A Universal Semantic-Geometric Representation for Robotic Manipulation

Anonymous Author(s)
Affiliation
Address
email

Abstract: Robots rely heavily on sensors, especially RGB and depth cameras, to perceive and interact with the world. RGB cameras record 2D images with rich semantic information while missing precise spatial information. On the other side, depth cameras offer critical 3D geometry data but capture limited semantics. Therefore, integrating both modalities is crucial for learning representations for robotic perception and control. However, current research predominantly focuses on only one of these modalities, neglecting the benefits of incorporating both. To this end, we present Semantic-Geometric Representation (SGR), a universal perception module for robotics that leverages the rich semantic information of large-scale pre-trained 2D models and inherits the merits of 3D spatial reasoning. Our experiments demonstrate that SGR empowers the agent to successfully complete a diverse range of simulated and real-world robotic manipulation tasks, outperforming state-of-the-art methods significantly in both single-task and multi-task settings. Furthermore, SGR possesses the unique capability to generalize to novel semantic attributes, setting it apart from the other methods.

Keywords: Representation Learning, Robotic Manipulation

1 Introduction

In the field of robotics, sensors play an indispensable role in enabling robots to perceive and interact with the world [1]. Among the various types of sensors, RGB cameras and depth cameras stand out for their accessibility and affordability [2]. RGB cameras capture high-resolution 2D images that are rich in semantic information, providing a high-level understanding of the scene [3, 4]. However, it is hard to use them to reason about complex spatial relationships. On the other hand, depth cameras provide 3D geometry information, which is crucial for accurate fine-grained manipulation [5, 6, 7]. Nevertheless, they have limited semantic understanding. As shown in Figure 1, it is of utmost importance in robot learning to utilize these two complementary modalities to develop general-purpose robots that can learn a diverse set of manipulation skills. In this paper, we ask the following question: *in the new era of pre-trained large vision models, how can we learn a representation for robots that integrates both semantic understanding and 3D spatial reasoning?*

A growing body of research is focused on utilizing pre-trained 2D vision models to learn visual representations that possess general knowledge of our world [8, 9, 10, 11, 12, 13]. This high-level semantic information is invaluable for generalizable robotic control. However, the current emphasis has largely been on pre-training on huge volumes of 2D images [14, 15], despite the fact that our world is intrinsically 3D. When confronted with the immense complexity of real-world environments, a robot with only 2D prior knowledge may face challenges such as partial occlusion and geometric shape comprehension [16, 17]. Therefore, it is essential to develop models that can handle 3D visual data to enable robots to perform spatial reasoning effectively.

Submitted to the 7th Conference on Robot Learning (CoRL 2023). Do not distribute.

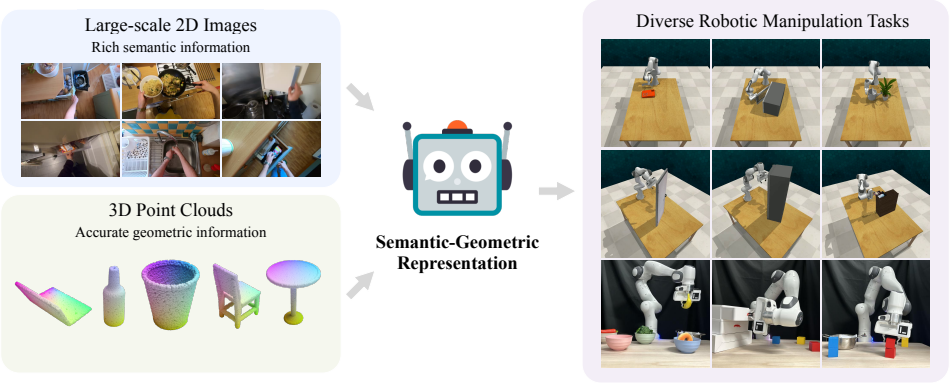


Figure 1: **Semantic-Geometric Representation (SGR):** Leveraging semantic information from massive 2D images and geometric information from 3D point clouds, we present a representation model SGR that enables the robots to solve a range of simulated and real-world manipulation tasks.

One natural and intuitive solution is to develop pre-trained 3D vision models. However, despite some attempts in the vision community [18, 19, 20, 21], current models are not yet sufficiently transferable to robotic manipulation tasks due to two main reasons. *(i)* The expensive data acquisition and labor-intensive annotation process result in a lack of large-scale 3D datasets [22, 23]. *(ii)* 3D point clouds with sparse structural patterns do not provide diversified semantics compared to colorful 2D images [24], thereby constraining the generalization capacity of 3D pre-training.

To address the challenges associated with using 2D or 3D vision models independently, we propose **Semantic-Geometric Representation (SGR)**, a hybrid perception module for robotics that captures both high-level semantics and low-level spatial patterns. As illustrated in Figure 2, we first utilize a large vision foundation model [25, 26, 27], pre-trained on massive amounts of internet data, to encode semantic feature maps from 2D images. Secondly, the context-rich 2D feature vectors are back-projected into 3D space and combined with the point cloud features that are extracted from point clouds using a point-based network [28, 29]. These fused features are fed into a number of set abstraction (SA) blocks [30], which jointly model the cross-modal interaction between 2D semantics and 3D geometry information. Finally, based on the output representations from the SA blocks, we predict the robotic action to execute.

We learn control policy through behavior cloning [31] on a broad spectrum of robotic manipulation tasks from the Robot Learning Benchmark (RLBench) [32]. In both single-task and multi-task learning scenarios, our SGR consistently outperforms a wide range of 2D and 3D representations, such as R3M [12], CLIP [26], and PERACT [33], by a significant margin. This striking performance demonstrates the effectiveness of integrating semantic and geometric representations synergistically. Furthermore, SGR stands as the *only* approach capable of generalizing to previously unseen semantic attributes, including colors and shapes. To further validate its effectiveness, we assess SGR using a Franka Emika Panda robot, training a multi-task agent across 8 real-world tasks. Remarkably, SGR significantly surpasses a strong baseline (PERACT) on tasks that involve visual distractors. All these findings provide compelling evidence that SGR possesses the potential to serve as a universal representation for general-purpose robot learning.

