Toward Future Spatial Representations for Manipulation Robots: A Survey

Hao Chen, Yuzhen Chen, Yuqing Luo

***Abstract*—In high–frequency closed loops that fuse multi-view RGB-D and tactile observations, efficiently *maintaining, querying, and dynamically updating* three-dimensional spatial representations under tight GPU memory budgets has become the central bottleneck for robotic manipulation in open environments. Existing *explicit* methods—voxels, meshes, point clouds, and distance fields—and *neural implicit* fields—Neural SDF, NeRF, and 3-D Gaussian Splatting—each excel in certain aspects, yet few representations jointly satisfy the control-critical requirements of memory efficiency, gradient availability, uncertainty expression, dynamic maintainability, and millisecond-level generation speed for large-scale, highly dynamic scenes. From a control-oriented viewpoint, this paper proposes a quantitative evaluation framework, systematically surveys and compares the contributions and costs of both representation families in real closed-loop settings, and employs visualization experiments to unveil the latent geometric structures embedded in end-to-end policies. The analysis distills open problems such as unified uncertainty modeling and global high-frequency updates. Finally, we outline a hybrid geometric–semantic roadmap—*explicit local caching plus a global neural field*—intended to serve as a design guideline for the next generation of general-purpose manipulation systems.**

I. Introduction

As robotic systems transition from constrained assembly lines to open environments such as homes and public service areas, the diversity of manipulated objects—in terms of geometry and physical properties—has expanded dramatically. Transparent glassware, deformable fabrics, semi-fluid ingredients, and daily items with intricate mechanical structures are now commonplace. To maintain robust, safe, and real-time action execution at high closed-loop frequencies from multi-modal observations, including multi-view RGB-D and tactile inputs, the bottleneck of robotic manipulation has shifted from the fundamental challenge of scene reconstruction to the efficient maintenance, querying, and dynamic updating of three-dimensional representations within tight GPU memory constraints. Currently, research efforts evaluate representations in isolation, emphasizing reconstruction errors, occupancy accuracy, or task-specific success rates independently, resulting in fragmented evaluation criteria. Meanwhile, new control methodologies such as trajectory optimization and diffusion-based policies impose stricter demands on representations—necessitating differentiability, confidence modeling, and high-frequency updates—highlighting an increasing gap between traditional geometric representations and emerging control paradigms.

This paper explicitly focuses on robotic manipulation, reassessing the strengths and limitations of 3D spatial representations through a control-oriented lens. Our contributions include:

- Introducing a quantitative evaluation framework from a control perspective, using unified metrics to systematically benchmark current geometric representations based on their practical contributions and computational costs within closed-loop robotic systems.
- Providing a comprehensive analysis of both explicit and neural implicit spatial representations—including voxels, meshes, point clouds, signed distance fields (SDF), neural SDFs, Neural Radiance Fields (NeRF), and 3D Gaussian Splatting (3DGS)—and employing visualization experiments to reveal potential internal spatial structures emerging within end-to-end learned policies.
- Identifying open problems and proposing hybrid design principles that address critical gaps such as global high-frequency updates and confidence modeling.

The remainder of this article is organized as follows: Section II elaborates the theoretical requirements of spatial representations in manipulation tasks. Section III reviews and quantitatively benchmarks explicit representations. Section IV discusses neural implicit fields in the context of robotic control and explores latent spatial structures inherent in end-to-end policies. Section V synthesizes existing limitations and outlines promising directions for future research. Finally, Section VI summarizes our key findings and contributions.

II. Preliminary

In typical robotic manipulation tasks, robot-environment interactions can be formally described as a Partially Observable Markov Decision Process (POMDP). The latent three-dimensional state x_t evolves due to robot actions u_t and is partially revealed through subsequent observations z_{t+1}. The belief distribution, given by

$$b_{t+1} = \eta\, P(z_{t+1} \mid x_{t+1}) \sum_{x_t} P(x_{t+1} \mid x_t, u_t)\, b_t,$$

where η is a normalization constant, captures all the robot's uncertainty regarding the geometric and physical aspects of the environment. Consequently, the optimal control policy depends explicitly on the belief rather than raw observations, formulated as $u_t = \pi(b_t)$.

Let Φ_t denote the internal 3-D representation at time t. From an information-theoretic perspective, robotic manipulation imposes five core requirements on Φ_t. First, **memory efficiency**: extensive unused representations significantly waste GPU memory without contributing to mutual information.

Thus, Φ_t demands sparsification, hierarchical structures, or adaptive resolution to maximize compression ratio R without sacrificing control-relevant mutual information. Second, **differentiability**: optimization-based planners frequently leverage gradients such as $\nabla d(x)$ or $\partial J/\partial \Phi$; non-differentiable or discretely queried representations disrupt gradient flow, effectively reducing Fisher information and degrading convergence rate and performance. Third, **uncertainty quantification**: risk-sensitive control explicitly requires collision probability or trajectory variance; a deterministic geometry without covariance systematically underestimates risk, violating chance constraints. Lastly, **dynamic compatibility** and **generation/query speed**: the perception-control pipeline acts as a bandwidth-limited communication channel, and additional latency τ introduces noise, reducing effective signal-to-noise ratio. Achieving a closed-loop high rate requires representation construction, rendering, and gradient queries to occur within milliseconds.

These five theoretical constraints, compression ratio, gradient informativeness, risk estimation, real-time responsiveness, and channel capacity, jointly define the design boundaries for high-performance 3D representations. The subsequent sections will quantitatively benchmark explicit geometric and neural implicit representations along these dimensions, motivating the development of hybrid representation architectures.

