Supplementary Material

Method	Simulation Displacement ↓	Penetration Volume ↓	Penetration Distance ↓	Contact Ratio ↑
D-VQVAE [37]	**1.61**	0.94	5.17	98
FastGrasp [28]	1.83	**0.91**	**2.39**	**98**

Table 4. Comparative experiments to evaluate the impact of different autoencoder structures.

Dataset	train set	val set	test set
Acc	98.11%	97.90%	98.48%

Table 5. Classification accuracy experiment.

Method	TTA	Sample optimization	Ours
Time	7.09s	1.73s	0.45s

Table 6. Different optimization method's inference efficiency.

6. Overview of Material

This supplementary material presents our detailed experiments, implementation, and additional visualizations. Sec. 7 investigates the impact of autoencoder architectures on hand representation. Sec. 9 elaborates on the automatic data annotation engine, followed by the text generation pipeline in Sec. 9.1. In Sec. 9.2, we present the object affordance prediction pipeline with experimental analysis. Sec. 10 and Sec. 11 discuss parameter sensitivity and training objectives. Additional affordance visualizations are provided in Sec. 12. Sec. 13 presents our evaluation scheme for assessing semantic consistency. Finally, Sec. 14 and Sec. 15 demonstrate AffordGrasp's execution in simulation and real-world experiments.

7. Autoencoder Structure

We conduct a comparative analysis of various autoencoder architectures, as detailed in Tab. 4. Empirical results demonstrate that FastGrasp yields superior performance in terms of reconstruction fidelity. Furthermore, FastGrasp achieves a significantly higher compression ratio compared to D-VQVAE [37], enabling more efficient latent representation. Consequently, we adopt FastGrasp as the backbone autoencoder for our framework.

8. Inference Speed

We investigated the integration of test-time adaptation [9, 16] and gradient optimization [27, 32] during the diffusion sampling process. However, we observed that both techniques incurred a substantial computational overhead, significantly compromising the real-time inference efficiency, as shown in Tab. 6. Therefore, we exclude them from our final inference pipeline to ensure efficiency.

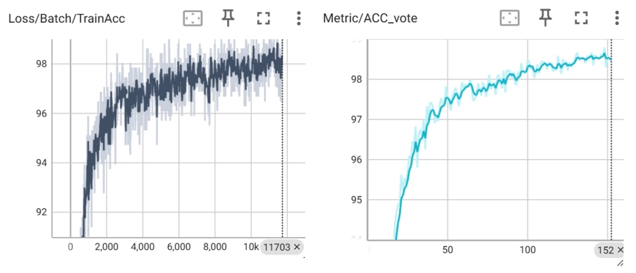


Figure 8. Training acc curve of the model in the AfforPose dataset.

9. Data Engine

Our data engine leverages existing datasets—such as OakInk [31] and GRAB [25]—which primarily contain paired hand–object data. We aim to enrich these datasets with additional modalities through automated annotation, ultimately constructing a dataset that includes hand meshes, object point clouds, textual descriptions, and object affordance labels.

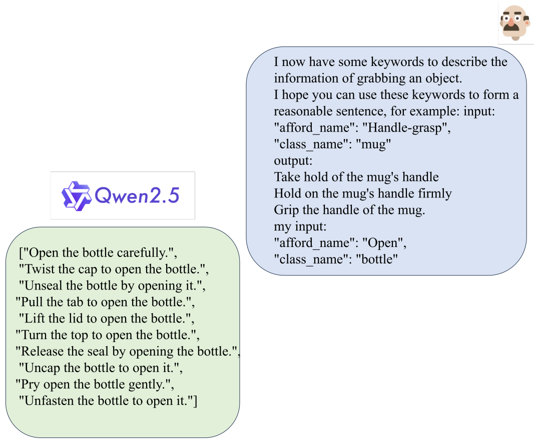
The data pipeline consists of two sequential modules with strict dependencies. The first module generates textual descriptions that characterize hand–object interactions, and the second module infers object affordances based on these generated descriptions. To simplify model training, we adopt an offline pipeline that directly outputs affordance labels. The workflow is strictly unidirectional: the second module is executed only after the successful completion of the first, preserving the dependency between text generation and affordance inference.

9.1. Text Instruction Generation

Semantic Affordance Prediction. The AffordPose dataset [8] provides rich annotations including hand, object, and semantic affordance labels (e.g., *Handle-grasp, No-grasp, Press, Lift, Wrap-grasp, Twist, Support, Pull, Lever, Null*). To leverage this, we design a classifier that takes the joint hand-object point cloud as input to predict the semantic affordance category corresponding to the grasping pose.

