RETHINKING PERCEPTUAL REPRESENTATIONS FOR 6-DOF GRASPING WITH STEREO SPIKE STREAMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Most robotic grasping pipelines first convert sensory observations into explicit 3D representations such as point clouds, which is a computational step not found in biological intelligence. While effective in many standard settings, this reconstruct-then-act paradigm can become a bottleneck under asynchronous sensing, where high dynamic range and strict latency constraints make intermediate reconstruction costly and potential losses for downstream control. This paper explores a fundamentally different, neuro-inspired paradigm for 6-DoF grasp detection. We introduce SpikeGrasp, a framework that mimics the feedforward visuomotor pathway, processing raw, asynchronous events from stereo spike cameras, similarly to retinas, to directly infer grasp poses. Our model fuses these stereo spike streams and uses a recurrent spiking neural network, analogous to high-level visual processing, to iteratively refine grasp hypotheses without ever reconstructing the point cloud. To validate the approach, we construct a benchmark spanning clutter, textureless objects, severe illumination changes, and occlusion, and compare direct inference against reconstruct then grasp baselines. Across synthetic and real high-dynamic-range scenes, SpikeGrasp achieves stronger accuracy, data efficiency, and better robustness under challenging sensing conditions. Our results suggest that explicit point-cloud reconstruction is not universally necessary for embodied manipulation, and that direct learning in the asynchronous observation space is a promising alternative for low-latency robot perception and control, particularly in complex real environments.

1 INTRODUCTION

Robotic grasping is a fundamental capability for embodied intelligence. Most existing 6-DoF grasp detection systems follow a geometry-first pipeline: sensory observations are first converted into explicit 3D representations, such as depth or point clouds, and grasp poses are then predicted from the reconstructed geometry (Fang et al., 2020; 2023; Gui et al., 2025). This paradigm is highly effective in many robotic manipulation settings because point clouds provide explicit spatial structure and are convenient for geometric reasoning. However, it also makes grasp prediction dependent on the quality of the intermediate reconstruction. When depth estimation or point-cloud generation becomes unreliable due to sensor noise, weak texture, reflective surfaces, or challenging illumination, reconstruction errors can propagate directly to downstream grasp planning (Mahler et al., 2017; Deng et al., 2020). This raises a broader question for embodied perception: is explicit 3D reconstruction always the most appropriate observation space for grasping?

This question becomes important under asynchronous sensing. Spike cameras provide a neuromorphic visual modality that records binary spike streams with high temporal resolution and high dynamic range (Zheng et al., 2026b). Unlike frame-based cameras, they naturally encode temporal changes through continuous spike activity, making them well suited for low-latency and visually challenging scenarios. A common way to use such data is to first reconstruct images, depth, or point clouds, and then apply conventional perception or grasping methods. While straightforward, this reconstruct-then-act strategy may discard task-relevant temporal information that are useful for manipulation. In contrast, biological visuomotor systems apparently do not rely on explicit point-cloud representations before acting. Instead, they transform continuous sensory streams into action-relevant signals through recurrent neural processing (Cao et al., 2025). Motivated by this view, we study whether 6-DoF grasp

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

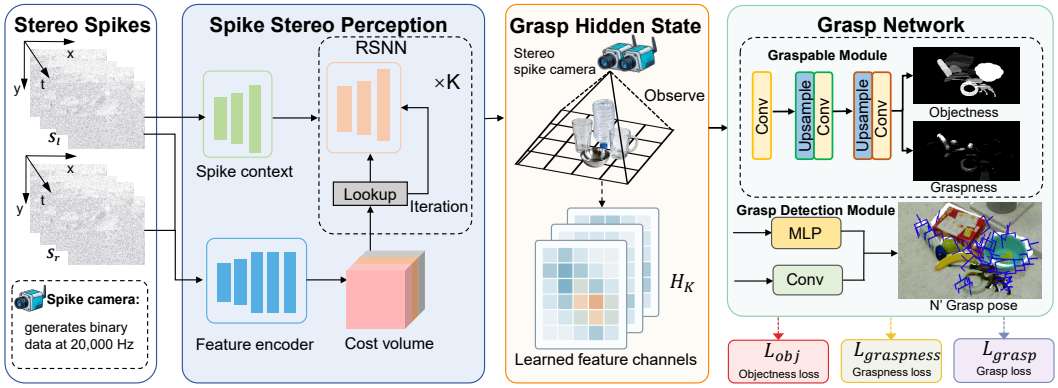


Figure 1: Overview of SpikeGrasp, the method for direct 6-DoF grasp pose detection from raw stereo spike streams. Stereo spike inputs are encoded by a shared-weight feature encoder to construct a disparity correlation volume. Meanwhile, contextual features extracted from the left spike input are used to initialize and guide the subsequent recurrent update process. After K recurrent updates, the learned feature tensor h^k is passed to the grasp decoder. The decoder predicts objectness and graspness and produces metric grasp translations, gripper orientations, widths, confidence scores, and predicts the final top-N grasp poses.

poses can be inferred directly from raw stereo spike streams without reconstructing intermediate geometric representations.

We introduce SpikeGrasp, a direct spike-to-grasp framework for 6-DoF grasp detection from stereo spike streams. Rather than treating spikes as an intermediate signal to be reconstructed into frames, or point clouds, SpikeGrasp operates directly in the asynchronous observation space. The stereo spike streams are first encoded into spike-native spatiotemporal features, from which cross-view correlation cues are constructed for binocular reasoning. A recurrent spiking module then iteratively updates a latent state that captures grasp-relevant scene structure and affordance cues. Finally, task-specific prediction heads decode this latent state into objectness, graspness, grasp scores, gripper widths, and 6-DoF grasp poses. This design is motivated by the efficiency of biological visuomotor processing, but its main objective is practical and testable: to preserve temporal and binocular information in raw spike streams while avoiding explicit geometric reconstruction during inference.

To evaluate this direction systematically, we construct SpikeGrasp-1B, a large-scale synthetic benchmark for 6-DoF grasp detection from stereo spike streams. The benchmark contains diverse cluttered tabletop scenes, high-dynamic-range visual conditions, and grasp annotations derived from large-scale object-level grasp labels. Experiments show that SpikeGrasp achieves stronger performance on the synthetic benchmark and demonstrates favorable generalization, especially under limited training data and challenging scene conditions. We further validate the method in real-world robot grasping experiments, where SpikeGrasp shows advantages over reconstruction-based baselines in cluttered and high-dynamic-range scenes. Our results suggest that explicit 3D reconstruction is not the only viable route to embodied manipulation. In asynchronous sensing regimes, direct learning in the spike observation space can provide an effective alternative to reconstruct then grasp pipelines. The main contributions of this paper are summarized as follows:

- We formulate 6-DoF grasp pose detection from stereo spike streams as a direct asynchronous observation-space learning problem, challenging the conventional reconstruct then grasp pipeline based on point clouds.
- We propose SpikeGrasp, an end-to-end spike-to-grasp framework that combines spike-native stereo encoding, to infer grasp relevant structure directly from raw stereo spike streams.
- We introduce SpikeGrasp-1B, a large-scale synthetic benchmark for spike-based 6-DoF grasp detection, enabling systematic evaluation under diverse objects, texture sparsity, clutter, and illumination conditions.
- We show through experiments that SpikeGrasp outperforms competing methods and provides stronger accuracy-robustness trade-offs in challenging high-dynamic-range scenes.

2 Related Work

Reconstruction-based Grasping. Most existing 6-DoF grasp detection methods rely on explicit geometric observations, typically depth maps or point clouds, before grasp reasoning is performed (ten Pas et al., 2017; Liang et al., 2019; Breyer et al., 2021; Qin et al., 2020; Ni et al., 2020). Early sampling and evaluation pipelines, such as GPD (ten Pas et al., 2017) and PointNetGPD (Liang et al., 2019), generate grasp candidates from geometric inputs and rank them with learned quality models. More recent methods directly regress grasp poses from explicit 3D observations using proposal-based, volumetric, point-based, or RGB-D fusion pipelines (Gou et al., 2021; Sundermeyer et al., 2021). Graspness guided approaches and context aware formulations further improve efficiency and robustness in cluttered scenes (Wang et al., 2021; Hoang et al., 2022; Liu et al., 2023; Chen et al., 2023; Qin et al., 2023). As a result, explicit 3D geometry has become the dominant observation space for robotic grasping (Iwase et al., 2025). However, effectiveness of this paradigm depends heavily on the quality of the intermediate reconstruction (Mahler et al., 2017; Deng et al., 2020). Under noise, low texture, fast motion, or difficult illumination, reconstruction errors can propagate to the final grasp estimate (Fang et al., 2023; Mahler et al., 2017; Deng et al., 2020). This limitation is relevant in asynchronous sensing settings, where forcing event-like observations into conventional geometric intermediates may discard task-relevant temporal information (Huo et al., 2024; Doshi et al., 2025).

Direct Perception to Grasping. An alternative line of work avoids explicit point clouds and predicts grasps directly from images, multi-view observations, or implicit scene representations (Jauhri et al., 2024; Dai et al., 2023). While early image-based methods are often efficient, many are limited to planar grasps action spaces. More recent approaches recover grasp-relevant structure through multi-view reasoning, neural rendering, or implicit 3D representations without constructing point clouds. These results suggest that accurate grasping does not always require explicit geometry, and that alternative observation spaces may be preferable when depth is ambiguous or expensive to reconstruct. This perspective is related to direct perception-to-action learning, where models infer task-relevant actions from sensory input without relying on designed intermediate representations (James et al., 2022; He et al., 2024; Zhou et al., 2025). In latency-sensitive embodied settings, this is advantageous because the representation best suited for reconstruction may not be the most informative one for action (Ma et al., 2024). In contrast to reconstruction-free grasping methods, which still depend on frame-based observations (Kim et al., 2016; Liu et al., 2025), our work studies direct 6-DoF grasp inference from stereo spike streams as a native asynchronous observation space.

Spike-based Perception for Embodied Vision. Neuromorphic visual sensing has enabled a wide range of perception tasks based on event or spike streams. For spike cameras, many works focus on reconstructing intensity images from asynchronous spikes using filtering, neuron-inspired models, optimization, and deep learning (Dong et al., 2022; Zhao et al., 2021). Spike streams have also been used for super-resolution, HDR imaging, and high-speed video interpolation, demonstrating advantages in extreme lighting and fast-motion scenes (Zhao et al., 2023; Zhou et al., 2020; Xia et al., 2023; Dong et al., 2024). These studies establish spike cameras as a promising sensing modality for conditions in which conventional frame-based imaging suffers from motion blur, saturation, or temporal under-sampling. Beyond reconstruction, spike-based representations have been applied to object detection, optical flow, and depth estimation (Zheng et al., 2026a; Zhao et al., 2026; Gao et al., 2026). This progress shows that spike streams can support not only low-level reconstruction, but also structured scene understanding and geometry related prediction. Despite this progress, spike-based embodied manipulation remains relatively underexplored, and existing neuromorphic works mainly target reconstruction or generic perception (Li et al., 2020). Moreover, standardized benchmarks and specific task architectures for grasping from stereo spike streams are limited.

3 Method

We introduce our end-to-end 6-DoF pose detection network based on stereo spike streams, which are illustrated in the Fig. 1. The following sections introduce the detailed components of SpikeGrasp.

3.1 Spike Stereo Encoding

A spike camera follows an integrate-and-fire sensing mechanism. For a pixel at location (x, y), incident irradiance is accumulated over time until a threshold is reached, at which point a binary

spike is emitted and the accumulator is reset. The accumulated charge can be written as

$$A(x, y, t) = \Big(\int_0^t I(x, y, s)\, ds \Big) \bmod \theta, \tag{1}$$

where $I(x, y, s)$ is the incident irradiance and θ is the firing threshold. Capturing a scene over a time interval of length T produces a binary spike tensor $S \in \{0, 1\}^{N \times H \times W}$, whose event density encodes the underlying luminance dynamics. Given left and right spike streams S^l and S^r, we first extract multi-scale feature maps using a shared convolutional backbone. A 7×7 convolution with stride 2 produces half-resolution features, followed by residual blocks that refine local spatiotemporal responses. A second downsampling stage yields quarter-resolution feature maps $f_{l,4}$ and $f_{r,4}$, which are used to build the stereo matching representation. In addition, we derive multi-scale features

$$f_{l,i}, f_{r,i} \in \mathbb{R}^{C_i \times H_i \times W_i} \mid i \in \{4, 8, 16\}, \tag{2}$$

which support subsequent coarse-to-fine refinement. This spike-native encoder converts sparse spike observations into a compact latent representation while retaining the temporal sensitivity of signals.

3.2 Cross-view Stereo Correlation

The correlation volume compares stereo features extracted from the input spike window. Eq. (3) correlates left and right feature positions along rectified scanlines and has no separate temporal-lag axis. Temporal information, to the extent retained by the encoder, is carried by its input features. For all-pairs horizontal matching, we write

$$C(y, x_l, x_r) = \sum_c F^l(c, y, x_l) F^r(c, y, x_r), \tag{3}$$

and the resulting tensor $C \in \mathbb{R}^{D \times H \times W}$ forms the initial cost volume. To capture context at multiple resolutions, we build a four-level correlation pyramid $\{C^{(1)}, \ldots, C^{(4)}\}$ by successively applying 1D average pooling along the dimension with kernel size 2 and stride 2. Each level halves the resolution while preserving spatial dimensions, enabling coarse to fine matching in subsequent stages.