III. Explicit Representations

In manipulation pipelines, explicit 3D representations can be viewed as discretized belief approximations: we project the probabilistic belief about the real scene onto a set of explicit geometric primitives (e.g., voxels, surfels, or point clouds), subsequently performing queries, collision checks, and path searches directly on these primitives. Such a structure is inherently easy to visualize and provide safety guarantee, and conveniently interfaces with classical motion planning algorithms.

Explicit 3D representations have evolved significantly, branching into multiple parallel research directions. Initially, voxel-based occupancy grids dominated early 3D geometric representations. To extend these grids efficiently to larger scales within limited memory budgets, Hornung et al. proposed OctoMap, encapsulating voxels in an octree structure and reducing the complexity of occupancy probability queries dramatically [1]. While traditional occupancy grids only support binary collision checks through probabilistic queries $p(\text{occ})$, upgrading them to Signed Distance Fields (SDFs) allows direct access to spatial gradients $\nabla d(x)$, which provides continuous and differentiable safety constraints for trajectory optimization. Early methods such as CHOMP embedded precomputed SDF values into trajectory cost functions, enabling analytic collision gradients for optimization [2]. GPMP2 further integrated these gradients into factor graph frameworks, achieving real-time optimization of full-arm trajectories at millisecond scales [3]. To handle dynamic environments, FIESTA incrementally updates the Euclidean Signed Distance Field (ESDF) at 20 Hz, allowing MPC-based obstacle avoidance controllers to reactively leverage dynamic collision gradients [4]. When natural-language instructions must be incorporated, VoxPoser maps semantic information from Vision-Language Models (VLMs) onto voxel-based value functions, enabling simultaneous satisfaction of multiple complex directives through a unified 3D potential field [5]. Collectively, voxel and distance-field representations effectively translate explicit geometric structures into continuous, differentiable safety constraints.

Meshes and surfel representations provide continuous surfaces crucial for precise contact-intensive tasks, such as polishing, sanding, or insertion. The Flexible Collision Library (FCL) computes real-time distances efficiently by mesh discretization and hierarchical bounding volume representations [6]. Dex-Net reconstructs object shapes from TSDF data into triangular meshes, then trains CNNs using physics-informed grasping matrices, resulting in robust grasp policies resilient to shape deformation and perception noise [7]. MeshDMP directly encodes curvature from CAD models into Dynamic Movement Primitives (DMPs), enabling robotic end-effectors to smoothly traverse arbitrary free-form surfaces [8]. Continuous-density surfel methods, such as Surfel-SLAM, maintain centimeter-level accuracy for contact sensing over complex terrains, effectively supporting robust mobile robot navigation [9]. The primary advantage of these representations lies in their explicit collision checks and continuous surface normal estimation, which significantly enhances reliability in contact-rich manipulation.

Point clouds preserve raw geometric details captured by sensors without the discretization artifacts introduced by voxels or meshes, making them particularly suitable for end-to-end geometric feature learning. Classical methods such as GPD rely on explicit geometric heuristics to rank candidate grasps [10], while more recent approaches, such as PointNet-GPD, directly estimate grasp quality using PointNet on local point sets [11]. DexPoint feeds sparse, full-scene point clouds into reinforcement learning policies, successfully enabling zero-shot grasping generalization [12]. PartManip further uses annotated multi-view point clouds indicating target parts to generalize manipulation across categories of knobs and drawers [13]. A common characteristic of these approaches is their direct utilization of high-resolution point clouds as primary inputs to action-generation networks, thereby trading some global geometric consistency for detailed local feature fidelity.

When manipulation tasks span multiple rooms or numerous objects, scene graphs elevate geometry to relational structures suitable for logical reasoning and hierarchical abstraction. Kimera DSG was the first to combine SLAM-based geometry, topology, and semantics into multilayered dynamic scene graphs, supporting efficient environmental understanding and task planning [14]. CG+ encodes physical constraints like "support/on" relationships directly within scene graphs, simplifying task planning into graph editing operations [15]. Graph Neural Network (GNN)-based manipulation policies formulate grasping and stacking tasks as interactions among object nodes and relational edges, directly predicting the next

object to manipulate, thereby offering lightweight yet effective generalization [16]. RoboEXP actively explores and expands an Action-Conditioned Scene Graph to facilitate multi-step kitchen scene arrangements [17]. These graph-based representations allow control policies to perform decision-making and learning directly at the relational abstraction level, eliminating the need to revert to detailed geometric reasoning at each step.

IV. Implicit / Neural Representations

In the previous section, we discussed explicit representations such as voxel grids, triangle meshes, and signed distance fields (SDFs). While these methods offer rapid construction, intuitive rendering, constant-time queries, and are inherently suitable for collision detection and safety-margin verification, they struggle to simultaneously satisfy several critical demands relevant to robotic manipulation. These include achieving high resolution at large scene scales, maintaining consistency across multiple viewpoints and sensors, providing differentiable gradients, enabling real-time incremental updates, and supporting probabilistic and multimodal data fusion. To address these limitations, researchers have begun compressing scene geometry into continuously differentiable neural functions. One prominent line of research, exemplified by NeRF, 3DGS, and Neural SDFs, offers compressed, smooth, and globally differentiable 3D scene representations. Another approach emerges implicitly within end-to-end visuomotor policies, where neural networks internally learn latent spatial representations bridging multi-view perception and motor actions. Could explicitly inserting or actively manipulating this latent space accelerate convergence, enhance cross-view generalization, and facilitate the integration of physical priors? The remainder of this section will explore these two research directions in depth.