2 Related Work

2D Representation Learning for Robotics. A wealth of prior approaches learn visual representations from *in-domain* data taken directly from the target environment and task. These techniques encompass a broad spectrum, ranging from data augmentation [34, 35, 36] and forward dynamics modeling [37, 38] to the utilization of task-specific information [39, 40]. However, recent studies shift their focus towards acquiring more generalized representations. They achieve this by employing vision models pre-trained on large-scale *out-of-domain* data as the cornerstones for visuo-motor

control [11, 41]. RRL [42], PIE-G [43], and MVP [10] demonstrating the effectiveness of supervised or self-supervised pre-trained representations for RL agents. PVR [9] and R3M [12] find that vision models pre-trained on real-world data enable data-efficient behavior cloning. VIP [44] proposes a pre-trained representation capable of producing dense reward signals. Hu et al. [8] conducts the first thorough empirical evaluation of pre-trained visual representations across different downstream policy learning methods. However, these approaches lack an explicit consideration of 3D geometry, which limits their capacity to develop highly accurate spatial manipulation skills in robotics.

3D Representation Learning for Robotics. The 3D perception module for robotics can be classified into the following three types: (1) Projection-based models: Many approaches utilize multi-view images, which represent the projections of a 3D environment onto various image planes, as direct inputs for robots [45, 46, 47]. However, a significant drawback of these methods is the loss of geometric information during the projection stage. (2) Voxel-based models: Another way to represent a 3D environment is voxelization, which extends the concept of 2D pixels [48, 49]. Recent works like C2FARM [50] and PERACT [33] employ a voxelized observation and action space for 6-DoF manipulation. Nevertheless, these models underutilize the sparsity of point sets in 3D and suffer from massive computational and memory costs. (3) Point-based models: In contrast to the previous two types, point-based models, such as PointNet [28], PointNet++ [30], directly and efficiently process point clouds. A plethora of approaches in robotics use PointNet or PointNet++ as a visual feature encoder [51, 52, 53, 54, 55, 56]. In this paper, we utilize point-based models to extract precise 3D geometric information from point clouds due to their effectiveness and efficiency.

Fuse 2D and 3D information. There is a rich body of literature dedicated to the study of sensor fusion, which aims to leverage the complementary nature of 2D and 3D information. In the field of autonomous driving, numerous previous works utilize camera-captured images and LiDAR point clouds to achieve accurate and robust 3D object detection [57, 58, 59, 60, 61]. Similarly, in the field of robotics, several prior studies explore the combination of color and depth information from RGB-D input to estimate object poses [62, 63, 64]. One notable approach, called DenseFusion [62], processes the two data sources individually and employs a fusion network to extract pixel-wise dense feature embeddings. However, one limitation of DenseFusion is its exclusive training on data from the target domain, leading to a failure in generalizing to novel objects. In contrast, our approach leverages visual representations pre-trained on large-scale images sourced from the web, empowering our robot to generalize to unseen attributes. This level of generalization is unattainable for a domain-specific method.

3 Method

In this section, we cover the different components of our approach. We start with a brief introduction of CLIP [26] and PointNeXt [28] in Section 3.1. Next, in Section 3.2, we present Semantic-Geometric Representation (SGR), a method that leverages the benefits of both pre-trained 2D models and 3D networks. Lastly, in Section 3.3, we demonstrate the integration of SGR into a behavior-cloning agent for 6-DoF manipulation tasks.

3.1 Background

Contrastive Language-Image Pre-training (CLIP) [26] has emerged as a simple yet powerful methodology for learning representations by pre-training on extensive image-text pairs from the web. Initially, CLIP employs separate modality-specific models to embed images and text. Subsequently, these vectors are projected into a shared embedding space and normalized. The InfoNCE loss [65] is calculated based on these final embeddings, utilizing corresponding images and captions as positive pairs and treating all non-matching images and captions as negative pairs. The learned visual representation not only exhibits impressive semantic discriminative power and zero-shot transferability [26], but also demonstrates exceptional generalization performance in robotic manipulation and navigation tasks [66, 67]. Although we utilize CLIP to extract semantic features, our perception module is also compatible with a wide range of pre-trained 2D vision models [68, 13].

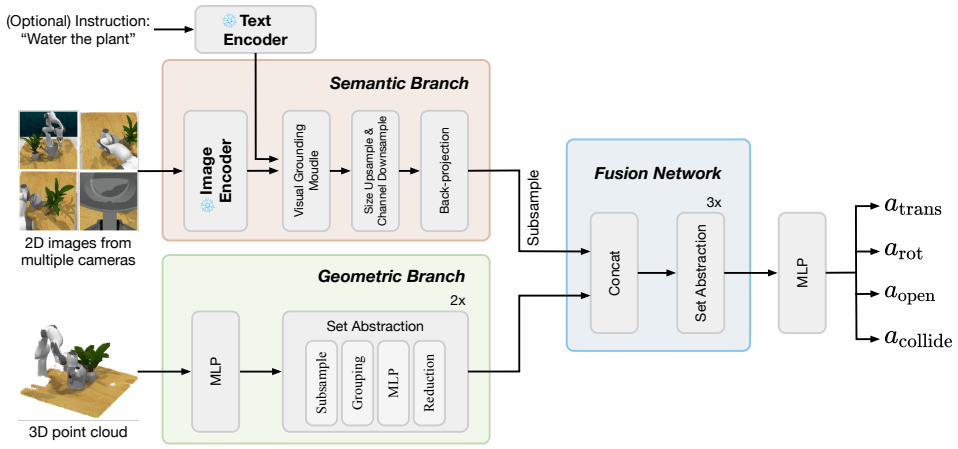


Figure 2: **Architecture.** The semantic branch generates point-wise semantic features from the multi-view images. Simultaneously, the geometric branch extracts geometric features from a point cloud. Finally, we fuse the two kinds of complementary features and predict the action to execute.

