2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) | 979-8-3315-9994-2/25/$31.00 ©2025 IEEE | DOI: 10.1109/CVPRW67362.2025.00017

SplatTouch: Explicit 3D Representation Binding Vision and Touch

Antonio Luigi Stefani, Niccolò Bisagno, Nicola Conci, and Francesco De Natale
University of Trento - Department of Information Engineering and Computer Science
{name.surname}@unitn.it

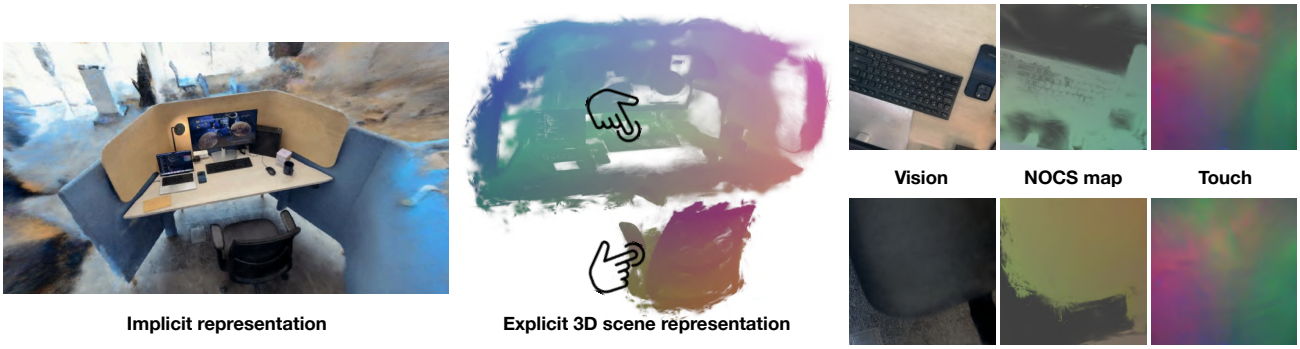

Figure 1. Starting from a set of images and sparse touch samples, we generate an implicit NeRF representation alongside a 3D Gaussian Splatting model as an explicit 3D scene representation. The explicit 3D model is then transformed into Normalized Object Coordinate Space (NOCS), enabling the extraction of NOCS maps, where each 3D position is color coded, as scene-level descriptors. This allow to effectively bind the images from the vision domain to the 3D localization of touch samples in the scene, allowing a step forward toward fully touchable 3D scenes.

Abstract

When compared to standard vision-based sensing, touch images generally captures information of a small area of an object, without context, making it difficult to collate them to build a fully touchable 3D scene. Researchers have leveraged generative models to create tactile maps (images) of unseen samples using depth and RGB images extracted from implicit 3D scene representations. Being the depth map referred to a single camera, it provides sufficient information for the generation of a local tactile maps, but it does not encode the global position of the touch sample in the scene. In this work, we introduce a novel explicit representation for multi-modal 3D scene modeling that integrates both vision and touch. Our approach combines Gaussian Splatting (GS) for 3D scene representation with a diffusion-based generative model to infer missing tactile information from sparse samples, coupled with a contrastive approach for 3D touch localization. Unlike NeRF-based implicit methods, Gaussian Splatting enables the computation of an absolute 3D reference frame via Normalized Object Coordinate Space (NOCS) maps, facilitating structured, 3D-aware tactile generation. This framework not only improves tactile sample prompting but also enhances 3D tactile localization, overcoming the local constraints of prior implicit approaches. We demonstrate the effectiveness of our method in generating novel touch samples and localizing tactile interactions in 3D. Our results show that explicitly incorporating tactile information into Gaussian Splatting improves multi-modal scene understanding, offering a significant step toward integrating touch into immersive virtual environments. The code is available at: https://github.com/mmlab-cv/SplatTouch

1. Introduction

As humans, we enjoy our daily interactions with objects in a multi-modal fashion, exploiting one or more senses at the same time, such as sight, hearing, and touch. When dealing with the virtualization of real-world in VR/AR/XR and robotic applications, the sensing experience should be faithfully replicated [1, 29]. The literature shows that sight and hearing have been extensively explored and integrated, as standard pipelines for collecting, processing, and delivering visual and auditory information are well established. Vicev-

2160-7516/25/$31.00 ©2025 IEEE
DOI 10.1109/CVPRW67362.2025.00017

ersa, tactile sensing has been lagging behind, with no standardized (or recognized de-facto) pipeline made available [26].

As of today, the collection of tactile data has primarily relied on two classes of sensors to map a small patch of the real world data into a 2D tactile map, or sample. More in detail, the literature refers to the use of either inertial sensors [16] —such as accelerometers, probes, and force sensors— or vision-based approaches [4, 6, 8, 34], which capture high-resolution 2D information in the form of haptic/tactile maps. The latter approach utilizes a camera to record the deformation of a membrane upon contact with a surface, as shown by the touch sample in Fig. 1.

The collection of tactile data from a 3D scene requires users to sample multiple data points at different spatial locations. Most datasets jointly collect visual and tactile information by using an RGB camera to record the capture of each touched point [13, 31], for visual matching and contextual purposes. However, this results in occlusion of the touched area, as the sensor and the human or robotic hand frequently obstruct the view of the area being sensed. In TaRF [7], these limitations are overcome by rigidly attaching the RGB camera and the tactile sensor. This setup ensures that each sample consists of an RGB image—suitable for training a Neural Radiance Field (NeRF)—along with a touch sample whose position in the 3D scene can be retrieved through its relationship with the camera.

However, while RGB images capture overlapping samples that provide contextual information about the scene, tactile maps present a significant challenge due to the sparsity of the collected data, which typically consists of non-overlapping discrete points. This discrete and sparse nature makes it difficult to construct a comprehensive scene model where haptic data is available for each 3D position of the observed scene.

