Grasp in Gaussians: Fast Monocular Reconstruction of Dynamic Hand–Object Interactions

AYCE IDIL AYTEKIN, MPI for Informatics, SIC & VIA Research Center, Germany
XU CHEN, Google, Switzerland
ZHENGYANG SHEN, Google, USA
THABO BEELER, Google, Switzerland
HELGE RHODIN, MPI for Informatics, Germany and Bielefeld University, Germany
RISHABH DABRAL, MPI for Informatics, SIC & VIA Research Center, Germany
CHRISTIAN THEOBALT, MPI for Informatics, SIC & VIA Research Center, Germany

arXiv:2604.12929v2 [cs.CV] 27 Aug 2026

Fig. 1. **Grasp in Gaussians (GraG):** Given a single monocular video of a hand interacting with an object, *GraG* reconstructs 3D geometry and pose of the hand and the object. Our method is designed to be efficient for long sequences, and can reconstruct in-the-wild captured examples.

We present **Grasp in Gaussians (GraG)**, a fast and robust method for reconstructing dynamic 3D hand–object interactions from a single monocular video. Unlike recent approaches that optimize heavy neural representations, our method focuses on tracking the hand and the object efficiently, once initialized from pretrained large models. Our key insight is that, given strong pretrained object and hand priors, accurate and temporally stable hand–object motion can be recovered using a compact Sum-of-Gaussians (SoG) representation, revived from classical tracking literature and integrated with generative Gaussian-based initializations. We initialize object pose and geometry using a video-adapted SAM3D pipeline, then convert the resulting dense Gaussian representation into a lightweight SoG via subsampling. This compact representation enables efficient and fast tracking while preserving geometric fidelity, with the pretrained priors and simple geometric/contact losses providing accurate hand-object placement. For the hand, we adopt a complementary strategy: starting from off-the-shelf monocular hand pose initialization, we refine hand motion using simple yet effective 2D joint, silhouette, depth, and contact losses, avoiding per-frame refinement of a detailed 3D hand appearance model while maintaining stable articulation. Extensive experiments on public benchmarks demonstrate that GraG reconstructs temporally coherent hand-object interactions on long sequences 4.4×–38.9× faster than prior work while preserving temporally coherent hand-object motion. Project page: GraG-page.

CCS Concepts: • **Computing methodologies → Reconstruction**.

Additional Key Words and Phrases: 3D reconstruction, hand-object interaction

1 Introduction

Our hands enable us to perform countless interactions with the physical world, from grasping everyday objects to manipulating tools with precision. Reconstructing such dynamic 3D interactions is a fundamental modeling task with crucial applications in robotics, where learning from human demonstration is key, and in augmented reality, for seamless virtual-physical interaction. Our goal is to reconstruct hand-object interaction (HOI) from a monocular video, which involves estimating the 3D geometry and pose of the hand and the object. In this work, we focus on the foundational setting of manipulating a single object with the dominant hand, which covers

Authors' Contact Information: Ayce Idil Aytekin, MPI for Informatics, SIC & VIA Research Center, Germany, aaytekin@mpi-inf.mpg.de; Xu Chen, Google, Switzerland, xche@google.com; Zhengyang Shen, Google, USA, zyshen@google.com; Thabo Beeler, Google, Switzerland, tbeeler@google.com; Helge Rhodin, MPI for Informatics, Germany and Bielefeld University, Germany, helge.rhodin@uni-bielefeld.de; Rishabh Dabral, MPI for Informatics, SIC & VIA Research Center, Germany, rdabral@mpi-inf.mpg.de; Christian Theobalt, MPI for Informatics, SIC & VIA Research Center, Germany, theobalt@mpi-inf.mpg.de.

a vast majority of functional daily tasks. Yet, performing such reconstruction is challenging, as heavy occlusions and depth ambiguity make it a fundamentally ill-posed problem. The challenge is further exacerbated when runtime considerations are introduced.

In the absence of clear, unoccluded visual cues, it becomes important to introduce appropriate priors to curb the ill-posedness of the task. However, past methods have done so with a variety of strong assumptions that often limit their practical applicability to open-world and unconstrained scenarios with complex backgrounds. For instance, many methods require a pre-scanned 3D object template [Fan et al. 2023; Hasson et al. 2020, 2021; Yang et al. 2021], making them unable to reconstruct novel unseen objects. While some template-free methods exist, they are often trained on datasets with a limited number of object instances [Hasson et al. 2019; Karunratanakul et al. 2020; Ye et al. 2022] or are restricted to specific, predefined object categories [Ye et al. 2023], leading to poor generalization and inaccurate geometry for out-of-distribution objects. Another line of work, in-hand object scanning [Hampali et al. 2023; Huang et al. 2022; Zhong et al. 2024], can reconstruct novel objects but typically assumes a rigid hand pose, failing to capture dynamic, dexterous articulations. Recent video-based approaches such as HOLD [Fan et al. 2024] and BIGS [On et al. 2025] enable category generalization by optimizing implicit surfaces or Gaussian representations. However, optimizing such volumetric representation across several frames is painstakingly slow (∼10 hours for HOLD, ∼4 hours for BIGS for a video with 100 frames), thereby limiting their use on long, in-the-wild video reconstruction.

To overcome these restrictions, we examine what modern priors can offer and what fast tracking requires. Recent image-to-3D models [Lai et al. 2025; Xiang et al. 2025] can produce plausible object geometry from a single view, but naïvely re-running such models frame-by-frame yields jittery poses and inaccurate shape under occlusion. In comparison, classical model-based human/object tracking formulations are fast and stable, but have historically required carefully engineered templates or multi-view capture [Sridhar et al. 2014; Zhong et al. 2000], which cannot directly generalize to hand-object reconstruction. In this paper, we show that we can combine the strengths of both worlds by *using a generative 3D prior to obtain a canonical object, and then performing lightweight tracking in a compact representation.*

Our method, dubbed *Grasp in Gaussians (GraG)*, operates in three stages. First, we select a small set of informative and diverse keyframes from the input video and reconstruct a canonical object using MV-SAM3D [Li et al. 2026], a multi-view extension of SAM3D [Chen et al. 2025]. MV-SAM3D represents the object as a set of canonical shape tokens (shared across views), which we decode once to obtain a dense 3D Gaussian representation of the object. Second, we extend SAM3D to videos by freezing the canonical shape and estimating per-frame object pose (rotation, translation, and scale) with temporal guidance in its latent space, which suppresses jitter and prevents occlusion-induced drift. While dense 3D Gaussians are expressive, directly optimizing a high-resolution set of Gaussians across all frames is unnecessary for tracking and becomes computationally expensive. At the same time, the appearance generated by image-to-3D models like SAM3D is often too inaccurate to be used for photogrammetric tracking objectives. Therefore, we sparsify the decoded dense Gaussians into a compact Sum-of-Gaussians (SoG) model [Stoll et al. 2011] and optimize a differentiable alignment objective that matches its 2D projection to an image SoG constructed from each frame via quad-tree color clustering. For the hand, we refine a Dyn-HaMR [Yu et al. 2025] initialization using 2D joint, depth, silhouette, and contact priors, achieving stable articulation without heavy optimization.

GraG adopts the reconstruction paradigm of having a strong initial prior followed by efficient tracking without additional training. Such a design makes our method particularly well-suited for long, unconstrained videos. Experimental results confirm that GraG achieves state-of-the-art performance on standard benchmarks and generalizes robustly to in-the-wild videos with unseen objects and diverse scenes (*c.f.* Fig. 1). We further demonstrate the practical applicability of GraG in robotic settings. Specifically, we use GraG to reconstruct hand–object interactions from real human demonstrations and use them as goal poses within a tracking-based control policy. Our method is highly efficient as we reduce the runtime of prior works from ∼ 3-10 hours to ∼16 minutes on long sequences. Our contributions are threefold:

- We propose a fast monocular HOI reconstruction system that leverages pretrained object and hand priors to scale to long, unconstrained videos while drastically reducing runtime.
- We extend SAM3D to videos by freezing the canonical shape and tracking only per-frame pose with temporal guidance, improving stability under occlusion.
- We introduce compact occlusion-aware SoG tracking for Gaussian-based object assets by sparsifying dense Gaussians and aligning projected object SoG to an image SoG built via quad-tree clustering.

2 Related Work

3D asset generation. Reconstructing 3D objects from images is a core problem in computer vision. While traditional multi-view methods like Structure from Motion (SfM) [Hartley and Zisserman 2011; Scharstein et al. 2002; Szeliski 2010; Tomasi and Kanade 1992] are effective, their performance degrades significantly in videos with heavy occlusion and limited viewpoints. To address these limitations, recent work has shifted towards learning-based, single-image 3D generation, trained on large-scale datasets like Objaverse [Deitke et al. 2023]. Diffusion models, in particular, have emerged as the state-of-the-art for generating high-fidelity 3D assets from a single 2D image [Chen et al. 2025; Xiang et al. 2025]. Recently, SAM3D [Chen et al. 2025] proposes pose estimation together with 3D asset generation by separating pose and shape space of the object in the image, resulting in high-quality posed 3D objects. To obtain a strong geometric prior for the object, we employ the multi-view extension of SAM3D.

Gaussian tracking. Gaussian primitives were used for human [Elhayek et al. 2015; Moeslund et al. 2006; Plankers and Fua 2003; Rhodin et al. 2016a; Stoll et al. 2011], hand [Bretzner et al. 2002; Sridhar et al. 2015], and object tracking [Ren et al. 2014] well before modern Gaussian Splatting (GS) [Kerbl et al. 2023]. SoG [Stoll et al. 2011] is especially useful because representing both the image and

3D template with Gaussians enables efficient matching via closed-form Gaussian overlap. Later tracking methods improved occlusion handling with ray tracing [Rhodin et al. 2016a, 2015], contour tracking [Rhodin et al. 2016b], alpha blending [Kerbl et al. 2023], point visibility [Rajič et al. 2025], or dynamic surface tracking [Zheng et al. 2025]. However, these methods often rely on multi-view or depth input, or optimize dense photorealistic Gaussian renderings [Cong et al. 2025; Liu et al. 2025b]. Opposed to increasing photorealism with GS, we revisit the SoG representation, as it alleviates sorting Gaussians back-to-front (for alpha blending) and reduces the computation cost with the sparse Gaussian image representation.

Monocular hand tracking under object occlusion. Hand-object occlusion is also a central challenge in monocular hand mesh recovery. HandOccNet [Park et al. 2022] injects hand information into occluded image regions with transformer modules, H2ONet [Xu et al. 2023] fuses non-occluded finger and orientation cues across nearby frames, and HaMeR [Pavlakos et al. 2024] uses a large transformer-based model for robust monocular hand reconstruction. Building on HaMeR, Dyn-HaMR [Yu et al. 2025] recovers temporally coherent hand motion and can use external camera intrinsics and extrinsics, making it a suitable initialization for our camera-registered hand trajectory. These works motivate using a strong hand prior under occlusion, but they recover only the hand and do not estimate the interacting object's geometry, scale, or trajectory. GraG uses monocular hand estimates as geometric anchors while reconstructing the object and its hand-relative motion over time.

Hand-object reconstruction. Dynamic HOI reconstruction is difficult due to severe mutual occlusion, fast object motion, and the absence of object templates in the wild. While datasets such as DexYCB [Chao et al. 2021] provide synchronized multi-view RGB-D captures for controlled evaluation, monocular video reconstruction requires reasoning under depth ambiguity. Early template-based or depth-dependent methods [Cao et al. 2021; Corona et al. 2020; Liu et al. 2021; Tekin et al. 2019; Yang et al. 2021] therefore have limited applicability outside controlled settings. Recent single-image methods such as EasyHOI [Liu et al. 2025a] and FollowMyHold [Aytekin et al. 2026] improve generalization by composing foundation models, but do not exploit temporal cues from videos. Video-based methods such as HOLD [Fan et al. 2024] and BIGS [On et al. 2025] optimize implicit or Gaussian scene representations, while MagicHOI [Wang et al. 2025] adds generative priors to complete occluded regions. However, these approaches remain computationally expensive and sensitive to initialization (commonly SfM-style bootstrapping [Sarlin et al. 2019, 2020]) under occlusion, textureless objects, and long sequences. Concurrent work, AGILE [Shi et al. 2026], addresses these issues by using agentic generation to obtain complete, high-texture, simulation-ready objects and then tracking them without SfM, while ForeHOI [Chen et al. 2026] reconstructs the object geometry from interaction videos in a feed-forward manner. In contrast, GraG focuses on efficient long-video HOI reconstruction: we reconstruct a canonical object from selected keyframes and then recover temporally stable hand-object motion with compact occlusion-aware SoG tracking, emphasizing fast hand-relative object pose recovery rather than photorealistic texture completion.

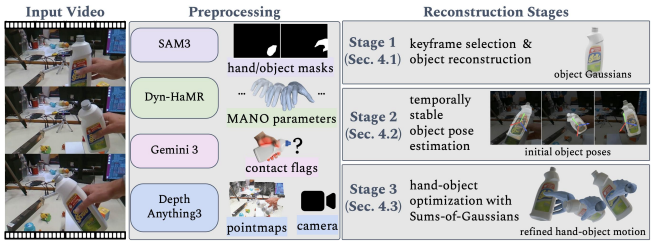

Fig. 2. **Method Overview.** Given a monocular video, we recover per-frame hand-object poses and geometry. We first preprocess the video to obtain masks, initial hand poses, per-frame binary contact information, pointmaps and camera intrinsics/extrinsics. Preprocessing is followed by three stages: Stage 1 (Section 4.1) for keyframe selection and object reconstruction, Stage 2 (Section 4.2) for estimating a temporally stable initial object trajectory (and scale) by adapting SAM3D to videos, and Stage 3 (Section 4.3) for hand and object pose and scale refinement via Sums-of-Gaussians tracking.

