TetSphere: Representing High-Quality Geometry with Lagrangian Volumetric Meshes

Anonymous

Abstract. We present TetSphere, an explicit, Lagrangian representation for reconstructing 3D shapes with high-quality geometry. In contrast to conventional object reconstruction methods, such as neural implicit representations (e.g., NeRF, NeuS) and Eulerian approaches (e.g., DMTet), which often struggle with high computational demands and suboptimal mesh quality, TetSphere utilizes an underused but highly effective geometric primitive—tetrahedral meshes. This approach directly yields superior mesh quality without relying on neural networks or post-processing. It deforms multiple initial tetrahedral spheres to accurately reconstruct the 3D shape through a combination of differentiable rendering and geometric energy optimization, resulting in significant computational efficiency. Serving as a robust and versatile geometry representation, TetSphere seamlessly integrates into diverse applications, including single-view 3D reconstruction, image-/text-to-3D content generation. Experimental results demonstrate that TetSphere outperforms existing representations, delivering faster optimization speed, enhanced mesh quality, and reliable preservation of thin structures.

1 Introduction

Reconstructing 3D geometry stands as a fundamental task in computer vision and graphics. The field has seen significant strides with the advent of diffusion models, showing exceptional capabilities in generating images. This success has spurred research into employing 2D generative methods for 3D reconstruction.

These advances have prominently featured the use of *Eulerian* representations – the geometry is defined based on spatial coordinates. Dreamfusion [67] introduces Score Distillation Sampling (SDS) for distilling geometry and appearance from 2D models and inspires the development of numerous 2D lifting methods [10,46,80,94]. Alongside this, efforts have been made on single-view reconstruction by first generating multi-view 2D images from a single input, which are then used to reconstruct 3D shapes [47,49,50,53,56,68]. Predominantly, both types of these methods employ *neural implicit* representations, such as Neural Radiance Fields (NeRF) [58] and Neural Implicit Surfaces (NeuS) [91], which are inherently Eulerian representations. In parallel, *explicit* representation methods such as Deep Marching Tetrahedra (DMTet) [77] and its variants [25, 52], present an alternative take on Eulerian representations. These methods utilize a deformable tetrahedral grid with signed distance values at grid vertices, offering advantages in capturing intricate geometric details and the integration of explicit shading materials. However, optimizing these models is both time- and

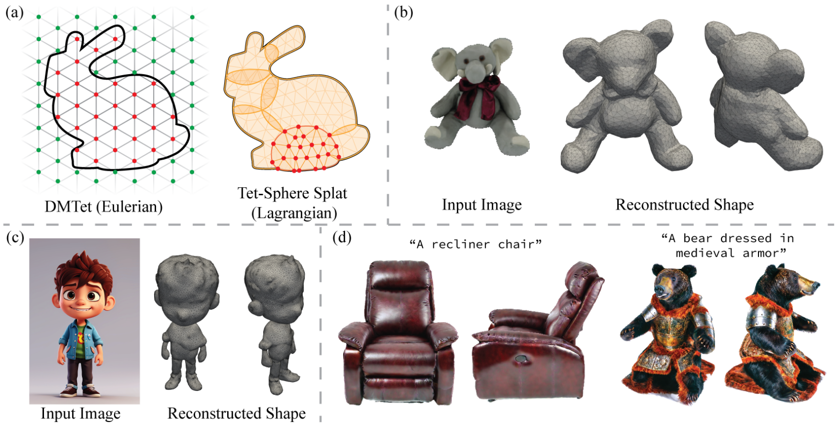


Fig. 1: (a) Eulerian v.s. Lagrangian geometry representations: DMTet employs a deformable tetrahedral grid, assigning signed distance values at vertices, whereas TetSphere reconstructs 3D shapes by directly deforming tet-spheres, enhancing computational and memory efficiency. TetSphere supports a range of applications, including (b) single-view 3D reconstruction from Google Scanned Objects dataset [19], (c) SDS-based image-to-3D generation, and (d) text-to-3D content creation.

memory-consuming, primarily due to the necessity of employing high-resolution grids that inherently contain numerous parameters. Moreover, Eulerian representations are vulnerable in modeling thin structures, often leading to floating artifacts. Extracting surfaces from these models is also required for subsequent applications like rendering and simulation, adding another layer of complexity.

To overcome these issues, we propose TetSphere, an *explicit, Lagrangian* geometry representation designed to construct high-quality meshes efficiently. Lagrangian representations, which track the movement of geometry primitives through space, are more efficient and accurate than Eulerian representations. An illustrative comparison between Eulerian and Lagrangian geometry representation is shown in Fig. 1 (a). Our TetSphere leverages the advantages of tetrahedral meshes – a volumetric, Lagrangian representation that remains relatively underexplored in the realm of 3D reconstruction. Akin to Gaussian splatting (GS) [41] which distributes point clouds, TetSphere “splats” deformed *tetrahedralized spheres* into 3D space to conform to the target object. The final 3D model is reconstructed by the union of these deformed tetrahedral spheres. Compared to GS, TetSphere imposes structured constraints between points owing to tetrahedralization and also offers a clear definition for the interior and boundary of the object. Compared to other Lagrangian representations, particularly surface mesh-based methods [1,65] which struggle with regularization issues leading to self-intersections and compromised mesh quality, TetSphere exhibits greater stability due to its volumetric nature. It also effectively handles complex topologies without the need for re-meshing, an often-necessary requirement in surface

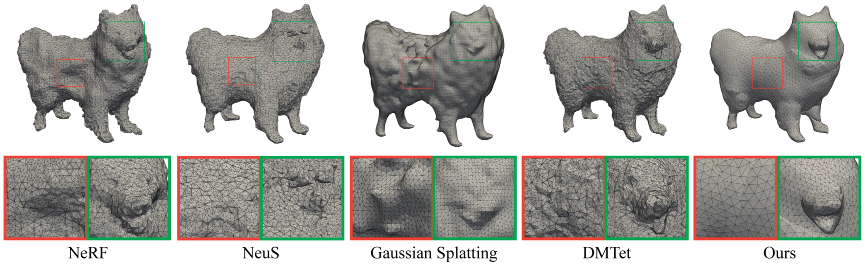


Fig. 2: Comparisons of mesh quality across NeRF, NeuS, GS, DMTet, and TetSphere, highlighting a critical yet overlooked criterion of 3D reconstruction. Existing methods are not inherently mesh-based but are derived through post-processing like Marching Cubes. On the contrary, our method demonstrates superior mesh quality without remeshing. Results are obtained from a single dog image as shown in Fig. 3.

mesh representations during optimization [65]. Overall, TetSphere offers the following advantages: **1)** its explicit representation ensures fast optimization speed (in a matter of minutes) and reduced memory cost, significantly quicker than Eulerian representations; **2)** its Lagrangian nature shows robustness to thin structures and reducing floating artifacts; **3)** it employs volumetric regularization, significantly improving the mesh quality; and **4)** it is capable of handling shapes with arbitrary topologies.

We further present a computational framework for TetSphere. This framework formulates the deformation of tetrahedron spheres as a geometric energy optimization, incorporating differentiable rendering loss, bi-harmonic energy of the deformation gradient field, and non-inversion constraints, to ensure robust and accurate reconstruction. Our framework is versatile, enabling seamless integration with applications including single-view object reconstruction, SDS-based image-to-3D generation, and text-to-3D generation. Notably, TetSphere's fast optimization and reduced computational demands offer significant benefits for the latter two tasks that require intensive time and memory.

In our evaluation, we underscore a frequently overlooked aspect of 3D reconstruction: the quality of the reconstructed mesh. The concept of *mesh quality* encompasses multiple attributes key to the usability of 3D models, such as the uniformity of surface triangles, the absence of undesired bumps, and manifoldness. Despite its paramount importance, particularly in rendering and simulation applications, the evaluation of mesh quality has not received adequate focus in the current 3D reconstruction research. Recognizing this oversight, we introduce three evaluation metrics to assess mesh quality. We conduct a comprehensive evaluation using the Google Scanned Objects (GSO) dataset [19]. Compared to state-of-the-art methods, our TetSphere exhibits superior performance when measured under these newly proposed criteria. It maintains competitive performance on other commonly employed metrics as well. Furthermore, we illustrate its utility in 3D content generation, where it produces qualitatively superior results compared to existing state-of-the-art methods.

2 Related Work

Eulerian and Lagrangian geometry representations. The differentiation between Eulerian and Lagrangian representations originates from computational fluid dynamics [14] but extends more broadly into computational geometry and physics. Eulerian representations characterize the geometry within a fixed spatial reference, usually utilizing grids or voxels to define the space. In contrast, Lagrangian methods track the movement of individual particles or elements, providing a dynamic representation of geometry that adapts to changes over time. Using fluid simulation as an analogy, an Eulerian view would analyze fluid presence at fixed points in space, whereas a Lagrangian perspective follows specific fluid particles. Neural implicit representations, such as DeepSDF [66], NeRF [59], and InstantNGP [62], are modern adaptations of Eulerian concepts, processing 3D positions as inputs to neural networks. These methods theoretically allow for infinite resolution through NN-based parameterization but can result in slow optimization speed due to the training of NN. Beyond implicit ones, explicit or hybrid Eulerian representations, such as DMTet [77], DefTet [24], and TetGAN [102], incorporate explicit irregular grids but can still cause substantial memory usage for high-resolution shapes. Gaussian splatting [82] exemplifies a Lagrangian approach by moving Gaussian point clouds in space. Our proposed TetSphere is an explicit Lagrangian representation, which can be viewed as introducing explicit constraints among points due to tetrahedral meshing, with enhanced efficiency and reconstruction quality.