To address this issue, generative models, such as Generative Adversarial Networks (GANs) [10, 21, 35] and diffusion models [7, 32], have been employed for cross-modal touch estimation in novel views. For example, given a touch sample from a portion of a wooden surface, the goal is to extend this knowledge to the entire surface by leveraging contextual RGB information. This is done using generated multi-scale RGB images and depth maps extracted from a NeRF model as contextual information for touch estimation [7]. However, depth maps provide a *relative* contextual information, as they refer to the local camera position; this turns out to be incomplete, when attempting to estimate the *absolute* position of a queried touch in the 3D scene.

In this work, we propose a generative approach to extend local tactile information and infer a more comprehensive understanding of the 3D scene. To achieve this, we introduce three key components: an explicit 3D scene representation using Gaussian Splatting, a generative model based on diffusion techniques, and a contrastive method for 3D touch sample localization. Unlike current implicit approaches that compute depth maps as local spatial scene information, our explicit representation via Gaussian splatting enables the computation of an *absolute* depth map, namely a Normalized Object Coordinate Space (NOCS) map [28], where RGB values correspond to precise 3D positions in space. By extracting NOCS maps, we provide the diffusion model with a structured 3D understanding of the scene, obtaining improved results on the generation of missing tactile samples. Moreover, we show how the tactile data generated can be used to train a contrastive framework, leading to improvements on the 3D tactile localization task, thanks to the scene-level information provided by NOCS.

Our contributions can be summarized as:

- Binding together visual and tactile domain using NOCS map as a global descriptor of the scene.
- Outperforming current approaches in the cross-modal touch generation task thanks to the NOCS representation.
- Introducing a novel contrastive learning between NOCS and touch domains for the 3D localization task.
- Establishing a novel 3D localization task, where the goal is to accurately retrieve the position of a queried touch sample in the 3D scene.

2. Related work

NOCS map as scene representation. First introduced in the context of object pose estimation [28], Normalized Object Coordinate Space (NOCS) aim at providing a shared canonical representation for all possible object instances within a category. NOCS maps have commonly indicated the 2D projection of the canonical representation on an image plane [28, 30]. In [15, 30], the authors use diffusion models to generate multiple NOCS maps of different viewpoints, effectively tricking generative models to provide a 2D representation of the 3D world. In our work we propose to apply the NOCS representation on GS at a scene level for a dual goal: (I) providing the diffusion model with a scene-level global context information for the cross-modal generation of novel touch samples, and (II) aiding the touch 3D localization task.

Cross-modal touch generation. Cross-modal touch generation aims at estimating the touch signal given an RGB image. Recently, this task has gained popularity in many real-world applications, since it allows to collect data according to one modality, and successively transfer the information to a different one [7, 9, 11, 20, 24, 32, 33]. Depending on the data at hand, the task can be divided into two main branches, considering whether the RGB images contain the human/robotic hand that is touching the object interest, thus effectively leading to the object of interest to be occluded in the RGB image [9, 10, 20, 24, 32], or not [7, 11]. Both classes of methods rely only on contextual RGB infor-

mation to infer the necessary information for generating the touch samples.

Recently, a novel line of works [7, 33, 35] has tried to combine different pieces of information from different domains. In [33] the authors bind touch to both visual and natural languages domains; however, they do not provide any 3D scene representation. In [35] NeRF-rendered RGB-D images are fed as inputs to a conditional Generative Adversarial Network model (cGAN) to generate tactile samples. Similarly, in [7] the authors extract the depth and visual data from a 3D scene representation, for a better output. The depth represents a local descriptor of the scene, contributing to the generation task, yet not allowing to retrieve the 3D position of the generated sample in the scene. In our work we include the scene information via the NOCS maps, thus providing the absolute 3D position in the scene.

3D representation and localization of tactile signals. The task of tactile signal localization involves accurately determining the position of a query touch signal within a scene. In [12], the problem is approached from an opposite perspective, leveraging the generated tactile signal to enhance the fine-grained reconstruction of 3D objects. In [11], given an object's mesh and different sensory observations of the contact position (such as visual images, impact sounds, or tactile readings), the goal is to predict the exact location on the mesh where the contact occurs using a simple regression model. In [7], rather than directly retrieving the touch location, a contrastive learning framework is used as a lookup table to infer the position of the RGB camera that captures the scene. In our work, we retrieve the precise position of the touch query, by introducing a novel contrastive learning framework that jointly aligns the tactile and 3D representation domains.

3. Method

Starting from paired image and touch samples, our goal is to:

1. register each pair of samples adopting an explicit 3D representation;
2. obtain a scene-level NOCS descriptor;
3. generate the tactile signal for any point in the scene;
4. retrieve the accurate 3D position of a given touch sample.

As for the first task, we leverage Gaussian Splatting (GS) [17], defined as $\mathcal{F}_1 : \mathcal{V} \rightarrow (p, \Sigma, \alpha, c)$. Given a video $\mathcal{V}$ of a static scene, GS reconstructs a set of 3D Gaussians, where each Gaussian is characterized by its position $p = (x, y, z)$, covariance matrix Σ, opacity α, and color c. From GS, we generate an image I and the corresponding camera parameters $[R|t]$, which is registered to the collected touch sample τ.

Next, given the explicit GS representation, we generate a scene-level Normalized Object Coordinate Space (NOCS) [28] map η for each image I such that $\mathcal{F}_2 : (I, [R|t], p, \Sigma, \alpha, c) \rightarrow \eta$.

To generate the missing touch samples of the visual 3D model, we introduce a diffusion model [25] $\mathcal{F}_3 : (I, \eta) \rightarrow \tau$ that generates the tactile signal τ at the center of an image I. funding, h We then define a contrastive learning framework [2] $\mathcal{F}_4 : \tau \rightarrow \mu$ that aims to retrieve the 3D position of a given query touch within the scene.

In the following sections, we describe in detail $\mathcal{F}_1$, $\mathcal{F}_2$, $\mathcal{F}_3$, and $\mathcal{F}_4$.

3.1. Touch and Visual Signal Registration

We construct a visual 3D reconstruction $\mathcal{F}_1$ using Gaussian Splatting (GS) from the video feed $\mathcal{V}$ of the scene. GS provides an explicit scene representation, as the position p of all Gaussians is known.