PointNet, PointNet++, and PointNeXt. PointNet [28] is a pioneering and classical model that directly operates on unordered point clouds, capable of learning global point cloud features through shared multi-layer perceptrons (MLPs) and a max-pooling operation. PointNet++ [30] extends PointNet by utilizing a hierarchical neural network to handle local and global features in a more efficient manner. PointNeXt [29] is a recent study that revisits PointNet++ with improved training and scaling strategies to enhance its performance to the state-of-the-art level. Because of the simplicity, efficiency, and effectiveness of PointNeXt, we use PointNeXt in our experiments to enable robots to obtain accurate 3D spatial information.

3.2 Semantic-Geometric Representation

Our objective is to design a universal perception module for robotics that incorporates various sensory inputs, such as RGB images, point clouds, and proprioception data (e.g., gripper open state). This module generates a **Semantic-Geometric Representation (SGR)** capable of capturing both high-level semantics and low-level spatial patterns. The resulting representations are utilized to predict robotic actions for manipulation tasks. Moreover, for language-conditioned tasks, our module can process natural language and ground semantic concepts present in the images. The module comprises three key components: (1) a semantic branch, (2) a geometric branch, and (3) a fusion network. Figure 2 provides an overview of our design.

Semantic branch. Given a set of RGB images $\{I_i\}_{i=1}^K$ obtained from K calibrated cameras, we first utilize a *frozen* pre-trained 2D model $\mathcal{G}$ (e.g., the visual encoder of CLIP) to extract multi-view image features $\{\mathcal{G}(I_i)\}_{i=1}^K$. If an language instruction S is provided for a task, we employ a pre-trained language model $\mathcal{H}$ (e.g., the language encoder of CLIP) to obtain the language encoding $\mathcal{H}(S)$. Following MaskCLIP [69] and PROGRAMPORT [70], we use a visual grounding module to align each feature map $\mathcal{G}(I_i)$ with the language encoding $\mathcal{H}(S)$, resulting in aligned feature maps $\{M_i\}_{i=1}^K$. For more details on the visual grounding module, please refer to Appendix B.1. Next, we upsample the visual feature maps $\{\mathcal{G}(I_i)\}_{i=1}^K$ (or aligned feature maps $\{M_i\}_{i=1}^K$) to the same size as the input images by bilinear interpolation and downsample the channels via 1×1 convolution, resulting in a set of features denoted as $\{F_i\}_{i=1}^K$, where $F_i \in \mathbb{R}^{H \times W \times C_1}$ and H, W represent the image size. These encoded 2D features, which encompass rich high-level semantics, are projected back into 3D space, yielding point-wise semantic features for the point cloud. We formulate them as $F^{3\text{D}} \in \mathbb{R}^{N \times C_1}$, where $N = K \times H \times W$.

Geometric branch. The raw point cloud, denoted as $P \in \mathbb{R}^{N \times 3}$, is generated by utilizing multi-view depth images and known camera parameters (i.e., camera intrinsics and extrinsics). Additionally, we incorporate robot proprioceptive information z by applying a linear layer, which produces a

D-dimensional vector that is attached to all the points. This augmented point cloud is represented as $P' \in \mathbb{R}^{N\times(3+D)}$. Subsequently, we employ a point-based network (e.g., a hierarchical PointNeXt) to process the point cloud P'. This processing step yields downsampled point geometric features, denoted as $G \in \mathbb{R}^{M\times C_2}$, where $M < N$. While these geometric features effectively capture local 3D structure and spatial characteristics, they lack the understanding of semantic information.

Fusion network. To integrate the two complementary branch, we begin by processing the point-wise semantic features F^{3D} using the same point subsampling procedure as the geometric branch. The subsampled semantic features are denoted as $F^{\text{3D}}_{\text{sub}} \in \mathbb{R}^{M\times C_1}$. Next, we fuse the semantic features and geometric features through channel-wise concatenation, expressed as $F^{\text{3D}}_{\text{fuse}} = \text{Concat}(F^{\text{3D}}_{\text{sub}}, G) \in \mathbb{R}^{M\times(C_1+C_2)}$. Finally, the fused features undergo several set abstraction blocks [30], enabling a cohesive modeling of the cross-modal interaction between 2D semantics and 3D geometric information. The resulting global feature represents our Semantic-Geometric Representation, which is utilized for action prediction. SGR back-projects 2D feature maps to 3D and fuses them with point cloud features directly in 3D space. This process fully utilizes the semantic information from pre-trained models and leads to better alignment with geometric information.

3.3 Robot Learning Framework

Behavior Cloning. Hu et al. [8] conduct a large-scale benchmarking study of pre-trained vision models using different policy learning methods. Their investigation show that evaluation based on reinforcement learning (RL) is highly variable and therefore unreliable, and further advocate for using more robust methods like behavior cloning (BC) [31]. Heeding this advice, we investigate the learning of 6-DoF manipulation tasks using BC. Inspired by PERACT [33], we formulate BC training as a task of predicting the "next best action". At each timestep, the robot receives an observation O comprising multi-view RGB images $\{I_i\}_{i=1}^{K}$, organised point cloud P, and proprioceptive data z. Leveraging the perception module introduced in Section 3.2, the robot extracts the SGR representation and predicts the next best action a in terms of target translation, rotation, and gripper state. Subsequently, this action is executed through a motion planner, and the process repeats until termination. Following James et al. [71, 50], all predicted robot actions are keyframe actions that capture bottleneck end-effector poses (i.e., joint velocities close to zero and unchanged gripper open state). This keyframe discovery strategy allows for the circumvention of direct predictions of long sequences of noisy actions, thus enabling more efficient policy learning. The robot is trained through supervised learning with input-keyframe action tuples from a dataset of expert demonstrations.