From traditional Signed Distance Fields (SDFs) to Neural SDFs, conventional voxel-based methods, such as TSDF and ESDF, directly provide surface normals and collision gradients. However, they encounter critical limitations, including memory consumption scaling cubically with voxel count, rigid spatial resolution, and difficulty encoding semantic or confidence information. Moreover, when objects move, voxel-based methods must recompute local distance fields entirely, severely compromising real-time performance. Neural SDF approaches address these issues by replacing voxel grids with multi-layer perceptrons (MLPs), using continuous latent vectors to encode shapes, thereby decoupling memory requirements from spatial resolution, and effectively integrating multi-view and multimodal information through a single back-propagation. For instance, iSDF achieves incremental updates of a SDF map about 30 Hz on GPUs, allowing grasp planners to directly utilize the distance function $d(x)$ and its gradient $\nabla d(x)$ to generate grasp candidates, which are then finalized by trajectory planners like RRT-Connect or MPPI [27]. NeuralFeels integrates tactile pixel measurements and depth images into a unified Neural SDF, updating gradients every 200 ms to modulate fingertip forces, thus forming a 5 Hz hybrid force-position closed-loop system capable of performing complex single-hand manipulation tasks [21]. Moreover, approaches like NDF and Relational NDF embed SE(3)-equivariant descriptors into SDF space, enabling gradient descent over descriptor energies to synchronize object-tool poses efficiently, successfully performing tasks such as hooking, insertion, and stacking without explicit 6-DoF searches [22], [23]. Across these systems, control strategies exploit Neural SDFs as differentiable cost functions, either explicitly integrating distance values and gradients into Model Predictive Control (MPC) constraints or embedding them into reinforcement learning states as continuous geometric channels.

In the domain of neural radiance fields, both NeRF and 3DGS methods rely on volumetric rendering integrals. NeRF employs an MLP-based network $f_\theta(x, d) \rightarrow (\sigma, c)$ to predict scene density σ and color c, while 3DGS explicitly accumulates Gaussian primitives $(\mu_i, \Sigma_i, \alpha_i, c_i)$, enabling rendering speeds exceeding 150 Hz. Robotic applications leveraging these techniques include iNeRF, which minimizes rendering residuals with gradient descent on 6-DoF poses, achieving sub-pixel visual-servoing precision [28]. Dex-NeRF and Evo-NeRF quickly reconstruct transparent objects and feed the rendered depths into grasp-quality networks like Dex-Net or self-supervised Q-networks, with Evo-NeRF incrementally updating its weights after each successful grasp, facilitating rapid sequential picking [24], [29]. GraspNeRF jointly trains a NeRF encoder and a 6-DoF grasping network, online generating grasp distributions directly from volumetrically rendered feature maps at 11 Hz [25]. Explicit GSplat-based methods, such as GS-SLAM, rapidly construct dense Gaussian maps at 8.43 Hz, which planners interpret as occupancy grids for ESDF updates, enabling mobile robot base controllers to perform obstacle avoidance at high update rate [30]. GaussianGrasper and GraspSplats directly sample grasp orientations from explicit Gaussian surface representations, eliminating traditional voxel or point-cloud projection steps [26], [31]. SparseGrasp dynamically modifies semantic Gaussian maps through "render-compare" loops after each grasp, enabling continuous grasp sequences and fast replanning [32]. Typically, neural radiance fields interface with control backends through either rendering-based re-projection—generating depth or feature images for conventional perception pipelines—or direct geometric/semantic queries, sampling surface points from explicit or implicit representations to estimate grasp feasibility and geometric gradients for torque control and trajectory optimization.

Differentiable geometric capabilities of neural fields significantly enhance precise robotic contact control. Specifically, normalized density gradients

$$n(x) = \frac{\nabla \sigma(x)}{|\nabla \sigma(x)|}$$

yield continuous surface normals and local nearest-distance information. Precision insertion systems directly incorporate these density gradients into energy formulations, using optimization methods like L-BFGS to refine gripper poses. Additionally, some studies embed density iso-surfaces ($\sigma < \tau$) into factor graphs of trajectory optimizers, such as CHOMP or

Representation Category	Representative Papers	MemEffcy	DiffAvail	Uncerty	DynCompat	G&Q Speed
Voxel / Occupancy Grid	SPARK/FLAME [18]	$\sim 40\%$	✗	✗	✗	5-10s / -
	VoxPoser [5]	$\sim 50\%$	✗	●	●	- / 200ms
	Continuous Mapping [9]	$\sim 70\%$	✗	✓	✓	- / 5s per scan
Mesh / Surfel Surface	FCL [6]	$\sim 50\%$	✗	✗	✓	- / 3.86ms
	MeshDMP [8]	$\sim 70\%$	✓	✗	●	N.E.S. / N.E.S.
	Surfel Mapping [9]	$\sim 90\%$	✗	✓	✓	- / 5s
	Shape Completion [19]	$\sim 40\%$	✗	✗	●	0.35 hours / 200ms
Point Cloud	GPD [10]	$\sim 40\%$	✗	✗	✗	N.E.S. /
	PointNetGPD [11]	$\sim 50\%$	✓	✗	●	2-3 hours / 10-20ms
	DexPoint [12]	$\sim 60\%$	✓	✗	●	2-3 hours / 10-20ms
	PartManip [13]	$\sim 60\%$	✓	✗	●	6 hours / 10-20ms
	Dex-Net 2.0 [7]	$\sim 60\%$	✓	✗	✗	48 hours / 800ms
	AnyGrasp [20]	$\sim 60\%$	✓	✗	✓	8 hours / 100ms
Distance Field	CHOMP [2]	$\sim 80\%$	✓	✗	●	5 s / 1-5 s
	GPMP2 [3]	$\sim 80\%$	✓	✗	●	20–50 ms / 30–70 ms
	FIESTA [4]	$\sim 75\%$	✓	✓	✓	25 ms / 3.5 ms
Scene / Contact Graph	Kimera DSG [14]	$\sim 70\%$	✓	✓	✗	- / 100ms
	CG+ [15]	$\sim 80\%$	✓	✓	✗	- / 1-2s
	GNN Policy [16]	$\sim 70\%$	✓	✗	✓	0.33 hours / 10ms
	RoboEXP [17]	$\sim 80\%$	✓	✓	✗	- / 200ms
Neural / Implicit Field	NeuralFeels [21]	$\sim 40\%$	✓	✓	✗	- / 200ms
	Neural Desc Fields [22]	$\sim 50\%$	✓	✗	✗	6-8 hours / 1-2s
	Relational NDF [23]	$\sim 40\%$	✓	✗	✗	1 hour / 1-2s
	Dex-NeRF [24]	$\sim 40\%$	✓	✗	✗	6 hours / 1-2s
	GraspNeRF [25]	$\sim 30\%$	✓	✗	✗	N.E.S. / 1-2s
	GraspSplats [26]	$\sim 50\%$	✓	✗	✗	N.E.S. / 1-2s

