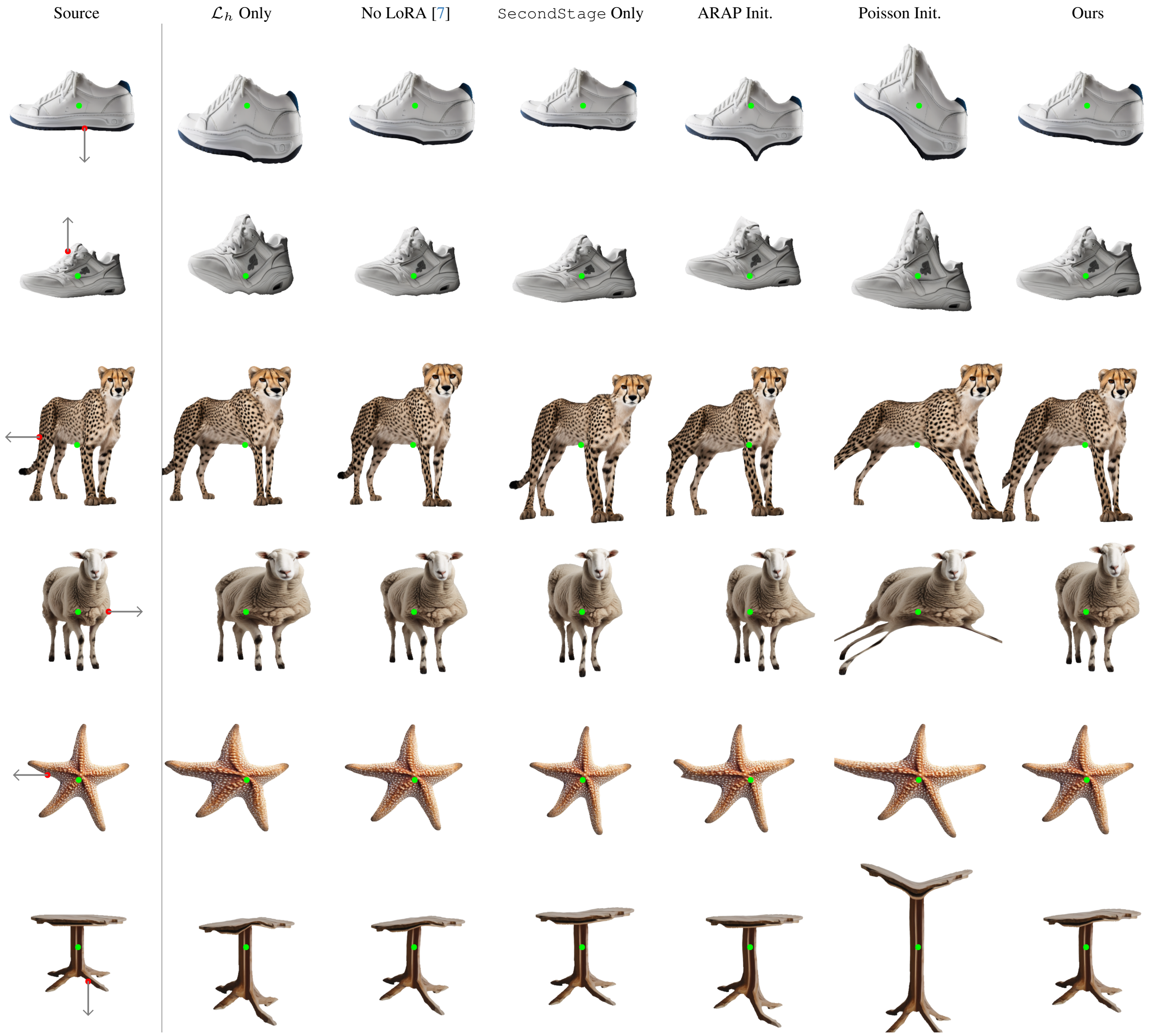


Figure S4. Ablation study for 2D mesh editing. We examine the impact of each design choice on deformation outputs, including the use of diffusion prior (the second column), LoRA finetuning (the third column), two-stage pipeline (the fourth column), and initialization strategies during the `FirstStage` (the fifth and sixth column).



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 [17] 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. S2, 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, as shown in Fig. S4.

S6. Additional Qualitative Results for 3D Shape Deformation

Fig. S5 summarizes outputs of 3D shape deformation with additional results. Here, ARAP [17] only enforces local rigidity and hence cannot produce smooth deformations intended by users. In the ninth row, ARAP [17] introduces a pointy end given an editing instruction that drags the bottom of a doll downward. Ours, however, elongates the entire geometry smoothly, producing a more visually plausible deformation. Another example displayed in the tenth row shows similar behaviors of