In TaRF [7], tactile and visual signals are captured simultaneously using a DIGIT [19] touch sensor rigidly attached to the RGB camera. To obtain the synthetic images I, depth maps D and their corresponding camera parameters $[R|t]$ that are visually aligned with the touch samples τ, the dataset leverages NeRF to synthesize virtual novel views.

Since both the NeRF model and our GS representation are derived from the same input images, the generated novel views are aligned to both models. Consequently, the synthetically generated images I are also well-aligned with our 3D model. We prefer to use these NeRF-generated images rather than generating new ones with GS, because NeRF is known to achieve superior image quality [36]. Thus, we obtain images I, depth maps D and touch samples τ that are registered to our GS.

3.2. Scene-Level NOCS Map Generation

We define a method $\mathcal{F}_2$ to explicitly compute a 3D representation using scene-level Normalized Object Coordinate Space (NOCS). The goal of the representation is to compute 2D NOCS maps η that are closely aligned with the images I and touch samples τ. The 2D NOCS maps effectively act as a 3D descriptor, binding the camera view I with the 3D position of the touch sample τ.

Unlike the implicit NeRF representation which would need a neural network to precisely estimate the NOCS representation [22], our explicit representation enables precise computation of scene-level descriptors.

The process, shown in Fig. 2, consists of four steps:

1. Outlier removal
2. Space normalization
3. Gaussian cloud recoloring
4. NOCS map generation

Outlier Removal. GS is optimized to obtain the lowest possible Peak Signal-to-Noise Ratio (PSNR) on input views, not on the 3D scene structure. This can result in

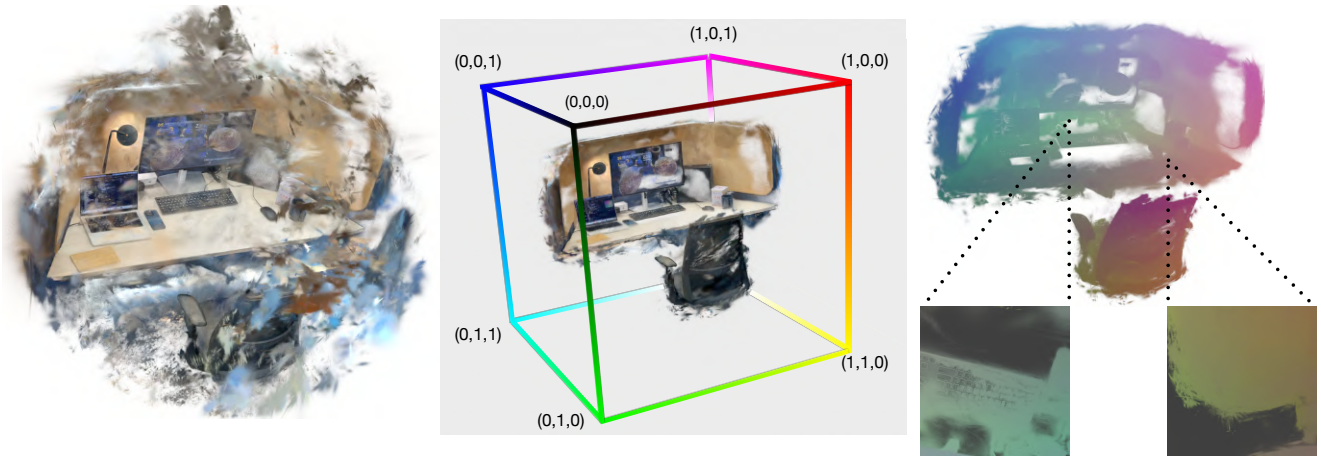

Figure 2. Our scene-level NOCS map generation pipeline. We start with the Gaussian Splatting (GS) representation of the scene, which initially contains outliers (left). The scene is refined and normalized within the NOCS framework (center), where each 3D position in space is mapped to an RGB color sample for visualization. The Gaussian Splatting scene is recolored based on the NOCS representation, allowing for the extraction of NOCS maps using the same camera parameters as the vision samples (right).

Gaussians, especially those in the background, being projected far from the scene center. Since tactile samples of the background are unavailable in the dataset, we remove all Gaussians whose position p is more than 0.5 meters away from the closest available touch sample.

Space Normalization. Since NOCS [28] is defined within a unit cube, we normalize each scene by scaling it so that its smallest bounding box has a diagonal of length one. This is achieved by selecting the minimum position as the bottom corner and the maximum position as the top corner, ensuring that the normalized positions p_n of all Gaussians vary between -1 and 1.

Gaussian Recoloring. Our representation allows each Gaussian in the scene to be defined within the NOCS. We recolor each Gaussian based on its normalized position, modifying the representation to $(p, \Sigma, \alpha, c(p_n))$. This ensures that each Gaussian is colored according to its position in normalized space.

NOCS Map Generation. To generate the NOCS map η, we render an image of the recolored GS using the same parameters $[R|t]$ that were used to generate the corresponding images I and depth map D. Thus, we obtain a tuple (I, D, τ, η), where each touch sample and RGB image refers to a NOCS map that encodes their displacement in the 3D space, explicitly binding the two domains.

3.3. Missing Touch Generation

To estimate the touch signal (represented as a haptic map) for any location in the scene, we train a diffusion model $\mathcal{F}_3$ to generate the tactile signal given the images I, depth maps D and NOCS maps η extracted in previous steps.

Similarly to [7], we employ a stable latent generative model [25] coupled with an autoencoder [5], as shown in Fig. 3.

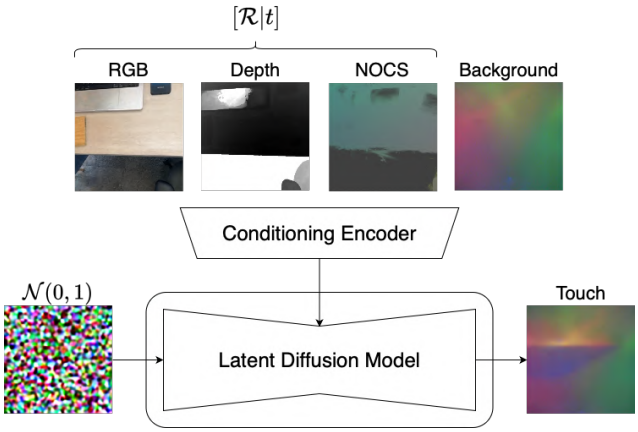

Figure 3. Our diffusion model for cross-modal touch generation. The conditioning vector encapsulates the aligned RGB, depth and NOCS maps to obtain the best possible representation of the novel touch sample τ at position $[R|t]$.