3.3 Recurrent Grasp Hypothesis Refinement

We maintain a hidden state field h^k that stores the current estimate of scene structure and grasp-relevant affordance cues. At iteration $k = 0, ..., K-1$, correlation features are sampled from the multi-scale pyramid according to the current state and fused with spike-context features. These inputs are then processed by a three-layer recurrent spiking neural network (RSNN) update operator, which predicts an increment Δh and updates the hidden state as

$$h^{k+1} = h^k + \Delta h. \tag{4}$$

The update module shares parameters across iterations and exchanges information across scales, allowing the model to progressively refine both visible and occluded structure. This iterative design enlarges the receptive field while preserving fine detail at high resolution. The recurrent module produces output spike trains that are decoded into real-valued hidden representations through membrane-potential accumulation. Specifically, if s_t denotes the output spike at time step t, the output membrane potential is computed as

$$v_k^{\text{out}} = \tau_{\text{out}} v_{k-1}^{\text{out}} + s_k, \qquad v_K^{\text{out}} = \tau_{\text{out}}^K v_0^{\text{out}} + \sum_{k=1}^{K} \tau_{\text{out}}^{K-k} s_k. \tag{5}$$

where τ_{out} is a decay factor. We use the final membrane potential to obtain the decoded hidden state as $h_K = \text{ReLU}(v_K^{\text{out}})$. This decoding strategy turns sparse recurrent spike activities into a dense feature field for subsequent grasp prediction.

3.4 Grasp Representation and Prediction Heads

Each grasp is represented in the camera frame as $\text{G} = (\mathbf{R}, \mathbf{t}, w)$, where $\mathbf{R} \in \mathbb{R}^{3\times3}$ is the gripper orientation, $\mathbf{t} \in \mathbb{R}^3$ is the translation, and $w \in \mathbb{R}$ is the gripper width. Rather than directly regressing

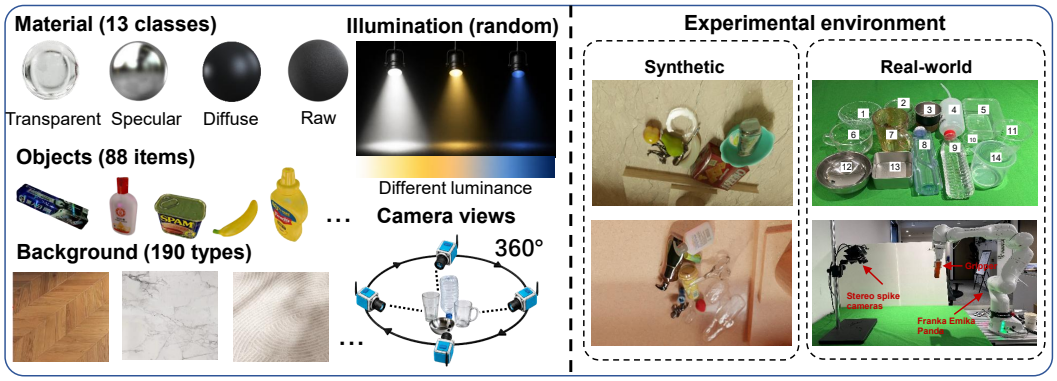


Figure 2: Overview of the SpikeGrasp-1B dataset composition. For each scene, we randomize the background, material, object and illumination intensity to increase diversity. The material of each object is randomly assigned from transparent, metallic, diffuse, or original appearances. In addition, each scene is captured from 360-degree surrounding viewpoints to generate diverse visual observations. Right part shows robot and object settings in the real-world grasping experiments.

the full rotation matrix, we adopt a 6D pose parameterization that decomposes orientation into a discrete viewpoint class and a continuous in-plane rotation. For each candidate location, the model predicts a feasible approach direction, an approach depth, an in-plane rotation angle, and a tight gripper width. This representation is easier to optimize and yields mechanically valid grasp poses.

From the final hidden state h^K, we first predict a dense graspability map using an encoder-decoder head. This head outputs a two-channel probability map $M \in \mathbb{R}^{2\times H\times W}$, where the first channel denotes objectness, i.e., whether a pixel belongs to a foreground object, and the second channel denotes graspness, i.e., how suitable that location is for grasp contact. Skip connections and multi-scale upsampling preserve fine spatial detail and help localize graspable regions in cluttered scenes.

The grasp decoder maps h^K and the predicted graspability outputs to a set of candidates $\{(R_i, t_i, w_i, q_i)\}$. Here, R_i and t_i specify the gripper pose in the left-camera frame, w_i is the gripper width, and q_i is the predicted grasp score. Candidate selection and pose decoding operate on the latent representation. No reconstructed depth map or point cloud is supplied to the decoder. The candidates are ranked using their predicted scores, followed by the stated grasp post-processing procedure.

3.5 Training Objectives

We quantify grasp quality by the minimum friction coefficient u_i required to guarantee force closure. A lower friction coefficient corresponds to a more stable grasp. Following the standard normalization used in graspness-based formulations, we convert u_i into a confidence score $q_i \in [0, 1]$,

$$q_i = \begin{cases} \dfrac{\ln(u_{\max}/u_i)}{\ln(u_{\max}/u_{\min})}, & \text{if the grasp is positive,} \\ 0, & \text{others,} \end{cases} \tag{6}$$

where $u_{\min}$ and $u_{\max}$ are empirical lower and upper bounds. This mapping assigns higher confidence to more stable grasps while suppressing infeasible candidates. The network uses a multi-task loss:

$$L = L_{obj} + \alpha\big(L_p + \gamma\, L_v\big) + \beta\big(L_s + L_w\big) + \lambda L_r, \tag{7}$$

where L_{obj} supervises the objectness map, L_p and L_v model point-wise and view-wise graspability, L_s and L_w are the grasp scores and gripper widths, respectively. The terms L_p and L_s are evaluated on object pixels, L_v is applied on predefined seed views, and L_w is computed for grasp hypotheses with positive ground-truth scores. Classification terms use softmax losses, whereas regression tasks use smooth-l_1 losses. L_r is the regularization term for RSNN. This objective jointly encourages accurate scene parsing, graspable region discovery, and precise 6-DoF grasp prediction. The loss terms supervise grasp prediction and are optimized jointly with the recurrent perception backbone.

4 EXPERIMENTS

Benchmark and Data Generation. We construct a synthetic benchmark SpikeGrasp-1B for spike-based 6-DoF grasp detection in Blender, witch contains over 1 billion grasp poses. The dataset contains 190 scenes in total, including 100 training scenes and 90 test scenes. To evaluate generalization under different levels of object overlap, the test set is further divided into three subsets: 30 scenes with seen objects, 30 scenes with unseen but similar objects, and 30 scenes with novel objects. Each scene provides 256 pairs of stereo spike streams together with scene-level supervision. The benchmark is built from 88 object models from GraspNet (Fang et al., 2020). We use physics-based simulation and extensive domain randomization over lighting, pose, camera viewpoint, and material properties to generate diverse HDR scenes and corresponding spike observations. This design aims to expose the model to large appearance and geometry variations while maintaining consistent grasp supervision. The overview of the synthetic dataset is shown in the Fig 2. Additional dataset statistics, the generation process, and further implementation details are provided in Appendix A.2.

Implementation Details. Our model is implemented with PyTorch and trained on Nvidia RTX 4090 GPUs for 18 epochs with Adam optimizer and batch size of 4. The learning rate is 2×10^{-4} in the first epoch and multiplied by 0.95 every epoch. During training, 16 update iterations for each sample.

4.1 MAIN RESULTS

We compare our method against representative grasp detection methods on the SpikeGrasp-1B benchmark. The compared methods include point-cloud-based: GraspNet (Fang et al., 2020), GraNet (Wang et al., 2023), GraspFast (Gui et al., 2025), Scale-Balanced-Grasp (Ma & Huang, 2023), GSNet (Wang et al., 2021), FineGrasp (Du et al., 2026) and EconomicGrasp (Wu et al., 2024). The RGB-D image-based: CenterGrasp (Chisari et al., 2024), HGGD (Chen et al., 2023) and ZeroGrasp (Iwase et al., 2025). The reconstruction-free or multi-view methods: RGB_matters (Gou et al., 2021), GraspNeRF (Dai et al., 2023) and ASGrasp (Shi et al., 2024). For fair comparison, all methods are trained on the same training split of SpikeGrasp-1B and evaluated on the same Seen, Similar, and Novel test subsets. Since most existing baselines are designed for conventional visual inputs, they still take reconstructed depth maps or multiple RGB images as input when adapted to our benchmark, whereas SpikeGrasp directly consumes raw stereo spike streams.

Grasp Pose Prediction. We report the standard grasp detection metrics AP, $AP_{0.8}$, and $AP_{0.4}$. As shown in Table 1, SpikeGrasp achieves the best AP, $AP_{0.8}$, and $AP_{0.4}$ on all three splits. In particular, our method reaches 71.45/82.46/62.36 AP/$AP_{0.8}$/$AP_{0.4}$ on Seen objects, 65.85/76.28/54.94 on Similar objects, and 36.42/45.56/25.28 on Novel objects, outperforming all competing methods under the same training protocol. These results indicate that direct grasp inference from stereo spike streams is a strong alternative to reconstruction-based pipelines on the proposed benchmark. More experimental results analysis are in Appendix A.3.

To visually illustrate the performance of our proposed SpikeGrasp algorithm in 6-DoF grasp pose detection, we performed a comparative visualization among our algorithms, as shown in Fig. 3. We visualize the top-50 grasp poses predicted by SpikeGrasp on representative scenes from the Seen, Similar, and Novel splits. Across a wide range of cluttered and HDR arrangements, our method produces grasp proposals that are better aligned with object surfaces and more consistently concentrated on graspable regions. Compared with competing methods, SpikeGrasp also yields more coherent confidence ranking, with higher score grasps typically corresponding to more physically plausible contacts. More visualization results are in the Appendix A.3.

Observation-Space Sufficiency Analysis. We assess grasp decodability on SpikeGrasp-1B by comparing reconstructed frames, frames augmented with spikes, reconstructed point clouds, and initial spike features using frozen upstream modules and a shared lightweight probe architecture. The latent variants also examine training choices: Latent + freeze fixes the backbone while training the grasp predictor. The Latent + joint combines grasp and auxiliary depth supervision, and Ours (default) reports the full model trained with the objective in Section 3.5. The default row therefore serves as a full-model reference within this comparison. Table 3 shows it achieves the highest AP on all splits. On Novel objects, Latent + freeze, Latent + joint, and Ours achieve 26.29, 30.80, and 36.42 AP, respectively. These results support effectiveness of the tested latent pipeline, with performance varying across training settings. Since upstream supervision and trainable components differ, the

Table 1: Quantitative comparison on the SpikeGrasp-1B benchmark. All methods are trained on the same training split and evaluated on subsets. We report the average precision AP, $AP_{0.8}$ and $AP_{0.4}$.

Method	Seen			Similar			Novel		
	AP	$AP_{0.8}$	$AP_{0.4}$	**AP**	$AP_{0.8}$	$AP_{0.4}$	**AP**	$AP_{0.8}$	$AP_{0.4}$
RGB_matters	16.89	22.47	11.23	15.05	19.76	6.19	7.38	8.78	1.32
CenterGrasp	17.86	21.51	13.97	13.31	15.74	10.81	5.94	7.35	3.43
GraspNet	19.29	23.20	15.18	15.67	18.35	13.06	7.90	9.73	4.58
GraspNeRF	32.13	39.43	23.77	30.24	36.17	24.87	12.04	15.21	5.86
GraNet	33.28	39.51	27.80	33.1	39.75	27.24	10.78	13.39	5.87
ASGrasp	38.91	45.20	34.64	32.51	38.55	28.32	13.58	16.94	7.22
GraspFast	41.73	51.56	31.93	39.48	46.95	34.18	17.40	22.08	9.08
Scale-Balanced-Grasp	45.81	53.96	40.35	39.09	45.99	35.03	16.15	20.24	8.45
HGGD	60.78	70.89	53.43	54.80	66.36	45.19	26.11	33.77	16.23
GSNet	64.30	76.39	53.20	56.05	69.05	45.45	26.85	33.41	14.39
FineGrasp	65.30	77.12	54.45	55.67	71.05	48.23	24.85	38.13	17.07
EconomicGrasp	66.18	79.33	56.73	62.04	75.30	51.41	32.99	41.72	17.80
ZeroGrasp	67.43	80.01	58.94	60.73	74.63	50.29	34.26	43.93	19.42
SpikeGrasp (Ours)	**71.45**	**82.46**	**62.36**	**65.85**	**76.28**	**54.94**	**36.42**	**45.56**	**25.28**

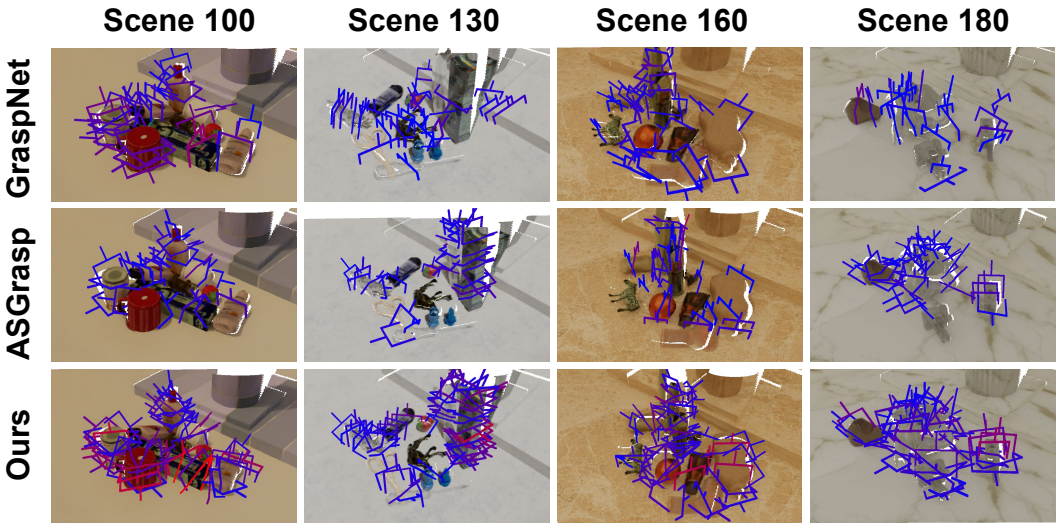


Figure 3: Qualitative comparison on representative scenes from the SpikeGrasp-1B benchmark. From top to bottom: point-cloud baseline, multi-view method, and SpikeGrasp. Compared with the baselines, SpikeGrasp produces grasp poses that are more consistently aligned with object surfaces and more concentrated on graspable regions, especially in cluttered or visually ambiguous scenes.