Training Details. Following the setup in PERACT [33], an action a consists of the 6-DoF pose (translation and rotation), gripper open state, and whether the motion-planner used collision avoidance to reach an intermediate pose: $a = \{a_{\text{trans}}, a_{\text{rot}}, a_{\text{open}}, a_{\text{collide}}\}$. For rotations, the ground-truth action is represented as a one-hot vector per rotation axis with R rotation bins: $a_{\text{rot}} \in \mathbb{R}^{(360/R)\times 3}$ ($R = 5$ degrees in our implementation). Open and collide actions are binary one-hot vectors: $a_{\text{open}} \in \mathbb{R}^2, a_{\text{collide}} \in \mathbb{R}^2$. The exception is translation, which is represented by a continuous 3D vector: $a_{\text{trans}} \in \mathbb{R}^3$. In this way, our loss objective is as follows:

$$\mathcal{L}_{\text{total}} = \lambda_1\mathcal{L}_{\text{trans}} + \lambda_2\mathcal{L}_{\text{rot}} + \lambda_3\mathcal{L}_{\text{open}} + \lambda_4\mathcal{L}_{\text{collide}}, \tag{1}$$

where $\mathcal{L}_{\text{trans}}$ is L1 loss for translation regression, and $\mathcal{L}_{\text{rot}}$, $\mathcal{L}_{\text{open}}$, and $\mathcal{L}_{\text{collide}}$ are cross-entropy losses for corresponding classification. Since these two kinds of losses have different scales, in our experiments, we set $\lambda_1 = 300$ and $\lambda_2 = \lambda_3 = \lambda_4 = 1$. We use AdamW [72] to optimize the model.

We utilize several data augmentation methods [29] to enhance the robustness and generalization of the model. During training, the translation is perturbed with ± 0.125 m, and colors are randomly replaced with zero values, which are called translation perturbations and color drop respectively. Additionally, we adopt the point resampling strategy in both training and evaluation. For more details, we refer to Appendix B.2.

Method	open microwave	open door	water plants	toilet seat up	phone on base	put books	take out umbrella	open fridge	**Average**
R3M	7.0	24.2	1.2	59.7	0.0	15.3	28.3	4.7	17.6
CLIP	9.1	68.4	4.1	40.1	7.3	38.0	66.7	7.4	30.2
$\text{PointNeXt}_{\text{ULIP}}$	9.4	60.0	10.4	63.0	35.6	41.8	49.8	15.3	35.7
$\text{PointNeXt}_{\text{LfS}}$	17.0	63.8	28.2	52.7	65.9	53.1	**95.1**	7.3	47.9
PERACT	26.9	47.8	**41.7**	67.1	0.0	63.7	94.4	10.1	44.0
SGR (Ours)	**52.6**	**80.2**	40.2	**80.1**	**81.1**	**84.8**	94.7	**34.8**	**68.6**

Table 1: **Single-Task Test Results.** All numbers represent percentage success rates averaged over 3 seeds. See Appendix E for standard deviation. On average, our SGR outperforms $\text{PointNeXt}_{\text{LfS}}$, the most competitive baseline, with an improvement of $1.43\times$.

4 Experiments

Our experiments are designed to answer the following questions: (1) To what extent does SGR outperform methods exclusively concentrating on semantic or geometric information individually? (2) Is it feasible to train a multi-task model capable of handling all tasks, and how well does it perform? (3) Can SGR exhibit generalization capabilities to previously unseen semantic attributes such as colors and shapes? (4) How well does SGR perform on real-world tasks?

4.1 Simulation Setup

Environment and Tasks. Our simulated experiments are conducted on RLBench [32], a large-scale benchmark and learning environment designed for vision-guided manipulation research. The visual observations are obtained from four RGB-D cameras positioned at the front, left shoulder, right shoulder, and wrist of a Franka Emika Panda robot. All cameras have a resolution of 128×128 and are noiseless. We use 8 tasks with 100 demonstrations per task for training. For more details about individual tasks and language-conditioned tasks, please see Appendix A.

Evaluation. We employ a meticulous evaluation protocol to minimize variance in our results. Specifically, for every task, we train an agent for 40,000 iterations and save checkpoints every 1,000 iterations. Then we evaluate the last 20 checkpoints for 25 episodes and mark the top 3 best-performing checkpoints. Lastly, we evaluate the top 3 checkpoints for 100 episodes and get the average success rates as our test performances. The experiments are conducted with 3 seeds.

Baselines. To verify the effectiveness of SGR, we compare with the following baselines:
2D representations: We evaluate **CLIP** [26] and **R3M** [12]. Notably, R3M is pre-trained on diverse human video datasets and showcases remarkable performance on robotic manipulation tasks. For fair comparisons, we utilize frozen CLIP or R3M to process RGB images, employ a separate 2D CNN to process depth images, and subsequently fuse the two resulting features together [73].
3D representations: We examine **PERACT** [33], **$\text{PointNeXt}_{\text{LfS}}$** [29], and **$\text{PointNeXt}_{\text{ULIP}}$** [74]. PERACT is a voxel-based model that utilizes a Perceiver Transformer [75] to process voxelized observations. As for the point-based models, we consider both a Learning-from-Scratch PointNeXt ($\text{PointNeXt}_{\text{LfS}}$) and a frozen PointNeXt that has been pre-trained by ULIP [74] ($\text{PointNeXt}_{\text{ULIP}}$). ULIP is a recent 3D representation learning method that aligns features from images, text, and point clouds in the same space, following the philosophy of CLIP. Importantly, irrespective of whether 2D or 3D representations are employed, all the baselines utilize the same input modalities (RGB-D images) to ensure fair and meaningful comparisons.