Unlike [7], the conditioning input of our diffusion model does not rely on multi-scale input, which provide an approximate and local 3D descriptor. Instead, we add the NOCS map generated by $\mathcal{F}_2$, which provides a precise scene-level descriptor. At training time, starting from gaussian noise $\mathcal{N}(0, 1)$, the diffusion model is conditioned by the NOCS map η, the RGB image I, the depth and the background image (haptic map produced by the sensor when not in contact) to try to estimate the haptic map τ.

At test time, given a novel location in the 3D scene, we first render the RGB I and depth from the NeRF scene, and the NOCS map η from the GS. We then use our diffusion

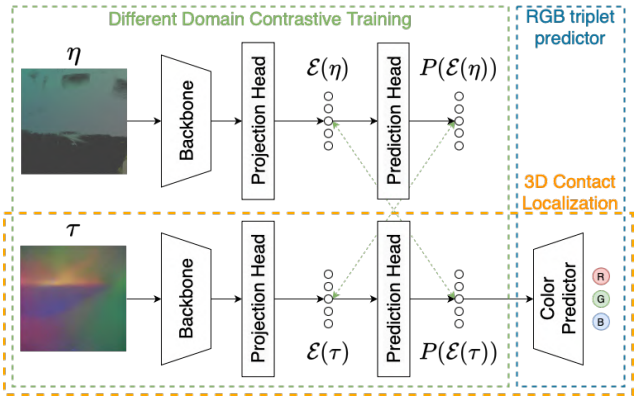

Figure 4. Our contrastive networks for 3D contact localization. NOCS and touch are jointly train to obtain representative feature. At test time, given a query touch, the 3D contact localization branch predicts the color that can be mapped to the 3D space thanks to the NOCS map representation.

model $\mathcal{F}_3$ to estimate the haptic map τ.

3.4. 3D Touch Localization

The task of touch localization in a scene is often formulated as a retrieval problem [7], where, given a query touch signal, the goal is to retrieve the closest image in the visual domain. Since images in the visual domain have known positions obtained from the structure from motion, the touch localization in 3D is taken as the RGB camera's position, and do not provide the exact touch position within the scene. Moreover, as also noted in [7], the relation between a touch sample and RGB camera location is not unique, as there are multiple positions at the same distance from which the touch sample can be collected. Instead, similar to touch localization on objects [11], our goal is to predict the precise touch position in the scene given a query touch.

More formally, the task is defined as follows: given an image I with its camera parameters $[R|t]$, a tactile signal τ, and the 3D representation of a scene (e.g., a point cloud or Gaussian Splatting), the model must localize the contact position on the 3D structure.

To achieve this, we introduce a contrastive learning framework $\mathcal{F}_4$ based on SimSiam [2], shown in Fig. 4. The objective of contrastive learning is to learn a representation that captures the relationships between the tactile maps and NOCS images, facilitating downstream tasks such as precise 3D localization. The training of our architecture is divided in two steps. In the first step, we train our framework, which consists of two branches—one for each domain. Each branch includes a backbone, a projection head, and a prediction head. The training objective is to minimize the negative cosine similarity between the two modalities:

$$\mathcal{L}(\eta,\tau) = -\frac{\mathcal{P}(\mathcal{E}(\eta))}{||\mathcal{P}(\mathcal{E}(\eta))||_2} \cdot \frac{\mathcal{E}(\tau)}{||\mathcal{E}(\tau)||_2} \tag{1}$$

where $\mathcal{E}(\cdot)$ is the transformation applied by the backbone and projection head and $\mathcal{P}(\cdot)$ is the transformation applied by the prediction head. In our case, each backbone consists of ResNet18, each projection head is a Multi-Layer Perceptron (MLP) with 3 layers of size $(512, 512, 128)$ and each prediction head is another MLP of size $(128, 64, 128)$. For further details, refer to [2].

Most methods contrastively train the visual and tactile domains [31, 33]. Differently, our approach jointly trains NOCS and tactile domains, embedding the NOCS map information within the learned feature space. This is crucial, as the NOCS map's color coding precisely encodes the 3D spatial position of the touch sample.

Thus, in the second step, given the latent space produced by our contrastive methods, we employ as a color predictor an MLP-based decoder to estimate the touch sample's position in space as an RGB triplet. The ground-truth RGB triplet corresponding to the touch location is obtained by averaging the NOCS map colors, which can be uniquely mapped to a 3D position in space. The color predictor MLP consists of three layers with size (128,64,3) and is trained using a standard L1 mean absolute error (MAE) loss.

At test time, given a query touch τ, the network predicts the RGB triplet that corresponds to the position of a Gaussian in the 3D space.

4. Experiments

Leveraging the global scene representation provided by NOCS to unify the visual and tactile domains, we conduct two lines of experiments: cross-modal touch generation for novel views, and 3D touch localization.

4.1. Implementation details

Gaussian splatting We train our GS using the standard SplatFacto-big model [27], with camera poses estimated via structure-from-motion. We use the images provided by TaRF [7], which amounts at roughly 1000 densely collected RGB images per scene on a NVIDIA RTX 4090 GPU.

Diffusion model. The diffusion model is optimized on each scene for 30 epochs by an Adam optimizer [18] with learning rate 10^{-6} and batch size 16 on a NVIDIA RTX 4090 GPU. At inference time, we perform 200 denoising steps with a 7.5 guidance scale. As common in cross-modal synthesis works [7, 23], we apply re-ranking of samples, by generating 16 images for each sample and picking the one identified by the contrastive model with the highest quality as our result.