The model utilizes a pre-trained PointBERT [33] backbone to process combined hand-object point clouds. The input pipeline consists of: 1) **Canonicalization:** Unifying the point clouds through coordinate system alignment; 2) **Sampling:** Downsampling to $N = 4096$ points via Farthest Point Sampling (FPS) to ensure consistent input dimensions; and 3) **Feature Extraction:** Learning joint features through the transformer backbone.

For the 10-class affordance prediction task, we employ



I now have some keywords to describe the information of grabbing an object.
I hope you can use these keywords to form a reasonable sentence, for example: input:
"afford_name": "Handle-grasp",
"class_name": "mug"
output:
Take hold of the mug's handle
Hold on the mug's handle firmly
Grip the handle of the mug.
my input:
"afford_name": "Open",
"class_name": "bottle"

["Open the bottle carefully.",
"Twist the cap to open the bottle.",
"Unseal the bottle by opening it.",
"Pull the tab to open the bottle.",
"Lift the lid to open the bottle.",
"Turn the top to open the bottle.",
"Release the seal by opening the bottle.",
"Uncap the bottle to open it.",
"Pry open the bottle gently.",
"Unfasten the bottle to open it."]

Figure 9. LLM generates the pipeline of language instructions.

the standard cross-entropy loss:

$$\mathcal{L}_{cls} = -\sum_{i=1}^{C} y_i \log(p_i), \tag{12}$$

where $C = 10$ denotes the number of affordance categories, y_i represents the ground-truth one-hot label, and p_i is the predicted probability for class i.

We employ an initial classifier trained on AffordPose to generate pseudo-labels for the unlabelled OakInk and GRAB datasets. The pipeline proceeds as follows:

1. **High-confidence Pseudo-labeling:** We select reliable predictions by filtering samples based on the output probability distribution, adhering to a strict confidence interval (referencing the 3σ principle).
2. **Human Validation:** To ensure label quality, we conduct manual verification on a subset of 100 randomly sampled instances.
3. **Iterative Refinement:** We adopt a self-training paradigm, jointly training on the original AffordPose data and the validated pseudo-labels. This process is repeated across multiple cycles to progressively expand label coverage until the target datasets are fully annotated.

Instruction Generation. We generate language instructions based on the obtained language affordance and the class name of the object. We use Qwen(Fig. 9) as our instruction generator, and we can get the corresponding instructions through automated methods.

9.2. Object Affordance Prediction

We generate object affordance representations using the annotated dataset to better align linguistic and geometric spatial embeddings. The model is trained on the AffordPose dataset, processing point clouds and language instructions to estimate per-point semantic adherence probabilities. This mapping associates textual semantics with localized operable regions in the point cloud through attention weights

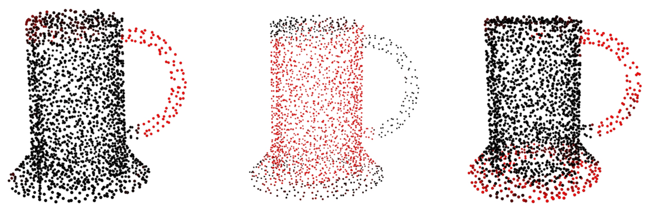


Figure 10. **Object Affordance Visualization.**
Left. Grip the handle of the mug.
Mid. Wrap your hand around the mug.
Right. Support the mug to prevent spills.

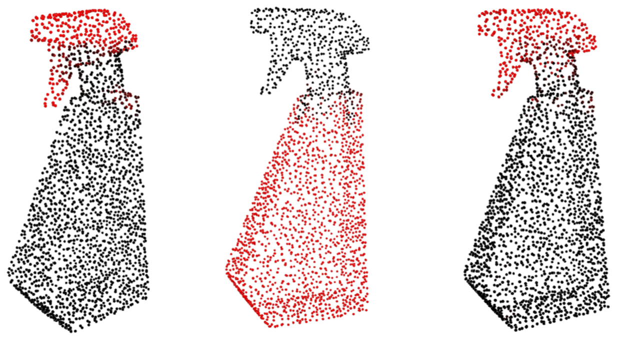


Figure 11. **Object Affordance Visualization.**
Left. Press the dispenser to avoid over-pouring.
Mid. Wrap your fingers around the dispenser for a secure hold.
Right. Support the bottle's handle to prevent spills.