Another line of work adds physics to the reconstruction. Hu et al. [2022] recover the interaction and its contact forces with an explicit physics solver from a single RGB-D sensor, and their follow-up, HOIC [Hu et al. 2024] uses deep reinforcement learning in a physics simulator with object compensation control and a surface-contact model to further improve physical correctness. These methods highlight the benefit of explicit physical reasoning, but address RGB-D reconstruction with physical motion/contact modeling. We instead keep tracking cheap with lightweight contact and penetration priors for efficient monocular tracking and leave stronger physical constraints to future work.

3 Background: SAM3D

Our approach to initialize object poses is built upon SAM3D [Chen et al. 2025],which is a generative model that reconstructs a 3D object from a single RGB image I, object mask M, and an optional scene pointmap P. Unlike previous methods that predict geometry in an implicit orientation [Lai et al. 2025; Xiang et al. 2025], SAM3D explicitly decouples the canonical *shape* S and texture F from the camera-space *layout* (R, τ, s), which represents the object's orientation, translation, and scale.

Concretely, the model approximates the conditional joint distribution $p(S, F, R, \tau, s \mid I, M, P)$ using a structured multi-modal transformer. Within this architecture, the object is represented by separate canonical shape tokens Z^S and low-dimensional layout tokens Z^P. Modality-specific projection layers map these tokens into a high-dimensional feature space, allowing cross-modal attention mechanisms to resolve the final geometry and pose.

4 Method

Given a monocular RGB video $\mathbf{I} = \{\mathbf{I}_t\}_{t=1}^{N}$ of a hand interacting with a rigid object, our goal is to recover temporally coherent 3D geometry and motion of the hand and the object. This task is highly underconstrained due to depth ambiguity and hand-object occlusions. To tackle it, we leverage the strong data-driven priors provided by generative 3D models like SAM3D and integrate them into the highly efficient classical tracking approach based on Sum-of-Gaussians.

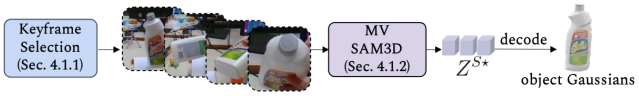

Fig. 3. **Stage 1** reconstructs a canonical object with MV-SAM3D by selecting keyframes and decoding shape tokens into dense Gaussians (Section 4.1).

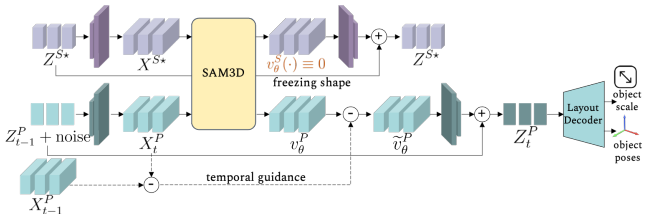

Fig. 4. **Stage 2: Temporally stable SAM3D pose estimation.** Data flow for per-frame pose estimation. **Top (Shape):** Canonical shape tokens $Z^{S\star}$ are projected to latents $X^{S\star}$ and processed by the transformer. The predicted shape velocity is zeroed to keep the geometry unchanged. **Bottom (Pose):** Previous frame's layout tokens Z_{t-1}^P are projected to latents X_t^P. The predicted velocity v_θ^P is modified by a temporal guidance term (Eq. 1) derived from the difference between current and previous frame latents (X_{t-1}^P). The guided velocity $\tilde{v}_\theta^P$ is mapped back to token space and added to the previous frame's tokens to produce Z_t^P for decoding.

Fig. 2 illustrates the overall schema of our method. Briefly, we first obtain a reliable canonical object representation from the video (Section 4.1), then estimate per-frame object pose (Section 4.2), and finally refine both hand and object trajectories with an explicit, occlusion-aware tracking objective (Section 4.3). This design is motivated by the observation that heavy per-frame 3D reconstruction is unnecessary once a strong canonical prior is available, whereas lightweight tracking can robustly propagate motion through ambiguous and occluded frames.

As preprocessing, we extract hand and object masks M_t^h and M_t^o using SAM3 [Carion et al. 2025], where object masks are prompted with labels from a large VLM [Team et al. 2025]. We estimate camera intrinsics/extrinsics (K_t, T_t) and pointmaps P_t with DepthAnything3 [Lin et al. 2025]. These camera parameters are then provided as external inputs to Dyn-HaMR [Yu et al. 2025] to initialize the hand trajectory. Because Dyn-HaMR regresses the MANO-based hand pose directly within the coordinate frame defined by (K_t, T_t), the hand and object geometries are natively expressed in the same coordinate system, eliminating the need for post-hoc spatial alignment. Finally, we obtain per-frame contact flags $c_t^{ho} \in \{0, 1\}$ using Gemini 3 [Google 2025]. Throughout our discussion, we use superscripts o and h to denote object and hand related variables, respectively.

4.1 Stage 1: Keyframe Selection and Object Reconstruction

Since our system's input is a video, multiple views of the same object are generally available over time, but occluded by the hand. Compared to single-frame reconstruction, these views provide complementary visual cues that can significantly improve object shape estimation. To use this information, and be efficient, we select a compact set of K keyframes $\mathcal{V} = \{\mathcal{V}_k\}_{k=1}^K \subset \mathbf{I}$ that are informative and diverse. These video keyframes are treated as multi-view inputs and fed to MV-SAM3D [Li et al. 2026], a multi-view extension of the state-of-the-art image-to-3D model SAM3D [Chen et al. 2025], yielding canonical object shape tokens and dense 3D Gaussians.

4.1.1 Keyframe selection. To select informative and diverse views, we adopt the saddle-balanced selection strategy from Depth Anything 3 (DA3) [Lin et al. 2025] using DINOv2 `[CLS]` features [Oquab et al. 2023]. We compute a balance cost based on normalized feature statistics (mean magnitude, variance, and inter-frame similarity) to filter out unstable or weak activations [Darcet et al. 2024]. Finally, a subset of K keyframes is selected greedily by trading off this stability cost against a diversity penalty to eliminate redundancy (Fig. 3). We provide more details in Section 10.4 of the Supplementary Material.

4.1.2 Object reconstruction. The K keyframes selected above are then used as input to MV-SAM3D [Li et al. 2026]. The output shape tokens, $Z^{S\star}$, from MV-SAM3D act as a canonical object representation, which we later decode into Gaussians and sparsify for fast tracking (Section 4.3.2).

4.2 Stage 2: Temporally Stable Object Pose Estimation

In Stage 2 (Fig. 4), we estimate the per-frame object layout (R_t^o, τ_t^o, s^o) while forcing the shape to remain constant over time. Following SAM3D's conditional flow matching [Chen et al. 2025], we initialize each frame with shared shape tokens $Z_t^S \leftarrow Z^{S\star}$ and zero-out the shape velocity field ($v_\theta^S(\cdot) \equiv 0$). This freezes the canonical shape while allowing the layout to evolve via cross-modal attention. This constraint is crucial for hand-object interactions, preventing severe occlusions and rapid appearance changes from corrupting the reconstruction. We illustrate this stage in Fig. 4.

4.2.1 Temporal guidance on pose latents. Even with a frozen shape, per-frame layout estimation can jitter or even flip orientation under weak visual cues such as symmetric objects and heavy occlusion. We therefore bias the per-frame layout velocity toward temporal consistency using a temporal guidance term. Let X_t^P be the current pose/layout latents projected using an MLP from Z_t^P and X_{t-1}^P the previous frame's latents. We modify the predicted velocity by

$$\tilde{v}_\theta^P = v_\theta^P - \lambda_{\text{temp}} (X_t^P - X_{t-1}^P), \tag{1}$$

applied to translation and rotation latents. Intuitively, Equation (1) discourages abrupt latent changes unless strongly supported by the current observation. At the end of Stage 2, we obtain an initial object trajectory $\{R_t^o, t_t^o, s^o\}_{t=1}^T$.

4.3 Stage 3: Hand–Object Tracking with SoG

So far, we have reconstructed the object shape with MV-SAM3D and arrived at an initial estimate of the object trajectory using temporal guidance on pose latents. Our formulations have been agnostic to the presence of the hand mesh. Additionally, the estimated object geometry is not guaranteed to be positioned appropriately in the scene. Therefore, we refine the object trajectory in short windows $\mathcal{W}$ by solving a compact tracking objective. At a high level, we maximize an image alignment energy E between a set of object

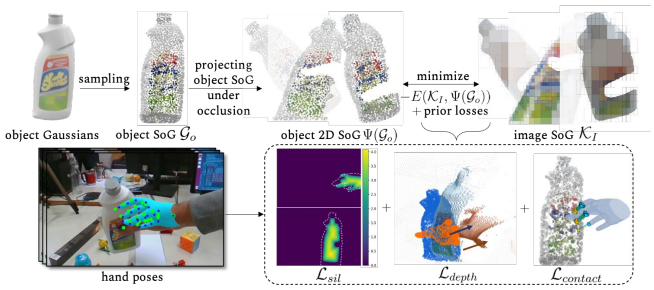

Fig. 5. **Stage 3** refines hand and object motion using an explicit tracking objective: we sparsify the dense Gaussians into a compact SoG and perform hand-aware SoG alignment with lightweight priors including silhouette loss $\mathcal{L}_{sil}$ to represent the distance to the silhouette edges, depth loss $\mathcal{L}_{depth}$ between estimated pointmap medians and hand-object medians, and contact loss $\mathcal{L}_{contact}$ between hand fingertips and object surface (Section 4.3). For the silhouette loss, we visualize the $log(DT_{hand}+1)$ and $log(DT_{object}+1)$.

Gaussians and the current frame approximated by Gaussians, while regularizing with lightweight geometric and interaction priors:

$$\min_{\mathcal{X}} \sum_{t\in\mathcal{W}} -E_t + \lambda_{\text{sil}}\mathcal{L}_{\text{sil}} + \lambda_{\text{depth}}\mathcal{L}_{\text{depth}} + \lambda_{\text{j2d}}\mathcal{L}_{\text{j2d}} + \lambda_{\text{contact}}\mathcal{L}_{\text{contact}} + \lambda_{\text{smooth}}\mathcal{L}_{\text{smooth}}. \tag{2}$$

where $\mathcal{L}_{\text{sil}}$ aligns the projected object and hand silhouettes with their image masks, $\mathcal{L}_{\text{depth}}$ anchors metric depth and couples object depth to hand depth in contact frames, $\mathcal{L}_{\text{j2d}}$ is the hand-joint reprojection loss, $\mathcal{L}_{\text{contact}}$ encourages physically plausible hand-object contact, and $\mathcal{L}_{\text{smooth}}$ suppresses temporal jitter. $\mathcal{X} = \{R_t^o, \tau_t^o, s^o, \Theta_t, R_t^h, \tau_t^h, s^h\}_{t\in\mathcal{W}}$ denotes the variables optimized in window $\mathcal{W}$. We illustrate our optimization approach in Fig. 5. It is worth noting that Stage 3 refines only the poses, not geometry; the canonical object shape from Stage 1 and the hand shape from Dyn-HaMR stay fixed. We optimize only the per-frame poses and translations, together with one global object scale and one global hand scale per sequence.

Critical to our approach is the image-object alignment term E_t, which is computed using Sum-of-Gaussians (SoG) tracking. By representing the object with a compact set of 3D Gaussians and encoding the masked object image with a compact set of 2D Gaussians, SoG tracking seeks to maximize their continuous overlap similarity. The SoG energy provides the main visible-object area matching signal; the silhouette term is only a guard against degenerate support, not a separate replacement for SoG alignment. We next define the main ingredients of E_t. For each frame t, we (i) build an *image SoG* from the image inside the object mask, and (ii) project a compact set of object Gaussians (*object SoG*) into the image under the current pose.

4.3.1 Image SoG. Given an image I_t, we seek a compact SoG $\mathcal{K}_{I_t}$ that represents coherent pixel regions inside the object mask M_t^o. A naive construction would assign one Gaussian to each pixel, but this is prohibitively expensive. Instead, we build a quad-tree over the object mask area to cluster pixels with similar color as shown in Fig. 6 (see [Stoll et al. 2011] for details).

4.3.2 Object SoG. We decode the canonical shape tokens $Z^{S\star}$ into a dense 3D Gaussian asset via SAM3D's Gaussian decoder. Generally, these photorealistic assets are very dense with hundreds

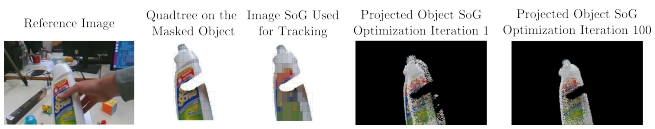

Fig. 6. **Approximating the image as Gaussians and projected object SoG.** From left to right: Reference frame, quad-tree build over the object area, image SoG used for tracking where each square represents a Gaussian with the same color, projected object SoG before optimization, projected object SoG at the end of optimization.

of thousands of Gaussians, which is unnecessary for tracking. We therefore sparsify the decoded asset using farthest-point sampling and keep 2000 Gaussians. The resulting object representation is a set of 3D Gaussians $\mathcal{G}_o = \{(\mu_j, \Sigma_j, c_j)\}_{j=1}^{N_o}$ where $\mu_j \in \mathbb{R}^3$ is the mean, Σ_j is the diagonal covariance, and c_j is the color. For faster convergence, we use an isotropic approximation by replacing each diagonal covariance with its average variance, following classical SoG tracking [Stoll et al. 2011]. Although exact perspective projection generally yields an anisotropic 2D footprint, we approximate the projected footprint to be isotropic. This enables the scalar closed-form Gaussian overlap in Eq. (5) for a smooth image-alignment loss. The anisotropic projection and its empirical trade-off are analyzed in Section 10.5 of the Supplementary Material.

Thus, SAM3D's dense Gaussians provide the canonical object geometry and appearance, while SoG provide the lightweight matching objective for tracking.