Contrastive visual-NOCS-tactile model. The contrastive model is trained on samples from each scene, applying random augmentation [3] with a factor of 3 per sample.

Training is conducted for 100 epochs using an Stochastic Gradient Descent (SGD) optimizer with a learning rate of 10^{-2} and a batch size of 8. After training the contrastive model, its weights are frozen, and the MLP for color prediction is subsequently trained for 1000 epochs using an SGD optimizer with a learning rate of 10^{-3} and a batch size of 8.

4.2. Cross-modal touch generation

The goal is to evaluate the ability of our model to generate a coherent haptic map for a novel viewpoint in the scene. For the TaRF dataset, we follow the same train/test split as in [7], thus each sequentially collected split of 50 samples is divided between train, validation and test samples with an 8|1|1 ratio. We use the standard metrics for the evaluation: PSNR, Structure Similarity (SSIM) [14] and Frechet Inception Distance (FID) [31]. All the metrics are computed between the generated touch sample and the ground truth.

We compare our solution against 4 methods: VisGel [21], VisGel(L1) as in [7], base TaRF [7] and custom TaRF*. VisGel corresponds to the GAN-based model from [21] and VisGel(L1) is the same model trained with an L1 loss. TaRF is the base diffusion-based model from [7], while TaRF* is the same model trained on data for a single scene. We train TaRF* on a single scene at a time to make the setup comparable to ours. The numerical results are obtained by averaging the results across the 16 training scenes.

Quantitative results. The quantitative results are reported in Table 1.

Our approach outperforms all other methods across all evaluated metrics. It achieves superior results in pixel-wise metrics such as PSNR and SSIM, which can be highly influenced by the generating position, as well as in the more general FID score, a standard metric for cross-modal generation tasks to measure the distribution of the generated data. TaRF*, trained on individual scenes, outperforms the version trained on the entire dataset (TaRF), but nonetheless its reliance on depth maps as local descriptors leads to weaker performance compared to our method, which benefits from the scene-level NOCS maps for a more robust representation.

Ablation studies. The quantitative results are reported in Table 2. The full model corresponds to the experiments incorporating re-ranking along with RGB, depth, and NOCS map conditioning. Re-ranking provides a slight performance improvement, demonstrating the model's ability to generate high-quality samples consistently. Removing depth from the conditioning results in performance degradation, indicating that a local descriptor still contributes to overall estimation accuracy. Removing RGB information leads to a decline in PSNR, as expected, given the crucial relevance of visual information provided by the image. Interestingly, removing the NOCS map slightly improves PSNR, as the network faces an easier task, not having to interpret the 3D position in space. On the other hand, removing NOCS degrades FID performances, suggesting that such information contributes to better capturing the overall data distribution.

Model	Ref.	PNSR ↑	SSIM ↑	FID ↓
VisGel(L1)	[21]	24.34	0.82	97.05
VisGel	[21]	23.66	0.81	130.22
TaRF	[7]	22.84	0.72	28.97
TaRF*	[7]	23.88	0.76	15.20
SplatTouch		**30.19**	**0.84**	**10.06**

Table 1. Quantitative results on cross-modal touch generation for novel views. Our approach achieves a lower FID score by more effectively preserving the distribution of real tactile data, while still outperforming the baselines on low-level PSNR and SSIM metrics.

Model	PNSR ↑	SSIM ↑	FID ↓
Full	30.19	**0.84**	**10.06**
No re-rank	28.49	0.83	13.11
No Depth	29.21	0.83	11.75
No RGB	27.82	0.82	10.15
No NOCS	**30.29**	**0.84**	10.63

Table 2. Ablation studies for the touch generation task. We choose the model with the lowest FID as it is the one capable of better capturing the overall data distribution thanks to the NOCS.

Qualitative results. We report the qualitative results of our experiments in Fig. 5. Our model demonstrates enhanced generation capabilities compared to TaRF, producing outputs that maintain a closer structural resemblance to the original sample. TaRF instead either blurs or overemphasizes finer details (see Fig. 5 columns 1-3 and column 4, respectively). It is interesting to note that the performance of both methods are comparable when the touch occurs on a flat surface. However, our approach demonstrates significant improvements in the presence of edges. This advantage stems from the ability of NOCS to explicitly model the 3D space, enabling the diffusion model to leverage this structured 3D information through its 2D representation.

4.3. 3D touch localization

Given a query touch sample, the goal is to predict its 3D position within the scene. To achieve this, we predict the RGB triplet corresponding to a NOCS map. Thanks to our NOCS mapping, this RGB value directly translates to a 3D position in space, associated with a Gaussian. Our approach feeds the query sample to the contrastive learning encoder to obtain a latent representation that is fed to the MLP decoder