Table I: Task-oriented comparison of 3-D representations. Representative methods from six geometric categories—Voxel/Occupancy Grid, Mesh/Surfel Surface, Point Cloud, Distance Field, Scene/Contact Graph, and Neural/Implicit Field—are projected into a five-dimensional metric space defined by **memory efficiency** (ratio of actively used to stored data), **differentiability** (availability of analytic gradients), **uncertainty** (built-in probabilistic information), **dynamic compatibility** (ability to remain valid under rapid scene changes), and **generation/query speed** (left: training or reconstruction time; right: single-query latency). ✓ indicates strong support, ● partial or conditional support, and ✗ little or no support. The table reveals that explicit representations rarely combine high **memory efficiency** with full **differentiability**, whereas current implicit methods, despite excelling in those two metrics, still lack unified uncertainty modeling and robust global updates for highly dynamic scenes. **N.E.S** denotes Not Explicitly Stated.

GPMP2, providing direct analytical collision gradients. Highly efficient CUDA implementations have reduced σ-$\nabla\sigma$ query times to near real-time, enabling density-gradient-driven MPC controllers to operate reliably for contact-rich tasks.

Explicit or neural-field-based geometric representations perform exceptionally well in tasks involving rigid components and everyday objects with clear visual-geometric priors. However, these methods often struggle when confronted with complex objects lacking stable geometric cues or direct visual features, such as deformable cloth, transparent or reflective containers, and liquids. To overcome this limitation, researchers have shifted towards end-to-end visuomotor policies, which directly map raw sensory observations to control signals, bypassing explicit scene reconstruction altogether.

Since the pioneering work of Levine et al., who directly mapped raw visual images to joint torques for simple rigid-object manipulation tasks [33], end-to-end strategies have rapidly evolved from early single-task, single-modality behavior cloning approaches to general-purpose multi-task, multi-modal foundational models. Recent models such as BC-Z [34] and Diffusion Policy [35], despite still requiring robot-specific training, have demonstrated that with sufficiently large and diverse datasets, a single policy network can achieve impressive zero-shot generalization across dozens of tabletop manipulation skills. More advanced approaches, such as Gato [36] and RoboCat [37], have further unified visual observations, proprioception, textual instructions, and target images into a single Transformer model, enabling rapid few-shot transfer across drastically different robotic hardware and tasks. The most recent development, RT-2 [38], combines internet-scale vision-language pretraining with real-world action annotations, empowering mobile manipulators to perform complex open-vocabulary instructions through multimodal reasoning, such as "pick up the hardest fruit" or "place the object shaped like a cup into the blender."

Despite these notable advancements, the sensor-to-action pipelines of existing end-to-end models remain largely opaque "black boxes." It remains unclear whether large-scale visuo-

motor models internally construct explicit or implicit three-dimensional spatial representations. If such internal representations indeed exist, determining how to effectively reveal, interpret, and guide them constitutes an open research question. This inherent opacity significantly limits the interpretability, verifiability, and reliability of end-to-end models in precision manipulation tasks that demand rigorous spatiotemporal reasoning.

A. Visualising Attention in the Diffusion Policy

a) Motivation: Diffusion Policy maps a horizon of visual observations to an action sequence by *implicitly* unrolling a denoising process in a high-dimensional latent space. While this formulation is powerful, the resulting policy is essentially a *black box*: it is unclear *which* spatial cues in the input images drive the predicted actions. In the context of PushTImageEnv, this opacity raises several practical questions:

- Does the policy truly attend to the T-shaped block, whose motion is the task's primary objective?
- Does it additionally—or alternatively—rely on contextual features of the surrounding scene (e.g. table edges, shadows)?
- Or has the network failed to ground its decisions at all, producing actions from spurious or random visual signals?

Answering these questions first helps provide interpretability checks—if the policy ignores the target block, it will likely fail to generalize to new scene layouts. Second, analyzing the results can provide actionable feedback for model design and data collection. If we find the policy overly reliant on background textures, we can apply targeted data augmentation or architectural regularization accordingly. Thus, we insert Grad-CAM hooks into the visual encoder, backpropagating action-prediction gradients to the original pixel space for visualization. This experiment will enable us to verify whether the diffusion-based policy has learned semantically consistent visual alignment or if further intervention is required.

b) Hook layer: The vision encoder follows a ResNet-18 backbone. We attach a Grad-CAM hook to the second 3×3 convolution of the last residual block in `layer3`:

$$\begin{aligned}\mathcal{A} &= \texttt{layer3[-1].conv2} \in \mathbb{R}^{C\times H\times W},\\ &\text{with } C=512,\ H=W=7\end{aligned} \tag{1}$$

This layer preserves spatial layout while being the deepest feature map with the strongest task semantics.