AP gaps describe the evaluated representation and training settings. Reconstructed point-cloud performance remains competitive on Seen objects but degrade more noticeably on harder subsets, confirming explicit reconstruction is useful but not always sufficient for asynchronous grasp inference.

4.2 Ablation Study

We perform ablations on the synthetic SpikeGrasp-1B benchmark to isolate the contribution of the major components of SpikeGrasp. All variants are trained on the same training split and evaluated on the Seen, Similar, and Novel test subsets using the same protocol as in the main experiments.

Graspability Supervision. We first study the effect of the graspability head by removing the objectness (O) module, the graspness (G) module, or both. As shown in Table 2, removing either branch leads to a clear performance drop on all three subsets, while removing both causes a much larger degradation. In particular, the full model outperforms the variant without graspness by 26.67 AP on Seen objects and 24.99 AP on Similar objects, and it more than doubles the performance on

Table 2: Ablation studies on the synthetic SpikeGrasp-1B benchmark. We report AP on the Seen, Similar, and Novel subsets.

Model Variant	Seen	Similar	Novel
w/o O	36.14	34.41	15.54
w/o G	44.78	40.86	17.28
w/o O + G	25.85	22.08	14.43
w/o spike	43.55	38.08	21.60
w/o adaptive	58.28	50.87	30.60
w/o correlation	51.33	44.88	27.31
w/o iterative	41.67	36.43	19.76

Table 3: Grasp AP for representation probes and latent training variants on SpikeGrasp-1B. Ours (default) is the full-model reference.

Representation	Seen	Similar	Novel
Frame-based	35.70	22.42	17.81
Frame + spike	40.73	26.42	19.01
Point-cloud	44.98	30.70	20.63
Initial feature	47.13	31.28	23.51
Latent + freeze	60.17	48.27	26.29
Latent + joint	64.82	55.19	30.80
Ours (default)	**71.45**	**65.85**	**36.42**

the Novel split. This confirms that explicit supervision of object-centered and grasp-centered regions is critical for guiding downstream pose prediction in cluttered scenes.

Adaptive Spike Neuron Dynamics. We next replace the spike-driven recurrent refinement module with a Vanilla RNN of similar capacity but without spike dynamics. This change reduces AP showing that the performance gain is not simply due to recurrence or model size. Instead, the spike-based recurrent update plays an important role in accumulating and refining grasp-relevant evidence from asynchronous observations. To evaluate the importance of adaptive temporal dynamics, we replace the adaptive spiking neurons in the refinement module with standard LIF neurons while keeping the rest of architectures unchanged. The result suggests that adaptive neuron dynamics improve temporal evidence integration and are especially helpful when generalizing beyond the training distribution.

Architecture Controls. We further introduce two additional controls. First, we remove the stereo correlation volume and replace it with a simpler feature fusion scheme. Then, we disable iterative refinement and perform grasp prediction directly from the initial hidden state, in order to measure the value of hidden state updates. As shown in Table 2, the full model achieves the best performance in all cases, indicating that the advantage of SpikeGrasp arises from the joint effect of spike-native input, stereo correlation, and recurrent refinement rather than from a single design choice.

4.3 Real-World Robot Experiments

To evaluate the performance and generalizability of methods in the real world, we train methods on the synthetic dataset and conduct grasping experiments. Our robotic setup uses a Franka Panda robot and a UMI soft finger (Chi et al., 2024). The robot setup and objects for experiments are in Fig. 2 (b).

Experimental Protocol. We consider four representative scene conditions: isolated objects, moderate clutter, heavy clutter, and HDR scenes. The isolated setting contains one or two target objects with minimal occlusion. The moderate and heavy clutter settings contain multiple objects with increasing overlap and mutual occlusion. The HDR setting includes strong illumination variation, reflective or low-texture surfaces, and visually challenging objects for which depth reconstruction is less stable. For each condition, we evaluate different tabletop scenes with 4 grasp attempts per scene. We compare SpikeGrasp against several representative conventional baselines: the point-cloud-based grasp detector, multi-view and depth restoration methods. A grasp is counted as successful if the target object is lifted from the table and held stably for at least 2 seconds without visible slip or collision induced failure. Declutter rate is average percentage of removed objects in all rounds.

Results and Analysis. Table 4 reports both grasp success rate and declutter rate across four real-world settings. SpikeGrasp achieves the strongest overall performance on both metrics. In terms of grasp success, SpikeGrasp reaches an overall success rate of 85.0%, outperforming the point-cloud baseline by 19.2 points, outperforming the depth restoration baseline by 11.7 points and the multi-view baseline by 5.8 points. The gain is particularly clear in heavy clutter and HDR scenes, where our method exceeds the point-cloud baseline by 16.7 and 46.6 points, respectively. A similar trend is observed for declutter rate: SpikeGrasp attains 82.5% overall, compared with 61.3% for the point-cloud baseline and 76.7% for the multi-view method. The gap between methods is relatively small in isolated scenes, where both explicit geometry and multi-view reasoning remain effective. In HDR

Table 4: Real-world robot grasp success rate (%) / declutter rate (%) across different scene conditions.

Method	Isolated	Moderate Clutter	Heavy Clutter	HDR Scenes	Overall
AnyGrasp (Fang et al., 2023)	86.7 / 83.3	75.0 / 70.0	65.0 / 58.3	36.7 / 33.3	65.8 / 61.3
SwinDR (Dai et al., 2022)	85.0 / 81.7	73.3 / 70.0	61.7 / 58.3	73.3 / 70.0	73.3 / 70.0
ASGrasp (Shi et al., 2024)	83.3 / 81.7	78.3 / 76.7	75.0 / 71.7	80.0 / 76.7	79.2 / 76.7
GraspGen (Murali et al., 2026)	88.3 / 86.7	75.0 / 71.7	65.0 / 58.3	36.7 / 33.3	66.3 / 62.5
SSDC-Grasp (Tong et al., 2026)	81.7 / 80.0	71.7 / 68.3	61.7 / 58.3	70.0 / 66.7	71.3 / 68.3
SpikeGrasp (Ours)	88.3 / 86.7	**86.7 / 83.3**	**81.7 / 78.3**	**83.3 / 81.7**	**85.0 / 82.5**

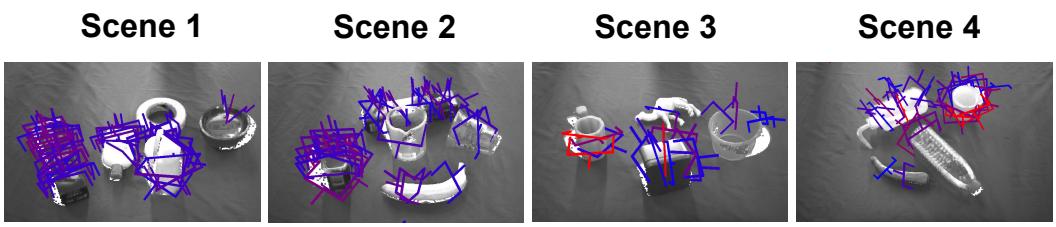


Figure 4: The real-world qualitative results of SpikeGrasp. We show representative grasp predictions and execution outcomes across isolated, cluttered, and HDR scenes. These examples demonstrate that SpikeGrasp transfers from synthetic training to diverse real environments and can produce stable object-centric grasp poses under varying scene complexity.

scenes, the point-cloud baseline degrades most severely. These results indicate that direct inference in the spike observation space is especially advantageous when reconstruction-based pipelines are more vulnerable to missing depth, reflective surfaces, unstable geometry, or severe occlusion.

Figure 4 shows real-world grasping examples of SpikeGrasp across different scene conditions. For each case, SpikeGrasp predicts a 6-DoF grasp pose from the fixed stereo spike camera observations. The results show that the model can transfer from synthetic training to real-world manipulation scenarios and produce stable grasp poses under varying object layouts and illumination conditions. In particular, SpikeGrasp remains effective in scenes where visual appearance is challenging for conventional frame- or depth-based sensing, such as reflective surfaces, strong illumination changes, and cluttered object arrangements. More real robot experiments can be found in the Appendix A.3. The videos captured by real robot testing are in the Supplementary Materials.

5 Conclusion

We presented SpikeGrasp, a framework for direct 6-DoF grasp detection from raw stereo spike streams without reconstructing intermediate point clouds. By combining spike-native stereo encoding, cross-view correlation, and recurrent refinement, the proposed model directly maps asynchronous binocular observations to grasp relevant structure and final 6-DoF grasp poses. Experiments show that SpikeGrasp outperforms competing methods on the synthetic benchmark and also demonstrates clear advantages over conventional reconstruction-based baselines in real HDR scenes. These results suggest that direct learning in the spike observation space is a viable alternative to geometry-first pipelines for embodied manipulation. In addition, we introduce a synthetic benchmark for spike-based 6-DoF grasp detection, which we hope will support future research on neuromorphic perception, low latency robot manipulation, and sim-to-real transfer.

Limitations. Although we conduct 240 real-world grasp trials, training and most controlled evaluations still rely on synthetic data. Consequently, the simulated spike streams, object layouts, and supervision signals may not fully reflect the noise characteristics, calibration errors, and illumination variations of real spike cameras, leaving the sim-to-real gap as a challenge. In addition, while real-world experiments demonstrate the feasibility of direct spike-to-grasp inference and outperform reconstruction-based baselines, the evaluation scale remains limited for broad deployment. Finally, our current focus is grasp pose detection rather than full closed-loop manipulation. Future work will address limitations by collecting larger real stereo spike grasping datasets, improving sim-to-real adaptation, and extending SpikeGrasp to closed-loop manipulation in dynamic real-world settings.

AI USE STATEMENT

In this work, we used generative AI tools for cleaning and reformatting the dataset and assisting with the interpretation of results. We have not used generative AI tools for generating synthetic data sets, developing theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, proposing or refining hypotheses, designing or providing feedback on research methodology or experiments, or implementing methods, and assisting with translation and supporting qualitative and thematic data analysis are not applicable to this work. Additionally, we used generative AI tools for identifying relevant literature and summarizing existing work during the related-work survey, creating and editing software code, and editing the manuscript to improve readability. We have reviewed all AI-assisted work: all literature identified with AI assistance was manually retrieved and read in full by the authors to confirm its relevance and to rule out hallucinated references, all LLM-assisted code was verified and tested for correctness by two authors, and all AI- suggested text edits were checked against the original manuscript by the corresponding author. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

To facilitate reproduction of our results, we provide descriptions of our methodology. All experiments were conducted under PyTorch with CUDA, and the environment is specified. We show model architectures, hyperparameters, training protocols, data-splitting strategies, and evaluation metrics in the main body and Appendix. All experiments were performed with fixed random seeds. We will release our complete dataset, code repository, and demo.

REFERENCES

Michel Breyer, Jen Jen Chung, Lionel Ott, Roland Siegwart, and Juan Nieto. Volumetric grasping network: Real-time 6 dof grasp detection in clutter. In *Proceedings of the 2020 Conference on Robot Learning*, volume 155 of *Proceedings of Machine Learning Research*, pp. 1602–1611, 2021.

Sheng-Hao Cao, Xin-Yong Han, Zhi-Ping Zhao, Jian-Wen Gu, Tian-Zi Jiang, and Shan Yu. Hand position fields of neurons in the premotor cortex of macaques during natural reaching. *Nature Communications*, 16(1):3489, 2025.

Siang Chen, Wei Tang, Pengwei Xie, Wenming Yang, and Guijin Wang. Efficient heatmap-guided 6-dof grasp detection in cluttered scenes. *IEEE Robotics and Automation Letters*, 8(8):4895–4902, 2023.

Cheng Chi, Zhenjia Xu, Chuer Pan, Eric Cousineau, Benjamin Burchfiel, Siyuan Feng, Russ Tedrake, and Shuran Song. Universal Manipulation Interface: In-The-Wild Robot Teaching Without In-The-Wild Robots. In *Proceedings of Robotics: Science and Systems*, July 2024.

Eugenio Chisari, Nick Heppert, Tim Welschehold, Wolfram Burgard, and Abhinav Valada. Centergrasp: Object-aware implicit representation learning for simultaneous shape reconstruction and 6-dof grasp estimation. *IEEE Robotics and Automation Letters*, 9(6):5094–5101, 2024.

Qiyu Dai, Jiyao Zhang, Qiwei Li, Tianhao Wu, Hao Dong, Ziyuan Liu, Ping Tan, and He Wang. Domain randomization-enhanced depth simulation and restoration for perceiving and grasping specular and transparent objects. In *Computer Vision – ECCV 2022*, pp. 374–391, 2022.

Qiyu Dai, Yan Zhu, Yiran Geng, Ciyu Ruan, Jiazhao Zhang, and He Wang. Graspnerf: Multiview-based 6-dof grasp detection for transparent and specular objects using generalizable nerf. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1757–1763, 2023.