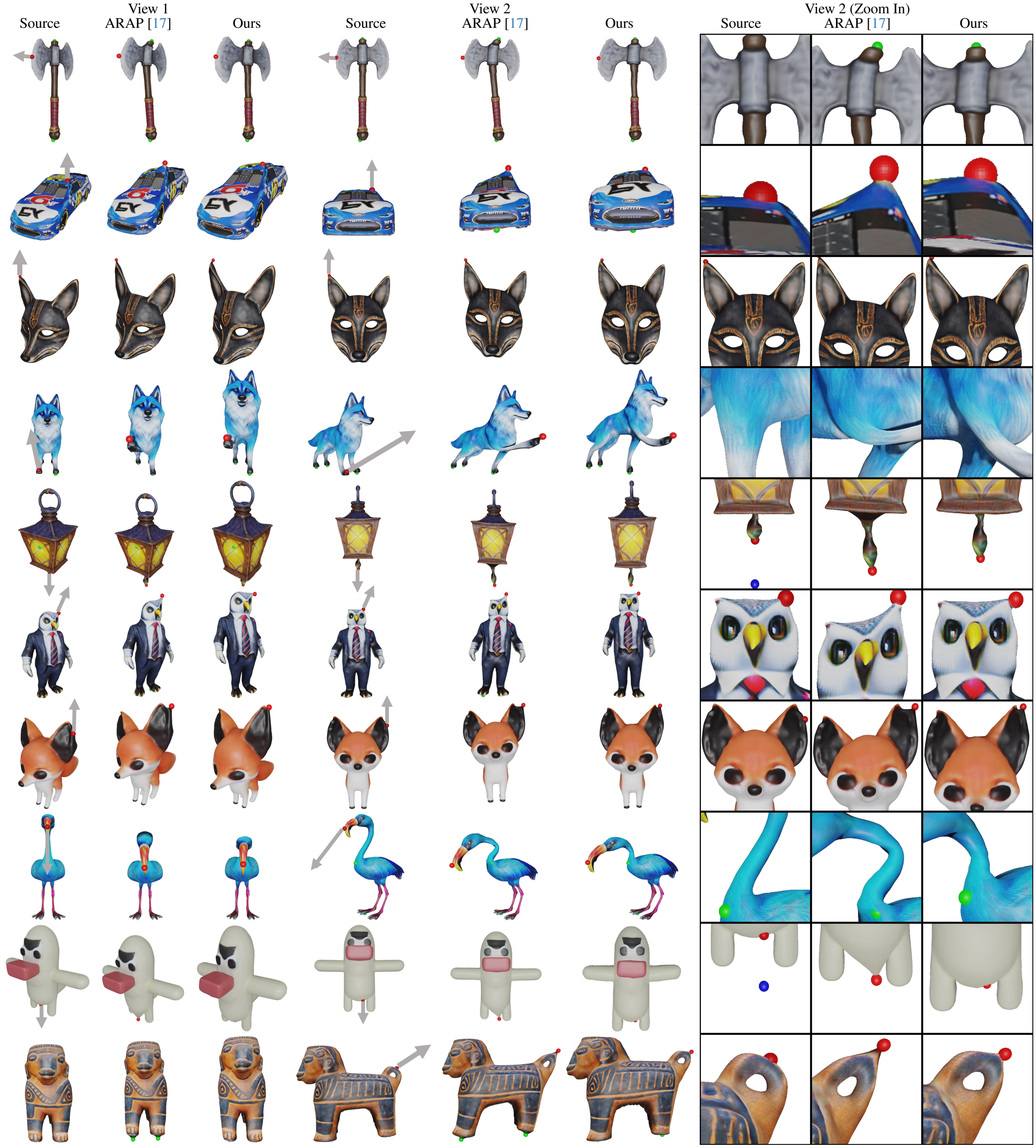


Figure S5. Additional qualitative results from 3D shape deformation. We visualize the source shapes and their deformations made using ARAP [17] and ours by following the instructions each of which specifies a handle (*red*), an edit direction denoted with an arrow (*gray*), and an anchor (*green*). We showcase the rendered images captured from two different viewpoints, as well as one zoom-in view highlighting local details.

ARAP [17] and ours, respectively. Here, unlike ARAP [17], the proposed method adjusts the overall proportion of the statue as the handle located at the tail is translated, while preserving the smooth and round geometry near the handle.

S7. Complex 3D Deformation Examples

In addition to the ability to optimize Jacobian fields using diffusion priors offered by the linearity of Poisson solvers, we can directly propagate local transforms, additionally defined at handle vertices, to Jacobians of neighboring faces by employing geodesic distances as weights [4]. This allows for more dramatic deformations illustrated in Fig. S6, involving limb articulations, large bending, and the use of multiple handles and anchors. As represented in the *Panda* (the seventh, eighth, ninth columns) example, our framework can handle large pose variations, useful in downstream applications, such as animation.