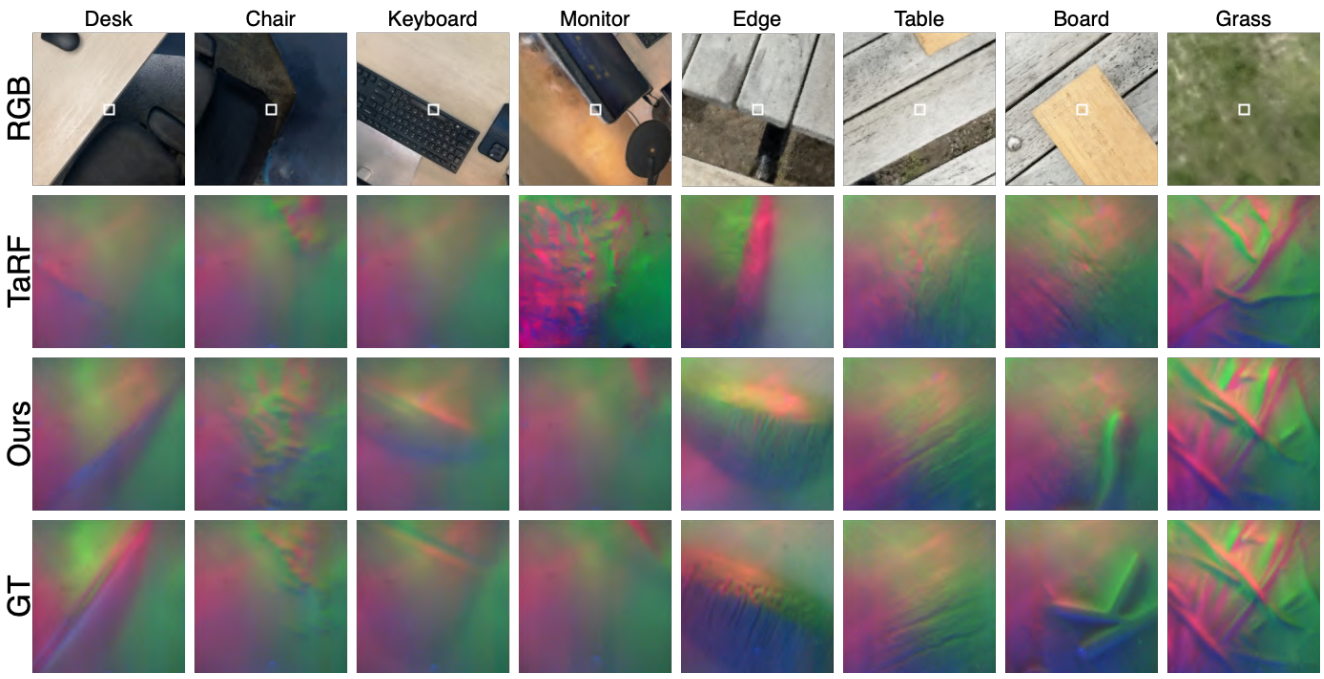

Figure 5. Qualitative results on the cross-modal touch generation task. Since the touched sample approximately corresponds to the center of the RGB image, we have highlighted the touched area with a square for better visualization. Our model demonstrates an improved ability to interpret the contextual information provided by the RGB images. For example, in the desk/keyboard sample, the surrounding context appears similar, leading TaRF to generate nearly identical touch samples. In contrast, our method effectively captures the subtle variations in texture and material, resulting in a more diverse and context-aware generation. A similar trend is observed in the edge/table/board images, where our method successfully captures part of the board's pattern. Notably, the board is a Braille surface used as a calibration object with fine details in the dataset, further demonstrating our model's improved ability to leverage fine-grained visual details to refine touch predictions.

that predicts the RGB value. We evaluate our approach against four baselines: (I) Random: we randomly select a Gaussian in the 3D space and take its position as the predicted one. Notably, this differs from selecting a completely random position in the scene, as Gaussians belong to touchable surfaces, (II) RGB+Touch: our MLP model trained on features extracted via contrastive learning between the visual and touch domains, (III) NOCS+Touch: our contrastive learning model trained only on real data, leveraging the NOCS-touch relationship, (IV) NOCS+Touch (Augmented): our contrastive method trained with additional synthetic samples generated using our generative approach. The augmentation consists of 25% additional samples per scene (e.g., for the office scene, which originally has 1,000 touch samples, we generate an extra 250). For evaluation, we measure the Euclidean distance between the predicted touch position and the ground truth.

Quantitative results. The quantitative results are reported in Table 3. (I) Random achieves the worst performance, as randomly selecting a touch position is ineffective. However, since the average scene size is approximately 2×2 meters and Gaussians are relatively dense, the error does not escalate as dramatically as it would in larger scenes. (II) RGB+Touch underperforms compared to the other methods, as RGB images alone do not provide precise 3D positional information like NOCS maps do. (III) NOCS+Touch improves upon (II), benefiting from the structured 3D representation offered by NOCS. (IV) NOCS+Touch (Augmented) surpasses (III), confirming that our generative model effectively produces meaningful synthetic samples, which further refine the 3D localization task.

	Model	Training data	Distance (cm)
I	Random	-	56.47
II	RGB+Touch	Real	22.44
III	NOCS+Touch	Real	13.02
IV	**NOCS+Touch**	**Real+Aug**	**11.65**

Table 3. Quantitative results for the 3D localization of a touch sample task.

Qualitative results. We show the qualitative results in Fig. 6. Our method effectively retrieves the correct sample position. Even when it fails, it still lands on surfaces of sim-