Xinke Deng, Yu Xiang, Arsalan Mousavian, Clemens Eppner, Timothy Bretl, and Dieter Fox. Self-supervised 6d object pose estimation for robot manipulation. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3665–3671, 2020.

Yanchen Dong, Jing Zhao, Ruiqin Xiong, and Tiejun Huang. 3d residual interpolation for spike camera demosaicing. In *2022 IEEE International Conference on Image Processing (ICIP)*, pp. 1461–1465, 2022.

Yanchen Dong, Ruiqin Xiong, Jian Zhang, Zhaofei Yu, Xiaopeng Fan, Shuyuan Zhu, and Tiejun Huang. Super-resolution reconstruction from bayer-pattern spike streams. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24871–24880, 2024.

Ria Doshi, Homer Rich Walke, Oier Mees, Sudeep Dasari, and Sergey Levine. Scaling cross-embodied learning: One policy for manipulation, navigation, locomotion and aviation. In *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 496–512, 2025.

Yun Du, Mengao Zhao, Tianwei Lin, Yiwei Jin, Chaodong Huang, and Zhizhong Su. Finegrasp: Towards robust grasping for delicate objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 4428–4435, 2026.

Hao-Shu Fang, Chenxi Wang, Minghao Gou, and Cewu Lu. Graspnet-1billion: A large-scale benchmark for general object grasping. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Hao-Shu Fang, Chenxi Wang, Hongjie Fang, Minghao Gou, Jirong Liu, Hengxu Yan, Wenhai Liu, Yichen Xie, and Cewu Lu. Anygrasp: Robust and efficient grasp perception in spatial and temporal domains. *IEEE Transactions on Robotics*, 39(5):3929–3945, 2023.

Zhuoheng Gao, Yihao Li, Jiyao Zhang, Rui Zhao, Tong Wu, Hao Tang, Zhaofei Yu, Hao Dong, Guozhang Chen, and Tiejun Huang. Spikestereonet: A brain-inspired framework for stereo depth estimation from spike streams. In *The Fourteenth International Conference on Learning Representations*, 2026.

Minghao Gou, Hao-Shu Fang, Zhanda Zhu, Sheng Xu, Chenxi Wang, and Cewu Lu. Rgb matters: Learning 7-dof grasp poses on monocular rgbd images. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 13459–13466, 2021.

Haiyuan Gui, Shanchen Pang, Xiao He, Luqi Wang, Xue Zhai, Shihang Yu, and Kuijie Zhang. Graspfast: Multi-stage lightweight 6-dof grasp pose fast detection with rgb-d image. *Pattern Recognition*, 161:111318, 2025.

Haoran He, Chenjia Bai, Ling Pan, Weinan Zhang, Bin Zhao, and Xuelong Li. Learning an actionable discrete diffusion policy via large-scale actionless video pre-training. In *Advances in Neural Information Processing Systems*, volume 37, pp. 31124–31153, 2024.

Dinh-Cuong Hoang, Johannes A. Stork, and Todor Stoyanov. Context-aware grasp generation in cluttered scenes. In *2022 International Conference on Robotics and Automation (ICRA)*, pp. 1492–1498, 2022.

Mingxiao Huo, Mingyu Ding, Chenfeng Xu, Thomas Tian, Xinghao Zhu, Yao Mu, Lingfeng Sun, Masayoshi Tomizuka, and Wei Zhan. Human-oriented Representation Learning for Robotic Manipulation. In *Proceedings of Robotics: Science and Systems*, July 2024.

Shun Iwase, Muhammad Zubair Irshad, Katherine Liu, Vitor Guizilini, Robert Lee, Takuya Ikeda, Ayako Amma, Koichi Nishiwaki, Kris Kitani, Rares Ambrus, and Sergey Zakharov. Zerograsp: Zero-shot shape reconstruction enabled robotic grasping. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pp. 17405–17415, 2025.

Stephen James, Kentaro Wada, Tristan Laidlow, and Andrew J. Davison. Coarse-to-fine q-attention: Efficient learning for visual robotic manipulation via discretisation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13739–13748, 2022.

Snehal Jauhri, Ishikaa Lunawat, and Georgia Chalvatzaki. Learning any-view 6dof robotic grasping in cluttered scenes via neural surface rendering. In *Robotics: Science and Systems XX*, RSS2024, 2024.

Hanme Kim, Stefan Leutenegger, and Andrew J. Davison. Real-time 3d reconstruction and 6-dof tracking with an event camera. In *Computer Vision – ECCV 2016*, pp. 349–364, 2016.

Bin Li, Hu Cao, Zhongnan Qu, Yingbai Hu, Zhenke Wang, and Zichen Liang. Event-based robotic grasping detection with neuromorphic vision sensor and event-grasping dataset. *Frontiers in Neurorobotics*, Volume 14 - 2020, 2020.

Hongzhuo Liang, Xiaojian Ma, Shuang Li, Michael Görner, Song Tang, Bin Fang, Fuchun Sun, and Jianwei Zhang. Pointnetgpd: Detecting grasp configurations from point sets. In *2019 International Conference on Robotics and Automation (ICRA)*, pp. 3629–3635, 2019.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision – ECCV 2014*, pp. 740–755, 2014.

Xiaozheng Liu, Yunzhou Zhang, He Cao, Dexing Shan, and Jiaqi Zhao. Joint segmentation and grasp pose detection with multi-modal feature fusion network. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1751–1756, 2023.

Zibin Liu, Banglei Guan, Yang Shang, Yifei Bian, Pengju Sun, and Qifeng Yu. Stereo event-based, 6-dof pose tracking for uncooperative spacecraft. *IEEE Transactions on Geoscience and Remote Sensing*, 63:1–13, 2025.

Haoxiang Ma and Di Huang. Towards scale balanced 6-dof grasp detection in cluttered scenes. In *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 2004–2013, 14–18 Dec 2023.

Haoxiang Ma, Modi Shi, Boyang Gao, and Di Huang. Active perception for grasp detection via neural graspness field. In *Advances in Neural Information Processing Systems*, volume 37, pp. 38122–38141, 2024.

Jeffrey Mahler, Jacky Liang, Sherdil Niyaz, Michael Laskey, Richard Doan, Xinyu Liu, Juan Aparicio, and Ken Goldberg. Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics. In *Robotics: Science and Systems XIII*, RSS2017, 2017.

Adithyavairavan Murali, Balakumar Sundaralingam, Yu-Wei Chao, Jun Yamada, Wentao Yuan, Mark Carlson, Fabio Ramos, Stan Birchfield, Dieter Fox, and Clemens Eppner. Graspgen: A diffusion-based framework for 6-dof grasping with on-generator training. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2026.

Peiyuan Ni, Wenguang Zhang, Xiaoxiao Zhu, and Qixin Cao. Pointnet++ grasping: Learning an end-to-end spatial grasp generation algorithm from sparse point clouds. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3619–3625, 2020.

Ran Qin, Haoxiang Ma, Boyang Gao, and Di Huang. Rgb-d grasp detection via depth guided learning with cross-modal attention. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8003–8009, 2023.

Yuzhe Qin, Rui Chen, Hao Zhu, Meng Song, Jing Xu, and Hao Su. S4g: Amodal single-view single-shot se(3) grasp detection in cluttered scenes. In *Proceedings of the Conference on Robot Learning*, volume 100 of *Proceedings of Machine Learning Research*, pp. 53–65, 2020.

Jun Shi, Yong A, Yixiang Jin, Dingzhe Li, Haoyu Niu, Zhezhu Jin, and He Wang. Asgrasp: Generalizable transparent object reconstruction and 6-dof grasp detection from rgb-d active stereo camera. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5441–5447, 2024.

Martin Sundermeyer, Arsalan Mousavian, Rudolph Triebel, and Dieter Fox. Contact-graspnet: Efficient 6-dof grasp generation in cluttered scenes. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 13438–13444, 2021.

Andreas ten Pas, Marcus Gualtieri, Kate Saenko, and Robert Platt. Grasp pose detection in point clouds. *The International Journal of Robotics Research*, 36(13-14):1455–1473, 2017.

Ling Tong, Kechen Song, Hongkun Tian, Yunhui Yan, Fei Shen, and Xian Tao. Ssdc-grasp: A joint semantic segmentation and depth completion approach for robotic grasping of transparent objects. *IEEE Transactions on Instrumentation and Measurement*, 75:1–12, 2026.

Chenxi Wang, Hao-Shu Fang, Minghao Gou, Hongjie Fang, Jin Gao, and Cewu Lu. Graspness discovery in clutters for fast and accurate grasp detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15964–15973, 2021.

Haowen Wang, Wanhao Niu, and Chungang Zhuang. Granet: A multi-level graph network for 6-dof grasp pose generation in cluttered scenes. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 937–943, 2023.

Xiao-Ming Wu, Jia-Feng Cai, Jian-Jian Jiang, Dian Zheng, Yi-Lin Wei, and Wei-Shi Zheng. An economic framework for 6-dof grasp detection. In *Computer Vision – ECCV 2024*, pp. 357–375, November 2024.

Lujie Xia, Jing Zhao, Ruiqin Xiong, and Tiejun Huang. Svfi: Spiking-based video frame interpolation for high-speed motion. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(3): 2910–2918, 2023.

Jiyao Zhang, Weiyao Huang, Bo Peng, Mingdong Wu, Fei Hu, Zijian Chen, Bo Zhao, and Hao Dong. Omni6dpose: A benchmark and model for universal 6d object pose estimation and tracking. In *Computer Vision – ECCV 2024*, pp. 199–216, 2024.

Jing Zhao, Jiyu Xie, Ruiqin Xiong, Jian Zhang, Zhaofei Yu, and Tiejun Huang. Super resolve dynamic scene from continuous spike streams. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2533–2542, 2021.

Jing Zhao, Ruiqin Xiong, Jian Zhang, Rui Zhao, Hangfan Liu, and Tiejun Huang. Learning to super-resolve dynamic scenes for neuromorphic spike camera. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(3):3579–3587, 2023.

Junwei Zhao, Shiliang Zhang, Lei Ma, Zhaofei Yu, and Tiejun Huang. Spikingsim: A bio-inspired spiking simulator. In *2022 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 3003–3007, 2022.

Rui Zhao, Ruiqin Xiong, Dongkai Wang, Shiyu Xuan, Jian Zhang, Xiaopeng Fan, and Tiejun Huang. Spike camera optical flow estimation based on continuous spike streams. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(4):4756–4770, 2026.

Yajing Zheng, Chengen Li, Jiyuan Zhang, Zhaofei Yu, and Tiejun Huang. Snntracker: Online high-speed multi-object tracking with spike camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(1):624–638, 2026a.

Yajing Zheng, Jiyuan Zhang, Rui Zhao, Jianhao Ding, Shiyan Chen, Weijian Wu, Ruiqin Xiong, Zhaofei Yu, and Tiejun Huang. Spikecv: open a continuous computer vision era. *Science China Information Sciences*, 69(3), 2026b.

Chu Zhou, Hang Zhao, Jin Han, Chang Xu, Chao Xu, Tiejun Huang, and Boxin Shi. Unmodnet: Learning to unwrap a modulo image for high dynamic range imaging. In *Advances in Neural Information Processing Systems*, volume 33, pp. 1559–1570, 2020.

Pei Zhou, Ruizhe Liu, Qian Luo, Fan Wang, Yibing Song, and Yanchao Yang. Autocgp: Closed-loop concept-guided policies from unlabeled demonstrations. In *International Conference on Learning Representations*, volume 2025, pp. 44815–44842, 2025.

A TECHNICAL APPENDICES AND SUPPLEMENTARY MATERIAL

The appendix provides additional implementation details and extended experimental results for the SpikeGrasp framework introduced in the main paper. It is organized into four distinct sections: The Theory Analysis and Method in Appendix A.1, Details of the Dataset in Appendix A.2, More Experimental Results in Appendix A.3, Reproducibility and Implementation Details in Appendix A.4 and Broader Impacts in Appendix A.5.

A.1 Theory Analysis and Method Details

A.1.1 Problem Statement

We study direct 6-DoF grasp detection from a single window of stereo spike observations. Each camera provides N_{read} consecutive binary readouts at $f_{\mathrm{cam}} = 20{,}000\,\mathrm{Hz}$, corresponding to a readout interval of $\Delta t_{\mathrm{cam}} = 1/f_{\mathrm{cam}} = 50\,\mu\mathrm{s}$. Unless otherwise specified, we use $N_{\mathrm{read}} = 50$, giving an input duration of $W_{\mathrm{input}} = N_{\mathrm{read}}\Delta t_{\mathrm{cam}}$. The left and right cameras cover the same observation interval. We define t_{ref} as the end of this interval and write the input tensors as

$$\mathbf{S}^v(t_{\mathrm{ref}}) = \left(\mathbf{S}_n^v\right)_{n=1}^{N_{\mathrm{read}}} \in \{0,1\}^{N_{\mathrm{read}}\times H\times W}, \qquad v \in \{l, r\}. \tag{8}$$

here, H and W are the spatial dimensions, and $\mathbf{S}_n^v \in \{0,1\}^{H\times W}$ is the n-th binary readout from camera v. The readouts cover $(t_{\mathrm{ref}} - W_{\mathrm{input}}, t_{\mathrm{ref}}]$, and no observations acquired after t_{ref} are included in the input. The symbol N_{read} corresponds to N in Section 3.1.

Our objective is to predict grasp candidates directly from these tensors:

$$\widehat{\mathcal{G}} = f_\theta\big(\mathbf{S}^l(t_{\mathrm{ref}}), \mathbf{S}^r(t_{\mathrm{ref}})\big) = \{(\mathbf{R}_i, \mathbf{t}_i, w_i, q_i)\}_{i=1}^{N_g}, \tag{9}$$

where f_θ denotes SpikeGrasp and N_g is the number of predicted candidates. Each candidate contains a gripper orientation $\mathbf{R}_i$, a translation $\mathbf{t}_i$ in the left-camera frame, a gripper width w_i, and a grasp score q_i, as defined in Section 3.4. The grasp decoder operates on the learned latent representation.