Single image-to-3D object reconstruction. Single-image 3D reconstruction is an inherently ill-posed problem, and extensive research has been dedicated to addressing it [20, 23]. Early approaches utilized a combination of 2D image encoders and 3D decoders trained on 3D data with both explicit representations, including voxels [12,13,88,97,98], meshes [26,90], and point clouds [21,22,29,55], and implicit representations such as NeRF [38, 61, 103], SDF [60, 66, 100], and occupancy networks [5, 57]. Many of these methods were trained on categorized 3D datasets such as 3D templates [27, 40, 72] and semantics [45], yet faced challenges in generalizing to unseen categories. Recently, an active research direction has been leveraging 2D generative models for 3D reconstruction, including the use of SDS and supplementary losses [17, 30, 46, 49, 56, 76, 80, 99]. Alternative approaches train view-conditioned 2D diffusion models to directly produce multi-view images for 3D reconstruction [9,48–50,53,85,86]. Our method adopts a similar strategy but introduces an explicit and Lagrangian geometry representation to overcome the limitation of prior representations. The recent introduction of large-scale 3D dataset propelled feed-forward large reconstruction models [8, 15, 16, 33, 41, 75, 81, 92, 96, 101], which directly reconstruct triplane-based NeRF [8] or 3D Gaussians [41]. Their feed-forward inference significantly

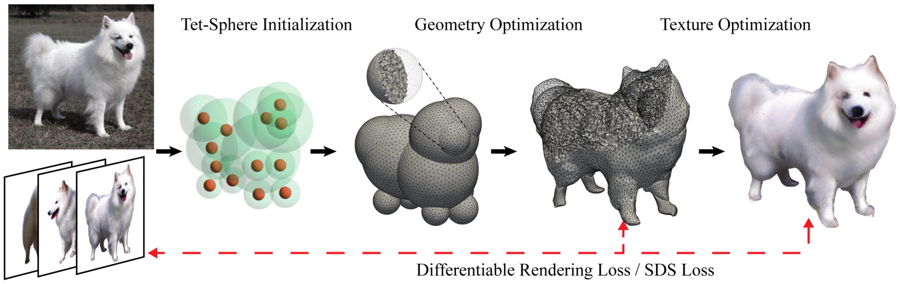

Tet-Sphere Initialization
Geometry Optimization
Texture Optimization
Differentiable Rendering Loss / SDS Loss

Fig. 3: Overall pipeline: From multi-view images generated from a single image, TetSphere selects 3D feature points to initialize a set of tetrahedral spheres. The geometry optimization stage deforms these spheres to reconstruct the object. Texture optimization is then applied to obtain the texture or physical material of the object.

accelerates the speed of 3D object reconstruction, but often at a sacrifice of relatively low resolution and geometry quality, as well as, coupled geometry and materials.

Text-to-3D Content Generation. The recent success of text-to-2D image models has spurred a growing interest in generating 3D output from text input. In light of limited text-annotated 3D datasets, methods have been developed to leverage the pre-trained text-to-image models to reason between 2D renderings of 3D models and text descriptions. Early works [35, 42] adopt the pre-trained CLIP model [70] to supervise the generation by aligning the clip text and image embeddings. More recently, 2D diffusion-based generative models [71, 73] have powered direct supervision in the image/latent space and achieved superior 3D qualities [11, 46, 67, 84, 93]. Notably, DreamFusion [67] introduces the Score Distillation Sampling (SDS) for supervising the NeRF [58] optimization using diffusion priors as a score function (i.e., by minimizing the added noise and the predicted noise under the text condition). Follow-up works have since been proposed to improve score sampling formula with Perturb-and-Average and variational method [89, 93], better noise sampling schedules [34], various 3D representations [11, 46, 87], text prompts [2], 3D consistency [32, 44], and prior quality with dedicated diffusion models [69, 78]. Our method leverages the diffusion model used in [69] for text-to-3D shape generation, which is a Normal-Depth diffusion model trained on the large-scale LAION dataset, but replaces the geometry representation with our TetSphere.

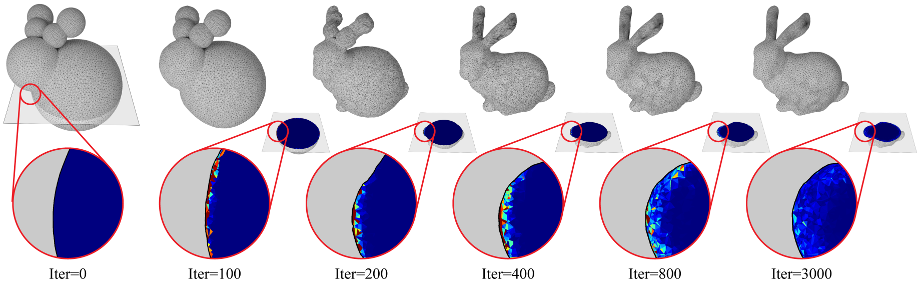


Fig. 4: TetSphere with deforming tetrahedral spheres. Color-coded regions represent the bi-harmonic energy values (red: high, blue: low) across tetrahedra, one of the geometric regularizations employed in our deformation optimization process.

3 Overview

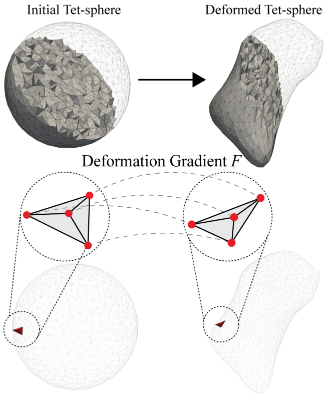
Our approach takes multi-view images as inputs. We initiate TetSphere using a set of tetrahedral spheres with different centers and sizes (radii). These spheres are chosen to ensure that, collectively, the initial spheres approximately cover the target shape under different input views. Subsequently, these spheres undergo a two-stage optimization process to precisely reconstruct the 3D object. The first stage, geometry optimization, deforms the tetrahedral spheres through the minimization of rendering losses and two geometric regularization energies. The second stage optimizes the surface texture or Physically Based Rendering (PBR) materials of the tetrahedral spheres. Both stages leverage differentiable rasterizers. The rendering loss typically incorporates supervision from ground truth images or SDS. An illustration of the overall pipeline is shown in Fig. 3.

4 Tet-Sphere

At the core of our method is TetSphere, an explicit, Lagrangian representation for 3D shapes. TetSphere positioning itself alongside Gaussian splatting [41], which is also a neural-network free Lagrangian representation by representing shapes using Gaussian point clouds as primitives. Although the use of point cloud is notably general, it lacks local constraints within individual Gaussian kernels, resulting in an undefined boundary between the interior and exterior of the object. Converting these Gaussian kernels into explicit meshes necessitates surface reconstruction, which often leads to uneven, bumpy surfaces due to the inherent limitation in defining surface from point clouds.

We employ tetrahedral spheres as our primitive of choice. Unlike point clouds, tetrahedral meshes inherently enforce structured local connectivity between points owing to tetrahedralization. This preserves the geometric integrity of the 3D shape and also enhances the surface quality by imposing geometric regularization across the entire mesh interior. We formulate the reconstruction of shapes through TetSphere as a deformation of tetrahedron spheres. Starting from a set of tetrahedral spheres, we adjust the positions of their vertices to align the rendered images of these meshes with the corresponding target multi-view images. The vertex movement is constrained by two geometric regularizations on the tetrahedral meshes, derived from the field of geometry processing [4]. These constraints, penalizing the nonsmooth deformation (via bi-harmonic energy) and preventing the inversion of mesh elements (via local injectivity), are proven to be effective, ensuring that the resulting tetrahedral meshes are of superior quality and maintain structural integrity. Fig. 4 illustrates the iterative process of TetSphere with deforming tetrahedral spheres.

4.1 Tetrahedral Sphere Primitive



The primitive of TetSphere is a tetrahedralized sphere, called *tet-sphere*, with N vertices and T tetrahedra. By applying principles from the Finite Element Method (FEM) [79], the mesh of each sphere is composed of tetrahedral elements, where each tetrahedron constitutes a 3D discrete piecewise linear volumetric entity. We denote the position vector of all vertices of the i-th deformed sphere mesh as $x_i \in \mathbb{R}^{3N}$. The deformation gradient of the j-th tetrahedron in the i-th sphere is denoted as $\mathbf{F}_{\mathbf{x}}^{(i,j)} \in \mathbb{R}^{3\times 3}$, which quantitatively describes how each tetrahedron's shape transforms [79]. Essentially, the deformation gradient $\mathbf{F}_{\mathbf{x}}^{(i,j)}$ serves as a measure of the spatial changes a tetrahedron undergoes from its original configuration to its deformed state. Refer to the inset figure for a visual explanation and the Supplementary Material for an in-depth derivation.

Rather than using a single sphere, TetSphere utilizes a collection of spheres to accurately represent arbitrary shapes. Consequently, the complete shape of TetSphere is the union of all spheres. By adopting multiple spheres, we ensure that each local region of a shape is detailed independently, enabling a highly accurate representation. Moreover, it allows for the representation of shapes with arbitrary topologies. Such a claim is theoretically guaranteed by the paracompactness property of manifold shapes [37].

Using tetrahedral spheres offers several technical benefits compared with prevalent representations for object reconstruction, as demonstrated in Fig. 2:

- Compared to neural representations (e.g., NeRF, NeuS), our tetrahedral representation does not rely on neural networks, thus inherently accelerating the optimization process.
- Compared to Eulerian representations (such as DMTet), our approach entirely circumvents the necessity for iso-surface extraction—an operation that

often degrades mesh quality owing to the predetermined resolution of the grid space. Furthermore, Eulerian methods rely on level-set functions, a dependency that can lead to undesirable floating and noisy artifacts given sparsely populated or unpopulated grid values during the training phase. The deformation process of our representation inherently prevents individual spheres from breaking, thereby eliminating floating noises.

- Compared to other Lagrangian representations, such as triangle meshes and Gaussian point clouds, our method offers a volumetric representation through the use of tetrahedral meshes. It is more robust to thin shapes, where surface meshes often face the risk of self-penetration, and to fragile structures, where point-based methods suffer from. Furthermore, unlike Gaussian point clouds, which require additional steps to produce a mesh, our tetrahedral representation naturally forms a mesh. Each tetrahedron also imposes constraints among vertices, leading to superior mesh quality.