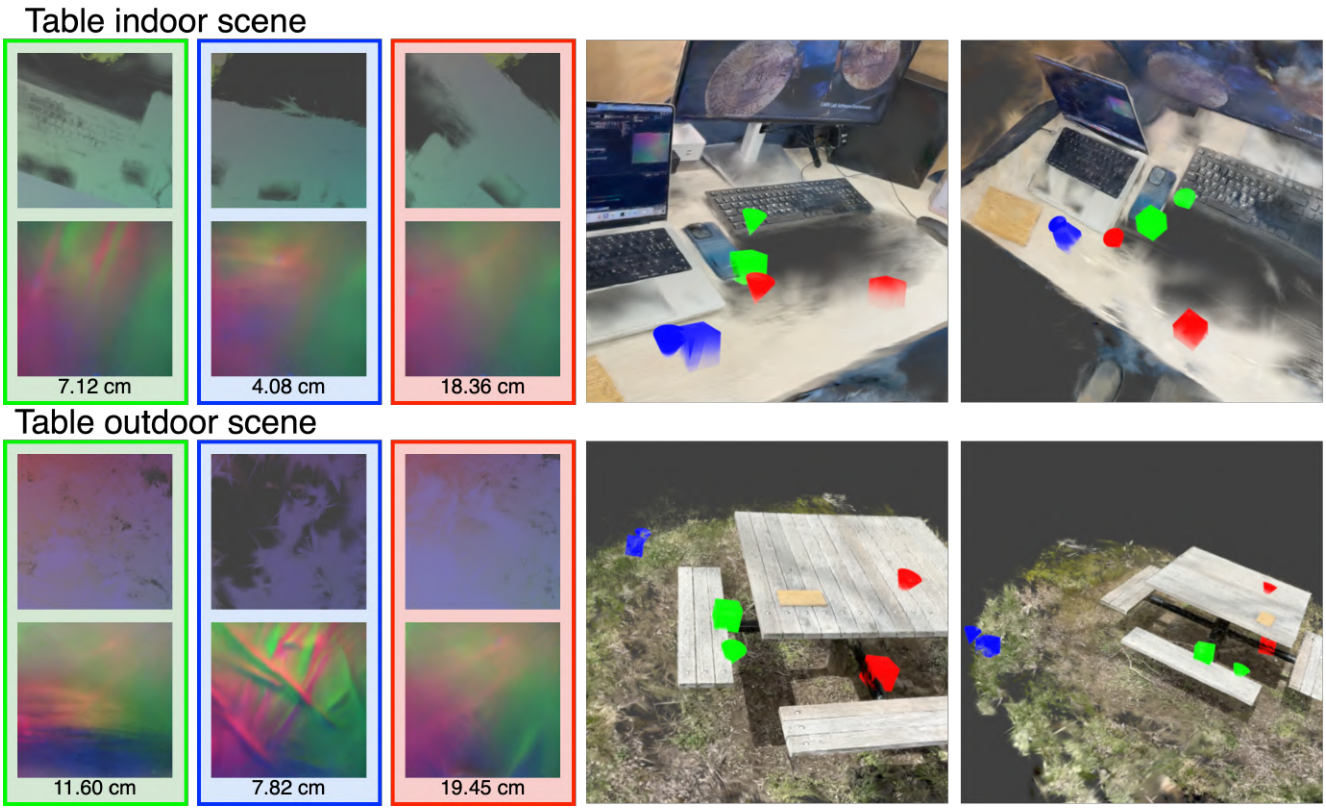

Figure 6. Qualitative results on 3D touch estimation task. On the left, the query samples, the relative NOCS and estimated errors. On the right, their actual location in the scene as seen from different viewpoints. The colored cones Δ represent the ground truth positions and the cubes $\square$ the predicted ones. Our framework consistently estimates the 3D position of query touch samples. The blue and green samples in both scenes are predicted close to their real positions. However, the red sample highlights two areas for potential improvement, primarily due to the noisiness of the Gaussian reconstruction. In the first scene, the position is predicted on the same material surface but at a different location. This is likely caused by Gaussians at the center of the desk being reconstructed further away in the scene and subsequently removed during the noise filtering step—an issue stemming from the challenge of reconstructing featureless surfaces. In the second case, the Gaussian is projected far from the surface but is not filtered out in the noise removal step, resulting in the sample being predicted further away than expected. Addressing these challenges could further refine the accuracy of 3D touch localization.

ilar material, indicating that the extracted features successfully capture both local sample relationships and broader material properties. In the context of VR/AR/XR, our approach can be utilized to generate tactile signals for every part of a touched scene, representing a step toward fully touchable virtual environments. While this work presents a complete processing pipeline for integrating touch and vision, a significant research gap remains in translating this gathered information into usable haptic feedback for users, which we aim to explore in future work.

5. Conclusions

We have presented a novel pipeline that integrates the touch and visual domains through an explicit 3D representation. We have demonstrated how this representation facilitates tasks such as cross-domain tactile generation and 3D touch localization. Our work takes a significant step towards creating scenes that can not only be *seen* but also *touched*, as every point in the 3D scene can be mapped to a haptic map. In future work, we aim to explore whether touch samples can aid the RGB domain in creating more refined 3D representations, such as GS and NeRF, at a micro level.

Limitations. Misalignment arising from multiple sources, such as camera calibration, GS and NeRF, noise removal and the data collection protocol, may affect the results. This issue can be addressed by designing sensors that can jointly capture visual and tactile data in a sequential manner, and would greatly be of help especially in the 3D localization task.

Potential negative impact. The training data currently available, is collected in an urban environment with a relatively low diversity of materials; thus it may not fully represent a wide range of scenarios, potentially introducing bias into the model.

Funding

We acknowledge the support of the MUR PNRR project iNEST- Interconnected Nord-Est Innovation Ecosystem (ECS00000043) funded by the European Union under NextGenerationEU. This work was also supported by the European Union - Next Generation EU under the Italian National Recovery and Resilience Plan (NRRP), Mission 4, Component 2, Investment 1.3, CUP C69J24000180004, partnership on "Telecommunications of the Future" (PE00000001 - program "RESTART").

References

[1] Sepehr Alizadehsalehi, Ahmad Hadavi, and Joseph Chuenhuei Huang. From bim to extended reality in aec industry. *Automation in construction*, 116:103254, 2020. 1

[2] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15750–15758, 2021. 3, 5

[3] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 702–703, 2020. 5

[4] Shaowei Cui, Rui Wang, Jingyi Hu, Chaofan Zhang, Lipeng Chen, and Shuo Wang. Self-supervised contact geometry learning by gelstereo visuotactile sensing. *IEEE Transactions on Instrumentation and Measurement*, 71:1–9, 2021. 2

[5] Mucong Ding, Kezhi Kong, Jingling Li, Chen Zhu, John Dickerson, Furong Huang, and Tom Goldstein. Vq-gnn: A universal framework to scale up graph neural networks using vector quantization. *Advances in Neural Information Processing Systems*, 34:6733–6746, 2021. 4

[6] Won Kyung Do and Monroe Kennedy. Densetact: Optical tactile sensor for dense shape reconstruction. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 6188–6194. IEEE, 2022. 2

[7] Yiming Dou, Fengyu Yang, Yi Liu, Antonio Loquercio, and Andrew Owens. Tactile-augmented radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26529–26539, 2024. 2, 3, 4, 5, 6

[8] Yipai Du, Guanlan Zhang, and Michael Yu Wang. 3d contact point cloud reconstruction from vision-based tactile flow. *IEEE Robotics and Automation Letters*, 7(4):12177–12184, 2022. 2

[9] Yu Fang, Xuehe Zhang, Wenqiang Xu, Gangfeng Liu, and Jie Zhao. Bidirectional visual-tactile cross-modal generation using latent feature space flow model. *Neural Networks*, 172: 106088, 2024. 2