The framework comprises stereo spike encoding, cross-view stereo correlation, recurrent refinement, and grasp prediction heads. As described in Section 3.2, the correlation volume compares spatial feature positions across the two views, and temporal evidence is carried by the features encoded from the input window. Network inference operates on windowed binary tensors rather than processing each sensor spike as a separate asynchronous update. We distinguish the camera readout index $n = 1, \ldots, N_{\mathrm{read}}$ from the refinement index $k = 0, \ldots, K-1$. By default, $K = 16$, and all refinement updates use features from the same input window. The neuron-state updates are internal network computations and do not represent additional camera readouts. Consequently, changing K does not change W_{input}.

A.1.2 Spike Noise Simulation

In real imaging scenarios, the number of photons reaching the sensing chip through diffuse reflection varies randomly over time, resulting in non-uniform spike intervals. To model this effect, we introduce a Poisson process to represent diffuse noise during spike generation. Let γ denote the average number of noise photons arriving at a pixel within a unit time δt ($\delta t = 2/\mathrm{CLK}$), and let T_r denote the readout interval (Zhao et al., 2022). The corresponding probability mass function is defined as

$$P(k \mid T_r, \gamma) = \frac{(\gamma T_r)^k}{k!} e^{-\gamma T_r}, \quad k = 0, 1, 2, \ldots \tag{10}$$

This model captures the random fluctuations in photon arrivals and helps produce more realistic variations in spike intervals.

In addition to diffuse noise, inherent sensor noise may trigger spikes even in the absence of illumination. To characterize this effect, we cover the spiking camera with a lens cap and record the emitted spikes over a fixed duration (e.g., 1 s). We then compute the inter-spike interval (ISI) distribution of the recorded data. The ISI distribution can be well approximated by a Gaussian model with mean $\mu = 140$ and standard deviation $\sigma = 50$:

$$N(k \mid \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{k-\mu}{\sigma}\right)^2\right), \quad k = 0, 1, 2, \ldots \tag{11}$$

This Gaussian term is used to refine the spike-interval distribution obtained from the firing frequency computed in Eq. 11, thereby improving the realism of the simulated spike streams.

A.1.3 THE ADAPTIVE LEAKY INTEGRATE-AND-FIRE MODEL

Common RSNN models often neglect biologically relevant dynamics that evolve over longer time scales. To better capture such behavior, we incorporate neuronal adaptation into our recurrent spiking architecture. This design is motivated by experimental evidence, including observations from the Allen Brain Atlas, showing that many excitatory neurons exhibit adaptive responses with diverse time constants. Following this motivation, we adopt the adaptive leaky integrate-and-fire dynamics used in (Gao et al., 2026) and restate the update equations here for completeness. These neuronal dynamics are an inherited component of the perception backbone. Their parameters are optimized under the training objective specified for each experimental variant.

$$\begin{aligned} \boldsymbol{h_t} &= \boldsymbol{\alpha_t} \odot \boldsymbol{v_{t-1}} + (1 - \boldsymbol{\alpha_t}) \odot \left(W_{\text{rec}} \boldsymbol{s_{t-1}} + W_f \boldsymbol{s_t^{(l-1)}} \right), \\ \boldsymbol{v_t^{\text{th}}} &= \boldsymbol{\beta_t} \odot v_{\text{peak}}, \\ \boldsymbol{s_t} &= \theta \left(\boldsymbol{h_t} - \boldsymbol{v_t^{\text{th}}} \right), \\ \boldsymbol{v_t} &= \boldsymbol{h_t} - \boldsymbol{\gamma_t} \odot \boldsymbol{s_t} \odot \boldsymbol{v_t^{\text{th}}}, \end{aligned} \tag{12}$$

where $\boldsymbol{v_t}$ denotes the membrane potential, $\boldsymbol{h_t}$ is the pre-spike membrane state, $\boldsymbol{v_t^{\text{th}}}$ is the adaptive firing threshold, and $\theta(\cdot)$ is the Heaviside step function. Here, W_{rec} and W_f are the recurrent and feedforward synaptic weights, implemented as convolutional kernels, and $\boldsymbol{s_t} \in \{0, 1\}$ denotes the binary spike output. The adaptive variables ($\boldsymbol{\alpha_t}, \boldsymbol{\beta_t}, \boldsymbol{\gamma_t} \in [0, 1]$) control membrane retention, threshold modulation, and soft reset strength, respectively. In this way, the neuron can dynamically adjust its temporal integration, excitability, and post-spike reset behavior. Such a mechanism is more consistent with biological findings that neuronal properties, including membrane time constants, firing thresholds, and reset dynamics, vary with internal state and contextual input rather than remaining fixed.

A.1.4 ALGORITHMIC PIPELINE AND PSEUDOCODE

Algorithm 1 summarizes the inference procedure of SpikeGrasp. Given stereo spike streams, the model first extracts multi-scale spike features, constructs a cross-view correlation pyramid, iteratively refines the hidden state using the RSNN update module, and finally decodes graspable regions and 6-DoF grasp hypotheses.

Algorithm 1 SpikeGrasp Inference

Require: Stereo spike streams S^l, S^r, number of refinement steps K
1: Extract multi-scale spike features f^l, f^r
2: Construct stereo correlation volume C
3: Build correlation pyramid $\{C^{(1)}, \ldots, C^{(4)}\}$
4: Initialize hidden state h^0
5: **for** $k = 0$ to $K - 1$ **do**
6: Sample correlation features conditioned on h^k
7: Fuse sampled correlation with spike context features
8: Update hidden state using RSNN:
9: $h^{k+1} = h^k + \Delta h^k$
10: **end for**
11: Decode objectness and graspness maps from h^K
12: Select candidate latent features using the implemented graspability rule
13: Decode (R_i, t_i, w_i, q_i) for each candidate in the left-camera view
14: Apply the specified grasp ranking and post-processing
15: **return** the top-ranked grasp set.

A.1.5 REAL-WORLD ROBOT PROTOCOL AND CAMERA CALIBRATION

Robot Platform. In the real-world experiments, SpikeGrasp uses a fixed external stereo rig composed of two 001T-Gen2 spike cameras, while the RGB-D/depth-based baselines use an Intel RealSense D435i RGB-D camera. We conduct real-world grasping experiments using a Franka Panda robot

equipped with a UMI soft finger. All methods share the same low-level execution controller and the same grasp execution protocol. The predicted grasp pose is first transformed into the robot base frame, followed by a pre-grasp motion, straight line approach, gripper closure, and object lifting.

Although the robot, gripper, execution controller, and success criterion are shared, the sensor observations are not identical. This experiment is interpreted as a system-level comparison. In the same-spike controls, all compared pathways use the same stereo spike rig, calibration, acquisition settings, and temporal support. Robot trials are organized in matched scene-layout blocks with scene resets and a prespecified method order or randomization schedule.

Sensor Placement. Our stereo spike camera system is deployed in a fixed eye-to-hand configuration. Specifically, the binocular spike cameras are rigidly mounted in front of the robot workspace, facing the manipulation area from the opposite side of the robot arm. The stereo rig remains fixed during the entire experiment, and its optical axes are approximately parallel to the tabletop workspace.

Stereo spike camera calibration. We first calibrate the intrinsic parameters of the left and right spike cameras and the rigid transformation between the two cameras. The stereo rig is then rectified so that binocular matching and cross-view correlation are performed in a common rectified coordinate system. We use the left spike camera as the reference camera and denote its coordinate frame as $\mathcal{F}_{C_s}$. The model predicts each grasp pose in this spike-camera frame:

$$G^{C_s} = (R_g^{C_s}, t_g^{C_s}, w), \tag{13}$$

where $R_g^{C_s}$, $t_g^{C_s}$, and w denote the gripper orientation, translation, and width, respectively.

Eye-to-Hand Calibration for experiments. Since the stereo spike rig is fixed in the scene, we estimate a constant transformation from the spike camera data to the robot base frame:

$$T_{C_s}^B \in SE(3), \tag{14}$$

where $\mathcal{F}_B$ denotes the robot base frame. This eye-to-hand calibration is obtained by observing a calibration target placed at known robot frame poses and solving for the rigid camera-to-base transformation. During deployment, the grasp pose predicted by SpikeGrasp is transformed to the robot base frame by

$$T_g^B = T_{C_s}^B T_g^{C_s}. \tag{15}$$

The transformed grasp T_g^B is then passed to the shared robot execution module.

Execution Protocol. For each predicted grasp pose T_g^B, the robot first moves to a pre-grasp pose that shares the same orientation as the target grasp pose. The pre-grasp position is obtained by offsetting the target position by $d_{\text{pre}} = 10$ cm along the negative approach direction:

$$t_{\text{pre}}^B = t_g^B - d_{\text{pre}} a_g^B, \tag{16}$$

where a_g^B is the gripper approach direction in the robot base frame. The robot then executes a straight line approach from the pre-grasp pose to the target grasp pose, closes the gripper, and lifts the object vertically by 10 cm. This protocol ensures that different methods are evaluated with the same grasp execution strategy after pose prediction.

Success and Declutter Criteria. A grasp trial is counted as successful if the target object is lifted from the tabletop and held stably for at least 2 seconds without visible slip or collision-induced failure. For all scenes, we additionally report the declutter rate. A trial is counted as a successful decluttering action if the target object is removed from the scene while avoiding severe disturbance, object drop, or unintended displacement of neighboring objects.

Comparison Protocol. To make the comparison as controlled as possible, we use the same robot, gripper, scene layouts, object sets, success definition, and execution controller across all methods. We also report results separately for isolated objects, moderate clutter, heavy clutter, and HDR scenes to reveal how each sensing configuration behaves under different scene difficulties.

A.2 Details of the Dataset

A conventional active stereo depth system typically comprises an infrared projector, a pair of IR stereo cameras and a color camera. We use the method from (Dai et al., 2022; Zhang et al., 2024) that

Table 5: Domain randomization factors used in SpikeGrasp-1B.

Factor	Randomization range / choice
Object pose	Random translation and rotation on tabletop
Camera viewpoint	Random azimuth and elevation around the scene
Lighting	Random intensity, direction, and HDR exposure
Material	Raw, Diffuse, specular, transparent, mixed materials
Background	Random tabletop and environment textures
Clutter level	Random number and placement of objects
Occlusion	Specific position occlusion

Table 6: Statistics of the SpikeGrasp-1B synthetic dataset.

Dataset	Scene Indexes	Number of objectness	Number of graspness	Number of depth	Number of Spike Frames	Number of Scenes
Train	0–99	25,600	25,600	25,600	51,200	100
Seen	100–129	7,680	7,680	7,680	15,360	30
Similar	130–159	7,680	7,680	7,680	15,360	30
Novel	160–189	7,680	7,680	7,680	15,360	30

replicates this process, containing light pattern projection, capture, and segmentation. The simulator is implemented based on the Blender rendering engine. To further bridge the domain gap between synthetic and real-world data, we adopt domain randomization strategies by randomizing object textures, material properties, object layouts, floor textures, and different lighting conditions. Table 5 summarizes the randomization ranges used in our simulator.

A.2.1 Synthetic Dataset

Existing grasp datasets evaluate RGB-D or point-cloud observation spaces, but they cannot answer whether raw asynchronous spike streams are sufficient for 6-DoF grasp inference. We construct a large-scale synthetic dataset SpikeGrasp-1B with domain randomization and a dedicated spike simulation pipeline, which is designed to evaluate this observation space question by pairing stereo spike streams with grasp annotations under controlled clutter, material, viewpoint, and HDR conditions. The dataset consists of a training subset and a test subset. These datasets are built from 88 distinct objects with randomized material properties, including raw, specular, transparent, and diffuse surfaces. The training set contains 51,200 spike streams and 25,600 objectness and graspness maps generated from 100 scenes. The test set contains 46,080 spike streams and 23,040 maps from 90 additional scenes. It is further divided into three parts: Seen, Similar, and Novel subsets, each with 30 scenes. For every scene, we provide 3 to 9 million grasp poses, resulting in more than 1 billion grasps in total, together with the binocular camera pose for each grasping scene. The objectness is a binary foreground mask generated from scene segmentation, and graspness is a continuous scene-level score computed by aggregating collision-free object-level grasps projected into the scene. We do not threshold graspness into a binary label for the main graspness regression and thresholding is only used to select positive candidates for score and width supervision. In addition, to enable fair comparison with baseline methods that take depth as input, we also render aligned depth images for all scenes at the same resolution as the spike data and other annotations. Detailed dataset statistics are summarized in Table 6, and representative examples are shown in Fig. 5.

The object meshes in our scenes are taken from the GraspNet-1Billion dataset (Fang et al., 2020), and the grasp poses are kept consistent with the original annotations. To generate diverse viewpoints, we sample camera poses uniformly on a virtual sphere with a fixed radius of 0.6,m and use them to form continuous camera trajectories. Neighboring viewpoints are separated by a constant angular increment of 3°, which ensures smooth transitions along the trajectory. The objects are rotated accordingly to maintain consistent spatial alignment.