4.2 Tet-Sphere as Shape Deformation

To reconstruct the geometry of the target object, we deform the initial tet-spheres by changing their vertex positions. This process is governed by two primary goals: ensuring the deformed tet-spheres align with the input multi-view images and maintaining high mesh quality that adheres to necessary geometry constraints.

To maintain the mesh quality, we leverage bi-harmonic energy – defined as an energy quantifying smoothness throughout a field, as drawn from the literature on geometry processing [6] – to the deformation gradient field. This geometric regularization ensures the smoothness of the deformation gradient field across the deformation process, thus preventing irregular mesh or bumpy surfaces. It's important to highlight that this bi-harmonic regularization does *not* lead to over-smoothness of the final result. This is because the energy targets the deformation gradient field, which measures the *relative* changes in vertex positions, rather than the *absolute* positions themselves. Such an approach allows for the preservation of sharp local geometric details, akin to techniques used in physical simulations [95]. Furthermore, we introduce a geometric constraint to guarantee local injectivity in all deformed elements [74]. This ensures that the elements maintain their orientation during the deformation, avoiding inversions or inside-out configurations. This constraint can be mathematically expressed as $\det(\mathbf{F}_{\mathbf{x}}^{(i,j)}) > 0$. Importantly, these two terms – bi-harmonic energy for smoothness and local injectivity for element orientation – are universally applicable to any tetrahedral meshes, stemming from their fundamental basis in geometry processing [4].

Let $\mathbf{x} = [x_1, ..., x_M] \in \mathbb{R}^{3NM}$ denote the positions of vertex across all M tet-spheres, and $\mathbf{F}_{\mathbf{x}} \in \mathbb{R}^{9MT} = [\mathrm{vec}(\mathbf{F}_{\mathbf{x}}^{(1,1)}), ..., \mathrm{vec}(\mathbf{F}_{\mathbf{x}}^{(M,T)})]$ denote the flattened deformation gradient fields of all tet-spheres. In the bi-harmonic energy, the Laplacian matrix is defined based on the connectivity of the tetrahedron faces, denoted as $\mathbf{L} \in \mathbb{R}^{9MT \times 9MT}$. This matrix is block symmetric, where each block

$\mathbf{L}_{pq} \in \mathbb{R}^{9\times 9}, p \neq q$ is set to a negative identity matrix $-I$ if the p-th and q-th tetrahedron shares a common triangle; or kI for $\mathbf{L}_{pp}$, where k is the number of neighbors of the p-th tetrahedron. The deformation of the tet-spheres is formulated as an optimization problem:

$$\begin{aligned} \min_{\mathbf{x}} \quad & \mathbf{\Phi}(R(\mathbf{x})) + ||\mathbf{L}\mathbf{F}_{\mathbf{x}}||_2^2 \\ \text{s.t.} \quad & \det(\mathbf{F}_{\mathbf{x}}^{(i,j)}) > 0,\ \forall i \in \{1,...,M\},\ j \in \{1,...,T\}, \end{aligned} \tag{1}$$

where $R(\cdot)$ is the rendering function, $\mathbf{\Phi}(\cdot)$ is the rendering loss matching the deformed tetrahedral spheres with the input images. The second term regulates the bi-harmonic energy across the deformation gradient field. The non-inversion constraint ensures that tetrahedrons maintain their orientation. To manage this constrained optimization, we reformulate it variationally by incorporating the non-inversion hard constraint as a soft penalty term into the objective,

$$\min_{\mathbf{x}} \quad \mathbf{\Phi}(R(\mathbf{x})) + w_1||\mathbf{L}\mathbf{F}_{\mathbf{x}}||_2^2 + w_2 \sum_{i,j} (\min\{0, \det(\mathbf{F}_{\mathbf{x}}^{(i,j)})\})^2, \tag{2}$$

allowing for optimization via standard gradient descent solvers.

In the proposed optimization framework, two considerations have been outlined. 1) The adaptive loss function $\mathbf{\Phi}(\cdot)$ is designed to be flexible, supporting a variety of metrics, including l_1 for color images, MSE for depth images, cosine embedding loss for normal images, or SDS loss. 2) Due to the tetrahedron being a linear element, the deformation gradient $\mathbf{F}_{\mathbf{x}}^{(i,j)}$ is a linear function of $\mathbf{x}$, making the bi-harmonic energy a quadratic term. 3) The weights w_1 and w_2 are dynamically adjusted using a cosine scheduler. We provide details of the scheduler's hyperparameters in the Supplemental Material.

5 Optimization Framework

The overall framework of optimization is illustrated in Fig. 3. The geometry optimization follows the shape deformation as detailed in Sect. 4. In this section, we introduce the remaining two components: initialization of TetSphere and texture optimization.

5.1 Tet-Sphere Initialization

Given multi-view images as inputs, we select feature points to initialize the 3D center positions of the tet-spheres. We aim to achieve a uniform distribution of these tet-spheres within the object, ensuring comprehensive coverage of the silhouette depicted in the multi-view images.

We introduce an algorithm, silhouette coverage, inspired by Coverage Axis [18] to automatically select initial centers of tet-spheres for an arbitrary shape. This process begins with the construction of a coarse voxel grid, initially assigning a zero value to each voxel. By projecting these voxels onto the image spaces

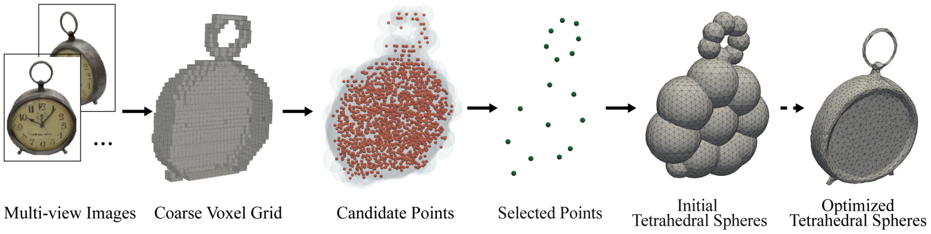


Fig. 5: Feature point selection via silhouette coverage algorithm. Our method automates feature point selection from a sparse voxel constructed from multi-view images. By solving a linear programming problem, we determine the feature points and initialize tetrahedral spheres for subsequent geometry optimization.

using the same camera poses as the input multi-view images, voxels within the foreground of all images are marked with a value of 1. These voxel positions are identified as candidate positions of tet-sphere centers. From these marked positions, the objective is to pick a minimal subset of candidates that ensure all candidates are fully encapsulated by tet-spheres centered on these points. This involves placing uniform spheres of varying radius values at all candidate points and choosing a minimal subset that collectively covers all the candidate points. We formulate a linear programming problem to efficiently perform the selection. The detailed formulation is provided in the Supplementary Material. Fig. 5 shows the pipeline of the silhouette coverage algorithm. In our implementation, with a voxel grid resolution 300 × 300 and $n = 20$, the whole tet-sphere initialization completes in $\sim$1 minute on average.

5.2 Texture/PBR Material Optimization

The final stage of our reconstruction process focuses on optimizing the texture or material properties. Our TetSphere method, with its explicit representation, allows textures and materials to be directly applied to the surface vertices and faces of the tet-spheres. This enables the use of sophisticated material models, such as Disney's principled BRDF [7], with physically-based rendering.

Material optimization is facilitated through the use of differentiable rasterizers [43], which adjust the textures or materials to closely match the input multi-view colored images. A significant advantage of TetSphere is that the deformation of tetrahedral spheres does not alter the surface topology. Unlike methods such as DMTet, which require isosurface extraction and subsequent texture parameterization at each step due to potential changes in the underlying shape, our method necessitates only a single texture parameterization at the beginning of optimization. This parameterization remains consistent throughout the process, significantly enhancing the efficiency of texture optimization.

For scenarios with dense input views, we have found that using textured images as optimization variables is straightforward and yields high-quality results.

Table 1: Single-View reconstruction results on the GSO Dataset: Evaluating reconstruction accuracy with Chamfer Distance (Cham.) and Volume IoU, alongside new mesh quality metrics: Area-Length Ratio (ALR), Manifoldness Rate (MR), and Connected Component Discrepancy (CC Diff.). TetSphere demonstrates superior performance on these criteria and maintains competitive reconstruction accuracy.

Method	Mesher	Cham.↓	Vol. IoU↑	ALR↑	MR(%)↑	CC Diff.↓
Realfusion [56]	Marching Cubes	0.0819	0.2741	0.0561	100%	47.7
Magic123 [68]	Marching Tets	0.0516	0.4528	0.0383	100%	13.7
One-2-3-45 [47]	Marching Cubes	0.0629	0.4086	0.0574	96%	0.83
Point-E [64]	Marching Cubes	0.0426	0.2875	0.2421	100%	18.38
Shap-E [39]	Marching Tets	0.0436	0.3584	0.1236	100%	9.03
Zero123 [49]	Marching Cubes	0.0339	0.5035	0.0543	100%	0.18
SyncDreamer [51]	Marching Cubes	**0.0261**	0.5421	0.0201	10%	0.3
Wonder3d [53]	Marching Cubes	0.0329	0.5768	0.0281	100%	**0.0**
Open-LRM [31]	Marching Cubes	0.0285	0.5945	0.0252	100%	**0.0**
DreamGaussian [83]	Marching Cubes	0.0641	0.3476	0.0812	100%	237.4
Ours	N/A	0.0351	**0.6317**	**0.3665**	100%	**0.0**

In cases with sparse input views, we adopt a two-layer multilayer perceptron (MLP) that takes the surface vertex positions as inputs and outputs the material parameters, a practice in line with existing methods [69, 80].

6 Experiments and Results

We conduct experiments on three applications to demonstrate TetSphere's versatility and effectiveness: single-view reconstruction, image-to-3D shape generation, and text-to-3D shape generation. We choose single-view reconstruction as it allows for comprehensive evaluation leveraging the well-established benchmark on the Google Scanned Objects (GSO) dataset. The focus on the latter two applications is because the existing methods are known for their high computational costs, specifically in terms of GPU quantity and memory capacity. We show experiments on these applications to demonstrate that TetSphere effectively addresses and mitigates these issues.