4.3.3 SoG alignment term. Now, we define the alignment energy as

$$E\big(\mathcal{K}_{I_t}, \Psi_t(\mathcal{G}_o)\big) = \sum_{i\in\mathcal{K}_{I_t}} \min\Big(\sum_{j\in\Psi_t(\mathcal{G}_o)} \upsilon_{j,t}\, E_{ij},\ E_{ii}\Big). \tag{3}$$

where $\Psi_t(\mathcal{G}_o)$ denotes the 2D object Gaussians obtained by projecting $\mathcal{G}_o$ into frame t using the current object pose and camera. The term E_{ij} is the color-weighted overlap between an image Gaussian i and a projected model Gaussian j, and E_{ii} is the self-overlap of i used for occlusion handling. A key difference from Gaussian Splatting is the simplified occlusion handling, which allows computing Gaussian similarities without sorting and in parallel (sums) instead of sequential, back-to-front alpha blending. Finally, $\upsilon_{j,t} \in \{0, 1\}$ gates out object Gaussians that are occluded by the hand (Section 4.3.4).

The pairwise overlap for two 2D Gaussians $\mathcal{B}_i$ and $\mathcal{B}_j$ is

$$E_{ij} = d(c_i, c_j) \int_\Omega \mathcal{B}_i(u)\, \mathcal{B}_j(u)\, du, \tag{4}$$

where (c_i, c_j) are the colors, $\Omega \subset \mathbb{R}^2$ is the image plane and $d(c_i, c_j)$ is a color kernel (see Section 10.4 of the Supplementary Material for details). With isotropic covariances, the overlap has closed form:

$$\int_\Omega \mathcal{B}_i(u)\mathcal{B}_j(u)\, du = 2\pi \frac{\sigma_i^2\sigma_j^2}{\sigma_i^2+\sigma_j^2} \exp\Big(-\frac{\|\mu_i-\mu_j\|^2}{2(\sigma_i^2+\sigma_j^2)}\Big). \tag{5}$$

4.3.4 Hand-aware SoG alignment. Unlike full-body tracking [Stoll et al. 2011], we leverage hand masks to restrict object-to-image matching to visible regions, preventing optimization drift during occlusions. Specifically, we exclude occluded Gaussians from the alignment using the visibility gate $\upsilon_{j,t} = \mathbf{1}\Big[M_t^h\Big(\text{round}(\mu_{j,t}^{2D})\Big) = 0\Big]$

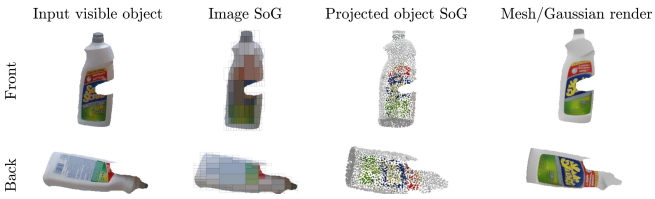

Fig. 7. **Comparing mesh/Gaussians-based optimization to SoG.** We show the differences between objectives in optimization using mesh/Gaussian rendering vs using SoG. Since MV-SAM3D/SAM3D has difficulty generating high-quality texture, specifically texture with text, trying to optimize the object pose with the rendered mesh/Gaussians does not work well, as shown in Table 2. In contrast, the SoG-based tracking formulation matches image regions more approximately and thus, even with inaccurate texture, we can still optimize the object pose.

where M_t^h is the binary hand mask and $\mu_{j,t}^{2D}$ is the projected mean of object Gaussian j.

4.3.5 Geometric and interaction priors. The alignment term E_t commits the projected object *interior* to image evidence, but cannot, on its own, resolve hand articulation, metric depth, or contact under occlusion. We therefore couple it with a small set of priors that bias the optimizer toward physically and temporally consistent solutions.

We pull projected hand and object boundaries toward the predicted-mask contours through a 2D signed distance-transform field, as shown in Fig. 5. Let Φ_t^x denote the signed distance transform of mask M_t^x, with $x \in \{o, h\}$, and let $\mathcal{B}_t^x$ be the visible boundary of the rendered shape (MANO silhouette vertices for the hand; projected object Gaussians on the rendered alpha contour for the object). Then, the silhouette loss can be defined as:

$$\mathcal{L}_{\text{sil}} = \sum_{t \in \mathcal{W}} \sum_{x \in \{o,h\}} \frac{1}{|\mathcal{B}_t^x|} \sum_{u \in \mathcal{B}_t^x} \left|\Phi_t^x(u)\right| \tag{6}$$

This boundary-only loss yields dense gradients for pose and articulation without requiring asset's photometric quality.

Dyn-HaMR [Yu et al. 2025] already provides strong temporal hand tracking and predicts per-frame hand poses $\{\mathbf{\Theta}_t, R_t^h, \tau_t^h\}_{t=1}^N$, where $\mathbf{\Theta}_t$ denotes articulated joint pose parameters, R_t^h the global root orientation, and τ_t^h the global translation. Therefore, we do not employ SoG tracking for the hand; instead, we regularize the initialized MANO trajectory with lightweight geometric priors: 2D joint reprojection, depth consistency, and local contact. We denote the joint reprojection term by $\mathcal{L}_{\text{j2d}}$ and provide its full formulation in Section 10.4 of the Supplementary Material To resolve depth ambiguity while staying robust to noisy monocular depth near occlusion boundaries, we use the pointmap P as an aggregate depth anchor rather than a dense objective. Let D_t^o and D_t^h denote rendered depths for the object and hand, and let D_t^{pm} denote the pointmap depth. We summarize these maps by median depths in the object and hand masks: $z_t^x = \text{median}(D_t^x[M_t^x])$ and $z_t^{\text{pm},x} = \text{median}(D_t^{\text{pm}}[M_t^x])$ for $x \in \{o, h\}$. Our depth term is

$$\mathcal{L}_{\text{depth}} = \sum_{t \in \mathcal{W}} \left(\left\| z_t^h - z_t^{\text{pm},h} \right\|_2^2 + \left\| z_t^o - z_t^{\text{pm},o} \right\|_2^2 + c_t^{ho} \left\| z_t^o - z_t^h \right\|_2^2 \right) \tag{7}$$

The first two terms anchor hand and object to the pointmap's metric scale. The third term is active in contact frames and ties the object depth to the hand, reducing failures where the object pointmap alone gives a visually plausible but grasp-inconsistent depth.

To ensure physical plausibility, we introduce a surface-aware contact loss $\mathcal{L}_{\text{contact}}$ that encourages designated hand contact zones [Hasson et al. 2019] to align with the object surface while penalizing interpenetration. We additionally use a smoothness loss $\mathcal{L}_{\text{smooth}}$ that penalizes temporal acceleration in translation and rotation. Detailed mathematical formulations of these tracking priors, along with their associated hyperparameters and weights in Eq. (2), are provided in Sections 10.2 and 10.4 of the Supplementary Material.

This optimization is highly lightweight because both the image and the object are processed as compact SoGs: the image is summarized by a quad-tree of 2D Gaussians, and the object is downsampled to at most 2000 Gaussians via farthest point sampling. This design revives the efficiency of classical SoG tracking [Stoll et al. 2011], while leveraging generative 3D models to obtain an accurate canonical object and a strong initialization in the wild.

5 Experiments

5.1 Datasets

HO3Dv3 Dataset. HO3Dv3 [Hampali et al. 2021] contains monocular videos of single-hand object interactions. We use the 18 sequences provided in the HOLD [Fan et al. 2024] evaluation set.

HOT3D. HOT3D [Banerjee et al. 2025] is an egocentric dataset with accurate 3D poses and shapes of hands and objects. Since the dataset does not provide ground-truth annotations for its test set, we instead select 18 sequences from the training set. Specifically, we extract segments from the full training videos that contain only single hand–object interactions with different objects. Before running the methods on HOT3D, the fisheye videos are undistorted.

5.2 Metrics

We decode shape tokens into meshes via SAM3D's decoder and evaluate following HOLD [Fan et al. 2024]. Hand accuracy is measured via root-relative MPJPE (mm). To evaluate object template quality independent of pose, we ICP-align meshes to compute Chamfer distance (CD, cm) and F-score at 10 mm (F10) [Fan et al. 2024]. To measure object pose and shape relative to the hand in 3D, we subtract the predicted hand root from the object vertices and compute the hand-relative CD for the object (CD_h) [Fan et al. 2024]. We also report Success Rate (SR) where a sequence is deemed as a failure if the method fails to output a valid result such that when initialization breaks down (notably for SfM-based pipelines) or when tracking degrades severely such that the hand-relative CD reaches 1000 cm. As methods operate with different clip lengths, we report normalized runtime (hours per 100 frames) on a single NVIDIA RTX4090 GPU to quantify efficiency, including preprocessing and optimization.

5.3 Implementation Details

We optimize each sequence with AdamW [Loshchilov and Hutter 2017] in a sliding-window manner: running 100 iterations per window of 8 frames (stride 7), sequentially covering the full video.

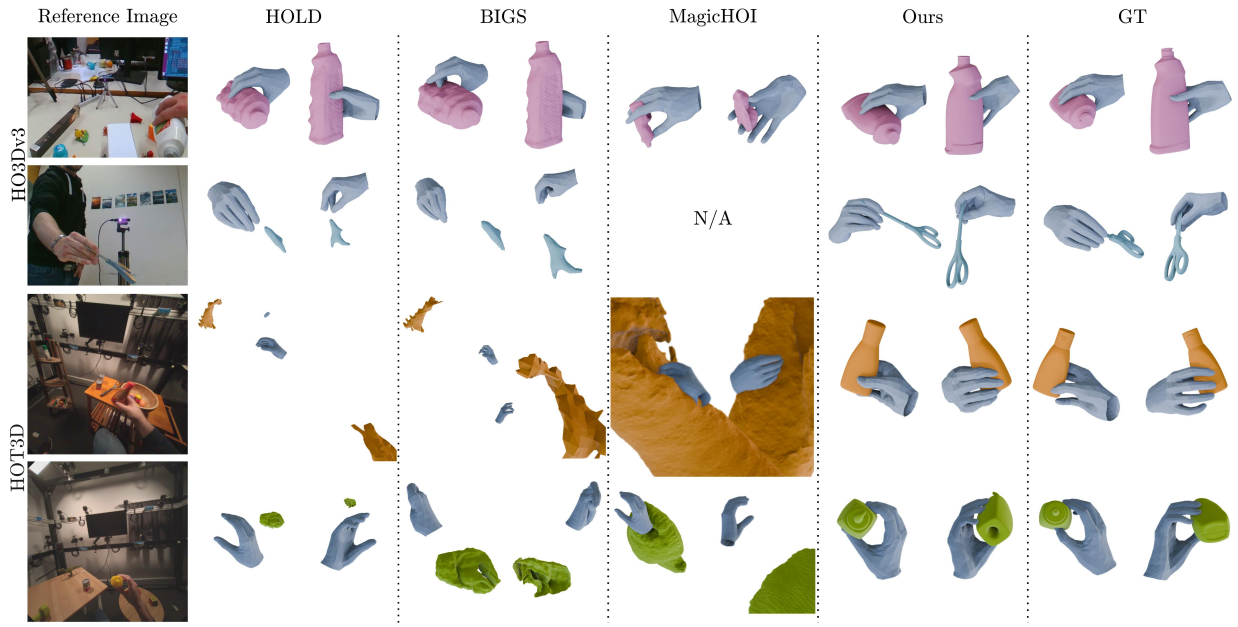

Fig. 8. **Qualitative Comparison.** We compare the output of GraG with previous SoTA works HOLD, BIGS, and MagicHOI on HO3D (first 2 rows) and HOT3D (last 2 rows). In the 2nd row 4th column, MagicHOI fails to produce a valid reconstruction; we therefore report it as N/A. Overall, GraG preserves sharper object geometry and yields more plausible hand poses (with fewer interpenetrations), while being substantially more efficient.

Within each window, we optimize the hand and object pose parameters, while keeping the canonical object and hand shape fixed after initialization. On a single RTX4090 GPU, for a 100-frame sequence, Stages 1-2 take ~6 minutes on average and Stage 3 takes ~10 minutes. See Section 10.2 of the Supplementary Material for the hyper-parameters.

5.4 Results

Compared SoTA Methods. We compare GraG to state-of-the-art methods including HOLD [Fan et al. 2024], BIGS [On et al. 2025], and MagicHOI [Wang et al. 2025], following HOLD's evaluation protocol. HOLD is a NeRF-based method that jointly optimizes hand–object scenes using SDF representations. BIGS reconstructs HOI scenes using 3D Gaussians with a triplane MLP and SDS loss. MagicHOI extends HOLD by leveraging novel-view synthesis models to hallucinate occluded object regions. Note that while MagicHOI is designed for short clips specifically, we still include the comparison for the sake of completeness. These are full-system comparisons: GraG benefits from stronger object and hand priors (SAM3D, Dyn-HaMR) that were not available in HOLD or BIGS. The quantitative gains reflect both these priors and our lightweight tracking formulation.

Quantitative Results. We report quantitative results in Table 1. On HO3Dv3, our method achieves the best object reconstruction quality, reducing *CD* by 23.9% over BIGS and 34.6% over HOLD while increasing F10 to 97.2%, partly due to the stronger SAM3D-based object prior. GraG also obtains the second-best MPJPE (9.38 mm) and preserves a 100% success rate. MagicHOI obtains the lowest MPJPE on successful clips, but its lower success rate and high CD_h indicate unstable hand-object alignment on full sequences. Notably, GraG is also substantially faster, requiring 0.27 hours per 100 frames compared to 1.2 for MagicHOI, 3.6 for BIGS, and 10.5 for HOLD.