4.2 Simulation Results

Single-Task Performance. Table 1 presents a comprehensive comparison of success rates achieved through single-task imitation learning using different representations. Initially, we observe a general underperformance of 2D representations compared to their 3D counterparts. This finding emphasizes the critical role of 3D geometric information in accomplishing fine-grained manipulation tasks. Among the 2D representations, CLIP demonstrates superior performance compared to R3M, suggesting that pre-trained R3M exhibits less transferability to the RLBench environment. Regard-

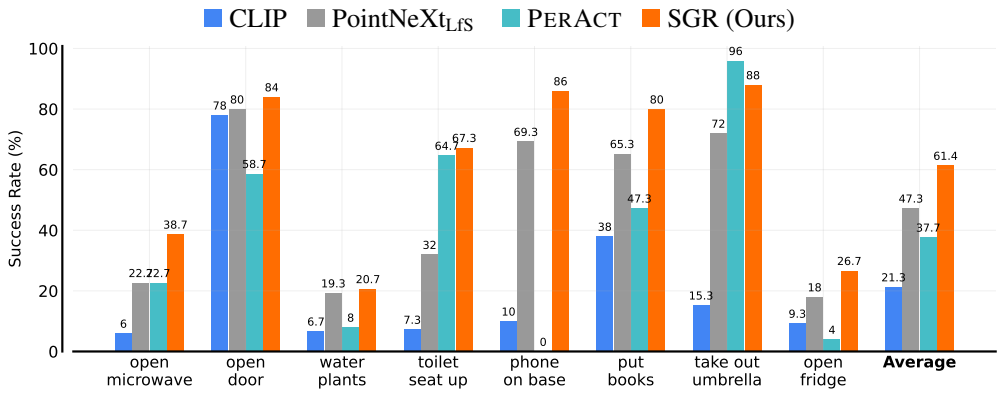


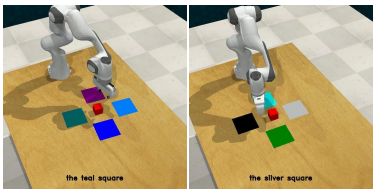
Figure 3: **Multi-Task Test Results.** We report the success rates averaged over 3 seeds. We observe that across 8 tasks SGR consistently outperforms baselines like CLIP, PointNeXt$_{\text{LfS}}$, and PERACT.

ing 3D representations, a Learning-from-Scratch PointNeXt (PointNeXt$_{\text{LfS}}$) outperforms its frozen counterpart pre-trained using ULIP (PointNeXt$_{\text{ULIP}}$). We hypothesize that this discrepancy arises due to the current focus of 3D pre-training methods on learning from small object-level datasets [22], resulting in representations that are less suitable for scene-level environments typically encountered in robotic manipulation tasks. Furthermore, PointNeXt$_{\text{LfS}}$ achieves comparable average results to PERACT while significantly reducing both computational and memory costs. Ultimately, our SGR proves to be the most effective representation, outperforming the second-best method by over 20% on average and yielding superior performance in 6 out of 8 tasks. These results highlight the benefits of capturing both 2D semantics and 3D spatial patterns for robotic manipulation.

Multi-Task Performance. In practical scenarios, our objective is to ensure that robots acquire a broad range of skills, rather than focusing on a single skill alone. With this goal in mind, we train a single agent on all 8 tasks, while also providing language instructions for each task to enhance the robot's ability to differentiate between them. For detailed information on the training of this language-conditioned behavior-cloning agent, please refer to the Appendix B.2. We compare our method, SGR, with three strong baselines and summarize the results in Figure 3. SGR demonstrates exceptional performance in multi-task scenarios, achieving an average accuracy of 61.4%. This outcome underscores the advantageous synergistic effect achieved by considering both semantic and geometric information, even in diverse multi-task settings. However, it is surprising to note that the performance of all multi-task models declines to varying extents when compared to their single-task counterparts. Notably, this phenomenon has also been observed in PERACT [33] and other recent studies [76, 77]. While multi-task training holds great promise for knowledge sharing among related tasks and facilitating more efficient learning, it presents numerous optimization challenges in practice [78, 79]. We hypothesize that employing advanced optimization algorithms [80, 81, 82] may help alleviate the negative transfer effects. We leave this exploration as future work.

Generalizing to Novel Attributes. A remarkable feature of human manipulation skill is that, once humans have learned to manipulate a category of objects, they are able to manipulate even unseen objects, regardless of their variations in color and shape. To evaluate whether an agent exhibits a similar level of generalization, we introduce three tasks: 1) reaching to a colored ball (`reach target`), 2) sliding a block onto a colored square target (`slide block`), and 3) reaching to a specified shape (`reach shape`). Notably, the `reach target` and `slide block` tasks examine the agent's ability to generalize across different colors, with training conducted on 10 colors and testing on 10 unseen colors. The `reach shape` task assesses the ability to generalize across various shapes, where training encompasses 3 shapes while testing involves 2 unseen shapes. Language instructions are employed to specify the desired color or shape. See Appendix A for more task details.

Figure 4 demonstrates that SGR achieves significantly higher performance when confronted with *unseen* colors or shapes, highlighting its remarkable generalization ability. This exceptional capacity primarily stems from the semantic branch of SGR (a pre-trained CLIP), which has undergone extensive exposure to a wide range of semantic attributes during its pre-training process on millions



Seen Unseen

Method	reach target		slide block		reach shape	
	Seen	Unseen	Seen	Unseen	Seen	Unseen
R3M	20.2	5.5	11.0	3.0	78.1	6.7
CLIP	42.0	17.7	49.0	16.3	95.5	0.2
$\text{PointNeXt}_{\text{ULIP}}$	-	-	-	-	45.9	6.7
$\text{PointNeXt}_{\text{LfS}}$	99.7	4.0	96.5	6.2	99.8	1.8
DenseFusion	71.8	9.5	26.5	16.5	99.3	3.8
SGR (Ours)	94.0	**36.8**	97.7	**43.0**	99.9	**27.7**

Figure 4: **Left:** For the `slide block` task, we train the agent on 10 colors and evaluate it on 10 unseen colors. Please see Appendix A for `reach target` and `reach shape` tasks. **Right:** Our SGR is the only method that demonstrates the ability to generalize across various colors and shapes. $\text{PointNeXt}_{\text{ULIP}}$ only accepts point coordinate input, thus the first two tasks are not applicable to it.

of image-caption pairs. In contrast, if we initialize the semantic branch randomly and train it from scratch, the resulting model would resemble DenseFusion [62]. As depicted in Figure 4, DenseFusion exhibits limited generalization ability. However, it is crucial to recognize that relying solely on semantic information proves inadequate. Even prominent pre-trained models such as CLIP and R3M encounter challenges in achieving satisfactory performance on the *seen* split of certain tasks, largely due to their inability to capture precise 3D spatial information.