For the first two applications, the input is a single image from which a textured 3D shape is reconstructed. For text-to-3D generation, the input is a text prompt. All the output targets are textured 3D shapes. For single-view reconstruction, We conduct a quantitative evaluation of our method on GSO dataset, comparing its performance with other state-of-the-art methods. The evaluation focuses on both the reconstruction accuracy and the geometry quality. In addition, we show a series of qualitative comparisons using web-collected images or text prompts. The supplementary Material provides the implementation details.

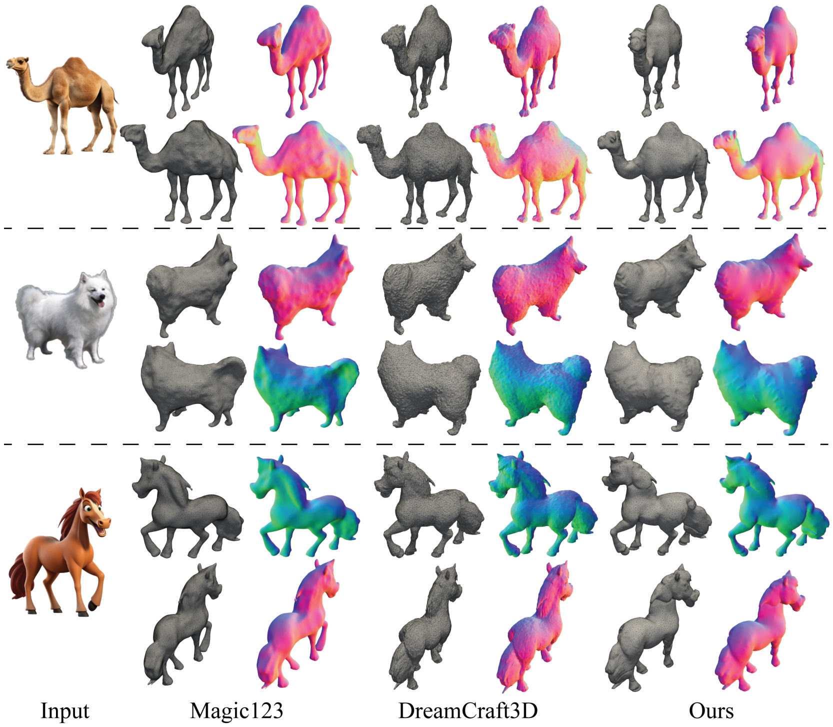


Fig. 6: Qualitative comparison on image-to-3D generation with surface mesh visualizations and rendered normal maps (1/2). Our generated meshes exhibit minimal noise and high quality, featuring more regular triangle meshing.

6.1 Baselines and Evaluation Protocol

Baselines. For single-view reconstruction, we quantitative compare with several state-of-the-art methods: RealFusion [56], Magic123 [68], One-2-3-45 [47], Point-E [64], Shap-E [39], Zero123 [49], SyncDreamer [50], Wonder3d [53], Open-LRM [31,33], and DreamGaussian [83]. For image-to-3D generation, we compare TetSphere with Magic123 and Dreamcraft3D [80]. Both are multi-stage methods starting with NeRF optimization followed by DMTet for optimizing mesh and texture. For text-to-3D, our comparison focuses on RichDreamer, a state-of-the-art method known for incorporating PBR materials and generating shapes, showcasing notable results in 3D generation from text prompts.

Evaluation Datasets. For single-view reconstruction, following prior research [51, 53], we use the GSO dataset for our evaluation, covering a broad range of everyday objects. The evaluation dataset aligns with that used by SyncDreamer and Wonder3D, featuring 30 diverse objects, from common household items to animals. For image-to-3D generation, we include a variety of internet-collected

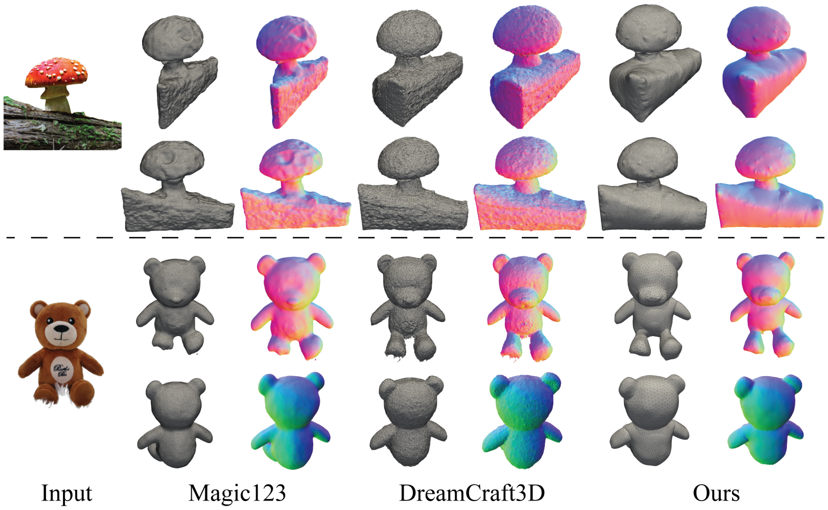


Fig. 7: Qualitative comparison on image-to-3D generation with surface mesh visualizations and rendered normal maps (2/2). Our generated meshes exhibit minimal noise and high quality, featuring more regular triangle meshing.

images with different styles in our evaluation. For text-to-3D generation, we employ text prompts from the official implementation of RichDreamer.

Metrics. To assess the accuracy of single-view reconstruction, we use two commonly used metrics: Chamfer Distances (Cham.) and Volume IoU, comparing the ground-truth shapes with the reconstructed ones. Following established practices, we use the rigid Iterative Closest Point (ICP) [3] algorithm to align the generated shapes with their ground-truth counterparts before metric calculation.

While Cham. and Volume IoU effectively gauge volumetric and point-based shape conformity, they fall short of evaluating mesh quality. Recognizing this gap, we introduce three additional metrics to comprehensively assess the geometry quality of the generated shapes: 1) **Area-length Ratio (ALR):** This metric computes the average ratio of a triangle's area to its perimeter (scaled by a constant coefficient) within the surface mesh. Values range from 0 to 1, where meshes with higher ALR values contain mostly equilateral triangles, thereby indicating superior triangle quality; 2) **Manifoldness Rate (MR):** Manifoldness verifies if a mesh qualifies as a closed manifold. Non-manifold meshes can manifest anomalies, such as edges shared by more than two faces, vertices connected by edges but not by a surface, isolated vertices and edges, or open boundaries, which can cause significant problems in downstream applications such as simulation and rendering. We report the percentage of manifold shapes within the evaluation dataset as MR; and 3) **Connected Component Discrepancy (CC Diff.)** from the ground-truth shape: This measure identifies the presence of floating

Table 2: Comparison of memory cost and run-time speed on image-to-3D generation with SDS loss. We report the maximal batch size of 256×256 images that can occupy a 40GB A100 and the run-time speed for training with batch size 4.

Method		Maximal Batch Size↑	Speed↑ (#iter./s)
NeRF	Make-it-3D [84]	4	1.22
	Magic123 [68]	4	1.03
NeuS	SyncDreamer [50]	48	1.8
DMTet	DreamCraft3D [80]	8	1.23
	GeoDream [54]	8	1.30
3D GS	DreamGaussian [83]	80	4.43
Ours		**120**	**6.59**

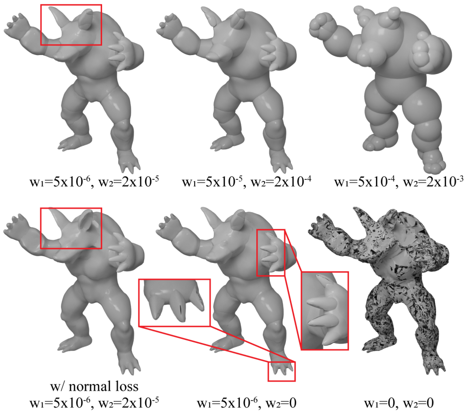

w_1=5x10^{-6}, w_2=2x10^{-5}
w_1=5x10^{-5}, w_2=2x10^{-4}
w_1=5x10^{-4}, w_2=2x10^{-3}
w/ normal loss
w_1=5x10^{-6}, w_2=2x10^{-5}
w_1=5x10^{-6}, w_2=0
w_1=0, w_2=0

Fig. 8: Analysis on energy coefficients for geometry optimization. Dark regions indicate flipping of the surface triangles.

artifacts or structural discontinuities within the mesh, highlighting the integrity and cohesion of the reconstructed shape.

6.2 Results

Single-view Reconstruction. We use the 8 colored images and normal maps generated by Wonder3d to reconstruct the shapes. Table 1 shows the comparison results. Our TetSphere technique excels beyond baseline methods in terms of mesh quality, while also achieving competitive levels of reconstruction accuracy (Fig. 1(b)). This improvement is attributed to the regularizations embedded within the TetSphere optimization. These results demonstrate the superior capabilities of our representation when compared to existing ones.

Image-to-3D Shape Generation. Fig. 6 and 7 illustrate the comparison results on image-to-3D shape generation. Our approach outperforms Magic123 and DreamCraft3D in terms of mesh quality, achieving smoother surfaces for broad regions such as animal bodies while retaining local sharp details in areas like eyes and noses. Magic123 tends to produce overly smooth surfaces that sometimes deviate from the correct geometry, as observed with the mushroom and the teddy bear in Fig. 7. DreamCraft3D suffers from noisy and bumpy surface meshes, indicating poor mesh quality. Furthermore, we also highlight the computational efficiency of TetSphere in Table 2. Compared to various geometry representations, TetSphere stands out for its minimal memory usage and achieves the fastest run-time speed. This efficiency underscores the benefits of TetSphere's explicit and Lagrangian properties.