Table 1. **Quantitative comparison.** We report object reconstruction, hand accuracy, success rate, and runtime on HO3Dv3 and HOT3D. We highlight the best and second-best results. All methods are averaged over the *successful results* only. (*)BIGS results on HO3Dv3 differ from those reported by [On et al. 2025]; we obtain them by running the authors' open-source implementation with the reported hyperparameters.

Dataset	Method	CD [cm] ↓	F10 [%] ↑	MPJPE [mm] ↓	CD_h [cm] ↓	Success Rate [%] ↑	Runtime [h] ↓
HO3Dv3	HOLD	0.78	92.0	23.4	**4.27**	**100**	10.5
	BIGS*	0.67	94.1	23.9	7.61	**100**	3.60
	MagicHOI	1.58	73.7	**4.35**	129	67.0	1.20
	Ours	**0.51**	**97.2**	9.38	5.02	**100**	**0.27**
HOT3D	HOLD	3.14	53.3	21.9	122	50.0	10.5
	BIGS	2.92	51.7	36.9	84.2	50.0	3.60
	MagicHOI	22.6	33.6	**18.4**	97.1	55.6	1.20
	Ours	**0.67**	**94.7**	20.6	**9.37**	**100**	**0.27**

On the more challenging HOT3D dataset, characterized by egocentric viewpoints, rapid camera motion, and heavy hand occlusions, our method further improves both object reconstruction and hand–object alignment. We reduce *CD* by 77.1% over BIGS and 78.7% over HOLD, and achieve the best hand-object alignment, reducing CD_h by 88.9% over the strongest baseline. Our method also achieves a 100% success rate, whereas SfM-based baselines drop to 50–56%. In long egocentric sequences, COLMAP-based initialization is sensitive to frame sampling and often produces unstable camera poses, which leads to drift or failed reconstructions. Consequently, methods that depend on such initialization (HOLD, BIGS, and MagicHOI)

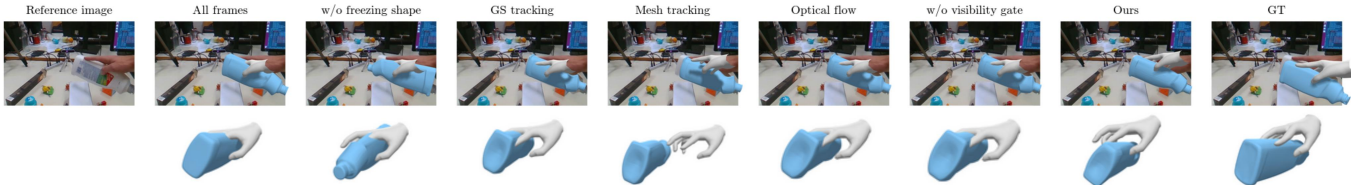

Fig. 9. **Ablation experiments.** We visualize how key design choices affect reconstruction quality (two representative views per setting: camera view and back view). Using all frames during reconstruction yields an inaccurate canonical object shape and scale, leading to implausible grasps. Running Stage 2 SAM3D pose estimation without freezing the shape tokens destabilizes per-frame pose estimates (note that Stage 3 still utilizes the Stage 1 canonical object in this experiment). Replacing our compact SoG refinement with dense Gaussian Splatting (GS) tracking is slower and often fails to converge under the same iteration budget. Similarly, replacing SoG with optical flow causes tracking to collapse during hand-object occlusion, degrading the final reconstruction. Without the visibility gate, SoG alignment is biased by hand-occluded regions, producing incorrect object poses. Our full model achieves the highest fidelity, closely matching the ground truth.

frequently fail to converge or produce inconsistent object tracking. Specifically, MagicHOI is designed for short-clip evaluation; when applied to our full-length HO3Dv3 and egocentric HOT3D sequences, its COLMAP-based initialization frequently becomes unstable, reducing its success rate.

Qualitative Results. As shown in Fig. 8, our method accurately reconstructs hands and objects from video, preserving finer geometric details and more coherent spatial placement than prior SoTA methods. Specifically on HOT3D dataset, our approach demonstrates superior performance; it recovers thin structures (e.g., scissors) and maintains plausible object scale under occlusion, whereas competing methods either fail to reconstruct meaningful object geometry or diverge during optimization, largely due to their reliance on COLMAP/SfM initialization. Our results exhibit fewer misalignments and more realistic grasping configurations. See Fig. 13 in the Supplementary Material for overlays on reference images.

6 Ablations

We validate our design choices by ablating on two HO3Dv3 sequences and two HOT3D sequences.

Results are presented in Table 2, Fig. 9 and the Supplementary Material. For Stage 1, balanced keyframes give better object reconstruction than single frame, random keyframes or using all frames, showing that view quality is more important than simply increasing the number of SAM3D inputs, as also shown in Fig. 10. In Stage 2, freezing the canonical shape and adding temporal guidance improve hand-relative object pose error, indicating more stable video-level pose adaptation. Most importantly, the "only foundation models" ablation evaluates the same SAM3D object and Dyn-HaMR hand priors without our Stage 3 refinement, and yields much higher hand-object error (CD_h=20.62). The gap to the full model (CD_h=7.65) comes from the whole Stage 3, which uses both the pretrained priors and the SoG alignment, and should not be read as the effect of SoG alone. When we drop only the SoG tracking term but keep the Stage 3 priors, CD_h is already 10.45, which shows that the priors recover most of the error after "only foundation models", and SoG closes the remaining gap (from 10.45 to 7.65). The priors are what make the hand-relative pose accurate, while the compact hand-aware SoG mainly provides an efficient tracking signal. We further replace SoG with dense Gaussian rendering, mesh rendering, and Lucas–Kanade[Lucas and Kanade 1981] optical-flow tracking baselines. Dense Gaussian and mesh rendering optimize RGB render-and-compare losses from the SAM3D asset, but are sensitive to generated texture and hand occlusion. Optical flow tracks projected object surface points through the video, but tracks are often lost under occlusion, fast rotation, and large appearance changes. All alternatives underperform the full SoG objective, mesh and dense Gaussian tracking is also slower on the ablation set. Among losses, contact is the most critical for hand-object alignment; depth, silhouette, and visibility gating improve robustness to occlusion.

Table 2. **Ablation study.** We ablate key components of our method on 4 sequences, 2 sequences from HO3Dv3 and 2 sequences from HOT3D. The first block ablates Stage 1 object reconstruction, the second block ablates Stage 2 SAM3D pose adaptation, and the third block ablates Stage 3 tracking objectives. For Stage 2 and Stage 3 variants, the same canonical object reconstructed with balanced keyframes is used, so CD and F10 remain unchanged. The "w/o freezing shape" row does not freeze the shape latent during SAM3D pose adaptation, but still uses the Stage 1 object during Stage 3 tracking. Runtime is normalized per 100 frames.

Ablation	CD [cm] ↓	F10 [%] ↑	MPJPE [mm] ↓	CD_h [cm] ↓	Runtime [h] ↓
random $K = 4$ keyframe selection	0.69	93.9	11.5	9.36	0.27
single view from first frame	0.73	92.1	18.2	22.73	0.27
using all frames	0.68	93.6	11.5	9.95	0.30
w/o freezing shape	0.45	98.8	11.4	10.81	0.28
w/o temporal guidance (λ_{temp}=0)	0.45	98.8	11.5	10.60	0.27
only foundation models (w/o Stage 3)	0.45	98.8	12.1	20.62	**0.10**
Gaussian Splatting tracking instead of SoG	0.45	98.8	11.8	10.46	1.20
mesh rendering instead of SoG	0.45	98.8	13.6	12.60	2.40
optical flow tracking instead of SoG	0.45	98.8	11.8	11.45	0.24
opt. w/o SoG refinement (only prior losses)	0.45	98.8	11.5	10.45	0.21
w/o visibility gate (no hand-occlusion gating)	0.45	98.8	11.5	9.35	0.27
w/o depth loss $\mathcal{L}_{\text{depth}}$	0.45	98.8	11.4	12.70	0.27
w/o silhouette loss $\mathcal{L}_{\text{sil}}$	0.45	98.8	11.5	10.71	0.25
w/o contact loss $\mathcal{L}_{\text{contact}}$	0.45	98.8	11.5	19.58	0.21
full model	**0.45**	**98.8**	**11.3**	**7.65**	0.27

7 Application

To illustrate a downstream robotic use case, we interface GraG with SimToolReal [Kedia et al. 2026], an object-centric framework for zero-shot dexterous tool manipulation. SimToolReal represents a task as a sequence of desired 6D object goal poses, and its policy observes the current robot/object state together with the relative

Fig. 10. **Effect of keyframe selection on object reconstruction.** Random keyframe selection and using all frames for reconstruction degrade the canonical reconstruction (shape/scale) compared to our balanced keyframe selection; ground truth shown for reference.

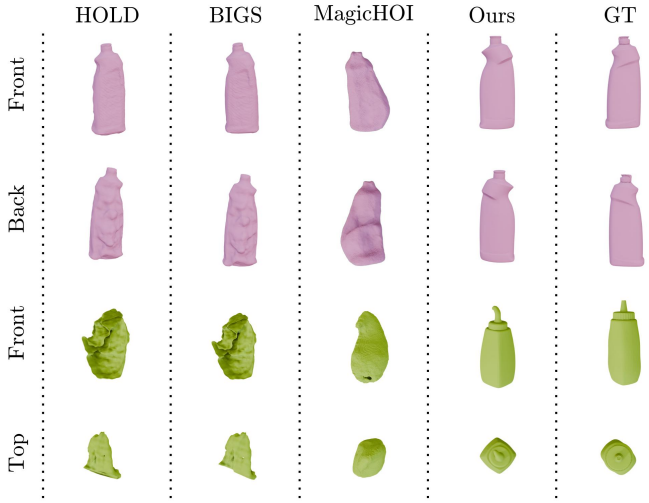

Fig. 11. **Object-only reconstruction comparison.** Front/back views of reconstructed object geometry; ground truth shown for reference.

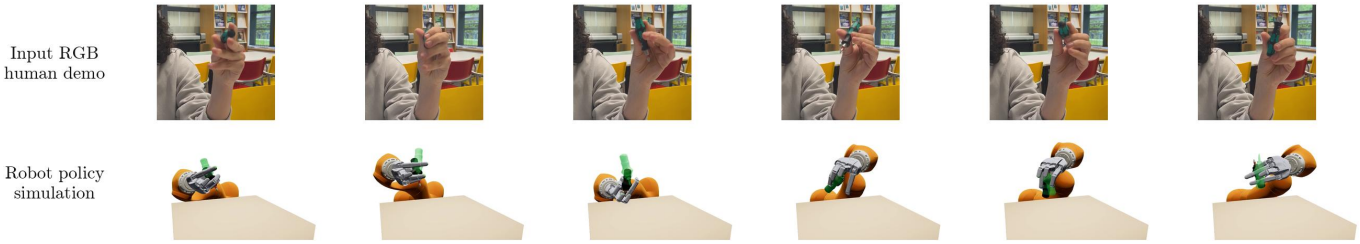

Fig. 12. **Human-to-robot tool manipulation from GraG object trajectories.** Top: monocular RGB frames from a human short-screwdriver demonstration. Bottom: corresponding visualizations of the SimToolReal policy observations. The green screwdriver indicates the current goal pose obtained from GraG's reconstructed 6D object trajectory (Zoom in for best view). The policy receives the robot state, estimated object pose, and goal pose; once the current object pose is sufficiently close to the goal, the goal advances to the next pose in the sequence. This visualization renders the policy inputs and goal-tracking behavior, rather than a physics-based real-robot execution.

object-to-goal keypoints. We record a monocular human demonstration of an in-house short screwdriver, reconstruct its 6D object trajectory with GraG, and convert this trajectory into the world-frame goal-pose sequence used by SimToolReal. We then evaluate the pretrained short-screwdriver policy on the corresponding vertical-spinning task, and visualize it in Fig. 12 and the supplementary video. As shown in the supplementary video, SimToolReal successfully manipulates the screwdriver based on the object geometry and poses specified by GraG. It is noteworthy that while robust, incorrect object tracking contributes to as many as 44% failure cases in SimToolReal. Although demonstrated qualitatively, GraG's ability to effortlessly integrate with SimToolReal demonstrates the generalization potential of our method.

8 Limitations

While using Depth Anything 3 pointmaps as aggregate metric anchors minimizes depth error, severe pointmap failures can still bias object scale and translation. Similarly, our color-weighted SoG alignment is robust to texture imperfections but remains vulnerable to illumination changes, shadows, specularities, and weak object texturing. Our alignment term approximately projects 3D primitives to isotropic 2D footprints. This can lose fidelity under large viewpoint changes or partial cropping near the image boundary, where exact perspective projection may produce elliptical 2D footprints. We use this approximation to have a simple but closed-form scalar overlap. Extending both object and image SoGs to anisotropic 2D covariances or a hybrid version would be an interesting future direction.

Our pipeline also depends on high-quality hand and object masks, which can be challenging to recover under extreme occlusion. Finally, the current system is restricted to single-hand manipulation of

a single rigid object. We leave bimanual interactions and deformable objects for the future work to explore.

9 Conclusion

We introduced Grasp in Gaussians (**GraG**), an efficient and scalable framework for reconstructing dynamic 3D hand-object interactions from a single monocular video. We showed that naïvely combining the 2D/3D foundational models can produce suboptimal results, particularly in terms of optimization runtime. We addressed this by re-introducing the idea of Sum-of-Gaussian tracking and showed how a sparse set of Gaussians can be tracked efficiently. Experiments demonstrate that, as a full system built on modern object and hand priors, GraG improves reconstruction quality and robustness while substantially reducing runtime compared to prior video-based HOI reconstruction methods. In future work, we aim to integrate physical plausibility of the reconstruction to improve robustness and extend the pipeline to multiple hands and interacting objects.

Acknowledgments

This project was supported by the Saarbrücken Research Center for Visual Computing, Interaction, and AI. We would like to thank the anonymous reviewers for constructive comments and suggestions.