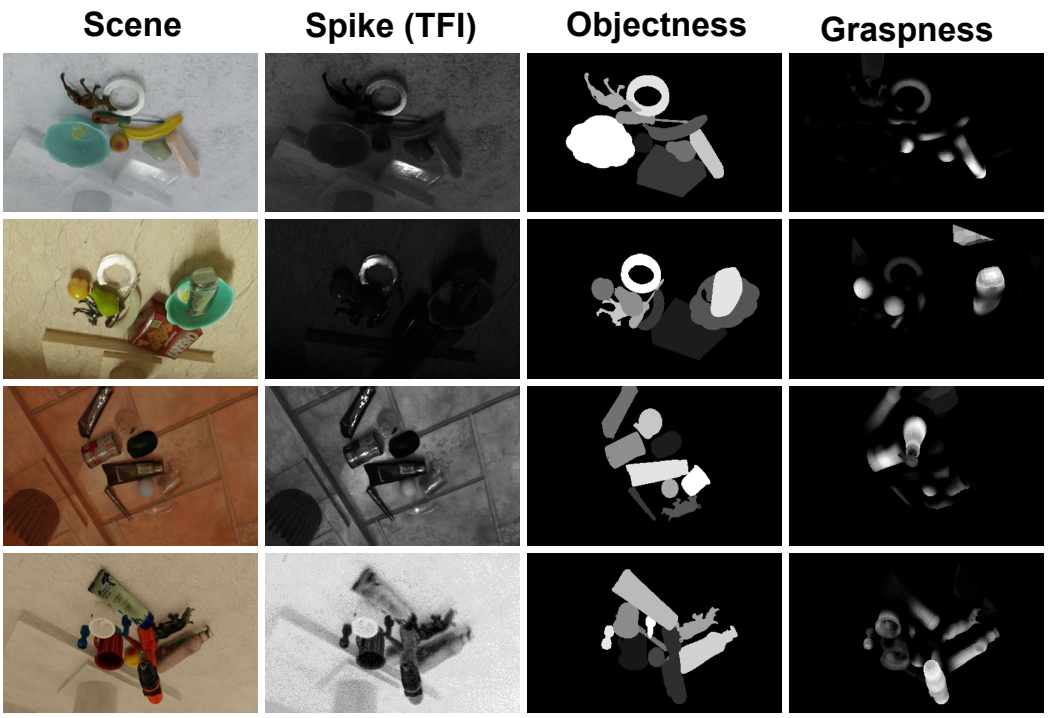


Figure 5: Visualize the results of examples in the synthetic dataset. From left to right are the scene map, TFI method reconstruction image of spike stream, objectness and graspness maps of the scene.

A.2.2 Spike Stream Generation

We generate synthetic spike streams directly from the original left and right HDR scene radiance, rather than from reconstructed LDR frames. Specifically, for each pixel location (x, y), we maintain an accumulator $A(x, y, t)$ that integrates the incoming HDR photon response over time. A spike is emitted once the accumulated response reaches a preset threshold θ, after which the residual charge is retained for subsequent integration. The spike generation process is written as

$$A(x, y, t) = A(x, y, t-1) + I(x, y, t), \qquad S(x, y, t) = \mathbb{I}[A(x, y, t) \geq \theta], \tag{17}$$

followed by

$$A(x, y, t) \leftarrow A(x, y, t) - S(x, y, t)\,\theta, \tag{18}$$

where $I(x, y, t)$ denotes the HDR photon response at time t, $S(x, y, t) \in \{0, 1\}$ is the generated spike, and $\mathbb{I}[\cdot]$ is the indicator function. The left and right spike streams are simulated independently but in temporal correspondence, yielding binocular spike observations that preserve the native high-dynamic-range and high-temporal-resolution characteristics of the original scene.

To reduce the domain gap between synthetic and real measurements, we further inject diffuse reflection noise and sensor noise during simulation from Eq. 10 and 11. In particular, we apply small random perturbations to the photon accumulation process and firing thresholds, and introduce mild temporal jitter in spike emission. This noise-aware simulation produces more realistic sparsity patterns and temporal fluctuations, making the generated spike distribution better aligned with real spike camera observations in HDR environments. As a result, the synthetic stereo spike streams provide a more faithful training signal for robust stereo spike-based grasp learning.

A.2.3 Graspness Generation

A cluttered scene contains multiple objects and unrelated backgrounds. Therefore, it is important to generate the scene-level graspness in advance in the dataset. We computed the graspness map for each frame using segmentation and scene maps, as well as object-level grasp scores from the dataset, using the method similar to GSNet (Wang et al., 2021).

To address collisions, we reconstruct the scene using 3D object models and their 6-DoF poses. For each grasp $g_k^{i,j}$, we perform the collision check and assign a binary label $c_k^{i,j} \in \{0, 1\}$. We then compute scene-level graspness by aggregating collision-free, high-quality grasps:

$$\begin{aligned}\tilde{s}_i^{\mathrm{p}} =& \frac{\sum_{j=1}^{V}\sum_{k=1}^{L}\mathbf{1}(q_k^{i,j} > c)\cdot\mathbf{1}(c_k^{i,j})}{\sum_{j=1}^{V}|\mathcal{G}_{i,j}|}, i = 1, ..., N,\\ \tilde{s}_i^{\mathrm{v}} =& \Big\{\frac{\sum_{k=1}^{L}\mathbf{1}(q_k^{i,j} > c)\cdot\mathbf{1}(c_k^{i,j})}{|\mathcal{G}_{i,j}|}\Big|1 \le j \le V\Big\}, i = 1, ..., N.\end{aligned} \tag{19}$$

where $q_k^{i,j}$ is the grasp quality, τ is a quality threshold, V is the number of views, L the number of grasps per view, and $\mathcal{G}_{i,j}$ the candidate set at the scene point i and view j. Next, we project object points into the scene using their 6-DoF poses and, for each scene point, transfer graspness by nearest-neighbor association with the projected object points.

Finally, we normalize the scores per scene to obtain coherent maps.

$$\begin{aligned}\mathcal{S}^p &= \Big\{\frac{\tilde{s}_i^p - \min(\tilde{\mathcal{S}}^p)}{\max(\tilde{\mathcal{S}}^p) - \min(\tilde{\mathcal{S}}^p)}\Big|i = 1, ..., N\Big\},\\ \mathcal{S}^v &= \Big\{\frac{\tilde{s}_i^v - \mathbf{min}(\tilde{\mathcal{S}}^v)}{\mathbf{max}(\tilde{\mathcal{S}}^v) - \mathbf{min}(\tilde{\mathcal{S}}^v)}\Big|i = 1, ..., N\Big\},\end{aligned} \tag{20}$$

where $\mathbf{min}(\cdot)$ denotes column-wise minimum:

$$\mathbf{min}(\tilde{\mathcal{S}}^v) = \Big\{\min_{i=1}^{N}\tilde{s}_{i(j)}^v\Big|j = 1, ..., V\Big\}, \tag{21}$$

and so does $\mathbf{max}(\cdot)$. An example of the resulting scene-level graspness map is shown in Fig. 5.

A.2.4 Evaluation Metrics

Before scoring, we apply SE(3) non-maximum suppression (NMS) to the predicted grasps and keep at most $N_{\text{per obj}}$ grasps per object. Let $\mathcal{G}_t = \{(\hat{G}_i, s_i)\}_{i=1}^{M_t}$ be the predictions for frame t, sorted by confidence s_i (descending) after NMS and per-object capping.

True-positive test (given friction μ). For each $\hat{G}_i$, define

$$y_i^{(\mu)} \in \{0, 1\} \tag{22}$$

to be 1 if $\hat{G}_i$ (i) is assigned to a scene object, (ii) is collision-free, and (iii) satisfies force-closure under Coulomb friction coefficient μ; otherwise 0.

Precision at the cutoff k.

$$\mathrm{Prec}_t^{(\mu)}(k) \;=\; \frac{1}{k}\sum_{i=1}^{k} y_i^{(\mu)}, \qquad k = 1, \ldots, K_t, \tag{23}$$

where $K_t = \min(50,\, M_t)$. We chose the top 50 score of grasp poses predicted in this paper.

Frame-level average precision at μ.

$$\mathrm{AP}_\mu(t) \;=\; \frac{1}{K_t}\sum_{k=1}^{K_t}\mathrm{Prec}_t^{(\mu)}(k). \tag{24}$$

Dataset-level average precision (AP) at μ. For a test set of T frames,

$$\overline{\mathrm{AP}}_\mu \;=\; \frac{1}{T}\sum_{t=1}^{T}\mathrm{AP}_\mu(t). \tag{25}$$

$\mathbf{AP}_{0.8}$ and $\mathbf{AP}_{0.4}$. These are $\overline{\mathrm{AP}}_\mu$ evaluated at $\mu = 0.8$ and $\mu = 0.4$, respectively.

Overall AP (multi-μ average). Given a friction set M (e.g. $\{0.2, 0.4, 0.6, 0.8, 1.0, 1.2\}$),

$$\mathrm{AP} \;=\; \frac{1}{|M|} \sum_{\mu \in M} \overline{\mathrm{AP}}_{\mu}. \tag{26}$$

Specify the NMS thresholds in translation and rotation as 0.03 m and $\pi/6$ rad, respectively, the per-object cap $N_{\text{per obj}}$, the friction set M, and the results are averaged in the synthetic dataset. We followed the standard evaluation protocols using COCO (Lin et al., 2014) metrics for segmentation, and AP metrics (Wang et al., 2021) for pose estimation.

A.2.5 Dataset Format

SpikeGrasp-1B will be released with stereo spike streams, scene-level objectness and graspness annotations, camera parameters, object poses, and grasp labels. Each scene is organized as a folder containing left and right spike data, metadata files, segmentation masks, and grasp annotations. The dataset also provides train/test split files for Seen, Similar, and Novel evaluation. The released package will include documentation describing data fields, file formats, preprocessing scripts, and evaluation protocols.

A.3 More Experimental Results

A.3.1 Observation-Space Motivation and Empirical Sufficiency Probe

We use the data processing inequality (DPI) only as a motivation for comparing observation spaces, rather than as a proof that direct spike-to-grasp prediction must be superior. Let S denote the raw stereo spike stream, $Z = R(S)$ an intermediate representation produced by a reconstruction module, such as a frame, depth map, or point cloud, and G the grasp variable of interest, including scene-level graspness and 6-DoF grasp quality. A reconstruct-then-grasp pipeline can be written as

$$S \xrightarrow{R} Z \xrightarrow{h} \hat{G}, \tag{27}$$

whereas our model learns a direct mapping from S to $\hat{G}$. Since Z is obtained from S, DPI gives

$$I(G; Z) \le I(G; S). \tag{28}$$

However, this inequality should be interpreted only as a motivation. It indicates that reconstruction cannot introduce additional information about G beyond what is already present in S, but it does not imply that reconstruction necessarily harms task performance. If Z is sufficiently informative for grasp prediction, then a reconstruct-then-grasp pipeline may still perform well. Therefore, rather than treating DPI as a theoretical proof, we evaluate task-level sufficiency empirically through controlled probing experiments.

We consider four observation spaces: reconstructed frames, reconstructed depth/point clouds, initial spike features before recurrent refinement, and the final recurrent latent state of SpikeGrasp. For each observation space, we freeze the corresponding upstream encoder or reconstruction module and train the same lightweight prediction probe to estimate scene-level graspness and 6-DoF grasp scores. The probe architecture, training schedule, and supervision are kept identical across all observation spaces. This design controls for downstream predictor capacity and isolates the effect of the observation space itself.

To quantify task-level sufficiency, let Z denote one candidate observation space and let $H^\star$ denote the final recurrent latent representation of SpikeGrasp. We define the probing performance on observation Z as

$$\mathrm{AP}_{\mathrm{probe}}(Z) = \mathrm{AP}\big(g_{\phi_Z}(Z)\big), \tag{29}$$

where g_{ϕ_Z} is the probe trained under the shared probing protocol. We then define an empirical sufficiency gap relative to the SpikeGrasp latent representation:

$$\Delta_{\mathrm{AP}}(Z) = \mathrm{AP}_{\mathrm{probe}}(H^\star) - \mathrm{AP}_{\mathrm{probe}}(Z), \tag{30}$$

and its normalized version as

$$\Delta_{\mathrm{norm}}(Z) = \frac{\mathrm{AP}_{\mathrm{probe}}(H^\star) - \mathrm{AP}_{\mathrm{probe}}(Z)}{\mathrm{AP}_{\mathrm{probe}}(H^\star)}. \tag{31}$$

Table 7: Observation-space sufficiency gap on SpikeGrasp-1B. We freeze each representation and train the same lightweight prediction probe. Δ_{suf} is computed with respect to the full SpikeGrasp model.

Representation	Grasp AP			Δ_{suf}		
	Seen	Similar	Novel	Seen	Similar	Novel
Reconstructed frame	35.70	22.42	17.81	35.75	43.43	18.61
Frame + spike	40.73	26.42	19.01	30.72	39.43	17.41
Reconstructed point-cloud	44.98	30.70	20.63	26.47	35.15	15.79
Initial spike features	47.13	31.28	23.51	24.32	34.57	12.91
SpikeGrasp (default)	**71.45**	**65.85**	**36.42**	**0.00**	**0.00**	**0.00**

Table 8: Normalized sufficiency gap $\overline{\Delta}_{\mathrm{suf}}$ on SpikeGrasp-1B. Values indicate the relative AP loss compared with the full SpikeGrasp model.

Representation	Seen	Similar	Novel
Reconstructed frame (stereo)	50.03%	65.95%	51.10%
Frame + spike	43.00%	59.88%	47.80%
Reconstructed point-cloud	37.05%	53.38%	43.36%
Initial spike features	34.04%	52.50%	35.45%
SpikeGrasp (default)	**0.00%**	**0.00%**	**0.00%**

By construction, $\Delta_{\mathrm{AP}}(H^\star) = 0$. For this reason, the gap should be interpreted as a task-level diagnostic relative to a chosen reference representation, rather than as independent theoretical evidence. We therefore report both the absolute probing AP and the relative gap.