c) Explanation target: Let the predicted action sequence be $\hat{\mathbf{a}} = [\hat{\mathbf{a}}_0, \dots, \hat{\mathbf{a}}_{H-1}] \in \mathbb{R}^{H\times d}$ with horizon H and action dimension d=2. We explain the *x–coordinate of the first step*

$$t \;=\; \hat{a}_{0,x} \in \mathbb{R}, \tag{2}$$

corresponding to `target=(0,1)` in the Captum interface.

d) Grad–CAM.: Given the stacked input observations $\mathbf{I} = \{I_{t-H_{\text{obs}}+1}, \dots, I_t\}$ and agent poses $\mathbf{p} = \{p_{t-H_{\text{obs}}+1}, \dots, p_t\}$, the vision encoder produces a feature map $F = [F^1, \dots, F^C]$. Channel weights are

$$\alpha_k \;=\; \frac{1}{H\,W}\sum_{i=1}^{H}\sum_{j=1}^{W}\frac{\partial t}{\partial F_{ij}^k}, \tag{3}$$

yielding the heat-map

$$L = \mathrm{ReLU}\Big(\sum_{k=1}^{C}\alpha_k F^k\Big) \in \mathbb{R}^{H\times W}. \tag{4}$$

We upsample L to 96×96 via bilinear interpolation and overlay it with the original RGB frame using a `jet` colour map.

e) Gradient flow through the diffusion process.: An action sequence is produced by reversing a T-step diffusion:

$$\mathbf{x}^{(T)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \qquad \mathbf{x}^{(\tau-1)} = \mathcal{D}_\theta\big(\mathbf{x}^{(\tau)}, \tau, \mathbf{o}\big), \;\; \tau = T, \dots, 1,$$

where the conditioning vector $\mathbf{o} = [F, \mathbf{p}]$ concatenates vision features F and agent poses $\mathbf{p}$.

Let the scalar explanation target be $t = \hat{a}_{0,x}$ and define the *step-wise gradient*

$$g^{(\tau)} \frac{\partial t}{\partial \mathbf{x}^{(\tau)}}, \qquad \tau = 0, \dots, T.$$

Applying the chain rule to every reverse step yields the compact recursion

$$g^{(\tau)} \;=\; g^{(\tau-1)}\,\underbrace{\frac{\partial \mathbf{x}^{(\tau-1)}}{\partial \mathbf{x}^{(\tau)}}}_{J_\tau}, \qquad J_\tau \in \mathbb{R}^{d\times d}.$$

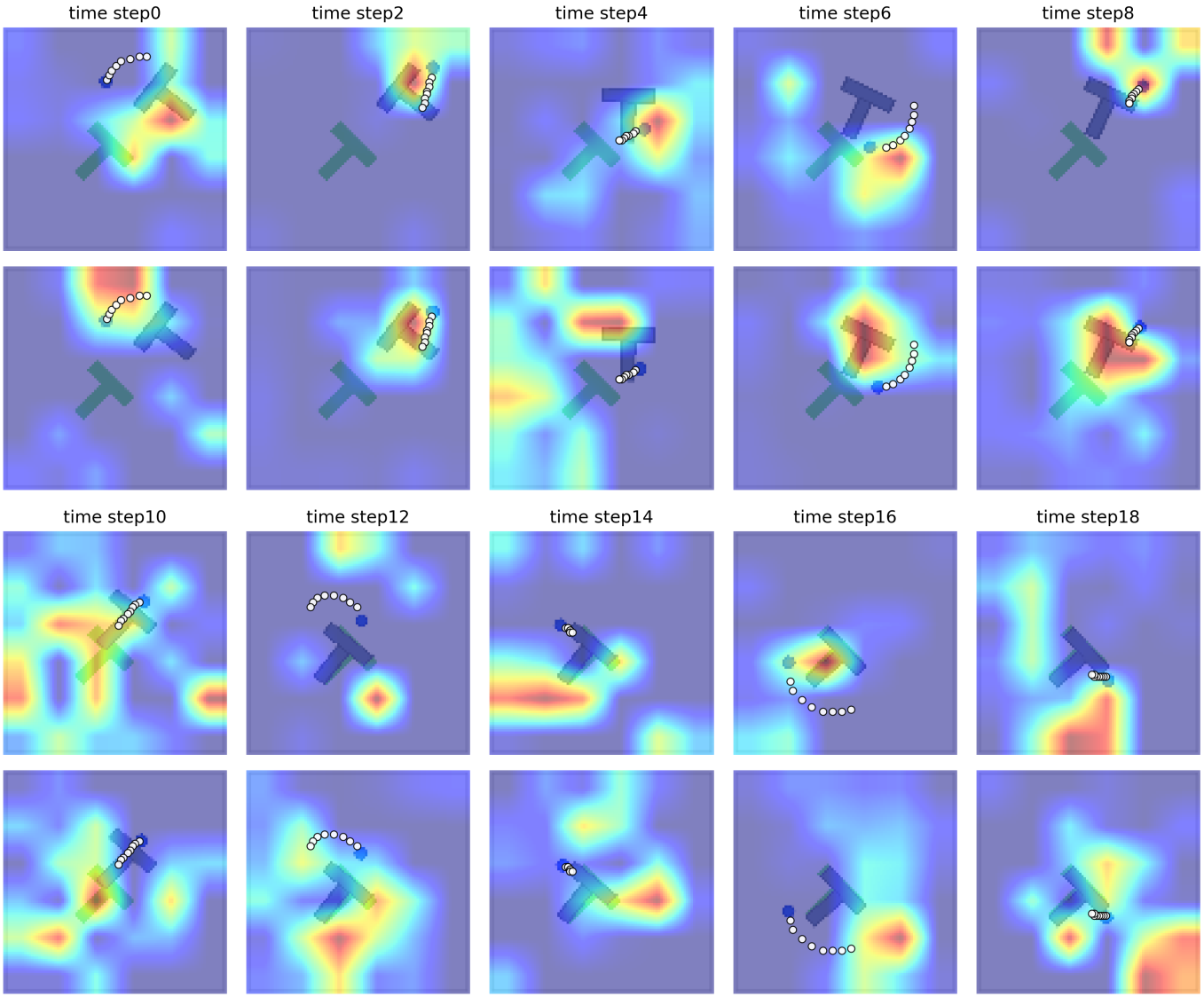
Hence the gradient required by Grad-CAM for the feature map F is