References

Ayce Idil Aytekin, Helge Rhodin, Rishabh Dabral, and Christian Theobalt. 2026. Follow My Hold: Hand-Object Interaction Reconstruction Through Geometric Guidance. In *2026 International Conference on 3D Vision (3DV)*. 795–805. doi:10.1109/3DV69130.2026.00081

Prithviraj Banerjee, Sindi Shkodrani, Pierre Moulon, Shreyas Hampali, Shangchen Han, Fan Zhang, Linguang Zhang, Jade Fountain, Edward Miller, Selen Basol, et al. 2025. Hot3d: Hand and object tracking in 3d from egocentric multi-view videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7061–7071.

L. Bretzner, I. Laptev, and T. Lindeberg. 2002. Hand gesture recognition using multi-scale colour features, hierarchical models and particle filtering. In *Automatic Face and Gesture Recognition*. 423–428. doi:10.1109/AFGR.2002.1004190

Zhe Cao, Ilija Radosavovic, Angjoo Kanazawa, and Jitendra Malik. 2021. Reconstructing hand-object interactions in the wild. In *Proceedings of the IEEE/CVF international conference on computer vision*. 12417–12426.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. 2025. Sam 3: Segment anything with concepts. *arXiv preprint arXiv:2511.16719* (2025).

Yu-Wei Chao, Wei Yang, Yu Xiang, Pavlo Molchanov, Ankur Handa, Jonathan Tremblay, Yashraj S Narang, Karl Van Wyk, Umar Iqbal, Stan Birchfield, et al. 2021. Dexycb: A benchmark for capturing hand grasping of objects. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 9040–9049.

Xingyu Chen, Fu-Jen Chu, Pierre Gleize, Kevin J Liang, Alexander Sax, Hao Tang, Weiyao Wang, Michelle Guo, Thibaut Hardin, Xiang Li, et al. 2025. Sam 3d: 3dfy anything in images. *arXiv preprint arXiv:2511.16624* (2025).

Yuantao Chen, Jiahao Chang, Chongjie Ye, Chaoran Zhang, Zhaojie Fang, Chenghong Li, and Xiaoguang Han. 2026. ForeHOI: Feed-forward 3D Object Reconstruction from Daily Hand-Object Interaction Videos. *arXiv preprint arXiv:2602.06226* (2026).

Xiaoyan Cong, Angela Xing, Chandradeep Pokhariya, Rao Fu, and Srinath Sridhar. 2025. DyTact: Capturing Dynamic Contacts in Hand-Object Manipulation. *arXiv preprint arXiv:2506.03103* (2025).

Enric Corona, Albert Pumarola, Guillem Alenya, Francesc Moreno-Noguer, and Grégory Rogez. 2020. Ganhand: Predicting human grasp affordances in multi-object scenes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 5031–5041.

Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. 2024. Vision transformers need registers. In *International Conference on Learning Representations*, Vol. 2024. 2632–2652.

Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. 2023. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 13142–13153.

A. Elhayek, E. Aguiar, A. Jain, J. Tompson, L. Pishchulin, M. Andriluka, C. Bregler, B. Schiele, and C. Theobalt. 2015. Efficient ConvNet-based Marker-less Motion Capture in General Scenes with a Low Number of Cameras. In *CVPR*.

Zicong Fan, Maria Parelli, Maria Eleni Kadoglou, Xu Chen, Muhammed Kocabas, Michael J Black, and Otmar Hilliges. 2024. Hold: Category-agnostic 3d reconstruction of interacting hands and objects from video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 494–504.

Zicong Fan, Omid Taheri, Dimitrios Tzionas, Muhammed Kocabas, Manuel Kaufmann, Michael J Black, and Otmar Hilliges. 2023. ARCTIC: A dataset for dexterous bimanual hand-object manipulation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 12943–12954.

Google. 2025. Gemini 3 Flash Preview. https://ai.google.dev/gemini-api/docs/models/gemini-3-flash-preview. Model identifier: `gemini-3-flash-preview`.

Shreyas Hampali, Tomas Hodan, Luan Tran, Lingni Ma, Cem Keskin, and Vincent Lepetit. 2023. In-hand 3d object scanning from an rgb sequence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 17079–17088.

Shreyas Hampali, Sayan Deb Sarkar, and Vincent Lepetit. 2021. Ho-3d_v3: Improving the accuracy of hand-object annotations of the ho-3d dataset. *arXiv preprint arXiv:2107.00887* (2021).

Richard Hartley and Andrew Zisserman. 2011. *Multiple View Geometry in Computer Vision* (2 ed.). Cambridge University Press, Cambridge, England.

Yana Hasson, Bugra Tekin, Federica Bogo, Ivan Laptev, Marc Pollefeys, and Cordelia Schmid. 2020. Leveraging photometric consistency over time for sparsely supervised hand-object reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 571–580.

Yana Hasson, Gül Varol, Cordelia Schmid, and Ivan Laptev. 2021. Towards unconstrained joint hand-object reconstruction from rgb videos. In *2021 International Conference on 3D Vision (3DV)*. IEEE, 659–668.

Yana Hasson, Gul Varol, Dimitrios Tzionas, Igor Kalevatykh, Michael J Black, Ivan Laptev, and Cordelia Schmid. 2019. Learning joint reconstruction of hands and manipulated objects. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 11807–11816.

Haoyu Hu, Xinyu Yi, Zhe Cao, Jun-Hai Yong, and Feng Xu. 2024. Hand-object interaction controller (hoic): Deep reinforcement learning for reconstructing interactions with physics. In *Acm siggraph 2024 conference papers*. 1–10.

Haoyu Hu, Xinyu Yi, Hao Zhang, Jun-Hai Yong, and Feng Xu. 2022. Physical interaction: Reconstructing hand-object interactions with physics. In *SIGGRAPH Asia 2022 conference papers*. 1–9.

Di Huang, Xiaopeng Ji, Xingyi He, Jiaming Sun, Tong He, Qing Shuai, Wanli Ouyang, and Xiaowei Zhou. 2022. Reconstructing hand-held objects from monocular video. In *SIGGRAPH Asia 2022 Conference Papers*. 1–9.

Daniel Sungho Jung and Kyoung Mu Lee. 2025. Learning Dense Hand Contact Estimation from Imbalanced Data. *Advances in Neural Information Processing Systems* (2025).

Korrawe Karunratanakul, Jinlong Yang, Yan Zhang, Michael J Black, Krikamol Muandet, and Siyu Tang. 2020. Grasping field: Learning implicit representations for human grasps. In *2020 International Conference on 3D Vision (3DV)*. IEEE, 333–344.

Kushal Kedia, Tyler Ga Wei Lum, Jeannette Bohg, and C Karen Liu. 2026. SimToolReal: An Object-Centric Policy for Zero-Shot Dexterous Tool Manipulation. *arXiv preprint arXiv:2602.16863* (2026).

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics* 42, 4 (July 2023). https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/

Zeqiang Lai, Yunfei Zhao, Haolin Liu, Zibo Zhao, Qingxiang Lin, Huiwen Shi, Xianghui Yang, Mingxin Yang, Shuhui Yang, Yifei Feng, et al. 2025. Hunyuan3D 2.5: Towards High-Fidelity 3D Assets Generation with Ultimate Details. *arXiv preprint arXiv:2506.16504* (2025).

Baicheng Li, Dong Wu, Jun Li, Shunkai Zhou, Zecui Zeng, Lusong Li, and Hongbin Zha. 2026. MV-SAM3D: Adaptive Multi-View Fusion for Layout-Aware 3D Generation. arXiv:2603.11633 [cs.CV] https://arxiv.org/abs/2603.11633

Haotong Lin, Sili Chen, Jun Hao Liew, Donny Y. Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. 2025. Depth Anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:2511.10647* (2025).

Shaowei Liu, Hanwen Jiang, Jiarui Xu, Sifei Liu, and Xiaolong Wang. 2021. Semi-supervised 3d hand-object poses estimation with interactions in time. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 14687–14697.

Xingyu Liu, Pengfei Ren, Qi Qi, Haifeng Sun, Zirui Zhuang, Jing Wang, Jianxin Liao, and Jingyu Wang. 2025b. Generalizable Hand-Object Modeling from Monocular RGB Images via 3D Gaussians. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Yumeng Liu, Xiaoxiao Long, Zemin Yang, Yuan Liu, Marc Habermann, Christian Theobalt, Yuexin Ma, and Wenping Wang. 2025a. EasyHOI: Unleashing the Power of Large Models for Reconstructing Hand-Object Interactions in the Wild. In *Proceedings of the Computer Vision and Pattern Recognition Conference*. 7037–7047.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101* (2017).

Bruce D Lucas and Takeo Kanade. 1981. An iterative image registration technique with an application to stereo vision. In *IJCAI'81: 7th international joint conference on Artificial intelligence*, Vol. 2. 674–679.

Thomas B. Moeslund, Adrian Hilton, and Volker Krüger. 2006. A Survey of Advances in Vision-based Human Motion Capture and Analysis. *CVIU* 104, 2 (2006), 90–126. doi:10.1016/j.cviu.2006.08.002

Jeongwan On, Kyeonghwan Gwak, Gunyoung Kang, Junuk Cha, Soohyun Hwang, Hyein Hwang, and Seungryul Baek. 2025. BIGS: Bimanual Category-agnostic Interaction Reconstruction from Monocular Videos via 3D Gaussian Splatting. In *Proceedings of the Computer Vision and Pattern Recognition Conference*. 17437–17447.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. 2023. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193* (2023).

JoonKyu Park, Yeonguk Oh, Gyeongsik Moon, Hongsuk Choi, and Kyoung Mu Lee. 2022. Handoccnet: Occlusion-robust 3d hand mesh estimation network. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 1496–1505.

Georgios Pavlakos, Dandan Shan, Ilija Radosavovic, Angjoo Kanazawa, David Fouhey, and Jitendra Malik. 2024. Reconstructing hands in 3d with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9826–9836.

R. Plankers and P. Fua. 2003. Articulated soft objects for multiview shape and motion capture. *PAMI* 25, 9 (2003), 1182–1187.

Frano Rajič, Haofei Xu, Marko Mihajlovic, Siyuan Li, Irem Demir, Emircan Gündoğdu, Lei Ke, Sergey Prokudin, Marc Pollefeys, and Siyu Tang. 2025. Multi-view 3d point tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 59–68.

Carl Yuheng Ren, Victor Prisacariu, Olaf Kaehler, Ian Reid, and David Murray. 2014. 3D Tracking of Multiple Objects with Identical Appearance Using RGB-D Input. In *3DV*. 47–54. doi:10.1109/3DV.2014.39

Helge Rhodin, Christian Richardt, Dan Casas, Eldar Insafutdinov, Mohammad Shafiei, Hans-Peter Seidel, Bernt Schiele, and Christian Theobalt. 2016a. Egocap: egocentric marker-less motion capture with two fisheye cameras. *ACM Transactions on Graphics (TOG)* 35, 6 (2016), 1–11.

Helge Rhodin, Nadia Robertini, Dan Casas, Christian Richardt, Hans-Peter Seidel, and Christian Theobalt. 2016b. General automatic human shape and motion capture using volumetric contour cues. In *European conference on computer vision*. Springer, 509–526.

Helge Rhodin, Nadia Robertini, Christian Richardt, Hans-Peter Seidel, and Christian Theobalt. 2015. A versatile scene model with differentiable visibility applied to generative pose estimation. In *Proceedings of the IEEE international conference on computer vision*. 765–773.

Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. 2019. From coarse to fine: Robust hierarchical localization at large scale. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 12716–12725.

Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. 2020. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 4938–4947.

D Scharstein, R Szeliski, and R Zabih. 2002. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. In *Proceedings IEEE Workshop on Stereo and Multi-Baseline Vision (SMBV 2001)* (Kauai, HI, USA). IEEE Comput. Soc.

Jin-Chuan Shi, Binhong Ye, Tao Liu, Xiaoyang Liu, Yangjinhui Xu, Junzhe He, Zeju Li, Hao Chen, and Chunhua Shen. 2026. AGILE: Hand-Object Interaction Reconstruction from Video via Agentic Generation. *arXiv preprint arXiv:2602.04672* (2026).

Srinath Sridhar, Franziska Mueller, Antti Oulasvirta, and Christian Theobalt. 2015. Fast and Robust Hand Tracking Using Detection-Guided Optimization. In *CVPR*. http://handtracker.mpi-inf.mpg.de/projects/FastHandTracker/

Srinath Sridhar, Helge Rhodin, Hans-Peter Seidel, Antti Oulasvirta, and Christian Theobalt. 2014. Real-time hand tracking using a sum of anisotropic gaussians model. In *2014 2nd International Conference on 3D Vision*, Vol. 1. IEEE, 319–326.

Carsten Stoll, Nils Hasler, Juergen Gall, Hans-Peter Seidel, and Christian Theobalt. 2011. Fast articulated motion tracking using a sums of gaussians body model. In *2011 international conference on computer vision*. IEEE, 951–958.

Richard Szeliski. 2010. *Computer Vision*. Springer, London, England.

V Team, Wenyi Hong, Wenmeng Yu, et al. 2025. GLM-4.5 V and GLM-4.1 V-Thinking: Towards Versatile Multimodal Reasoning with Scalable Reinforcement Learning. *arXiv preprint arXiv:2507.01006* (2025).

Bugra Tekin, Federica Bogo, and Marc Pollefeys. 2019. H+ o: Unified egocentric recognition of 3d hand-object poses and interactions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 4511–4520.

Carlo Tomasi and Takeo Kanade. 1992. Shape and motion from image streams under orthography: a factorization method. *Int. J. Comput. Vis.* 9, 2 (Nov. 1992), 137–154.