Table 7 summarizes the results, and Table 8 reports the normalized sufficiency gap. The final recurrent latent state of SpikeGrasp achieves the highest probing performance across the Seen, Similar, and Novel splits. The advantage becomes larger on the Similar and Novel subsets, suggesting that the spike-native recurrent representation preserves task-relevant cues more effectively under distribution shift. Reconstructed depth and point-cloud observations remain competitive on Seen objects, but degrade more noticeably on the harder subsets. These findings are consistent with the hypothesis that, under asynchronous sensing, explicit reconstruction can act as an observation bottleneck for grasp prediction. At the same time, they should be interpreted as empirical evidence from controlled probes rather than as a formal proof that direct prediction is always optimal. Overall, these results support the view that explicit reconstruction can become an observation bottleneck under asynchronous sensing, while also highlighting that this conclusion is based on empirical probing rather than theoretical guarantee.

A.3.2 Sim-to-real spike simulation experiment

Our simulator does not generate spikes from tone-mapped or reconstructed LDR frames. It applies an integrate-and-fire model directly to the original left and right HDR radiance. For each pixel, the photon response is accumulated until the firing threshold is reached, after which a spike is emitted and the residual charge is retained. Photon accumulation, firing thresholds, and emission times are perturbed to model diffuse-reflection noise, sensor variability, and temporal jitter.

To directly examine the sim to sensor consistency, we additionally compared simulated streams with real streams captured by the real stereo spike-camera rig under matched normal, low-light, and HDR conditions. Following the conventional validation protocol (Zhao et al., 2022), we evaluated the pixel-wise firing rate similarity ($M_1 \in [0, 1]$) and the Kullback–Leibler divergence ($M_2 \in [0, +\infty]$) between real and simulated firing distributions. The left and right streams were evaluated separately and averaged over the test scenes.

The high firing-rate similarity and low distribution divergence support the statistical consistency of the simulated and real spike streams across temporal windows. The task-level evidence is provided

Table 9: Simulation to real spike dataset experiment.

Temporal window (ms)	$M_1 \uparrow$	$M_2 \downarrow$
2.5	0.911	0.092
5.0	0.932	0.084
7.5	0.956	0.070
10.0	0.972	0.056

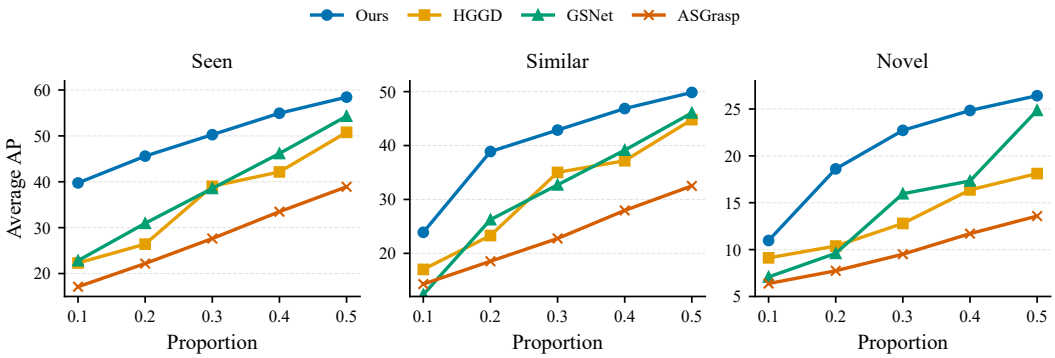


Figure 6: Data efficiency experiments on the SpikeGrasp-1B synthetic dataset. The horizontal axis indicates the proportion of the training set, while the vertical axes represent the value of AP.

by Appendix Table 11, where performance degrades gradually under increasing spike dropout, temporal jitter, and HDR perturbations, rather than depending on a single nominal simulation configuration. Moreover, after training on the synthetic benchmark, SpikeGrasp achieves an overall grasp success/declutter rate of 85.0/82.5% in multiple physical robot trials, including 83.3/81.7% in real HDR scenes. Together, these results suggest that the remaining discrepancy is small for the measured firing statistics and do not prevent effective task-level transfer. We therefore characterize the simulator as a physics-informed and empirically validated approximation, while interpreting Table 1 as controlled synthetic evidence complemented by the real-world results in Table 4.

A.3.3 Data Efficiency

To assess data efficiency, we trained our model and all baselines in randomly sampled subsets of the synthetic training set, ranging from 50% down to 10% in 10% increments, while evaluating on the full test set for fair comparison. As shown in Fig. 6, our method consistently exceeds all baselines at every training ratio, with margins widening as data become scarcer. This indicates a stronger generalization under limited supervision and highlights the robustness and data efficiency of the learned representations.

A.3.4 Scene-level Grasp

We present a complete 6-DoF grasp pose sequence for a single scene within the synthetic dataset, capturing the full motion trajectory of objects throughout the dynamic environment. The complete scene is captured by the camera system rotating around the sphere, and the multi-view scene lays the foundation for subsequent dynamic grasping. We selected 12 of 256 views in a scene as illustrated in Fig. 7, the predicted 6-DoF grasp sequences effectively reconstruct the continuous evolution of the scene, accurately reflecting both spatial structure and temporal consistency. These results demonstrate the model's ability to preserve coherent grasp across time, validating its effectiveness in handling visual inputs from different views. The supplementary materials in the zip file include continuously changing videos.

In addition, we conducted a qualitative comparison between SpikeGrasp and several advanced methods on the synthetic dataset. Our method consistently produces more accurate and visually coherent grasp poses in various scenes. Further qualitative results are provided to demonstrate the robustness and effectiveness of our approach.

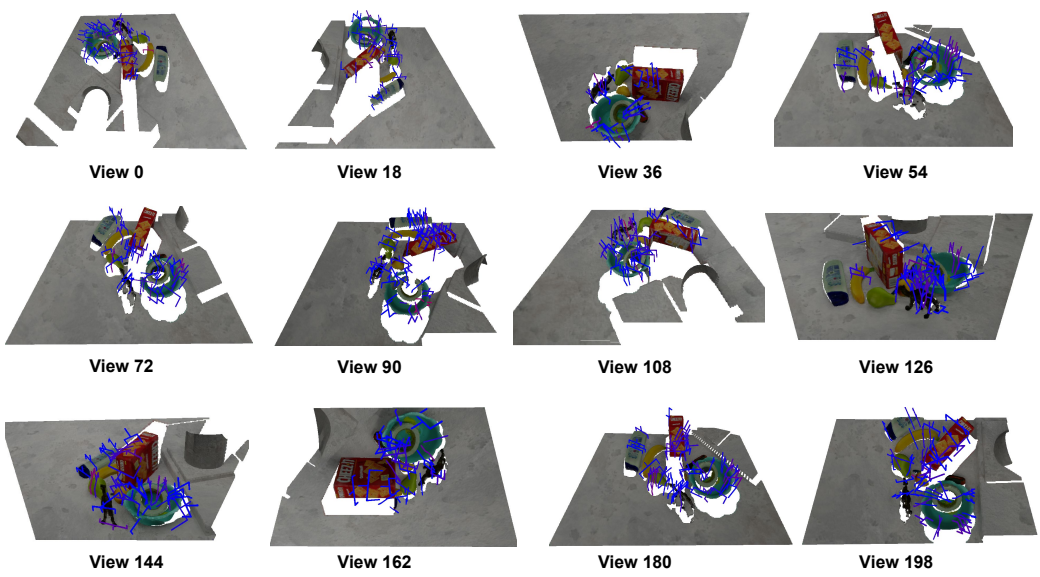


Figure 7: Visualization results of our method on the synthetic dataset in the single scene. We presented grasp poses predicted of 12 views in the scene.

A.3.5 More Grasp Results

To visually illustrate the performance of our proposed SpikeGrasp algorithm in grasp pose detection. Figure 8 provides additional qualitative results of SpikeGrasp on the synthetic SpikeGrasp-1B benchmark. The visualized scenes cover the Seen, Similar, and Novel subsets, as well as cluttered layouts with different object arrangements and levels of occlusion. Across these diverse cases, SpikeGrasp consistently predicts grasp poses that lie on object surfaces and concentrate around physically plausible contact regions. These results suggest that the learned spike-native representation generalizes beyond a small set of selected examples and remains stable across different object identities, viewpoints, and scene configurations. Compared with the limited qualitative examples in the main paper, this extended visualization is intended to show the breadth of the benchmark and the consistency of the proposed method. We visualize only the top-50 grasps for readability to avoid occluding the objects.

A.3.6 Robustness and Stress Experiments

We further evaluate SpikeGrasp under controlled perturbations to understand its robustness to sensing noise and scene difficulty. We consider five stress factors: random spike dropout, temporal jitter, stereo calibration perturbation, illumination shift, and clutter level. For each factor, we gradually increase the perturbation strength and evaluate AP on the validation split.

Table 10 summarizes the results. SpikeGrasp remains stable under moderate spike dropout and temporal jitter, suggesting that recurrent refinement can aggregate redundant temporal evidence from the spike stream. The largest degradation occurs under severe stereo misalignment and heavy occlusion, which directly affect cross-view correspondence.

A.3.7 More Ablation Results

In this section, we provide additional ablation results beyond the main paper. Unless otherwise specified, all variants are trained on the synthetic SpikeGrasp-1B benchmark using the same optimizer, learning schedule, and evaluation protocol as the full model. We report average precision (AP) on the Seen, Similar, and Novel subsets.

Effect of Recurrent Refinement Steps. We first study the effect of the number of recurrent refinement steps. Intuitively, the refinement module progressively integrates binocular matching cues

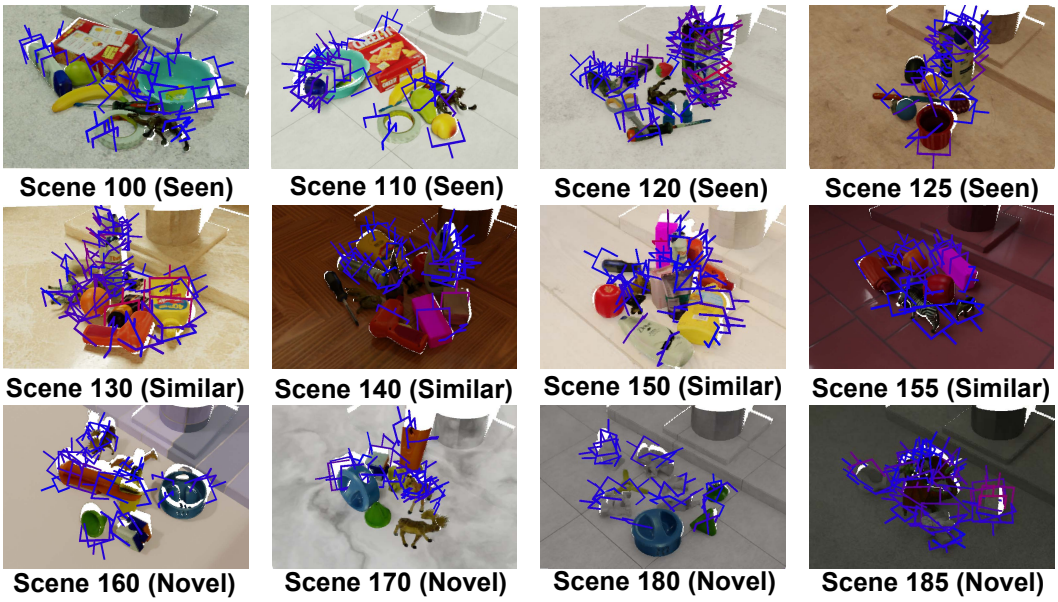


Figure 8: Additional qualitative results of SpikeGrasp on the SpikeGrasp-1B benchmark. We show representative grasp predictions across diverse synthetic scenes, including Seen, Similar, and Novel objects, as well as cluttered and visually challenging settings. For each scene, we visualize the top-50 grasp predictions of SpikeGrasp. These examples illustrate the diversity of the benchmark and the consistency of the proposed method across different scene conditions.

Table 10: Robustness under synthetic stress factors. we report the AP scores on Seen subset. AP_{Lx} refers to the AP at different disturbance levels.

Stress factor	AP_{L0}	AP_{L1}	AP_{L2}	AP_{L3}
Spike dropout	71.45	69.26	65.42	59.04
Temporal jitter	71.45	69.81	66.62	60.17
Stereo misalignment	71.45	65.17	57.13	46.44
HDR shift	71.45	68.41	64.43	56.59
Clutter level	71.45	68.02	58.67	50.37

and temporal evidence, so increasing the number of iterations should improve grasp prediction up to a certain point. To verify this, we vary the number of recurrent update steps $K \in \{1, 2, 4, 8, 16, 32\}$ while keeping all other components fixed.

Table 11 shows that performance improves consistently as the number of refinement steps increases. The largest gain occurs when increasing K from 1 to 16, indicating that iterative hidden state updates are important for recovering grasp-relevant structure from asynchronous observations. The improvement becomes smaller from 16 to 32 steps, suggesting that the benefit of additional refinement gradually saturates. This trend supports the design choice of using iterative recurrent updates in the full model.

Effect of Temporal Window Length. We next analyze the effect of the temporal window length of the stereo spike stream. Since the model operates directly on asynchronous observations, the input window controls the amount of temporal evidence available to the network. A very short window may fail to provide sufficient event support, while an overly long window may introduce redundant or less informative spikes.

We evaluate four input lengths, denoted by $N_{\text{read}} \in \{10, 30, 50, 70\}$, while fixing the rest of the architecture and training setup. The results are summarized in Table 12. We observe that larger input windows generally improve performance, especially from short to medium-length windows, confirming that direct spike-to-grasp inference benefits from richer temporal accumulation. However,

Table 11: Effect of the number of recurrent refinement steps K. Higher is better.