Text-to-3D Shape Generation. Fig. 9 shows the comparison results on text-to-3D generation, with additional results detailed in the Supplementary Materials. Our TetSphere is capable of producing slender structures, such as the

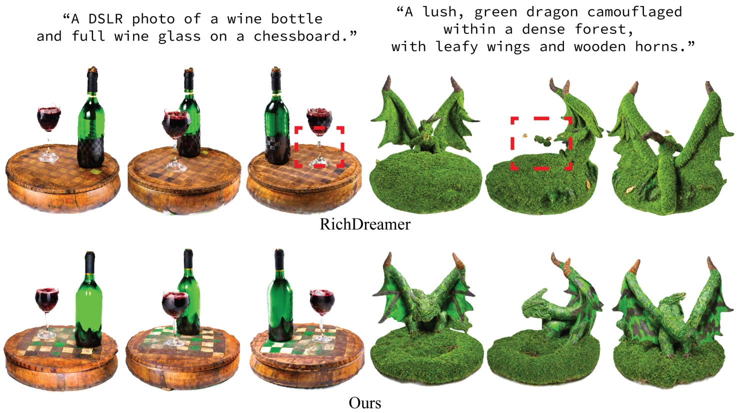

"A DSLR photo of a wine bottle and full wine glass on a chessboard."
"A lush, green dragon camouflaged within a dense forest, with leafy wings and wooden horns."
RichDreamer
Ours

Fig. 9: Results on text-to-3D shape generation. Our TetSphere excels in creating slender forms, demonstrating its strength in handling thin structures effectively.

dragon's head and the goblet. Furthermore, the results also demonstrate that the integration of TetSphere with SDS enhances the geometric detail of the generated shapes, resulting in both diverse and high-quality textures.

6.3 Analysis

Effects of Energy Coefficients. Fig. 8 demonstrates how different energy coefficients influence the reconstruction outcome. Larger coefficients lead to a smoother surface, but too-small coefficients may cause tetrahedron inversion. In our experiments, we choose $w_1 = 5 \times 10^{-6}, w_2 = 2 \times 10^{-5}$ and apply a cosine increase schedule to balance surface smoothness and structural integrity.
Dense-view Inverse Rendering. In the Supplementary Material, we show an additional application of TetSphere: inverse rendering with dense-sampled views and compare it with surface mesh representation [65]. Our method demonstrates fast and robust optimization results in this context. This further illustrates TetSphere's potential for broader reconstruction scenarios.

7 Conclusion

We introduced TetSphere, a geometry representation for the reconstruction of textured shapes with its tetrahedral mesh framework. This method addresses the limitations of existing reconstruction methods, such as the high computational cost of neural implicit representations and the suboptimal mesh quality inherent in Eulerian geometry methods. Future work could extend TetSphere to leverage direct 3D supervision with volumetric data. The current limitation of TetSphere lies in its inability to guarantee topology preservation due to the union of all tet-spheres. This underscores the necessity for future development of shape generation that can adhere to topology constraints.

References

1. Alliegro, A., Siddiqui, Y., Tommasi, T., Nießner, M.: Polydiff: Generating 3d polygonal meshes with diffusion models. arXiv preprint arXiv:2312.11417 (2023) 2
2. Armandpour, M., Sadeghian, A., Zheng, H., Sadeghian, A., Zhou, M.: Re-imagine the negative prompt algorithm: Transform 2D diffusion into 3d, alleviate janus problem and beyond (Apr 2023) 5
3. Arun, K.S., Huang, T.S., Blostein, S.D.: Least-squares fitting of two 3-d point sets. IEEE Trans. Pattern Anal. Mach. Intell. **9**(5), 698–700 (May 1987) 13
4. Bærentzen, J.A., Gravesen, J., Anton, F., Aanæs, H.: Guide to computational geometry processing: foundations, algorithms, and methods. Springer Science & Business Media (2012) 7, 8
5. Bian, W., Wang, Z., Li, K., Prisacariu, V.A.: Ray-onet: Efficient 3d reconstruction from a single rgb image. In: BMVC (2021) 4
6. Botsch, M., Sorkine, O.: On linear variational surface deformation methods. IEEE transactions on visualization and computer graphics **14**(1), 213–230 (2007) 8
7. Burley, B., Studios, W.D.A.: Physically-based shading at disney. In: Acm Siggraph. vol. 2012, pp. 1–7. vol. 2012 (2012) 10
8. Chan, E.R., Lin, C.Z., Chan, M.A., Nagano, K., Pan, B., De Mello, S., Gallo, O., Guibas, L., Tremblay, J., Khamis, S., Karras, T., Wetzstein, G.: Efficient geometry-aware 3D generative adversarial networks (Dec 2021) 4
9. Chan, E.R., Nagano, K., Chan, M.A., Bergman, A.W., Park, J.J., Levy, A., Aittala, M., De Mello, S., Karras, T., Wetzstein, G.: Generative novel view synthesis with 3D-Aware diffusion models (Apr 2023) 4
10. Chen, R., Chen, Y., Jiao, N., Jia, K.: Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation. arXiv preprint arXiv:2303.13873 (2023) 1
11. Chen, R., Chen, Y., Jiao, N., Jia, K.: Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation. arXiv preprint arXiv:2303.13873 (2023) 5
12. Chen, Z., Zhang, H.: Learning implicit fields for generative shape modeling. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5939–5948 (2019) 4
13. Choy, C.B., Xu, D., Gwak, J., Chen, K., Savarese, S.: 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. pp. 628–644. Springer (2016) 4
14. Chung, T.J.: Computational fluid dynamics. Cambridge university press (2002) 4
15. Deitke, M., Liu, R., Wallingford, M., Ngo, H., Michel, O., Kusupati, A., Fan, A., Laforte, C., Voleti, V., Gadre, S.Y., VanderBilt, E., Kembhavi, A., Vondrick, C., Gkioxari, G., Ehsani, K., Schmidt, L., Farhadi, A.: Objaverse-XL: A universe of 10m+ 3D objects (Jul 2023) 4
16. Deitke, M., Schwenk, D., Salvador, J., Weihs, L., Michel, O., VanderBilt, E., Schmidt, L., Ehsani, K., Kembhavi, A., Farhadi, A.: Objaverse: A universe of annotated 3D objects (Dec 2022) 4
17. Deng, C., Jiang, C.m., Qi, C.R., Yan, X., Zhou, Y., Guibas, L., Anguelov, D.: NeRDi: Single-View NeRF synthesis with Language-Guided diffusion as general image priors (Dec 2022) 4

18. Dou, Z., Lin, C., Xu, R., Yang, L., Xin, S., Komura, T., Wang, W.: Coverage axis: Inner point selection for 3d shape skeletonization. In: Computer Graphics Forum. vol. 41, pp. 419–432. Wiley Online Library (2022) 9
19. Downs, L., Francis, A., Koenig, N., Kinman, B., Hickman, R., Reymann, K., McHugh, T.B., Vanhoucke, V.: Google scanned objects: A High-Quality dataset of 3D scanned household items (Apr 2022) 2, 3
20. Fahim, G., Amin, K., Zarif, S.: Single-view 3d reconstruction: A survey of deep learning methods. Computers & Graphics **94**, 164–190 (2021) 4
21. Fan, H., Su, H., Guibas, L.: A point set generation network for 3D object reconstruction from a single image. arXiv [cs.CV] (Dec 2016) 4
22. Fan, H., Su, H., Guibas, L.J.: A point set generation network for 3d object reconstruction from a single image. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 605–613 (2017) 4
23. Fu, K., Peng, J., He, Q., Zhang, H.: Single image 3d object reconstruction based on deep learning: A review. Multimedia Tools and Applications **80**, 463–498 (2021) 4
24. Gao, J., Chen, W., Xiang, T., Jacobson, A., McGuire, M., Fidler, S.: Learning deformable tetrahedral meshes for 3d reconstruction. Advances In Neural Information Processing Systems **33**, 9936–9947 (2020) 4
25. Gao, W., Wang, A., Metzer, G., Yeh, R.A., Hanocka, R.: Tetgan: A convolutional neural network for tetrahedral mesh generation. arXiv preprint arXiv:2210.05735 (2022) 1
26. Gkioxari, G., Malik, J., Johnson, J.: Mesh r-cnn. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 9785–9795 (2019) 4
27. Goel, S., Kanazawa, A., Malik, J.: Shape and viewpoint without keypoints. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XV 16. pp. 88–104. Springer (2020) 4
28. González, Á.: Measurement of areas on a sphere using fibonacci and latitude–longitude lattices. Mathematical Geosciences **42**, 49–64 (2010) 22, 24
29. Groueix, T., Fisher, M., Kim, V.G., Russell, B.C., Aubry, M.: AtlasNet: A Papier-Mâché approach to learning 3D surface generation (Feb 2018) 4
30. Gu, J., Trevithick, A., Lin, K.E., Susskind, J., Theobalt, C., Liu, L., Ramamoorthi, R.: Nerfdiff: Single-image view synthesis with nerf-guided distillation from 3d-aware diffusion. In: ICML (2023) 4
31. He, Z., Wang, T.: Openlrm: Open-source large reconstruction models. `https://github.com/3DTopia/OpenLRM` (2023) 11, 12, 23
32. Hong, S., Ahn, D., Kim, S.: Debiasing scores and prompts of 2D diffusion for view-consistent Text-to-3D generation (Mar 2023) 5
33. Hong, Y., Zhang, K., Gu, J., Bi, S., Zhou, Y., Liu, D., Liu, F., Sunkavalli, K., Bui, T., Tan, H.: LRM: Large reconstruction model for single image to 3D (Nov 2023) 4, 12, 23
34. Huang, Y., Wang, J., Shi, Y., Qi, X., Zha, Z.J., Zhang, L.: DreamTime: An improved optimization strategy for Text-to-3D content creation (Jun 2023) 5
35. Jain, A., Mildenhall, B., Barron, J.T., Abbeel, P., Poole, B.: Zero-shot text-guided object generation with dream fields. CVPR (2022) 5
36. Jakob, W., Speierer, S., Roussel, N., Nimier-David, M., Vicini, D., Zeltner, T., Nicolet, B., Crespo, M., Leroy, V., Zhang, Z.: Mitsuba 3 renderer (2022), https://mitsuba-renderer.org 22
37. James, R.M.: Topology. Prentic Hall of India Private Limited, New delhi (2000) 7