$$\frac{\partial t}{\partial F} \;=\; \sum_{\tau=1}^{T} g^{(\tau-1)}\frac{\partial \mathbf{x}^{(\tau-1)}}{\partial F}, \tag{5}$$

where the Jacobians $\partial \mathbf{x}^{(\tau-1)}/\partial F$ arise from the explicit dependence of $\mathcal{D}_\theta$ on the conditioning vector $\mathbf{o}$.

We observed each column corresponds to one roll-out of our policy. Circular white markers depict the *16 future agent positions* that are produced in a single forward pass of the diffusion planner. The coloured background is a Grad-CAM heat-map computed on the second last convolutional layer of the vision encoder with respect to the *sum of all* $(x + y)$ *coordinates across the 8 predicted actions*. Red/yellow regions contribute most to the final motor command, whereas blue regions are largely irrelevant.

We feed the encoder with the two most recent frames; therefore every roll-out is visualised twice: *time step #0* shows the penultimate observation $(t-1)$, while *time step #1* shows the current observation (t). Notice how the areas of high attribution move consistently with the planned path—shifting from the upcoming contact point in frame #0 to the next



contact point in frame #1—indicating that the policy attends to semantically meaningful geometry (block edges and goal zone) rather than background texture. Across different columns the heat-map rotates and translates together with the scene layout, confirming that the learned visual features generalise to novel configurations instead of over-fitting to absolute pixel locations.

V. Comparative Insight & Open Issues

To transcend simplistic comparisons based solely on reconstruction accuracy or task success rates, we introduce five task-relevant metrics directly tied to closed-loop robotic manipulation in Tab. I: **memory efficiency**, **differentiability**, **uncertainty**, **dynamic compatibility**, and **generation/query speed**. By uniformly projecting representative methods from different geometric representation categories onto this five-dimensional metric space, we objectively highlight critical weaknesses and unnecessary redundancies, providing precise optimization targets for designing future hybrid geometric-semantic fields beyond mere qualitative pros-and-cons lists.

Reviewing the comparative analysis in Tab. I, we find that existing 3D representations struggle to simultaneously satisfy the five-dimensional requirements critical for robotic control. Specifically, voxel-based methods provide robust safety boundaries using occupancy probabilities, yet suffer from low utilization rates at high resolution, leading to substantial memory and bandwidth waste due to many unused voxels. Even high-frequency incremental ESDF approaches like FIESTA can only maintain real-time performance within small local windows [4]. Mesh or surfel representations offer continuous surface normals and curvature essential for precise contact tasks, but their differentiability relies on numerical approximations, and they require reconstruction or downsampling of bounding volume hierarchies after each scene update, hindering high-frequency closed-loop control. Point clouds directly expose raw geometric details to learning algorithms, achieving

high utilization and dynamic-friendliness, yet completely lack analytical gradients and explicit uncertainty measures. Thus, controllers must rely on black-box networks for discrete action scores, complicating safety margin validation in precision manipulation tasks. Distance fields come closest to an ideal scenario, as demonstrated by CHOMP [2] and GPMP2 [3], which efficiently query $d(x)$ and $\nabla d(x)$ at millisecond scales; however, their performance assumes static or low-frequency dynamic updates, limiting applicability in genuinely dynamic scenarios due to the computational burden of high-resolution reconstructions. Neural and implicit fields mitigate issues of differentiability and low utilization, exemplified by methods like NeuralFeels, achieving efficient high-frequency closed-loop querying and high element utilization [21]. However, these approaches still primarily focus on in-hand or static object manipulation, restricting their update frequencies to 1–2 Hz for larger-scale rapidly moving objects, and lack unified support for scene-level uncertainty representation and editing.

Consequently, we argue that existing explicit representations cannot simultaneously achieve high utilization and differentiable gradients, while implicit methods, despite their advantages in differentiability and utilization, have yet to comprehensively resolve global dynamic updating and unified uncertainty modeling. Addressing these gaps highlights the critical directions for developing next-generation hybrid geometric-semantic fields, combining explicit local caching with global neural representations.

In autonomous driving, large-scale systems such as Tesla's Occupancy Network and Waymo's Scene Transformer have demonstrated that aggregating multi-camera RGB observations into a unified Bird's-Eye-View (BEV) representation before passing them to upper-level planners effectively improves robustness and safety. This pipeline generally consists of three sequential stages. First, images from multiple viewpoints are mapped into a common vehicle-centric BEV coordinate system, integrating depth estimation, semantic segmentation, and temporal motion prediction, thereby achieving spatial alignment in both geometry and semantics. Next, perceptual noise and individual camera failures are encapsulated within sparse or dense occupancy/dynamic grids, ensuring that higher-level modules are decoupled from specific viewpoint dependencies. Finally, planning and control modules consume this unified 2D geometric-semantic plane to make decisions, thus providing viewpoint invariance and enabling engineers to conveniently embed rule checking, replay verification, and safety monitoring within the BEV space.

This conceptual framework also provides valuable inspiration for robotic manipulation. Purely image-based policies lacking explicit spatial alignment exhibit extreme sensitivity to viewpoint changes. Consequently, a BEV-like intermediate representation is essential to robustly isolate perceptual disturbances from downstream policy modules.