Shibo Wang, Haonan He, Maria Parelli, Christoph Gebhardt, Zicong Fan, and Jie Song. 2025. MagicHOI: Leveraging 3D Priors for Accurate Hand-object Reconstruction from Short Monocular Video Clips. *arXiv preprint arXiv:2508.05506* (2025).

Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. 2025. Structured 3D Latents for Scalable and Versatile 3D Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 21469–21480.

Hao Xu, Tianyu Wang, Xiao Tang, and Chi-Wing Fu. 2023. H2onet: Hand-occlusion-and-orientation-aware network for real-time 3d hand mesh reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 17048–17058.

Lixin Yang, Xinyu Zhan, Kailin Li, Wenqiang Xu, Jiefeng Li, and Cewu Lu. 2021. Cpf: Learning a contact potential field to model the hand-object interaction. In *Proceedings of the IEEE/CVF international conference on computer vision*. 11097–11106.

Yufei Ye, Abhinav Gupta, and Shubham Tulsiani. 2022. What's in your hands? 3d reconstruction of generic objects in hands. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 3895–3905.

Yufei Ye, Poorvi Hebbar, Abhinav Gupta, and Shubham Tulsiani. 2023. Diffusion-guided reconstruction of everyday hand-object interaction clips. In *Proceedings of the IEEE/CVF international conference on computer vision*. 19717–19728.

Zhengdi Yu, Stefanos Zafeiriou, and Tolga Birdal. 2025. Dyn-hamr: Recovering 4d interacting hand motion from a dynamic camera. In *Proceedings of the Computer Vision and Pattern Recognition Conference*. 27716–27726.

Chengwei Zheng, Lixin Xue, Juan Zarate, and Jie Song. 2025. GauSTAR: Gaussian Surface Tracking and Reconstruction. In *Proceedings of the Computer Vision and Pattern Recognition Conference*. 16543–16553.

Licheng Zhong, Lixin Yang, Kailin Li, Haoyu Zhen, Mei Han, and Cewu Lu. 2024. Color-NeuS: Reconstructing neural implicit surfaces with color. In *2024 International Conference on 3D Vision (3DV)*. IEEE, 631–640.

Yu Zhong, Anil K Jain, and M-P Dubuisson-Jolly. 2000. Object tracking using deformable templates. *IEEE transactions on pattern analysis and machine intelligence* 22, 5 (2000), 544–549.

Matthias Zwicker, Hanspeter Pfister, Jeroen Van Baar, and Markus Gross. 2001. Ewa volume splatting. In *Proceedings Visualization, 2001. VIS'01*. IEEE, 29–538.

M. Zwicker, H. Pfister, J. van Baar, and M. Gross. 2002. EWA splatting. *IEEE Transactions on Visualization and Computer Graphics* 8, 3 (2002), 223–238. doi:10.1109/TVCG.2002.1021576

10 Supplementary Material

For consistency across datasets and comparison methods, we resample all input videos to **10 fps** prior to preprocessing.

10.1 Additional Evaluation

Following ARCTIC [Fan et al. 2023], we also report interaction metrics in Table 4 such as Contact Deviation (CDev) that measures hand-object contact accuracy, Acceleration Error (ACC) to measure motion smoothness, Mean Relative-Root Position Error (MRRPE) to measure the root translation between hand and object, and Motion Deviation (MDev) to check when a hand moves an object, if the vertices of the hand and the object in stable contact move together. We don't report Average Articulation Error (AAE) since it is for object articulation, and articulated objects are out of our scope. For detailed explanations of these metrics, please check ARCTIC [Fan et al. 2023]. Please note that, for object acceleration error, we use a topology-agnostic variant rather than the original ARCTIC definition. Since different methods often output different object meshes, predicted and ground-truth meshes generally do not share vertex correspondence or indexing. Therefore, we measure object acceleration error by comparing the accelerations of object root trajectories over time.

Table 4 shows that GraG consistently improves interaction consistency, with the strongest gains on HOT3D. On HO3Dv3, our method achieves the best temporal interaction quality (best MDev and ACC_h), while remaining close to the strongest contact/pose baseline HOLD. This shows that our predictions are not only accurate in static contact proximity, but also more coherent over time during manipulation.

On HOT3D, the margin is larger: we substantially outperform prior methods in CDev, MDev, and $\text{MRRPE}_{h\to o}$. In particular, CDev drops from the 451–570 mm range of HOLD/BIGS to 23.8 mm, and $\text{MRRPE}_{h\to o}$ drops from 606–762 mm to 155 mm. These improvements indicate better hand-object coupling under more challenging conditions (dynamic camera, larger motion, and noisier geometry). HOT3D setting is particularly difficult for the comparison methods due to them depending on COLMAP/SfM initializations. ACC_o shows a complementary trend. MagicHOI obtains lower object acceleration error but with much worse CDev/$\text{MRRPE}_{h\to o}$ because its object trajectory is often overly smooth in long sequences. When COLMAP/SfM fails, the recovered camera/object motion can become nearly static, producing low acceleration despite incorrect object pose. Thus ACC_o alone does not indicate accurate object tracking. Our method GraG provides a better balance: low ACC_o together with strong contact and relative-pose consistency, which is more desirable for physically plausible manipulation.

Number of Gaussians trade-off. Increasing the Gaussian count substantially increases compute while giving little measurable improvement on this ablation set. Moving from Sparse SoG to the reduced SAM3D Gaussian asset increases runtime by about 4.44× and peak CUDA memory by about 2.3×, but changes reconstruction metrics only marginally. Using the full dense Gaussian asset is much more expensive: it is over two orders of magnitude slower than Sparse SoG, while yielding no meaningful gain in CD, F10, MPJPE, or hand-object reconstruction quality. This supports our design choice of optimizing a compact sparse isotropic SoG: it preserves the useful geometric signal from SAM3D while avoiding the optimization cost of tracking tens or hundreds of thousands of Gaussians.

Table 3. **Trade-off between Gaussian count, compute, memory, and reconstruction quality** on the ablation set. Runtime is normalized to hours per 100 frames as in the main paper setting. "Sparse SoG" is our default setting where we use farthest point sampling to subsample from the reduced SAM3D Gaussian asset and convert to isotropic Gaussians. "Reduced" denotes SAM3D's lower-count Gaussian decoder output, trained for downstream needs such as decoding fewer Gaussians for faster inference (see SAM3D's Supplementary Material for detailed information); "Full" denotes the dense SAM3D Gaussian decoder output. Storage reports the memory occupied by the object Gaussian parameter tensors, while Peak CUDA reports the maximum CUDA memory allocated during optimization. Thus, Storage measures representation compactness, whereas Peak CUDA captures practical optimization-time GPU memory.

Setting	Number of Gaussians	Runtime (h)	Storage (MB)	Peak CUDA (GB)	CD ↓	F10 ↑	MPJPE ↓	CD_h ↓
Sparse SoG	~ 2,000	0.27	0.11	2.4	0.45	98.8	11.3	7.65
Reduced	~ 60,000	1.20	3.20	5.5	0.44	99.1	11.3	7.50
Full	~ 300,000	34.1	16.0	6.2	0.44	99.1	11.3	7.50

Table 4. **Interaction-consistency metrics.** We report interaction metrics from ARCTIC [Fan et al. 2023] on HO3Dv3 and HOT3D, including Contact Deviation (CDev), Motion Deviation (MDev), hand/object acceleration error (ACC_h, ACC_o), and Mean Relative Root Position Error between the hand and object ($\text{MRRPE}_{h\to o}$). Lower is better for all metrics.

Dataset	Method	CDev [mm] ↓	MDev [mm] ↓	ACC_h [m/s^2] ↓	ACC_o [m/s^2] ↓	$\text{MRRPE}_{h\to o}$ [mm] ↓
HO3Dv3	HOLD [Fan et al. 2024]	15.8	10.8	11.3	30.2	38.6
	BIGS [On et al. 2025]	54.2	23.1	14.4	37.8	79.4
	MagicHOI [Wang et al. 2025]	533	21.5	15.4	10.3	747
	Ours	17.0	5.37	6.17	12.5	45.4
HOT3D	HOLD [Fan et al. 2024]	451	65.3	15.8	117	606
	BIGS [On et al. 2025]	570	85.6	24.1	136	762
	MagicHOI [Wang et al. 2025]	463	66.7	20.3	18.3	728
	Ours	23.8	15.7	10.5	27.4	155

10.2 Hyperparameters

We report the concrete settings used by our method for the components described in Section 4.3.

Windowed optimization. Stage 3 uses windows of $|\mathcal{W}| = 8$ frames with stride 7 (one frame overlap). Each window is optimized for 100 AdamW steps. We refine object pose and scale jointly with small hand-pose and hand-translation updates, while keeping the solution close to the SAM3D and Dyn-HaMR initialization through the priors in Eq. (2).

Optimizer settings. We use AdamW with $\beta = (0.9, 0.95)$ and weight decay 0, with per-variable learning rates: object translation 2.5×10^{-3}, object rotation 8×10^{-4}, object scale 10^{-3}, hand articulated pose 4×10^{-4}, hand translation 9×10^{-4}, hand global orientation 5×10^{-4}, and hand scale 2×10^{-4}. The object and hand shapes are fixed during Stage 3. We do not use gradient clipping; stability instead comes from value clamping after each step: the single global object scale is clamped to $[0.75,\ 2.0]\times$ its initialization and the hand scale to $[0.98,\ 1.04]\times$ its initialization, so large contact/depth corrections are absorbed by translation rather than scale. The object scale and the hand scale are single scalars shared over the whole sequence (not a per-window or per-frame parameter).

Table 5. **Contact-signal validation** against mesh-proximity ground truth. Balanced accuracy (Bal. Acc) averages the in-contact and no-contact class accuracies; the VLM flag is the stronger signal on both datasets, most clearly on the rare no-contact frames.

	HO3Dv3		HOT3D	
Contact signal	F1 ↑	Bal. Acc ↑	F1 ↑	Bal. Acc ↑
Gemini 3 (VLM, ours)	**0.99**	**0.80**	0.97	**0.66**
Mask-proximity (fallback)	0.96	0.70	**0.98**	0.59

Number of keyframes. We use $K = 4$ keyframes for object reconstruction.

Image SoG construction. The image SoG is built inside the object mask bounding box using a quadtree with maximum depth 6, color variance threshold 0.04, minimum cell size 8 pixels, bounding-box padding 2 pixels, and minimum valid-mask ratio 10^{-6}. We do not split uniform mask interiors. The isotropic image standard deviation is the geometric mean of the valid-pixel half extents in each leaf. RGB values are normalized to $[0, 1]$.

Object SoG projection. We sample 2000 object Gaussians by farthest-point sampling from the decoded SAM3D Gaussian asset. The object SoG uses isotropic 3D radii with factor $\gamma = 1.75$ as described in the method details below. For SoG matching, each projected object primitive considers the 64 nearest image SoG components. The color-kernel bandwidth is $\sigma_c = 0.15$.

Loss weights. The compact Stage 3 objective in Eq. (2) uses the following weights:

$$\lambda_{\text{sog}} = 0.05, \quad \lambda_{\text{j2d}} = 0.5, \quad \lambda_{\text{depth}} = 1000,$$
$$\lambda_{\text{sil}} = 100, \quad \lambda_{\text{contact}} = 1000, \quad \lambda_{\text{smooth}} = 100.$$

Depth and contact activation. The pointmap depth is used as a robust hand/object depth anchor. In contact frames, the object depth is additionally tied to the hand depth as in Eq. (7). The contact indicator c_t^{ho} is set from the VLM contact flag when available, and otherwise falls back to the mask-proximity test of Section 10.3; it gates the contact and depth-coupling terms. Please note that we used VLM contact flags throughout the experiments in the main paper. Mask-proximity flags are only used for contact-signal validation in Table 5.

Contact flag from the video VLM. We compute the contact flag once per sequence with `gemini-3-flash-preview` [Google 2025]. Instead of labeling frames one at a time, we pass the whole video to the model and ask when the hand is grasping the object, as a list of time intervals, and convert these intervals to per-frame flags c_t^{ho} using the sequence frame rate. We ask for intervals rather than per-frame labels because the start and end of a grasp are placed more consistently when the model reasons over the full sequence, which also removes the frame-to-frame flicker we saw with per-frame queries under occlusion. We use the following prompt, where `{subject}` is the interacting hand:

```
Analyze this video frame-by-frame and decide EXACTLY when {subject}
transitions from NOT holding (0) to HOLDING (1).

CRITICAL RULES:
1. Do NOT assume grasping at 0.0s unless the first visible frame clearly
   shows the fingers already around the object.
2. Reaching toward the object is NOT grasping.
3. Grasping begins ONLY at the FIRST frame where the fingers visibly enclose
   the object.
4. If uncertain, DELAY the start until enclosure is visually unambiguous.

Definitions:
- REACHING (0): {subject} moving toward or touching the object without
  enclosure.
- GRASPING (1): fingers of {subject} clearly wrapped around the object,
  or the object is clearly lifted while enclosed.

Step 1: Describe the timeline of states in order.
Step 2: Give the FIRST timestamp where grasping begins.
Step 3: Give when grasping ends (if it ends).

Return ONLY a JSON object in a ```json block:
{ "timeline_reasoning": string,
  "contact_intervals":
     [ {"start_sec": num, "end_sec": num} ],
  "no_contact_before_sec": num }

Also provided: duration_sec, sampled_num_frames, sampled_fps.
- Use exact timestamps; do not round aggressively.
- Do not assume holding from 0.0s unless visible.
- If grasping never occurs, return an empty list.
```

10.3 Contact Signal Validation

Our contact gate c_t^{ho} (Eq. (2)) is obtained from a video VLM (Gemini 3 [Google 2025]), which we prompt to identify per-sequence contact time intervals, which we then convert to per-frame flags. We validate this signal against ground-truth contact and against an image-space alternative for reproducibility.