Refinement steps K	Seen	Similar	Novel
1	55.60	52.62	23.09
2	59.27	55.02	27.95
4	63.09	59.11	29.36
8	68.57	61.37	33.74
16 (default)	71.45	65.85	36.42
32	**72.44**	**67.43**	**37.30**

the gains become marginal once the window is sufficiently long, which suggests that the model already captures most useful evidence at moderate temporal coverage.

These results provide further evidence that the advantage of SpikeGrasp does not come solely from its spatial architecture. Instead, the model genuinely benefits from the temporal structure of the raw spike observation stream.

Table 12: Effect of input-window length. N_{read} counts consecutive binary readouts per camera in one synchronized stereo window. At 20 kHz, $W_{\mathrm{input}} = N_{\mathrm{read}}/f_{\mathrm{cam}}$. The refinement count is fixed at $K = 16$.

N_{read}	W_{input} (ms)	Seen AP	Similar AP	Novel AP
10	0.5	58.46	55.18	25.81
30	1.5	65.88	61.29	31.75
50	2.5	71.45	65.85	36.42
70	3.5	**71.66**	**66.24**	**36.94**

Failure Case Analysis. To better understand the remaining errors of SpikeGrasp, we manually inspect failure cases on the synthetic benchmark. We find that most failures fall into three recurring categories. First, the model is more likely to fail in heavily occluded scenes, where cross-view correspondence becomes unreliable and graspable regions may be only partially visible. In these cases, the predicted top-ranked grasps may drift toward visible but suboptimal contact regions. Second, errors often occur in weak response regions with sparse spike activity. Since the model operates directly on asynchronous spike streams, insufficient event support can make both stereo matching and grasp refinement less stable. This effect is particularly evident for objects with low apparent motion or low local contrast. Third, failures also appear for objects with locally ambiguous grasp geometry, where multiple nearby contact regions are all plausible from the available signal. In such cases, the model may still predict a feasible grasp-like pose, but its ranking of top candidates becomes less reliable.

A.3.8 Simulation Evaluation

Before conducting real-world experiments, We conducted simulated grasping evaluations in NVIDIA Isaac Sim with a Franka Emika Panda and a Robotiq 2F-85 gripper. We construct 15 tabletop scenes, each populated with 6–8 objects instantiated from the CAD assets used to build our synthetic dataset, and randomly assign lighting conditions and object materials to the scenes. For each object, the policy predicts a 6-DoF end-effector pose in the spike camera signal sequences. We transform this pose to the world frame using known extrinsics, generate a pre-grasp by offsetting 10 cm along the approach axis, and execute a four-stage sequence: pre-grasp, approach, closure, and lift. Motion is performed via inverse kinematics with collision checks, and joint targets are tracked by Isaac's articulation controller at 120 Hz.

We allow up to three attempts per object. A trial is successful if the object is lifted by 10 cm and held stably for one second without slip or secondary contact. We report the success rates as the fraction of objects successfully grasped across all scenes. Trained only on synthetic data, our model achieves 96.3% on Seen objects, 90.8% on Similar objects, and 85.9% on Novel objects. These results show a strong transfer from synthetic training to simulation and support the robustness of our approach.

A.3.9 SELF-ABLATIONS OF INPUT, REPRESENTATION, AND TRAINING

We use variants of our own model to distinguish input configuration, prediction interface, training objective, and model components. Table 13 contains a core design: stereo spikes or stereo intensity, predicted point clouds or latent features, and frozen, grasp-only, or jointly supervised training. Additional variants evaluate depth-map and feature-augmented point-cloud interfaces, the auxiliary-depth gradient, and the components ablated in Table 2. All variants use the grasp-decoder architecture H, and each variant fits its own decoder weights.

S: identical stereo spikes; I: native simulated stereo intensity. F: frozen depth-pretrained backbone; G: grasp training; J: grasp plus auxiliary depth training; J-stop: auxiliary-depth gradients stopped at the backbone. G-obj, G-grn and G-both remove the indicated supervision branches. The RNN-update variant retains spike input.

Table 13: Self-ablation planning matrix on SpikeGrasp-1B. All variants use the same grasp-decoder architecture H. Higher AP is better.

ID	Input	representation	Train	Change	Seen	Similar	Novel
A. input, representation and training							
A01	S	Point cloud	F	Full	57.14	45.57	24.33
A02	S	Latent	F	Full	60.17	48.27	26.29
A03	S	Point cloud	G	Full	62.11	53.66	23.92
A04 (default)	S	Latent	G	Full	**71.45**	**65.85**	**36.42**
A05	S	Point cloud	J	Full	61.23	51.84	27.09
A06	S	Latent	J	Full	64.82	55.19	30.80
A07	I	Point cloud	G	Full	65.43	52.88	29.79
A8	I	Latent	G	Full	66.15	60.07	32.14
B. Representation and gradient-path controls							
B01	S	Depth map	G	Full	62.41	52.91	31.27
B02	S	Point cloud + latent	G	Full	64.01	55.31	26.62
C. Reported component ablations							
C01	S	Latent	G-obj	No Objectness	36.14	34.41	15.54
C02	S	Latent	G-grn	No Graspness	44.78	40.86	17.28
C03	S	Latent	G-both	No Both	25.85	22.08	14.43
C04	S	Latent	G	RNN update	43.55	38.08	21.60
C05	S	Latent	G	LIF update	58.28	50.87	30.60
C06	S	Latent	G	No correlation	51.33	44.88	27.31
C07	S	Latent	G	No refinement	41.67	36.43	19.76

Table 13 distinguishes sensing and prediction-pathway contributions through matched variants of our model. In the illustrative grasp-supervised comparisons, switching from intensity to spike input increases Novel AP by 3.20 with the point-cloud interface and 4.32 with the latent interface (13.5% relative for the latter). With spike input and supervision fixed, latent prediction adds 2.35/2.25/1.52 AP on Seen/Similar/Novel (4.4% relative on Novel), and the corresponding margins under joint supervision are 1.22/1.39/0.70 AP. The cross-input contrasts concern simulated sensing configurations and their necessary front ends, whereas the within-input contrasts assess the prediction pathway.

Training and component ablations examine how the model uses these observations. The reported Novel AP is 36.42 for the default model, 26.29 with a frozen backbone, and 30.80 with auxiliary depth supervision, and the illustrative stop-gradient control reaches 36.00. These comparisons examine task adaptation and sensitivity to the auxiliary objective. With spike input unchanged, replacing the RSNN update, removing stereo correlation, or disabling refinement reduces reported Novel AP to 21.60, 27.31, and 19.76, respectively. Objectness, graspness, and adaptive-neuron ablations also reduce AP.

Importantly, the spike sensor determines the input modality but does not prescribe a single downstream representation. Given the same spike observations, we compare reconstructed frames or point clouds, generic spike features, and our task-specific latent representation. These controlled comparisons provide direct evidence that the proposed representation contributes additional gains beyond those provided by the sensing modality itself.

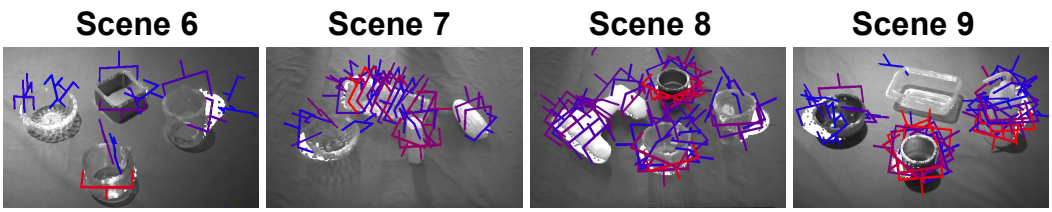


Figure 9: The more real-world qualitative results of SpikeGrasp.

Overall, the matched comparisons support complementary contributions from the sensing configuration, direct latent prediction, and task-oriented model design and training. The additional gains under identical spike inputs indicate that the system's performance cannot be attributed solely to the sensing modality. Within the evaluated grasp-prediction setting, these results support the integrated design of sensing, representation learning, and grasp inference, while demonstrating that explicit point-cloud reconstruction is not a necessary intermediate step for prediction.

A.3.10 More Real-World Robot Experiments

Figure 9 presents more real-world qualitative results of SpikeGrasp. We visualize representative cases from isolated, moderate-clutter, heavy-clutter, and high-dynamic-range scenes. Each scenes underwent 60 experiments, resulting in a total of 240 real-world graspe xperiments.

The point-cloud baseline is sensitive to missing or unstable geometry, especially in cluttered and HDR scenes where depth reconstruction may be incomplete or noisy. The multi-view method can alleviate some of these issues through view aggregation, but it may still fail when the available views are ambiguous or when implicit reconstruction does not preserve fine grasp-critical structure. By contrast, SpikeGrasp more consistently predicts object-centric and executable grasp poses from the stereo spike observations. This qualitative behavior is consistent with the real-world success and declutter rates reported in Table 4, and further supports the advantage of direct inference in the spike observation space under challenging real-world sensing conditions.

A.3.11 Real End-to-end latency Experiments

We measured batch size 1 latency on the same RTX 4090 for low-latency capturing experiments. "Network" is the model forward pass (for SpikeGrasp, includes all 16 recurrent refinement updates); "End-to-end" is wall-clock time from sensing start to the final ranked grasp pose, including transfer, preprocessing, reconstruction where applicable, inference, and ranking/NMS; "Sensing" latency is measured from the beginning of the observation interval until the corresponding spike or depth data become available to the host. The remaining time is dominated by host transfer, proposal decoding, collision filtering, and ranking/NMS. All required temporal context, reconstruction, transfers, inference, and post-processing are included in the appropriate interval. The specific experimental results are shown in Table 14.

Table 14: End to end time latency experiments in real-world scenes.

Pipeline	Sensing (ms)	Network Inference (ms) nd-to-end (ms)	
RGB-D → GraspNet	50	82.5	510.5
RGB-D → ZeroGrasp	50	202.2	740.8
Spike camera → reconstruction → grasp	2.5	80.2	478.2
Spike camera → SpikeGrasp	2.5	64.9	**440.4**

Thus, SpikeGrasp reduces wall-clock latency by 7.9% relative to the fastest RGB-D baseline and by 40.5% relative to the better performance pipeline. We additionally report the accuracy-latency trade-off over recurrent updates as Table 15.

Although the additional iterations K slightly increased inference time, AP performance steadily improved. These measurements show that the latency benefit comes from both fast spike acquisition and avoiding explicit reconstruction, despite recurrent refinement.

Table 15: Refinement iterations under a fixed input window. Each row uses one 50-readout stereo window.

Refinement K	Seen AP	Similar AP	Novel AP	Network (ms)	Total (ms)
1	55.60	52.62	23.09	34.3	405.8
4	63.09	59.11	29.36	40.2	415.7
8	68.57	61.37	33.74	47.7	426.3
16	71.45	65.85	36.42	64.9	440.4

A.4 Reproducibility and Implementation Details

Our model is implemented with PyTorch and trained on Nvidia RTX 4090 GPUs for 18 epochs with Adam optimizer and the batch size of 4. The learning rate is 2×10^{-4} in the first epoch and multiplied by 0.95 every epoch. During training, we apply 16 update iterations for each sample. In inference, we use one GPU for fast prediction.

Table 16 summarizes the training hyperparameters used in our experiments. Unless otherwise specified, all methods are trained using the same data split and evaluated with the same metric implementation. We will release anonymous code and instructions for reproducing the main experiments. The typical workflow consists of dataset preparation, model training, and evaluation.

Table 16: Training hyperparameters of SpikeGrasp.

Hyperparameter	Value
Optimizer	Adam
Initial learning rate	2×10^{-4}
Learning rate decay	0.95 per epoch
Batch size	4
Training epochs	18
Recurrent update steps	16
Surrogate gradient	$\sigma(x) = \frac{1}{1+\exp(-4x)}$
Loss weight	$\alpha = 10$, $\beta = 15$, $\gamma = 10$, $\lambda = 5$
GPU	NVIDIA RTX 4090
Framework	PyTorch

Reproducibility Statement. To support reproducibility and future research, we plan to publicly release the full implementation of SpikeGrasp, including model code, training and evaluation scripts, configuration files, dataset generation tools, and the SpikeGrasp-1B benchmark with its train/test splits and evaluation protocol. We will also provide documentation for data format, preprocessing, baseline adaptation, and metric computation, so that future work can reproduce our main results and extend the benchmark to other spike-based embodied perception tasks.

A.5 Broader Impacts

This work explores direct 6-DoF grasp detection from raw stereo spike streams, and may contribute to the broader development of neuromorphic perception for embodied intelligence. By avoiding explicit depth or point-cloud reconstruction during inference, SpikeGrasp provides an alternative sensing-to-action pathway for robotic manipulation, especially in scenarios where conventional visual pipelines can be affected by high-dynamic-range illumination, motion blur, sparse texture, or unstable depth estimation. In robotics, the proposed approach may support more robust perception for grasping, manipulation, and other low-latency embodied tasks. The accompanying SpikeGrasp-1B benchmark may also facilitate future research on spike-based visual representation learning, asynchronous stereo perception, and sim-to-real transfer for manipulation. More broadly, this study suggests that raw spike streams can serve not only as inputs for image reconstruction or low-level vision, but also as native observations for action-oriented scene understanding. At the same time, the current system is still limited by synthetic training data, domain shift to real spike cameras, and evaluation on a restricted

set of manipulation scenarios. Before deployment in safety-critical or industrial settings, future work should further validate the method across diverse hardware platforms, object categories, lighting conditions, and closed-loop robot control settings. Care should also be taken to ensure reliable failure detection and safe robot execution when operating in cluttered or human-shared environments.