4.3 Real-Robot Results

Finally, we evaluate the performance of SGR on real-world tasks. We design a set of 8 manipulation tasks using a Franka Emika Panda. See our anonymous website[1] for qualitative results that showcase the diversity of our tasks. We collect 8 demonstrations for each task and train a multi-task agent on all 8 tasks. Further details regarding the real robot setup can be found in Appendix D. In Table 2, we present the success rates and compare SGR with PERACT, one of the strong baselines we evaluated in simulation. SGR achieves an impressive average success rate of 59%, while PerAct attains 48%. However, it is important to note that our method is not flawless. The most common failure scenario involves predicting grasp poses that are already in proximity to the object but lack sufficient precision. We speculate that this occurs because the distribution of human demonstrations is inherently multi-modal, and our translation action (which outputs a point estimate) tends to learn the average of this distribution. To address this limitation, future research could explore integrating SGR into methods that possess multi-modal modeling capabilities [83, 84].

Success out of 10 Trials	SGR	PERACT
Hitting Ball	70%	30%
Putting Marker in Drawer	70%	60%
Moving Cup to Goal	70%	30%
Picking Red Block	60%	70%
Putting Banana in Pot	60%	60%
Putting Apple in Bowl	50%	70%
Opening Drawer	50%	30%
Pressing Handsan	40%	40%
Average	**59%**	48%

Table 2: Success rates of multi-task agents on real-world tasks.

5 Discussion

Limitations. Although we uncover the significance of harnessing both semantic and geometric information for robotic tasks, our current approach relies on modality-specific networks to extract these two complementary types of information and subsequently combines them using a straightforward method. However, we are thrilled by the prospect of employing a unified architecture, such as a multimodal Transformer [85], which would allow us to encode and fuse various modalities in a more elegant manner. This unified approach would also enable us to learn semantic-geometric representations during a pre-training phase, utilizing an wide range of data from real-world.

Conclusion. We present Semantic-Geometric Representation (SGR), a perception module for robotics that integrates high-level semantic understanding and low-level spatial reasoning. Through extensive evaluations in both single-task and multi-task learning scenarios, we observe remarkable performance improvements when employing SGR. Importantly, SGR exhibits an impressive ability to generalize to novel attributes, a pivotal factor in achieving general-purpose manipulation skills.

[1] `semantic-geometric-representation.github.io`

References

[1] K. S. Fu, R. C. Gonzalez, and C. G. Lee. *Robotics*. IEEE Computer Society Press, 1983.

[2] Realsense. https://www.intel.com/content/www/us/en/architecture-and-technology/realsense-overview.html, 2023.

[3] S. Dasari, F. Ebert, S. Tian, S. Nair, B. Bucher, K. Schmeckpeper, S. Singh, S. Levine, and C. Finn. Robonet: Large-scale multi-robot learning. *arXiv preprint arXiv:1910.11215*, 2019.

[4] F. Ebert, Y. Yang, K. Schmeckpeper, B. Bucher, G. Georgakis, K. Daniilidis, C. Finn, and S. Levine. Bridge data: Boosting generalization of robotic skills with cross-domain datasets. *arXiv preprint arXiv:2109.13396*, 2021.

[5] T. Schmidt, K. Hertkorn, R. Newcombe, Z. Marton, M. Suppa, and D. Fox. Depth-based tracking with physical constraints for robot manipulation. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 119–126. IEEE, 2015.

[6] D. Morrison, P. Corke, and J. Leitner. Learning robust, real-time, reactive robotic grasping. *The International journal of robotics research*, 39(2-3):183–201, 2020.

[7] Y. Cong, R. Chen, B. Ma, H. Liu, D. Hou, and C. Yang. A comprehensive study of 3-d vision-based robot manipulation. *IEEE Transactions on Cybernetics*, 2021.

[8] Y. Hu, R. Wang, L. E. Li, and Y. Gao. For pre-trained vision models in motor control, not all policy learning methods are created equal. *arXiv preprint arXiv:2304.04591*, 2023.

[9] S. Parisi, A. Rajeswaran, S. Purushwalkam, and A. Gupta. The unsurprising effectiveness of pre-trained vision models for control. In *International Conference on Machine Learning*, pages 17359–17371. PMLR, 2022.

[10] T. Xiao, I. Radosavovic, T. Darrell, and J. Malik. Masked visual pre-training for motor control. *arXiv preprint arXiv:2203.06173*, 2022.

[11] I. Radosavovic, T. Xiao, S. James, P. Abbeel, J. Malik, and T. Darrell. Real-world robot learning with masked visual pre-training. *arXiv preprint arXiv:2210.03109*, 2022.

[12] S. Nair, A. Rajeswaran, V. Kumar, C. Finn, and A. Gupta. R3m: A universal visual representation for robot manipulation. *arXiv preprint arXiv:2203.12601*, 2022.

[13] S. Karamcheti, S. Nair, A. S. Chen, T. Kollar, C. Finn, D. Sadigh, and P. Liang. Language-driven representation learning for robotics. *arXiv preprint arXiv:2302.12766*, 2023.

[14] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[15] C. Schuhmann, R. Vencu, R. Beaumont, R. Kaczmarczyk, C. Mullis, A. Katta, T. Coombes, J. Jitsev, and A. Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*, 2021.

[16] R. Cheng, A. Agarwal, and K. Fragkiadaki. Reinforcement learning of active vision for manipulating objects under occlusions. In *Conference on Robot Learning*, pages 422–431. PMLR, 2018.

[17] T. Chen, J. Xu, and P. Agrawal. A system for general in-hand object re-orientation. In *Conference on Robot Learning*, pages 297–307. PMLR, 2022.