Notably, the concept of BEV in manipulation contexts does not strictly imply a top-down viewpoint; rather, it encompasses any spatial bottleneck that compresses multi-view perceptions into a consistent coordinate frame while preserving task-relevant geometric-semantic information. For instance, a system employing 3D Gaussian Splatting can project semantic Gaussian clouds onto a plane defined in a hand-eye coordinate system, forming a continuously differentiable and editable BEV representation. Alternatively, in NeRF-based pipelines, discretizing feature volumes along the ray axis yields BEV tokens akin to those utilized by GraspNeRF. These representations share a fundamental trait: before entering the policy network, information from diverse viewpoints and modalities has been flattened into a unified spatial reference frame, compelling the network to first learn geometric-semantic alignment and subsequently infer actions.

VI. Conclusion

Over the past five years, robot–manipulation research has rapidly shifted from explicit geometric maps to implicit neural fields, from task-specific pipelines to generalist agents, and from vision-only inputs to multimodal world models. Structured scene representations such as OctoMap [1] and the family of 3-D Dynamic Scene Graphs and their manipulation extensions [14]–[17] have established an interpretable bridge between perception and high-level planning, yet their graph updates still lag behind the dynamics of fast human–robot interaction. Classical explicit geometries—voxels, meshes, point clouds, and distance fields—continue to trade memory efficiency against safety guarantees: occupancy-grid ESDF planners or their fast incremental variants offer reliable clearance estimates, whereas gradient-based motion planners such as CHOMP and GPMP2 [2], [3] provide analytic derivatives but incur costly high-resolution reconstructions. In contrast, implicit neural fields (NeuralFeels, Neural Descriptor Fields, Dex-NeRF, GraspSplats, and ISDF) [21], [22], [24], [26], [27] promise dense geometry, semantics, and differentiability in one continuous volume, yet real-time global updates and unified uncertainty remain open challenges. Finally, large-scale visuomotor policies and vision–language–action models—including Diffusion Policy [35], BC-Z [34], the "Generalist Agent" [36], RoboCat [37], and RT-2 [38]—have begun to endow robots with open-vocabulary instruction following and cross-task generalization, though tight integration with physically verifiable priors is still missing. We therefore envision a next-generation *hybrid geometric–semantic field* that fuses a local explicit cache for safety-critical collision checks, a global neural implicit volume for differentiable querying and semantic generalization, a dynamically updated scene graph for causal task reasoning, and a foundation world model for language grounding and continual self-improvement—together enabling robots to act precisely, robustly, and autonomously in truly open environments.

References

[1] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Autonomous Robots*, 2013, software available at https://octomap.github.io. [Online]. Available: https://octomap.github.io

[2] N. Ratliff, M. Zucker, J. A. Bagnell, and S. Srinivasa, "Chomp: Gradient optimization techniques for efficient motion planning," in *2009 IEEE international conference on robotics and automation*. IEEE, 2009, pp. 489–494.

[3] M. Mukadam, J. Dong, X. Yan, F. Dellaert, and B. Boots, "Continuous-time gaussian process motion planning via probabilistic inference," *The International Journal of Robotics Research*, vol. 37, no. 11, pp. 1319–1340, 2018.

[4] L. Han, F. Gao, B. Zhou, and S. Shen, "Fiesta: Fast incremental euclidean distance fields for online motion planning of aerial robots," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 4423–4430.

[5] W. Huang, C. Wang, R. Zhang, Y. Li, J. Wu, and L. Fei-Fei, "Voxposer: Composable 3d value maps for robotic manipulation with language models," *arXiv preprint arXiv:2307.05973*, 2023.

[6] J. Pan, S. Chitta, and D. Manocha, "Fcl: A general purpose library for collision and proximity queries," in *2012 IEEE international conference on robotics and automation*. IEEE, 2012, pp. 3859–3866.

[7] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, and K. Goldberg, "Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics," *arXiv preprint arXiv:1703.09312*, 2017.

[8] M. Dalle Vedove, F. J. Abu-Dakka, L. Palopoli, D. Fontanelli, and M. Saveriano, "Meshdmp: Motion planning on discrete manifolds using dynamic movement primitives," *arXiv e-prints*, pp. arXiv–2410, 2024.

[9] D. Droeschel, M. Schwarz, and S. Behnke, "Continuous mapping and localization for autonomous navigation in rough terrain using a 3d laser scanner," *Robotics and Autonomous Systems*, vol. 88, pp. 104–115, 2017.

[10] A. Ten Pas, M. Gualtieri, K. Saenko, and R. Platt, "Grasp pose detection in point clouds," *The International Journal of Robotics Research*, vol. 36, no. 13-14, pp. 1455–1473, 2017.

[11] H. Liang, X. Ma, S. Li, M. Görner, S. Tang, B. Fang, F. Sun, and J. Zhang, "Pointnetgpd: Detecting grasp configurations from point sets," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 3629–3635.

[12] Y. Qin, B. Huang, Z.-H. Yin, H. Su, and X. Wang, "Dexpoint: Generalizable point cloud reinforcement learning for sim-to-real dexterous manipulation," in *Conference on Robot Learning*. PMLR, 2023, pp. 594–605.

[13] H. Geng, Z. Li, Y. Geng, J. Chen, H. Dong, and H. Wang, "Partmanip: Learning cross-category generalizable part manipulation policy from point cloud observations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 2978–2988.

[14] A. Rosinol, A. Gupta, M. Abate, J. Shi, and L. Carlone, "3d dynamic scene graphs: Actionable spatial perception with places, objects, and humans," *arXiv preprint arXiv:2002.06289*, 2020.