38. Jang, W., Agapito, L.: CodeNeRF: Disentangled neural radiance fields for object categories (Sep 2021) 4
39. Jun, H., Nichol, A.: Shap-E: Generating conditional 3D implicit functions (May 2023) 11, 12, 23
40. Kanazawa, A., Tulsiani, S., Efros, A.A., Malik, J.: Learning category-specific mesh reconstruction from image collections. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 371–386 (2018) 4
41. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Transactions on Graphics **42**(4) (2023) 2, 4, 6
42. Khalid, N.M., Xie, T., Belilovsky, E., Tiberiu, P.: Clip-mesh: Generating textured meshes from text using pretrained image-text models. SIGGRAPH Asia 2022 Conference Papers (December 2022) 5
43. Laine, S., Hellsten, J., Karras, T., Seol, Y., Lehtinen, J., Aila, T.: Modular primitives for high-performance differentiable rendering. ACM Transactions on Graphics **39**(6) (2020) 10
44. Li, W., Chen, R., Chen, X., Tan, P.: SweetDreamer: Aligning geometric priors in 2D diffusion for consistent Text-to-3D (Oct 2023) 5
45. Li, X., Liu, S., Kim, K., De Mello, S., Jampani, V., Yang, M.H., Kautz, J.: Self-supervised single-view 3d reconstruction via semantic consistency. In: ECCV (2020) 4
46. Lin, C.H., Gao, J., Tang, L., Takikawa, T., Zeng, X., Huang, X., Kreis, K., Fidler, S., Liu, M.Y., Lin, T.Y.: Magic3d: High-resolution text-to-3d content creation. In: CVPR (2023) 1, 4, 5
47. Liu, M., Xu, C., Jin, H., Chen, L., Mukund, V.T., Xu, Z., Su, H.: One-2-3-45: Any single image to 3D mesh in 45 seconds without Per-Shape optimization (Jun 2023) 1, 11, 12, 23
48. Liu, M., Xu, C., Jin, H., Chen, L., T, M.V., Xu, Z., Su, H.: One-2-3-45: Any single image to 3d mesh in 45 seconds without per-shape optimization (2023) 4
49. Liu, R., Wu, R., Hoorick, B.V., Tokmakov, P., Zakharov, S., Vondrick, C.: Zero-1-to-3: Zero-shot one image to 3d object (2023) 1, 4, 11, 12, 23
50. Liu, Y., Lin, C., Zeng, Z., Long, X., Liu, L., Komura, T., Wang, W.: SyncDreamer: Generating multiview-consistent images from a single-view image (Sep 2023) 1, 4, 12, 14, 23
51. Liu, Y., Lin, C., Zeng, Z., Long, X., Liu, L., Komura, T., Wang, W.: Syncdreamer: Learning to generate multiview-consistent images from a single-view image. arXiv preprint arXiv:2309.03453 (2023) 11, 12
52. Liu, Z., Feng, Y., Black, M.J., Nowrouzezahrai, D., Paull, L., Liu, W.: Meshdiffusion: Score-based generative 3d mesh modeling. arXiv preprint arXiv:2303.08133 (2023) 1
53. Long, X., Guo, Y.C., Lin, C., Liu, Y., Dou, Z., Liu, L., Ma, Y., Zhang, S.H., Habermann, M., Theobalt, C., Wang, W.: Wonder3D: Single image to 3D using Cross-Domain diffusion (Oct 2023) 1, 4, 11, 12, 23, 24
54. Ma, B., Deng, H., Zhou, J., Liu, Y.S., Huang, T., Wang, X.: Geodream: Disentangling 2d and geometric priors for high-fidelity and consistent 3d generation. arXiv preprint arXiv:2311.17971 (2023) 14
55. Mandikal, P., Navaneet, K.L., Agarwal, M., Venkatesh Babu, R.: 3D-LMNet: Latent embedding matching for accurate and diverse 3D point cloud reconstruction from a single image (Jul 2018) 4
56. Melas-Kyriazi, L., Rupprecht, C., Laina, I., Vedaldi, A.: RealFusion: 360° reconstruction of any object from a single image (Feb 2023) 1, 4, 11, 12, 23

57. Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., Geiger, A.: Occupancy networks: Learning 3d reconstruction in function space. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4460–4470 (2019) 4
58. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. ECCV (2020) 1, 5
59. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM **65**(1), 99–106 (2021) 4
60. Mittal, P., Cheng, Y.C., Singh, M., Tulsiani, S.: Autosdf: Shape priors for 3d completion, reconstruction and generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 306–315 (2022) 4
61. Müller, N., Simonelli, A., Porzi, L., Bulo, S.R., Nießner, M., Kontschieder, P.: Autorf: Learning 3d object radiance fields from single view observations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3971–3980 (2022) 4
62. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. ACM Transactions on Graphics (ToG) **41**(4), 1–15 (2022) 4
63. Munkberg, J., Hasselgren, J., Shen, T., Gao, J., Chen, W., Evans, A., Müller, T., Fidler, S.: Extracting Triangular 3D Models, Materials, and Lighting From Images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 8280–8290 (June 2022) 24
64. Nichol, A., Jun, H., Dhariwal, P., Mishkin, P., Chen, M.: Point-E: A system for generating 3D point clouds from complex prompts (Dec 2022) 11, 12, 23
65. Nicolet, B., Jacobson, A., Jakob, W.: Large steps in inverse rendering of geometry. ACM Transactions on Graphics (TOG) **40**(6), 1–13 (2021) 2, 3, 15, 22, 25, 26
66. Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S.: Deepsdf: Learning continuous signed distance functions for shape representation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 165–174 (2019) 4
67. Poole, B., Jain, A., Barron, J.T., Mildenhall, B.: Dreamfusion: Text-to-3d using 2d diffusion. arXiv preprint arXiv:2209.14988 (2022) 1, 5
68. Qian, G., Mai, J., Hamdi, A., Ren, J., Siarohin, A., Li, B., Lee, H.Y., Skorokhodov, I., Wonka, P., Tulyakov, S., Ghanem, B.: Magic123: One image to high-quality 3d object generation using both 2d and 3d diffusion priors. arXiv preprint arXiv:2306.17843 (2023) 1, 11, 12, 14, 23
69. Qiu, L., Chen, G., Gu, X., Zuo, Q., Xu, M., Wu, Y., Yuan, W., Dong, Z., Bo, L., Han, X.: RichDreamer: A generalizable Normal-Depth diffusion model for detail richness in Text-to-3D (Nov 2023) 5, 11, 23, 24
70. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: ICML (2021) 5
71. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: CVPR (2022) 5
72. Roth, J., Tong, Y., Liu, X.: Adaptive 3d face reconstruction from unconstrained photo collections. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4197–4206 (2016) 4

73. Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E.L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., et al.: Photorealistic text-to-image diffusion models with deep language understanding. NeurIPS (2022) 5
74. Schüller, C., Kavan, L., Panozzo, D., Sorkine-Hornung, O.: Locally injective mappings. In: Computer Graphics Forum. vol. 32, pp. 125–135. Wiley Online Library (2013) 8
75. Shen, B., Yan, X., Qi, C.R., Najibi, M., Deng, B., Guibas, L., Zhou, Y., Anguelov, D.: GINA-3D: Learning to generate implicit neural assets in the wild. arXiv [cs.CV] (Apr 2023) 4
76. Shen, Q., Yang, X., Wang, X.: Anything-3D: Towards single-view anything reconstruction in the wild (Apr 2023) 4
77. Shen, T., Gao, J., Yin, K., Liu, M.Y., Fidler, S.: Deep marching tetrahedra: a hybrid representation for high-resolution 3d shape synthesis. In: NeurIPS (2021) 1, 4
78. Shi, Y., Wang, P., Ye, J., Long, M., Li, K., Yang, X.: MVDream: Multi-view diffusion for 3D generation (Aug 2023) 5
79. Sifakis, E., Barbic, J.: Fem simulation of 3d deformable solids: a practitioner's guide to theory, discretization and model reduction. In: Acm siggraph 2012 courses, pp. 1–50 (2012) 7, 22
80. Sun, J., Zhang, B., Shao, R., Wang, L., Liu, W., Xie, Z., Liu, Y.: DreamCraft3D: Hierarchical 3D generation with bootstrapped diffusion prior (Oct 2023) 1, 4, 11, 12, 14, 23
81. Tang, J., Chen, Z., Chen, X., Wang, T., Zeng, G., Liu, Z.: LGM: Large Multi-View gaussian model for High-Resolution 3D content creation (Feb 2024) 4
82. Tang, J., Ren, J., Zhou, H., Liu, Z., Zeng, G.: Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. arXiv preprint arXiv:2309.16653 (2023) 4
83. Tang, J., Ren, J., Zhou, H., Liu, Z., Zeng, G.: DreamGaussian: Generative gaussian splatting for efficient 3D content creation (Sep 2023) 11, 12, 14, 23
84. Tang, J., Wang, T., Zhang, B., Zhang, T., Yi, R., Ma, L., Chen, D.: Make-it-3d: High-fidelity 3d creation from a single image with diffusion prior. arXiv preprint arXiv:2303.14184 (2023) 5, 14
85. Tang, S., Chen, J., Wang, D., Tang, C., Zhang, F., Fan, Y., Chandra, V., Furukawa, Y., Ranjan, R.: MVDiffusion++: A dense high-resolution multi-view diffusion model for single or sparse-view 3D object reconstruction (Feb 2024) 4
86. Tang, S., Zhang, F., Chen, J., Wang, P., Furukawa, Y.: MVDiffusion: Enabling holistic multi-view image generation with Correspondence-Aware diffusion (Jul 2023) 4
87. Tsalicoglou, C., Manhardt, F., Tonioni, A., Niemeyer, M., Tombari, F.: TextMesh: Generation of realistic 3D meshes from text prompts (Apr 2023) 5
88. Tulsiani, S., Zhou, T., Efros, A.A., Malik, J.: Multi-view supervision for single-view reconstruction via differentiable ray consistency. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2626–2634 (2017) 4
89. Wang, H., Du, X., Li, J., Yeh, R.A., Shakhnarovich, G.: Score jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation. CVPR (2023) 5
90. Wang, N., Zhang, Y., Li, Z., Fu, Y., Liu, W., Jiang, Y.G.: Pixel2mesh: Generating 3d mesh models from single rgb images. In: Proceedings of the European conference on computer vision (ECCV). pp. 52–67 (2018) 4
91. Wang, P., Liu, L., Liu, Y., Theobalt, C., Komura, T., Wang, W.: NeuS: Learning neural implicit surfaces by volume rendering for multi-view reconstruction (Jun 2021) 1