[18] S. Xie, J. Gu, D. Guo, C. R. Qi, L. Guibas, and O. Litany. Pointcontrast: Unsupervised pre-training for 3d point cloud understanding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 574–591. Springer, 2020.

[19] X. Yu, L. Tang, Y. Rao, T. Huang, J. Zhou, and J. Lu. Point-bert: Pre-training 3d point cloud transformers with masked point modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19313–19322, 2022.

[20] Y. Pang, W. Wang, F. E. Tay, W. Liu, Y. Tian, and L. Yuan. Masked autoencoders for point cloud self-supervised learning. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part II*, pages 604–621. Springer, 2022.

[21] R. Zhang, Z. Guo, P. Gao, R. Fang, B. Zhao, D. Wang, Y. Qiao, and H. Li. Point-m2ae: multi-scale masked autoencoders for hierarchical point cloud pre-training. *arXiv preprint arXiv:2205.14401*, 2022.

[22] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[23] M. A. Uy, Q.-H. Pham, B.-S. Hua, T. Nguyen, and S.-K. Yeung. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1588–1597, 2019.

[24] R. Zhang, L. Wang, Y. Qiao, P. Gao, and H. Li. Learning 3d representations from 2d pre-trained models via image-to-point masked autoencoders. *arXiv preprint arXiv:2212.06785*, 2022.

[25] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

[26] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[27] L. Yuan, D. Chen, Y.-L. Chen, N. Codella, X. Dai, J. Gao, H. Hu, X. Huang, B. Li, C. Li, et al. Florence: A new foundation model for computer vision. *arXiv preprint arXiv:2111.11432*, 2021.

[28] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

[29] G. Qian, Y. Li, H. Peng, J. Mai, H. A. A. K. Hammoud, M. Elhoseiny, and B. Ghanem. Pointnext: Revisiting pointnet++ with improved training and scaling strategies. *arXiv preprint arXiv:2206.04670*, 2022.

[30] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017.

[31] D. A. Pomerleau. Alvinn: An autonomous land vehicle in a neural network. *Advances in neural information processing systems*, 1, 1988.

[32] S. James, Z. Ma, D. R. Arrojo, and A. J. Davison. Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 5(2):3019–3026, 2020.

[33] M. Shridhar, L. Manuelli, and D. Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. *arXiv preprint arXiv:2209.05451*, 2022.

[34] M. Laskin, A. Srinivas, and P. Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. In *International Conference on Machine Learning*, pages 5639–5650. PMLR, 2020.

[35] M. Laskin, K. Lee, A. Stooke, L. Pinto, P. Abbeel, and A. Srinivas. Reinforcement learning with augmented data. *Advances in neural information processing systems*, 33:19884–19895, 2020.

[36] I. Kostrikov, D. Yarats, and R. Fergus. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. *arXiv preprint arXiv:2004.13649*, 2020.

[37] C. Gelada, S. Kumar, J. Buckman, O. Nachum, and M. G. Bellemare. Deepmdp: Learning continuous latent space models for representation learning. In *International Conference on Machine Learning*, pages 2170–2179. PMLR, 2019.

[38] D. Hafner, T. Lillicrap, J. Ba, and M. Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019.

[39] R. Jonschkowski and O. Brock. Learning state representations with robotic priors. *Autonomous Robots*, 39:407–428, 2015.

[40] A. Zhang, R. McAllister, R. Calandra, Y. Gal, and S. Levine. Learning invariant representations for reinforcement learning without reconstruction. *arXiv preprint arXiv:2006.10742*, 2020.

[41] A. Majumdar, K. Yadav, S. Arnaud, Y. J. Ma, C. Chen, S. Silwal, A. Jain, V.-P. Berges, P. Abbeel, J. Malik, et al. Where are we in the search for an artificial visual cortex for embodied intelligence? *arXiv preprint arXiv:2303.18240*, 2023.

[42] R. Shah and V. Kumar. Rrl: Resnet as representation for reinforcement learning. *arXiv preprint arXiv:2107.03380*, 2021.

[43] Z. Yuan, Z. Xue, B. Yuan, X. Wang, Y. Wu, Y. Gao, and H. Xu. Pre-trained image encoder for generalizable visual reinforcement learning. *arXiv preprint arXiv:2212.08860*, 2022.

[44] Y. J. Ma, S. Sodhani, D. Jayaraman, O. Bastani, V. Kumar, and A. Zhang. Vip: Towards universal visual reward and representation via value-implicit pre-training. *arXiv preprint arXiv:2210.00030*, 2022.

[45] H. Liu, L. Lee, K. Lee, and P. Abbeel. Instruction-following agents with jointly pre-trained vision-language models. *arXiv preprint arXiv:2210.13431*, 2022.

[46] P.-L. Guhur, S. Chen, R. Garcia, M. Tapaswi, I. Laptev, and C. Schmid. Instruction-driven history-aware policies for robotic manipulations. *arXiv preprint arXiv:2209.04899*, 2022.

[47] Y. Seo, J. Kim, S. James, K. Lee, J. Shin, and P. Abbeel. Multi-view masked world models for visual robotic manipulation. *arXiv preprint arXiv:2302.02408*, 2023.

[48] D. Maturana and S. Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 922–928. IEEE, 2015.

[49] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser. Semantic scene completion from a single depth image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1746–1754, 2017.

[50] S. James, K. Wada, T. Laidlow, and A. 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*, pages 13739–13748, 2022.

[51] H.-S. Fang, C. Wang, M. Gou, and C. Lu. Graspnet-1billion: A large-scale benchmark for general object grasping. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11444–11453, 2020.

[52] M. Sundermeyer, A. Mousavian, R. Triebel, and D. Fox. Contact-graspnet: Efficient 6-dof grasp generation in cluttered scenes. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13438–13444. IEEE, 2021.

[53] W. Huang, I. Mordatch, P. Abbeel, and D. Pathak. Generalization in dexterous manipulation via geometry-aware multi-task learning. *arXiv preprint arXiv:2111.03062*, 2021.

[54] Y.-H. Wu, J. Wang, and X. Wang. Learning generalizable dexterous manipulation from human grasp affordance. *arXiv preprint arXiv:2204.02320*, 2022.