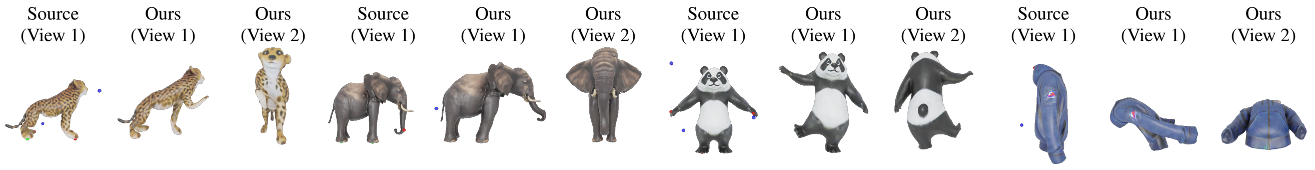
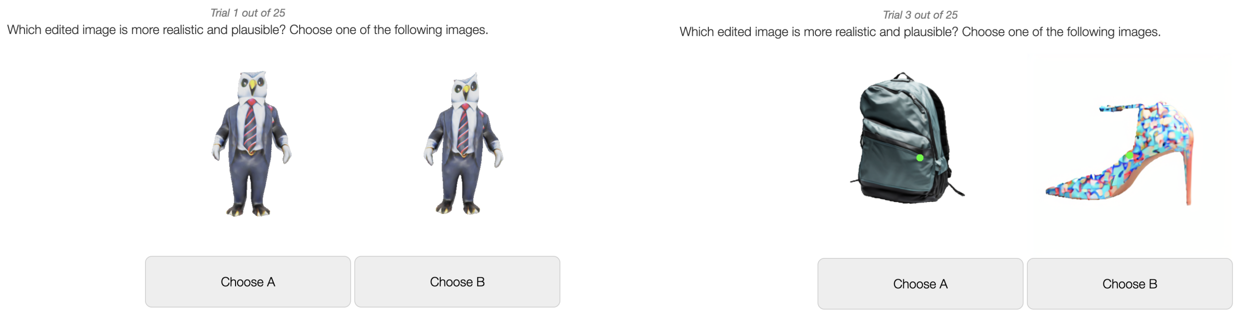


Figure S6. Examples of deforming source meshes using multiple handles and anchors. Best viewed in Zoom-in.

S8. User Study Results for 3D Shape Deformation



Trial 1 out of 25

Which edited image is more realistic and plausible? Choose one of the following images.

Trial 3 out of 25

Which edited image is more realistic and plausible? Choose one of the following images.

Figure S7. Examples of questionnaires displayed during the user study (3D shape deformation). During the user study, we asked the participants to evaluate 20 different result pairs from ARAP [17] and ours as shown on the left. To check whether a participant is focusing on the user study, we included 5 items for the vigilance test. As shown on the right, a vigilance test asks a participant to compare two images, with one of them containing noticeable artifacts.

Assessing the visual plausibility of 3D deformations is particularly challenging due to the difficulty in populating large-scale reference sets as we did for 2D meshes in Sec. S5. We further note that, unlike 3D generative models, computing image-based metrics such as CLIP-R score is non-trivial since it is hard to describe handle-based deformations solely using text prompts.

Therefore, we conduct a user study similar to the one presented in Sec. S5. We asked 47 user study participants on Amazon Mechanical Turk (MTurk) to compare rendered images of meshes deformed using ARAP [17] and ours. Each participant is provided with 20 image pairs and asked to select one image at each time given the question: "`Which edited image is more realistic and plausible? Choose one of the following images.`" An example of a questionnaire displayed to the participants is shown in Fig. S7 (left). We provide an example of vigilance tests, similar to the user study for 2D mesh editing, in Fig. S7 (right). As summarized in Tab. S4, the deformation produced by our method is preferred over the results from the baseline.

Methods	Preference (%) ↑
ARAP [17]	41.7
Ours	**58.3**

Table S4. **User study preference for 3D mesh deformation.** In a user study targeting users on Amazon Mechanical Turk (MTurk), the results produced using ours were preferred over the outputs from the baseline.

References

[1] Luma AI. Genie. 1

[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. 1

[3] 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. 1

[4] Yu et al. Mesh editing with poisson-based gradient field manipulation. *ACM Trans. Graph.*, 23(3), 2004. 9

[5] Hugging Face. Diffusers: State-of-the-art diffusion models for image and audio generation in PyTorch. 1

[6] Shuyang Gu, Jianmin Bao, Dong Chen, and Fang Wen. GIQA: Generated Image Quality Assessment. In *ECCV*, 2020. 3, 5

[7] 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. 1, 6, 7

[8] Alec Jacobson, Daniele Panozzo, et al. libigl: A simple C++ geometry processing library, 2018. 3

[9] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 1

[10] 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. 1

[11] Samuli Laine, Janne Hellsten, Tero Karras, Yeongho Seol, Jaakko Lehtinen, and Timo Aila. Modular Primitives for High-Performance Differentiable Rendering. *ACM TOG*, 2020. 1

[12] 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. 1, 2

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

[14] Daniel Ritchie. Rudimentary framework for running two-alternative forced choice (2afc) perceptual studies on mechanical turk. 3

[15] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-Resolution Image Synthesis with Latent Diffusion Models. In *CVPR*, 2022. 1

[16] 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, 5

[17] Olga Sorkine and Marc Alexa. As-Rigid-As-Possible Surface Modeling. *Proceedings of EUROGRAPHICS/ACM SIGGRAPH Symposium on Geometry Processing*, pages 109–116, 2007. 3, 4, 5, 7, 8, 9