92. Wang, P., Tan, H., Bi, S., Xu, Y., Luan, F., Sunkavalli, K., Wang, W., Xu, Z., Zhang, K.: PF-LRM: Pose-Free large reconstruction model for joint pose and shape prediction (Nov 2023) 4
93. Wang, Z., Lu, C., Wang, Y., Bao, F., Li, C., Su, H., Zhu, J.: Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. NeurIPS (2023) 5
94. Wang, Z., Lu, C., Wang, Y., Bao, F., Li, C., Su, H., Zhu, J.: Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. Advances in Neural Information Processing Systems **36** (2024) 1
95. Wen, J., Wang, B., Barbic, J.: Large-strain surface modeling using plasticity. IEEE Transactions on Visualization and Computer Graphics (2023) 8
96. Weng, Z., Liu, J., Tan, H., Xu, Z., Zhou, Y., Yeung-Levy, S., Yang, J.: Single-View 3D human digitalization with large reconstruction models (Jan 2024) 4
97. Xie, H., Yao, H., Sun, X., Zhou, S., Zhang, S.: Pix2Vox: Context-aware 3D reconstruction from single and multi-view images (Jan 2019) 4
98. Xie, H., Yao, H., Zhang, S., Zhou, S., Sun, W.: Pix2Vox++: Multi-scale context-aware 3D object reconstruction from single and multiple images. Int. J. Comput. Vis. **128**(12), 2919–2935 (Dec 2020) 4
99. Xu, D., Jiang, Y., Wang, P., Fan, Z., Wang, Y., Wang, Z.: NeuralLift-360: Lifting an in-the-wild 2D photo to a 3D object with 360° views (Nov 2022) 4
100. Xu, Q., Wang, W., Ceylan, D., Mech, R., Neumann, U.: DISN: Deep implicit surface network for high-quality single-view 3D reconstruction (May 2019) 4
101. Xu, Y., Tan, H., Luan, F., Bi, S., Wang, P., Li, J., Shi, Z., Sunkavalli, K., Wetzstein, G., Xu, Z., Zhang, K.: DMV3D: Denoising Multi-View diffusion using 3D large reconstruction model (Nov 2023) 4
102. Yang, S., Liu, J., Wang, W., Guo, Z.: Tet-gan: Text effects transfer via stylization and destylization. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 1238–1245 (2019) 4
103. Yu, A., Ye, V., Tancik, M., Kanazawa, A.: pixelNeRF: Neural radiance fields from one or few images (Dec 2020) 4

A Dense-view Inverse Rendering

Fig. 10, 11, and 12 show the results of 3D shape reconstruction from dense multi-view images. These multi-view images were captured by densely sampling camera positions around each object (employing 360 views sampled on a sphere based on a Fibonacci sequence [28]) and subsequently rendered using Mitsuba [36]. The chosen shapes for this demonstration are from the GSO dataset.

To establish the efficacy of our method, we compare its performance with that of the state-of-the-art surface mesh-based approach as described in [65] by applying both techniques to the aforementioned dense multi-view images. Our method utilizes a consistent hyperparameter set for all shapes undergoing $3,000$ optimization iterations, mirroring the parameters described in the main paper. In contrast, for the method in [65], we adhere to the parameters in the original paper—namely, using $\lambda = 19$ and a step size of 10^{-2}, and conducting $10,000$ iterations of optimization. This adjustment is necessary due to difficulties in achieving convergence within a few thousand iterations for this method.

While both methods effectively minimize the rendering loss, our method consistently delivers results of superior quality. Fig. 10 and 11 illustrate that our reconstructions are devoid of common artifacts such as wiggles or kinks, and showcase high-quality triangles, as evidenced by the wireframe representations. Furthermore, our approach demonstrates faster convergence and notable scalability advantages when compared to the second-order method in [65]. Additionally, as Fig. 12 indicates, our method is also capable of handling shapes with complex topologies, further underscoring its versatility and effectiveness.

B Additional Results

B.1 Qualitative Results on Single-view Reconstruction

Fig. 13 shows the results of single-view reconstruction performed on the GSO dataset. Our TetSphere demonstrates superior mesh quality, effectively capturing sharp geometric features, including the boundaries of the shoes.

B.2 More Results on Text-to-3D Shape Generation

Fig. 14 shows additional results of text-to-3D shape generation. These results highlight our method's capability to construct complicated material, such as reflections, by leveraging the explicit geometry representation.

C Tetrahedron and its Deformation Gradient

Following [79], we treat a tetrahedron as a piecewise linear element. The initial (undeformed) positions of the four vertices of a tetrahedron are denoted by $\mathbf{X} = [\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \mathbf{X}^{(3)}, \mathbf{X}^{(4)}]$, with each $\mathbf{X}^{(i)} \in \mathbb{R}^3$. Similarly, the positions of the four vertices of a deformed tetrahedron are represented by $\mathbf{x} = [\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \mathbf{x}^{(3)}, \mathbf{x}^{(4)}]$, where each $\mathbf{x}^{(i)} \in \mathbb{R}^3$. The deformation gradient $\mathbf{F} \in \mathbb{R}^{3\times 3}$, which quantifies the local deformation of the tetrahedron, is given by:

$$\mathbf{F} = \mathbf{D}_s \mathbf{D}_m^{-1}, \tag{3}$$

$$\mathbf{D}_s := \left[\mathbf{x}^{(1)} - \mathbf{x}^{(4)} \;\; \mathbf{x}^{(2)} - \mathbf{x}^{(4)} \;\; \mathbf{x}^{(3)} - \mathbf{x}^{(4)} \right], \tag{4}$$

$$\mathbf{D}_m := \left[\mathbf{X}^{(1)} - \mathbf{X}^{(4)} \;\; \mathbf{X}^{(2)} - \mathbf{X}^{(4)} \;\; \mathbf{X}^{(3)} - \mathbf{X}^{(4)} \right]. \tag{5}$$

The deformation gradient $\mathbf{F}$ essentially captures how a tetrahedron transforms from its initial state to its deformed state, encompassing both rotation and stretching effects.

D Formulation Details of TetSphere Initialization

Assuming there are a total of m candidate positions obtained from the coarse voxel grid (as shown in Fig. 5), our goal with TetSphere initialization is to select a subset of these candidate points such that the object's shape is adequately covered by tetrahedral spheres centered at these positions. We first initialize a sphere of fixed radius at each candidate position, where the radius is calculated as $\alpha r + \beta$, where r is the minimum distance from each candidate position to the voxel surface. We use $\alpha = 1.2, \beta = 0.07$ in all our examples. The objective is to select a subset of spheres that collectively cover all voxel positions.

We define a coverage matrix $\mathbf{D} \in \{0,1\}^{m\times m}$, where each element $d_{ji} \in \{0,1\}$ indicates whether voxel position j is covered by a sphere centered on candidate position i. A binary vector $\mathbf{v} \in \{0,1\}^m$ identifies selected candidate positions, with each element denoting the selection status of corresponding voxel positions. The selection of feature points is formulated as a mixed-integer linear programming problem:

$$\min_{\mathbf{v}} |\mathbf{v}| \quad \text{s.t.} \quad \mathbf{D}\mathbf{v} \geq \mathbb{1}, \tag{6}$$

where $|\cdot|$ is the l_1 norm, $\mathbb{1}$ is a vector of ones, and n is the predetermined maximum number of tetrahedral spheres. This optimization is efficiently solved using standard linear programming solvers.

E Implementation Details

Baselines. For the baseline methods, including RealFusion [56], Magic123 [68], One-2-3-45 [47], Point-E [64], Shap-E [39], Zero123 [49], SyncDreamer [50], Wonder3d [53], Open-LRM [31, 33], DreamGaussian [83], Dreamcraft3D [80], and RichDreamer [69], we follow their original implementations and use their publically available codebase for getting the results in this paper.

Our method. Our implementation of the TetSphere initialization algorithm is developed in C++ and makes use of the GUROBI linear programming solver.

The optimization of geometric energies is implemented using CUDA as a PyTorch extension to enhance computational efficiency. 1) For single-view reconstruction, Wonder3d [53] is employed to generate six multi-view images, including both color and normal images for each view, using predefined camera poses for reconstruction with TetSphere. A 2-layer MLP is utilized for texture representation. The optimization objective, $\mathbf{\Phi}(\cdot)$, encompasses both the rendering loss and the normal loss. The rendering loss consists of an l_1 norm on tone-mapped color and MSE on the alpha mask, along with a cosine loss on normals, similar to that described in [63]; 2) For the image-to-3D shape generation, multi-view images are obtained from the initial stage of DreamCraft3D (coarse NeRF fitting only), generating 360 views sampled on a Fibonacci sphere [28]. The texture optimization process directly employs a 2048×2048 2D texture image, circumventing the need for additional neural networks. Here, the optimization objective, $\mathbf{\Phi}(\cdot)$, focuses solely on the rendering loss; 3) For text-to-3D generation, the initial stage of RichDreamer [69] is used to obtain multi-view images. In this scenario, our TetSphere is optimized using both the rendering loss and the SDS loss. The SDS loss for geometry is calculated using the Normal-depth diffusion model as described in [69]. Likewise, the SDS loss for texture leverages the albedo diffusion model from the same work. A 2-layer MLP is implemented to parameterize and optimize the PBR material of the shape. To enhance robustness and efficiency, a cosine scheduler is applied to scale the coefficients of the geometry loss for all applications, formulated as $\eta = 4^{\sin(\frac{t\pi}{2n})}$, where t denotes the current iteration and T represents the total number of iterations.