[55] Y. Qin, B. Huang, Z.-H. Yin, H. Su, and X. Wang. Dexpoint: Generalizable point cloud reinforcement learning for sim-to-real dexterous manipulation. *arXiv preprint arXiv:2211.09423*, 2022.

[56] M. Liu, X. Li, Z. Ling, Y. Li, and H. Su. Frame mining: a free lunch for learning robotic manipulation from 3d point clouds. *arXiv preprint arXiv:2210.07442*, 2022.

[57] D. Xu, D. Anguelov, and A. Jain. Pointfusion: Deep sensor fusion for 3d bounding box estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 244–253, 2018.

[58] Z. Wang, W. Zhan, and M. Tomizuka. Fusing bird's eye view lidar point cloud and front view camera image for 3d object detection. In *2018 IEEE intelligent vehicles symposium (IV)*, pages 1–6. IEEE, 2018.

[59] M. Liang, B. Yang, S. Wang, and R. Urtasun. Deep continuous fusion for multi-sensor 3d object detection. In *Proceedings of the European conference on computer vision (ECCV)*, pages 641–656, 2018.

[60] M. Liang, B. Yang, Y. Chen, R. Hu, and R. Urtasun. Multi-task multi-sensor fusion for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7345–7353, 2019.

[61] S. Vora, A. H. Lang, B. Helou, and O. Beijbom. Pointpainting: Sequential fusion for 3d object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4604–4612, 2020.

[62] C. Wang, D. Xu, Y. Zhu, R. Martín-Martín, C. Lu, L. Fei-Fei, and S. Savarese. Densefusion: 6d object pose estimation by iterative dense fusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3343–3352, 2019.

[63] G. Zhou, Y. Yan, D. Wang, and Q. Chen. A novel depth and color feature fusion framework for 6d object pose estimation. *IEEE Transactions on Multimedia*, 23:1630–1639, 2020.

[64] C. Wang, R. Martín-Martín, D. Xu, J. Lv, C. Lu, L. Fei-Fei, S. Savarese, and Y. Zhu. 6-pack: Category-level 6d pose tracker with anchor-based keypoints. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10059–10066. IEEE, 2020.

[65] A. v. d. Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[66] M. Shridhar, L. Manuelli, and D. Fox. Cliport: What and where pathways for robotic manipulation. In *Conference on Robot Learning*, pages 894–906. PMLR, 2022.

[67] A. Khandelwal, L. Weihs, R. Mottaghi, and A. Kembhavi. Simple but effective: Clip embeddings for embodied ai. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14829–14838, 2022.

[68] C. Jia, Y. Yang, Y. Xia, Y.-T. Chen, Z. Parekh, H. Pham, Q. Le, Y.-H. Sung, Z. Li, and T. Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, pages 4904–4916. PMLR, 2021.

[69] C. Zhou, C. C. Loy, and B. Dai. Extract free dense labels from clip. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVIII*, pages 696–712. Springer, 2022.

[70] R. Wang, J. Mao, J. Hsu, H. Zhao, J. Wu, and Y. Gao. Programmatically grounded, compositionally generalizable robotic manipulation. *arXiv preprint arXiv:2304.13826*, 2023.

[71] S. James and A. J. Davison. Q-attention: Enabling efficient learning for vision-based robotic manipulation. *IEEE Robotics and Automation Letters*, 7(2):1612–1619, 2022.

[72] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[73] Y. Hu, R. Wang, K. Zhang, and Y. Gao. Semantic-aware fine-grained correspondence. In *European Conference on Computer Vision*, pages 97–115. Springer, 2022.

[74] L. Xue, M. Gao, C. Xing, R. Martín-Martín, J. Wu, C. Xiong, R. Xu, J. C. Niebles, and S. Savarese. Ulip: Learning unified representation of language, image and point cloud for 3d understanding. *arXiv preprint arXiv:2212.05171*, 2022.

[75] A. Jaegle, S. Borgeaud, J.-B. Alayrac, C. Doersch, C. Ionescu, D. Ding, S. Koppula, D. Zoran, A. Brock, E. Shelhamer, et al. Perceiver io: A general architecture for structured inputs & outputs. *arXiv preprint arXiv:2107.14795*, 2021.

[76] E. Parisotto, J. L. Ba, and R. Salakhutdinov. Actor-mimic: Deep multitask and transfer reinforcement learning. *arXiv preprint arXiv:1511.06342*, 2015.

[77] M. Crawshaw. Multi-task learning with deep neural networks: A survey. *arXiv preprint arXiv:2009.09796*, 2020.

[78] M. Hessel, H. Soyer, L. Espeholt, W. Czarnecki, S. Schmitt, and H. van Hasselt. Multi-task deep reinforcement learning with popart. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3796–3803, 2019.

[79] A. Kendall, Y. Gal, and R. Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7482–7491, 2018.

[80] T. Yu, S. Kumar, A. Gupta, S. Levine, K. Hausman, and C. Finn. Gradient surgery for multi-task learning. *Advances in Neural Information Processing Systems*, 33:5824–5836, 2020.

[81] S. Wu, H. R. Zhang, and C. Ré. Understanding and improving information transfer in multi-task learning. *arXiv preprint arXiv:2005.00944*, 2020.

[82] S. Liu, S. James, A. J. Davison, and E. Johns. Auto-lambda: Disentangling dynamic task relationships. *arXiv preprint arXiv:2202.03091*, 2022.

[83] N. M. Shafiullah, Z. Cui, A. A. Altanzaya, and L. Pinto. Behavior transformers: Cloning k modes with one stone. *Advances in neural information processing systems*, 35:22955–22968, 2022.

[84] T. Pearce, T. Rashid, A. Kanervisto, D. Bignell, M. Sun, R. Georgescu, S. V. Macua, S. Z. Tan, I. Momennejad, K. Hofmann, et al. Imitating human behaviour with diffusion models. *arXiv preprint arXiv:2301.10677*, 2023.

[85] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.