[10] Ruohan Gao, Yen-Yu Chang, Shivani Mall, Li Fei-Fei, and Jiajun Wu. Objectfolder: A dataset of objects with implicit visual, auditory, and tactile representations. *arXiv preprint arXiv:2109.07991*, 2021. 2

[11] Ruohan Gao, Yiming Dou, Hao Li, Tanmay Agarwal, Jeannette Bohg, Yunzhu Li, Li Fei-Fei, and Jiajun Wu. The objectfolder benchmark: Multisensory learning with neural and real objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17276–17286, 2023. 2, 3, 5

[12] Ruihan Gao, Kangle Deng, Gengshan Yang, Wenzhen Yuan, and Jun-Yan Zhu. Tactile dreamfusion: Exploiting tactile sensing for 3d generation. *Advances in Neural Information Processing Systems*, 37:29839–29863, 2025. 3

[13] Negin Heravi, Wenzhen Yuan, Allison M Okamura, and Jeannette Bohg. Learning an action-conditional model for haptic texture generation. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11088–11095. IEEE, 2020. 2

[14] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 6

[15] Takuya Ikeda, Sergey Zakharov, Tianyi Ko, Muhammad Zubair Irshad, Robert Lee, Katherine Liu, Rares Ambrus, and Koichi Nishiwaki. Diffusionnocs: Managing symmetry and uncertainty in sim2real multi-modal category-level pose estimation. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7406–7413. IEEE, 2024. 2

[16] Zhanat Kappassov, Juan-Antonio Corrales, and Véronique Perdereau. Tactile sensing in dexterous robot hands — review. *Robotics and Autonomous Systems*, 74:195–220, 2015. 2

[17] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. 3

[18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[19] Mike Lambeta, Po-Wei Chou, Stephen Tian, Brian Yang, Benjamin Maloon, Victoria Rose Most, Dave Stroud, Raymond Santos, Ahmad Byagowi, Gregg Kammerer, et al. Digit: A novel design for a low-cost compact high-resolution tactile sensor with application to in-hand manipulation. *IEEE Robotics and Automation Letters*, 5(3):3838–3845, 2020. 3

[20] Jet-Tsyn Lee, Danushka Bollegala, and Shan Luo. "touching to see" and "seeing to feel": Robotic cross-modal sensory data generation for visual-tactile perception. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 4276–4282. IEEE, 2019. 2

[21] Yunzhu Li, Jun-Yan Zhu, Russ Tedrake, and Antonio Torralba. Connecting touch and vision via cross-modal prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10609–10618, 2019. 2, 6

[22] Zhixiang Min, Bingbing Zhuang, Samuel Schulter, Buyu Liu, Enrique Dunn, and Manmohan Chandraker. Neurocs: Neural nocs supervision for monocular 3d object localiza-

tion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21404–21414, 2023. 3

[23] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International conference on machine learning*, pages 8821–8831. Pmlr, 2021. 5

[24] Samanta Rodriguez, Yiming Dou, William van den Bogert, Miquel Oller, Kevin So, Andrew Owens, and Nima Fazeli. Contrastive touch-to-touch pretraining. *arXiv preprint arXiv:2410.11834*, 2024. 2

[25] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 3, 4

[26] Antonio Luigi Stefani, Niccolò Bisagno, Andrea Rosani, Nicola Conci, and Francesco De Natale. Signal processing for haptic surface modeling: a review. *arXiv preprint arXiv:2409.20142*, 2024. 2

[27] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, et al. Nerfstudio: A modular framework for neural radiance field development. In *ACM SIGGRAPH 2023 conference proceedings*, pages 1–12, 2023. 5

[28] He Wang, Srinath Sridhar, Jingwei Huang, Julien Valentin, Shuran Song, and Leonidas J Guibas. Normalized object coordinate space for category-level 6d object pose and size estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2642–2651, 2019. 2, 3, 4

[29] Nannan Xi, Juan Chen, Filipe Gama, Marc Riar, and Juho Hamari. The challenges of entering the metaverse: An experiment on the effect of extended reality on workload. *Information Systems Frontiers*, 25(2):659–680, 2023. 1

[30] Chao Xu, Ang Li, Linghao Chen, Yulin Liu, Ruoxi Shi, Hao Su, and Minghua Liu. Sparp: Fast 3d object reconstruction and pose estimation from sparse views. In *European Conference on Computer Vision*, pages 143–163. Springer, 2024. 2

[31] Fengyu Yang, Chenyang Ma, Jiacheng Zhang, Jing Zhu, Wenzhen Yuan, and Andrew Owens. Touch and go: Learning from human-collected vision and touch. *arXiv preprint arXiv:2211.12498*, 2022. 2, 5, 6

[32] Fengyu Yang, Jiacheng Zhang, and Andrew Owens. Generating visual scenes from touch. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22070–22080, 2023. 2

[33] Fengyu Yang, Chao Feng, Ziyang Chen, Hyoungseob Park, Daniel Wang, Yiming Dou, Ziyao Zeng, Xien Chen, Rit Gangopadhyay, Andrew Owens, et al. Binding touch to everything: Learning unified multimodal tactile representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26340–26353, 2024. 2, 3, 5

[34] Wenzhen Yuan, Chenzhuo Zhu, Andrew Owens, Mandayam A Srinivasan, and Edward H Adelson. Shape-independent hardness estimation using deep learning and a gelsight tactile sensor. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 951–958. IEEE, 2017. 2

[35] Shaohong Zhong, Alessandro Albini, Oiwi Parker Jones, Perla Maiolino, and Ingmar Posner. Touching a nerf: Leveraging neural radiance fields for tactile sensory data generation. In *Conference on Robot Learning*, pages 1618–1628. PMLR, 2023. 2, 3

[36] Yiming Zhou, Zixuan Zeng, Andi Chen, Xiaofan Zhou, Haowei Ni, Shiyao Zhang, Panfeng Li, Liangxi Liu, Mengyao Zheng, and Xupeng Chen. Evaluating modern approaches in 3d scene reconstruction: Nerf vs gaussian-based methods. In *2024 6th International Conference on Data-driven Optimization of Complex Systems (DOCS)*, pages 926–931. IEEE, 2024. 3