Following ARCTIC [Fan et al. 2023], we label a frame as in-contact when the minimum hand-to-object vertex distance from the ground-truth meshes is below or equal to 3 mm, using each dataset's registered hand and object meshes.

As a VLM-free fallback for reproducibility, we also derive contact directly from segmentation masks: at each frame we dilate the binary hand and object masks with a 3×3 max-pooling kernel (with stride 1) and flag contact when the dilated masks overlap in image space.

Since the clips are contact-heavy (~97% of frames in contact), raw accuracy is misleading, so we report F1 and balanced accuracy (the mean of the in-contact and no-contact class accuracies), with the mask dilation chosen to maximize the baseline's balanced accuracy.

The VLM flag is the more reliable signal on both datasets (Table 5). The gap is clearest on the rare no-contact frames, which balanced accuracy captures: 0.80 vs. 0.70 on HO3Dv3 and 0.66 vs. 0.59 on HOT3D. Mask-proximity reaches a slightly higher F1 on HOT3D only because it labels almost every frame as contact. It struggles on the no-contact frames because it cannot see depth: as the hand approaches or releases the object, the two masks still overlap in the image while the surfaces are apart in depth, so it reports contact when there is none and would wrongly switch on our depth-coupling loss. The effect carries into reconstruction: swapping the VLM flag for the tuned mask-proximity flag raises CD_h from 5.02 ± 1.47 to 5.53 ± 2.32 cm on HO3Dv3 and from 9.37 ± 4.68 to 10.60 ± 6.50 cm on HOT3D. A tuned mask-proximity flag thus recovers most of the accuracy, but the VLM flag is better on the frames that matter and needs no tuning. The VLM flag can still hallucinate contact, but only on 1.2% of HO3Dv3 and 1.8% of HOT3D

frames. Exploring open-source video VLMs or dense hand-contact predictors (e.g., HACO [Jung and Lee 2025]) as reproducible contact cues is an interesting future direction.

10.4 Method Details

Keyframe Selection Details. To reconstruct a high-quality 3D object, we must select views that are both informative (reliable and stable) and diverse (covering different object angles). Following the saddle-balanced selection strategy of Depth Anything 3 (DA3) [Lin et al. 2025], we extract a global descriptor $f_t \in \mathbb{R}^d$ for each frame from the `[CLS]` token of the DA3 encoder, which uses a DINOv2 backbone [Oquab et al. 2023]. We compute the cosine similarity matrix $S_{t,s}$ between all pairs of frame descriptors and characterize each frame using three scalar statistics: the mean similarity to the remaining frames, $\bar{S}_t = \frac{1}{T-1}\sum_{s\neq t} S_{t,s}$, the feature magnitude $n_t = \|f_t\|_2$, and the feature variance $v_t = \mathrm{Var}(f_t)$ across descriptor dimensions.

Each statistic is independently min–max normalized over the sequence. We then define a balance cost that favors frames near the center of the normalized feature distribution: $C_{\text{bal}}(t) = \left|\bar{S}_t - 0.5\right| + |n_t - 0.5| + |v_t - 0.5|$. This midpoint discourages frames with extreme class tokens. In particular, unusually large feature norms or variances may reflect unstable activations in ViT features, a phenomenon also discussed in the context of register tokens [Darcet et al. 2024], while very small values tend to correspond to frames with weak or ambiguous visual signal. The balance cost therefore provides a simple criterion for selecting frames depending on their class tokens.

To avoid selecting redundant views, we combine this cost term with a diversity penalty:

$$\mathcal{V}^* = \arg\min_{\mathcal{V}\subset\mathbf{I},\,|\mathcal{V}|=K} \sum_{t\in\mathcal{V}} \left(C_{\text{bal}}(t) + \lambda_{\text{div}} \max_{s\in\mathcal{V}\setminus\{t\}} S_{t,s} \right), \tag{8}$$

where λ_{div} controls the trade-off between stability and diversity. We optimize this objective greedily: the first keyframe is chosen as the frame with the lowest cost, and each subsequent keyframe is selected by minimizing

$$C_{\text{bal}}(t) + \lambda_{\text{div}} \max_{s\in\mathcal{V}} S_{t,s}, \qquad t \notin \mathcal{V}. \tag{9}$$

This procedure selects frames whose descriptors are representative of the sequence, while avoiding views that are too similar to the keyframes already selected. Example keyframes can be seen in Fig. 3.

Flow Matching Guidance Details. In Stage 2, we modify the internal generative process of SAM3D [Chen et al. 2025] to achieve temporal stability.

- **Latent Space and Projections:** SAM3D represents object shape (Z^S) and layout (Z^P) as sets of discrete tokens. Prior to being processed by the SAM3D transformer, these tokens are mapped into a shared high-dimensional feature space via modality-specific *input projection* layers (MLPs). Let $X_t^P \in \mathbb{R}^D$ denote the layout latent state for frame t, where the dimension D is typically 1024.
- **Guidance in the Velocity Field:** SAM3D performs 3D reconstruction using *Conditional Flow Matching*. In this paradigm, the transformer does not predict the final pose directly; rather, it predicts a velocity field v_θ^P that defines the direction the current latent state X_t^P should move within the feature space to reach the target data distribution.
- **Guidance in the Velocity Field:** Our temporal guidance mechanism is applied directly within this 1024-dimensional space. Because velocity $v_\theta^P = \frac{dX_t^P}{dt}$ represents a displacement vector in the feature space, it is dimensionally equivalent to the latent state X_t^P. We compute the guided velocity $\tilde{v}_\theta^P$ as:

$$\tilde{v}_\theta^P \;=\; v_\theta^P \;-\; \lambda_{\text{temp}}\,(X_t^P - X_{t-1}^P), \tag{10}$$

  where $(X_t^P - X_{t-1}^P)$ acts as a *temporal spring* that discourages abrupt changes in orientation or translation between consecutive frames.
- **Output Projection and Token Update:** Following the guidance step, the guided velocity $\tilde{v}_\theta^P$ is passed through a modality-specific *output projection* MLP. This layer maps the 1024D velocity back into the original token space, yielding a token-space update ΔZ. This update is applied to the tokens of the previous frame:

$$Z_t^P \;=\; Z_{t-1}^P \;+\; \Delta Z. \tag{11}$$

  The updated tokens Z_t^P are then processed by the Layout Decoder to produce the physical rotation, translation, and scale[Chen et al. 2025]. By performing guidance at the latent velocity level, the model's self-attention mechanisms can resolve geometric ambiguities while adhering to a strong prior on temporal continuity.

At the end of Stage 2, there might still be some occasional orientation flips in the outputs. We perform a simple quaternion consistency check and reverse the out-of-order flips to handle them.

Image SoG. The image SoG used in Section 4.3.1 is built over the tight object-mask bounding box. Each quadtree leaf Ω_i stores only the pixels belonging to the object mask, $\mathcal{P}_i = \{u \in \Omega_i \mid M_t^o(u) = 1\}$. Its center, color, and weight are computed as

$$\mu_i = \frac{1}{|\mathcal{P}_i|}\sum_{u\in\mathcal{P}_i} u, \qquad c_i = \frac{1}{|\mathcal{P}_i|}\sum_{u\in\mathcal{P}_i} I_t(u), \qquad w_i = |\mathcal{P}_i|.$$

For an isotropic footprint, we compute the horizontal and vertical extents of valid pixels in the leaf, r_i^x and r_i^y, and set $\sigma_i = \sqrt{r_i^x r_i^y}$. The optimized image representation is therefore the compact set $\mathcal{K}_{I_t} = \{(\mu_i, \sigma_i, c_i, w_i)\}_{i=1}^{N_t}$. For visualization only, we rasterize these components back to the image plane with Gaussian-weighted color compositing.

Object SoG. The object SoG used in Section 4.3.2 is obtained by sparsifying the decoded SAM3D Gaussian asset with farthest-point sampling and converting each retained anisotropic Gaussian to an isotropic tracking primitive. Let $a_j \in \mathbb{R}^3$ denote the decoded Gaussian scale. We use the 3D radius

$$\rho_j = \gamma\,\frac{a_j^x + a_j^y + a_j^z}{3},$$

with $\gamma = 1.75$, yielding $\mathcal{G}_o = \{(x_j, \rho_j, c_j)\}_{j=1}^{N_o}$ and $N_o \leq 2000$. Under the current object pose and camera, each primitive projects to

$$\mu_{j,t}^{2D} = \Pi_t(s^o R_t^o x_j + \tau_t^o + \Delta\tau_t^o), \qquad \sigma_{j,t}^{2D} = \frac{\bar{f}_t\, s^o \rho_j}{z_{j,t}},$$

where $\bar{f}_t = (f_t^x + f_t^y)/2$ and $z_{j,t}$ is camera depth.

Color kernel for SoG. For two color vectors $c_i, c_j \in \mathbb{R}^3$, with RGB normalized to $[0, 1]$, we define

$$d(c_i, c_j) = \exp\left(-\frac{\|c_i - c_j\|_2^2}{\sigma_c^2}\right), \tag{12}$$

where σ_c controls how strongly color differences are penalized. We set $\sigma_c = 0.15$ in all experiments.

Stage 3 prior losses. This subsection gives the concrete auxiliary losses used with the SoG alignment term in Eq. (2). Since the initialization from Dyn-HaMR already provides strong hand tracking, we do not use SoG alignment for the hand. Instead, we regularize the hand trajectory with a lightweight 2D joint reprojection prior. Using the MANO joint regressor J, we obtain the 3D location of joint ℓ via $J_{t,\ell}(\Theta_t, R_t^h, \tau_t^h, s^h) \in \mathbb{R}^3$. Given detected 2D joints $\hat{u}_{t,\ell}$, the reprojection prior is

$$\mathcal{L}_{\text{j2d}} = \sum_{t \in \mathcal{W}} \sum_{\ell} \left\| \Pi\big(J_{t,\ell}(\Theta_t, R_t^h, \tau_t^h, s^h)\big) - \hat{u}_{t,\ell} \right\|_2^2. \tag{13}$$

For the depth loss, although pointmaps provide useful geometric cues, we use them only as aggregate anchors. In hand-object videos, object-side pointmaps are often noisy near occlusions or when the visible object region is small. A dense pointmap loss can then bias object pose or scale. We therefore use the compact depth term in Eq. (7): pointmap statistics anchor metric depth, while contact frames tie the object depth to the hand depth.

For the contact loss, let C_t be the selected MANO contact-zone vertices in frame t, and let $\mathcal{V}(H_t)$ and $\mathcal{V}(O_t)$ denote the vertices of the posed hand and object surfaces. The contact attraction term is

$$\mathcal{L}_{\text{near}} = \sum_{t \in \mathcal{W}} c_t^{ho} \frac{1}{|C_t|} \sum_{v \in C_t} \min_{y \in O_t} \|v - y\|_2^2.$$

To penalize interpenetration, let $\phi_{O_t}(x)$ and $\phi_{H_t}(x)$ be signed distances to the object and hand surfaces, positive outside and negative inside. We use

$$\mathcal{L}_{\text{pen}} = \sum_{t \in \mathcal{W}} c_t^{ho} \Bigg(\frac{1}{|\mathcal{V}(H_t)|} \sum_{v \in \mathcal{V}(H_t)} \max(0, -\phi_{O_t}(v))^2 + \frac{1}{|\mathcal{V}(O_t)|} \sum_{y \in \mathcal{V}(O_t)} \max(0, -\phi_{H_t}(y))^2 \Bigg).$$

The full contact loss is $\mathcal{L}_{\text{contact}} = \mathcal{L}_{\text{near}} + \lambda_{\text{pen}} \mathcal{L}_{\text{pen}}$ where $\lambda_{\text{pen}} = 5$. This combination pulls the active fingertip/contact region toward the object surface while discouraging either surface from passing through the other.

For temporal stability, we penalize hand/object translation acceleration and object angular acceleration:

$$\mathcal{L}_{\text{smooth}} = \sum_{t \in \mathcal{W}} \left(\|\Delta^2 \tau_t^h\|_2^2 + \|\Delta^2 \tau_t^o\|_2^2 + \lambda_R \|\Delta^2 \omega_t^o\|_2^2 \right),$$

where ω_t^o denotes the rotational velocity induced by consecutive object quaternions.