[15] Z. Jiao, Y. Niu, Z. Zhang, S.-C. Zhu, Y. Zhu, and H. Liu, "Sequential manipulation planning on scene graph," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 8203–8210.

[16] Y. Lin, A. S. Wang, E. Undersander, and A. Rai, "Efficient and interpretable robot manipulation with graph neural networks," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2740–2747, 2022.

[17] H. Jiang, B. Huang, R. Wu, Z. Li, S. Garg, H. Nayyeri, S. Wang, and Y. Li, "Roboexp: Action-conditioned scene graph via interactive exploration for robotic manipulation," *arXiv preprint arXiv:2402.15487*, 2024.

[18] C. Chamzas, Z. Kingston, C. Quintero-Peña, A. Shrivastava, and L. E. Kavraki, "Learning sampling distributions using local 3d workspace decompositions for motion planning in high dimensions," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 1283–1289.

[19] J. Varley, C. DeChant, A. Richardson, J. Ruales, and P. Allen, "Shape completion enabled robotic grasping," in *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2017, pp. 2442–2447.

[20] H.-S. Fang, C. Wang, H. Fang, M. Gou, J. Liu, H. Yan, W. Liu, Y. Xie, and C. Lu, "Anygrasp: Robust and efficient grasp perception in spatial and temporal domains," *IEEE Transactions on Robotics*, vol. 39, no. 5, pp. 3929–3945, 2023.

[21] S. Suresh, H. Qi, T. Wu, T. Fan, L. Pineda, M. Lambeta, J. Malik, M. Kalakrishnan, R. Calandra, M. Kaess *et al.*, "Neuralfeels with neural fields: Visuotactile perception for in-hand manipulation," *Science Robotics*, vol. 9, no. 96, p. eadl0628, 2024.

[22] A. Simeonov, Y. Du, A. Tagliasacchi, J. B. Tenenbaum, A. Rodriguez, P. Agrawal, and V. Sitzmann, "Neural descriptor fields: Se (3)-equivariant object representations for manipulation," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 6394–6400.

[23] A. Simeonov, Y. Du, Y.-C. Lin, A. R. Garcia, L. P. Kaelbling, T. Lozano-Pérez, and P. Agrawal, "Se (3)-equivariant relational rearrangement with neural descriptor fields," in *Conference on Robot Learning*. PMLR, 2023, pp. 835–846.

[24] J. Ichnowski, Y. Avigal, J. Kerr, and K. Goldberg, "Dex-nerf: Using a neural radiance field to grasp transparent objects," *arXiv preprint arXiv:2110.14217*, 2021.

[25] Q. Dai, Y. Zhu, Y. Geng, C. Ruan, J. Zhang, and H. 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)*. IEEE, 2023, pp. 1757–1763.

[26] M. Ji, R.-Z. Qiu, X. Zou, and X. Wang, "Graspsplats: Efficient manipulation with 3d feature splatting," *arXiv preprint arXiv:2409.02084*, 2024.

[27] J. Ortiz, A. Clegg, J. Dong, E. Sucar, D. Novotny, M. Zollhoefer, and M. Mukadam, "isdf: Real-time neural signed distance fields for robot perception," *arXiv preprint arXiv:2204.02296*, 2022.

[28] L. Yen-Chen, P. Florence, J. T. Barron, A. Rodriguez, P. Isola, and T.-Y. Lin, "inerf: Inverting neural radiance fields for pose estimation," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 1323–1330.

[29] J. Kerr, L. Fu, H. Huang, Y. Avigal, M. Tancik, J. Ichnowski, A. Kanazawa, and K. Goldberg, "Evo-nerf: Evolving nerf for sequential robot grasping of transparent objects," in *6th annual conference on robot learning*, 2022.

[30] C. Yan, D. Qu, D. Xu, B. Zhao, Z. Wang, D. Wang, and X. Li, "Gs-slam: Dense visual slam with 3d gaussian splatting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 19 595–19 604.

[31] Y. Zheng, X. Chen, Y. Zheng, S. Gu, R. Yang, B. Jin, P. Li, C. Zhong, Z. Wang, L. Liu *et al.*, "Gaussiangrasper: 3d language gaussian splatting for open-vocabulary robotic grasping," *IEEE Robotics and Automation Letters*, 2024.

[32] J. Yu, X. Ren, Y. Gu, H. Lin, T. Wang, Y. Zhu, H. Xu, Y.-G. Jiang, X. Xue, and Y. Fu, "Sparsegrasp: Robotic grasping via 3d semantic gaussian splatting from sparse multi-view rgb images," *arXiv preprint arXiv:2412.02140*, 2024.

[33] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *Journal of Machine Learning Research*, vol. 17, no. 39, pp. 1–40, 2016.

[34] E. Jang, A. Irpan, M. Khansari, D. Kappler, F. Ebert, C. Lynch, S. Levine, and C. Finn, "Bc-z: Zero-shot task generalization with robotic imitation learning," in *Conference on Robot Learning*. PMLR, 2022, pp. 991–1002.

[35] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," *The International Journal of Robotics Research*, p. 02783649241273668, 2023.

[36] S. Reed, K. Zolna, E. Parisotto, S. G. Colmenarejo, A. Novikov, G. Barth-Maron, M. Gimenez, Y. Sulsky, J. Kay, J. T. Springenberg *et al.*, "A generalist agent," *arXiv preprint arXiv:2205.06175*, 2022.

[37] K. Bousmalis, G. Vezzani, D. Rao, C. Devin, A. X. Lee, M. Bauzá, T. Davchev, Y. Zhou, A. Gupta, A. Raju *et al.*, "Robocat: A self-improving generalist agent for robotic manipulation," *arXiv preprint arXiv:2306.11706*, 2023.

[38] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn *et al.*, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," *arXiv preprint arXiv:2307.15818*, 2023.