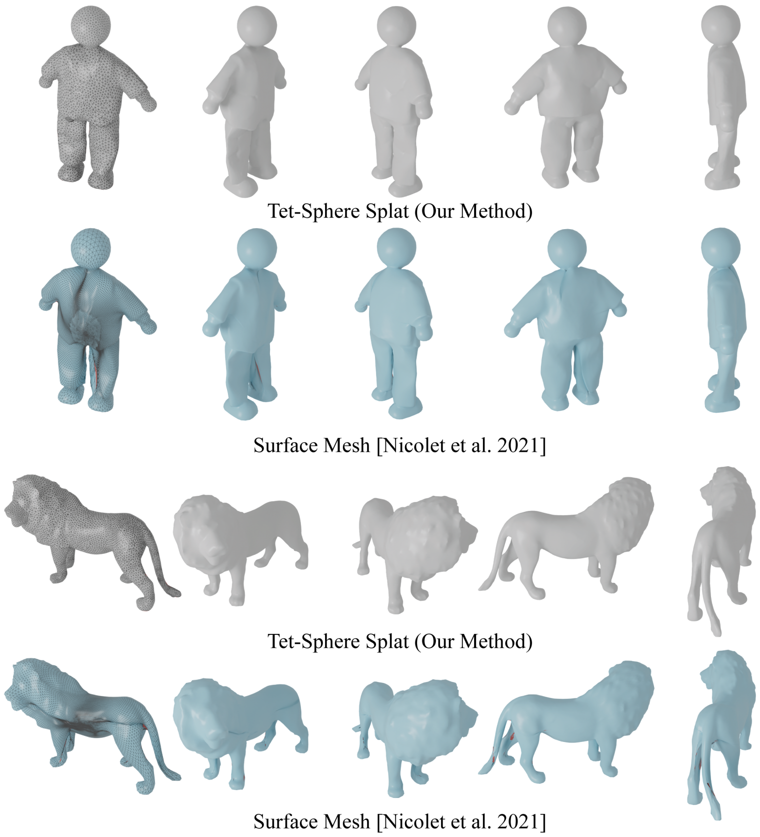
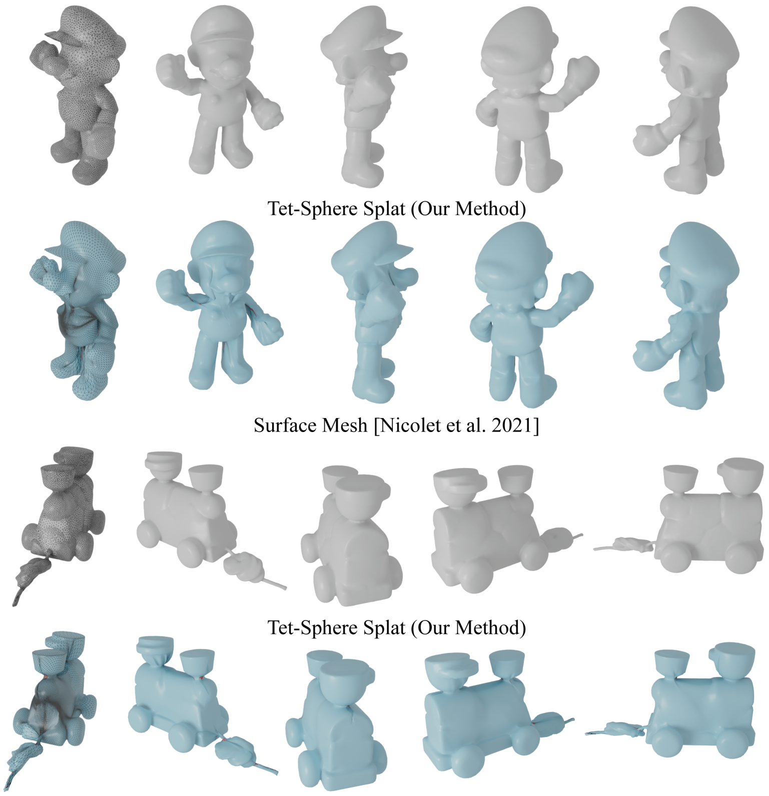

Tet-Sphere Splat (Our Method)
Surface Mesh [Nicolet et al. 2021]
Tet-Sphere Splat (Our Method)
Surface Mesh [Nicolet et al. 2021]

Fig. 10: Results on 3D shape reconstruction from dense multi-view images (1/2). Shapes reconstructed using our method are depicted in gray, whereas those reconstructed with the state-of-the-art method from [65] are presented in blue. While both methods effectively minimize the rendering loss, our technique does not produce artifacts, such as undesired wrinkles and flipping triangles (highlighted in red), and producing high-quality triangles, as illustrated by the wireframes.



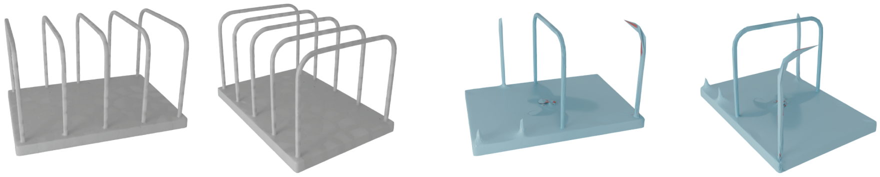
Tet-Sphere Splat (Our Method)

Surface Mesh [Nicolet et al. 2021]

Tet-Sphere Splat (Our Method)

Surface Mesh [Nicolet et al. 2021]

Fig. 11: Results on 3D shape reconstruction from dense multi-view images (2/2).



Tet-Sphere Splat (Our Method)

Surface Mesh [Nicolet et al. 2021]

Fig. 12: Our method is capable of handling shapes with complex topologies, an advantage that existing methods fail to achieve [65]. In this example, both shapes are computed by incorporating the rendering loss and normal loss.

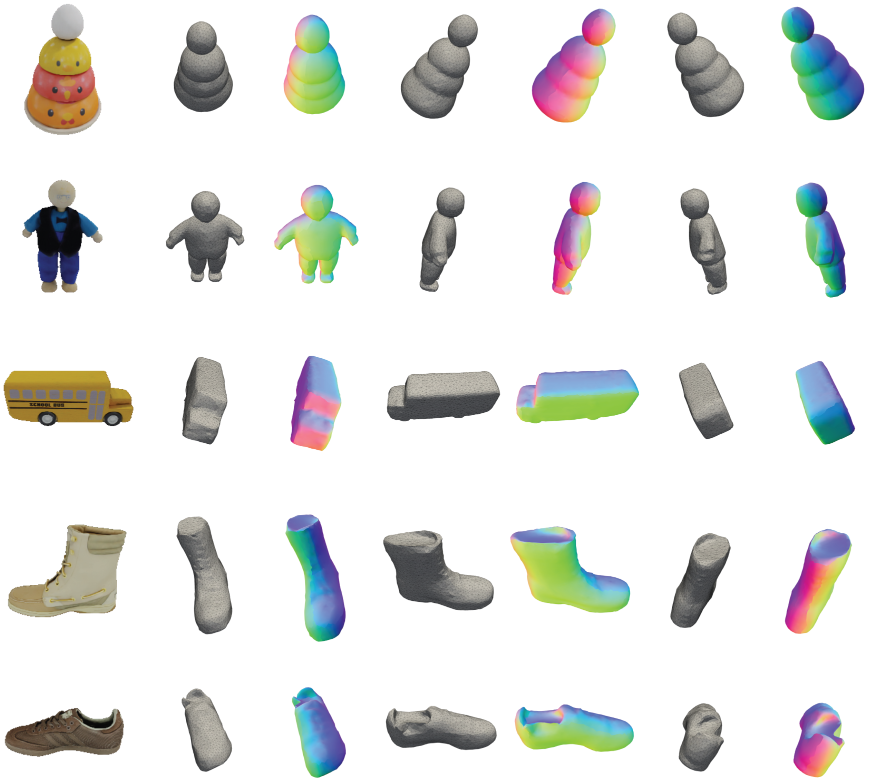
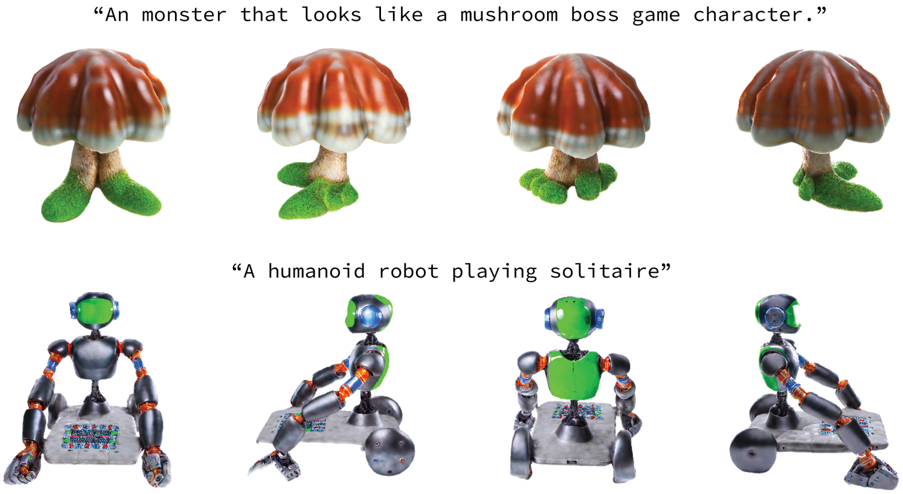


Fig. 13: Qualitative results on single-view reconstruction.



“An monster that looks like a mushroom boss game character.”

“A humanoid robot playing solitaire”

Fig. 14: More results on text-to-3D shape generation.