10.5 Isotropic vs. Anisotropic Object Gaussian Projection

In this section, we detail the projection approximation and the anisotropic projection ablation with a first-order perspective Jacobian. Zwicker et al. [2001]'s volume derivation maps camera space to three-dimensional ray space with a 3×3 Jacobian and obtains the 2D footprint covariance by dropping the depth row and column. Their surface derivation equivalently uses the direct 2×3 Jacobian of the camera-to-image map (Eq. 34 in [Zwicker et al. 2002]). Our implementation uses this direct image-space form. Let object Gaussian j have camera-space mean $\mu_j^c = (x, y, z)^T$ and isotropic world-space covariance $\sigma_j^2 \mathbf{I}_3$. The pixel projection is

$$\mu_j = \boldsymbol{\pi}(\mu_j^c) = \begin{bmatrix} f_x x/z + c_x \\ f_y y/z + c_y \end{bmatrix}. \tag{14}$$

Its image-space Jacobian is

$$\mathbf{A} = \frac{\partial \boldsymbol{\pi}}{\partial \mu_j^c} = \begin{bmatrix} f_x/z & 0 & -f_x x/z^2 \\ 0 & f_y/z & -f_y y/z^2 \end{bmatrix} \in \mathbb{R}^{2\times 3}. \tag{15}$$

Note that $\mathbf{A}$ is the derivative of the 2D pixel projection, not the 3×3 ray-space Jacobian of the volume formulation. Let $\mathbf{R}_{wc}$ denote the rotational part of the world-to-camera transform. The covariance implemented by the anisotropic branch is

$$\begin{aligned} \Sigma_j &= (\mathbf{A}\mathbf{R}_{wc})(\sigma_j^2 \mathbf{I}_3)(\mathbf{A}\mathbf{R}_{wc})^T \\ &= \sigma_j^2 \mathbf{A}\mathbf{A}^T, \end{aligned} \tag{16}$$

where the second equality follows because $\mathbf{R}_{wc}$ is orthonormal. This covariance can be written as

$$\Sigma_j = \sigma_j^2 \begin{bmatrix} \frac{f_x^2}{z^2}\left(1 + \frac{x^2}{z^2}\right) & \frac{f_x f_y x y}{z^4} \\ \frac{f_x f_y x y}{z^4} & \frac{f_y^2}{z^2}\left(1 + \frac{y^2}{z^2}\right) \end{bmatrix}. \tag{17}$$

Thus, perspective distortion produces an elliptical footprint in general. Our isotropic approximation replaces this covariance by a scalar screen-space variance. The anisotropic ablation uses Σ_j for projected model Gaussian j, while image Gaussian i remains isotropic with $\Sigma_i = \sigma_i^2 \mathbf{I}_2$. Following the anisotropic Gaussian overlap of Sridhar et al. [2014], the continuous spatial overlap of the unnormalized Gaussians is

$$\int_\Omega \mathcal{B}_i(u)\mathcal{B}_j(u)\, du = 2\pi \frac{\sqrt{|\Sigma_i|}\sqrt{|\Sigma_j|}}{\sqrt{|\Sigma_i + \Sigma_j|}} \cdot \mathrm{e}^{-\frac{1}{2}(\mu_i - \mu_j)^T (\Sigma_i + \Sigma_j)^{-1} (\mu_i - \mu_j)}. \tag{18}$$

Multiplying this spatial overlap by the same color kernel $d(c_i, c_j)$ gives E_{ij} as defined in Eq. (4). For isotropic model and image covariances, $\Sigma_j = \sigma_j^2 \mathbf{I}_2$ and $\Sigma_i = \sigma_i^2 \mathbf{I}_2$, Eq. (18) reduces exactly to the scalar expression in Eq. (5). This scalar form avoids per-pair 2×2 determinant and inverse evaluations, providing a smoother image alignment signal.

We repeat our optimization stage (Stage 3) using the described anisotropic projection and general overlap, while all other parameters remain unchanged. Results are reported in Table 6. Under this setting, Jacobian-based anisotropic projection results in a slightly

Table 6. **Gaussian-projection ablation.** Comparing our scalar closed-form overlap against full 2D anisotropic projection. Modeling Jacobian-based anisotropic projection with matrix overlap does not improve tracking accuracy (CD_h) and adds ∼11% runtime.

Projection	Overlap	HO3Dv3 CD_h[cm] ↓	HOT3D CD_h[cm] ↓	Runtime [h] ↓
Isotropic (ours)	Scalar closed-form	**5.02 ± 1.47**	**9.37 ± 4.68**	**0.27**
Anisotropic	Full matrix (2×2)	5.29 ± 2.04	10.43 ± 6.39	0.30

higher CD_h (from 5.02 to 5.29 on HO3Dv3, from 9.37 to 10.43 on HOT3D) and is ∼11% slower.

One possible explanation for the lack of improvement in the anisotropic variant's performance is that the spatial arrangement of 2000 primitives already captures much of the object-level shape needed for pose refinement (including elongated object structures such as scissors and screwdrivers). In this setting, modeling a more accurate elliptical footprint provides limited additional information. The general matrix overlap may also introduce a less smooth alignment landscape than the scalar overlap. A hybrid formulation of isotropy and anisotropy for boundaries or elongated object regions, and representing image SoG also as anisotropic Gaussians remains as future directions.

10.6 Evaluated Sequences

We evaluate our method on sequences from HO3Dv3 [Hampali et al. 2021] and HOT3D [Banerjee et al. 2025]. Specifically, we show which sequences and which timestamps used in Table 7 and Table 8.

Table 7. HO3Dv3 sequences for hand-object reconstruction. We directly use the same sequences from HOLD [Fan et al. 2024]. All of these sequences are only right-handed.

Object	**Sequence name**
bleach	ABF12
bleach	ABF14
potted meat	GPMF12
potted meat	GPMF14
cracker box	MC1
cracker box	MC4
power drill	MDF12
power drill	MDF14
sugar box	ShSu10
sugar box	ShSu12
mustard	SM2
mustard	SM4
mug	SMu1
mug	SMu40
banana	BB12
banana	BB13
scissors	GSF12
scissors	GSF13

10.7 Comparison Details

For HOLD [Fan et al. 2024], we employ the authors' publicly available pre-trained CVPR models for HO3D and use the available code to train for HOT3D. For BIGS [On et al. 2025], we use the official implementation from the project repository and optimize each stage using the hyperparameters specified in their Supplementary Material (Sec. A). To get meshes out of their optimized Gaussians, we employ Poisson Reconstruction with depth 6 after we remove the outliers within 100 neighbors with 2.0 standard deviation. For MagicHOI [Wang et al. 2025], we follow the official implementation from the project repository.

10.8 Detailed Results Discussion

GraG differs fundamentally from recent hand-object reconstruction methods such as HOLD, BIGS, and MagicHOI, which rely on COLMAP/SfM-based initialization followed by heavy per-scene optimization. These methods depend on stable multi-view correspondences to estimate camera poses and coarse geometry before performing joint reconstruction. However, such initialization is often unreliable in hand-object interaction videos due to severe hand occlusions, rapid camera motion, and small object projections, which are particularly common in datasets such as HOT3D. As a result, errors in the early SfM stage can propagate to later optimization stages, leading to degraded reconstruction quality and lower success rates.

MagicHOI further combines SfM initialization with generative priors and object inpainting to complete unseen regions. While effective under carefully selected inputs, achieving strong results typically requires manual selection of reconstruction frames and inpainting viewpoints. When the full video sequence is provided directly without such curation, the optimization becomes significantly more challenging and the reconstruction quality degrades, especially for longer sequences such as those in HO3Dv3 and HOT3D. In contrast, GraG is designed to operate directly on full video sequences without manual frame selection. Our method first identifies informative keyframes and reconstructs a canonical object representation using a foundation model, while adapting SAM3D-based object pose prediction to video sequences. The canonical object is then tracked across the sequence using a compact SoG representation with temporal and contact-aware constraints. By decoupling canonical shape reconstruction from per-frame motion estimation, GraG avoids repeated large-scale scene optimization and instead performs efficient low-dimensional tracking.

This design yields several advantages. First, GraG is substantially more robust to occlusion and unstable viewpoints, which explains its stronger performance on challenging datasets such as HOT3D where SfM-based methods often struggle. Second, reconstructing a canonical object geometry from informative keyframes preserves fine structures, making the approach more reliable for thin or partially observed objects that are difficult to recover through correspondence-based initialization. Finally, replacing heavy test-time optimization with lightweight tracking over a fixed canonical asset enables significantly faster reconstruction while maintaining higher accuracy on both HO3Dv3 and HOT3D.

Table 8. HOT3D clips used in our evaluations, reported with sequence identifiers, hand side, temporal boundaries, and object IDs. Temporal boundaries are reported as HOT3D timecode timestamps (TS) in nanoseconds.

Sequence name	**Hand side**	**Start TS** (ns)	**End TS** (ns)	**Object ID**
P0001_23fa0ee8	right	55003474858015	55015408224119	238686662724712
P0001_8d136980	right	54039408208193	54044341521984	238686662724712
P0001_b2bcbe28	left	54811408205972	54814874891157	258906041248094
P0001_f71fc9b1	right	66482274870136	66485208197200	249541253457812
P0010_8ff3e5c4	left	58695141597662	58697074869546	258906041248094
P0011_0c2d00ed	right	55563708197241	55572041548328	70709727230291
P0011_0c2d00ed	right	55660041543187	55661741528834	204462113746498
P0011_2255f410	right	56349641532049	56356641531612	249541253457812
P0011_ccc678c7	right	55922341542825	55926474871989	261746112525368
P0011_cd22f5e0	right	55211841541663	55215108194835	228358276546933
P0011_ff7cfc3a	right	55460874890915	55465808213471	225397651484143
P0011_ff7cfc3a	right	55518174857985	55522441524737	270231216246839
P0012_0a21e4c2	right	47121508235754	47124108213732	106957734975303
P0012_0a21e4c2	right	47145441536211	47157641593381	258906041248094
P0012_44c5f677	left	59321408185000	59324774850689	194930206998778
P0012_44c5f677	right	59420774854423	59423208211436	106434519822892
P0014_84ea2dcc	right	44789441528790	44791608213887	98604936546412
P0015_60573a3b	left	60819408201515	60825874885641	79582884925181

10.9 Why SoG Instead of Rendering or Optical Flow?

The dense MV-SAM3D Gaussian asset remains important: it defines the canonical geometry and appearance from which the compact object SoG is sampled. However, GraG does not optimize the full photorealistic asset during tracking. Stage 3 repeatedly re-poses a small set of Gaussian primitives and maximizes their closed-form overlap with the image SoG, using the hand mask to gate occluded support. This gives a region-based alignment signal with spatial extent, so the tracker still receives useful gradients under small pose errors, low texture, and partial occlusion.

This is different from three natural alternatives. First, mesh rendering and dense Gaussian RGB losses depend on view-consistent generated texture. As shown in Fig. 7, the generated asset often has useful geometry but unreliable front/back or side-specific texture because the multi-view reconstruction is not explicitly constrained by calibrated cameras. Photometric render-and-compare objectives can therefore optimize toward texture artifacts rather than pose. Second, sparse optical flow loses tracks under hand occlusion, fast rotations, and large appearance changes. Note that for the optical-flow baseline, we project object surface points using the current pose, track their 2D locations with iterative Lucas–Kanade[Lucas and Kanade 1981] flow, and optimize the rigid object pose so that the projected 3D points follow the tracked 2D positions. Table 2 reports the mesh/Gaussian rendering and optical-flow variants, which we outperform with GraG's SoG formulation.

10.10 Ablations

We conduct ablations on four sequences: ABF14 and GPMF12 from HO3Dv3, P0011_0c2d00ed and P0011_2255f410 from HOT3D. Unless stated otherwise, we keep the same training schedule and hyper-parameters as the full model (including all Stage 3 loss weights and the same number of optimization steps per window). Table 2 summarizes the effect of key components: (i) keyframe selection for canonical object reconstruction (Stage 1), (ii) video-level pose tracking with frozen shape and temporal guidance (Stage 2), and (iii) our SoG-based refinement and prior losses (Stage 3).

Attributing gains beyond foundation models. The "only foundation models" row keeps the SAM3D object reconstruction and DynHaMR hand motion, but removes Stage 3 refinement. This variant has the same reconstructed object quality, but much worse hand-object alignment (CD_h=20.62), showing that the final tracking stage is necessary beyond the pretrained priors. The "optimization w/o SoG refinement" row keeps the same Stage 3 optimization and the same geometric/contact priors, but removes the SoG image-alignment term. Its improvement over "only foundation models" shows that the priors correct a large part of the hand-object depth error, while the remaining gap to the full model shows the contribution of SoG tracking.

Full-benchmark SoG ablation. Table 2 ablates SoG on four sequences; here we repeat the priors-only vs. priors+SoG comparison on the full benchmark to show the effect of the SoG alignment term. These averages are over all 18 sequences from HO3Dv3 and 18 sequences from HOT3D, so they are not directly comparable to the four-sequence numbers in Table 2. Adding SoG lowers CD_h from 7.66 ± 4.93 to 5.02 ± 1.47 cm on HO3Dv3 and from 12.11 ± 6.39 to 9.37 ± 4.68 cm on HOT3D, and tightens the spread across sequences on both. The variance is larger on HOT3D, as expected from its stronger camera motion and heavier occlusion. As in the four-sequence study, the priors carry the hand-relative accuracy while SoG enables efficient tracking.

Alternative tracking objectives. The mesh, dense Gaussian, and optical-flow rows isolate whether the SoG objective itself is needed. Mesh and dense Gaussian rendering use the same generated object asset in a direct render-and-compare objective, while optical flow uses Lucas–Kanade tracks of projected object surface points. All three alternatives increase CD_h relative to full GraG. They are also

slower than GraG, which is consistent with our goal of using the Gaussian asset as a compact tracking template rather than as a dense renderer.

Loss terms. Removing contact produces the largest degradation in CD_h, confirming that contact is central for resolving hand-object relative depth under occlusion. Depth, silhouette, and visibility gating provide smaller but consistent gains by anchoring metric scale, aligning object masks, and preventing occluded object Gaussians from matching hand pixels.

10.11 MagicHOI Result Discrepancy

We observe a notable variance between the performance of MagicHOI in our experiments compared to the results reported in their original publication. This discrepancy is primarily attributed to differences in input sequence length and domain robustness. While the original MagicHOI study evaluates models on short, curated clips (2–3 seconds), our evaluation utilizes the original HO3D sequence lengths (20–30 seconds). Also, MagicHOI appears sensitive to temporal scale and viewpoint; it exhibits significantly lower performance when applied to longer, uncurated sequences and egocentric perspectives.

10.12 Additional Limitation Details

Our method uses Depth Anything 3 pointmaps as aggregate metric-depth anchors. Our depth and contact formulation reduces the effect of moderate object-depth errors: the object is not anchored only to its own predicted depth, but is also coupled to the hand depth in frames with contact. This improves hand-object alignment in cases where the object depth alone would place the object too far from the hand. However, severe pointmap failures can still affect the reconstruction. When the pointmap gives inconsistent metric depths for both the hand and object, or when masks/contact cues are unreliable, the available cues may agree on an incorrect metric scale or translation. In such cases, refinement can still produce an inaccurate hand-object relative pose.

Fig. 13. **More qualitative results.** We overlay rendered hand-object meshes on the input images for qualitative inspection and also show side views of the interactions.