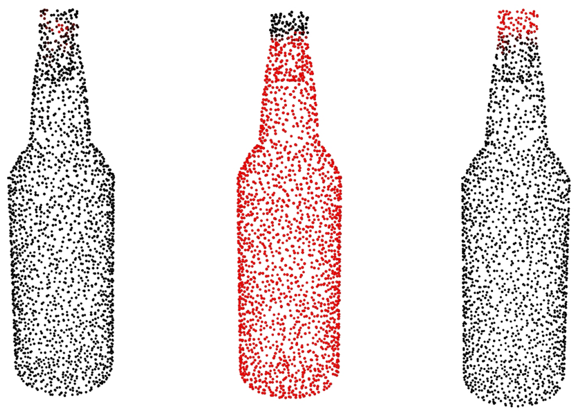


Figure 12. **Object Affordance Visualization.**
Left. Grip the handle of the bottle.
Mid. Wrap-grasp the bottle to prevent spill.
Right. twist the bottle to open it.

$p_{i,j} \in [0, 1]$, where $p_{i,j}$ indicates the focus priority for each spatial region during manipulation tasks. To address the class imbalance in affordance prediction, we optimize using a combined Focal Loss and Dice Loss objective:

	bag	bottle	dispenser	earphone	faucet	handle-bottle	jar	keyboard	knife	laptop	mug	pot	scissors
IoU↑	0.842	0.866	0.850	0.903	0.867	0.867	0.822	0.764	0.899	0.751	0.905	0.836	0.905
AUC↑	0.999	0.988	0.996	1.000	0.999	0.998	0.991	0.986	1.000	0.996	0.999	0.997	0.999
SIM↑	0.943	0.953	0.949	0.986	0.965	0.965	0.924	0.885	0.982	0.876	0.979	0.926	0.982
MAE↓	0.009	0.025	0.018	0.006	0.010	0.012	0.038	0.061	0.008	0.019	0.009	0.017	0.015

Table 7. Assessment for different object categories.

	Handle-grasp	Press	Lift	Wrap-grasp	Twist	Support	Pull	Lever	OVERALL
IoU↑	0.889	0.786	0.872	0.910	0.807	0.723	0.717	0.870	0.855
AUC↑	1.000	0.993	1.000	0.991	0.998	0.990	0.999	0.999	0.996
SIM↑	0.975	0.903	0.965	0.975	0.923	0.851	0.836	0.967	0.948
MAE↓	0.009	0.032	0.001	0.030	0.013	0.029	0.006	0.009	0.019

Table 8. Assessment for different action.

$$\mathcal{L}_{\text{focal}} = -\frac{1}{N_{\text{points}}} \sum_{i=1}^{N} \sum_{j} \Big[\alpha \, (1 - p_{i,j})^{\gamma} \, g_{i,j} \log(p_{i,j}) + (1-\alpha) \, p_{i,j}^{\gamma} \, (1 - g_{i,j}) \log(1 - p_{i,j}) \Big], \tag{13}$$

$$\mathcal{L}_{\text{dice}} = \frac{1}{N} \sum_{i=1}^{N} \left(2 - \text{Dice}_{\text{pos}}^{(i)} - \text{Dice}_{\text{neg}}^{(i)} \right), \tag{14}$$

$$\text{Dice}_{\text{pos}}^{(i)} = \frac{2 \sum_j p_{i,j} g_{i,j}}{\sum_j p_{i,j} + \sum_j g_{i,j} + \epsilon}, \tag{15}$$

$$\text{Dice}_{\text{neg}}^{(i)} = \frac{2 \sum_j (1 - p_{i,j})(1 - g_{i,j})}{\sum_j (1 - p_{i,j}) + \sum_j (1 - g_{i,j}) + \epsilon}, \tag{16}$$

$$\mathcal{L} = \mathcal{L}_{\text{focal}} + \lambda \mathcal{L}_{\text{dice}}, \tag{17}$$

Here, λ is a balancing hyperparameter, N is the batch size, N_{points} is the total number of points in the batch, $p_{i,j}$ is the predicted probability for point j in sample i, and $g_{i,j} \in \{0, 1\}$ is its ground-truth. For $\mathcal{L}_{\text{focal}}$, α and γ are the standard balancing factor and focusing parameters. For $\mathcal{L}_{\text{dice}}$, we use a symmetric formulation by averaging the Dice coefficients for positive (Dice_{pos}) and negative (Dice_{neg}) classes, stabilizing training for imbalanced cases. ϵ is a small constant for numerical stability."

We validate our affordance prediction model through comprehensive benchmarking against state-of-the-art 3D affordance learning methods [1, 13], with detailed results in Tab. 7 and Tab. 8. Our evaluation employs four established metrics: Area Under the Curve (AUC), Mean Intersection Over Union (mIoU), Similarity (SIM), and Mean Absolute Error (MAE), ensuring rigorous comparison across critical performance dimensions.

AUC (Area Under the Curve): In evaluation, AUC measures how effectively the model distinguishes affordance from non-affordance regions within objects. By analyzing classification performance across threshold variations, this metric captures the model's capacity to identify functionally relevant object parts under diverse conditions.

mIoU (Mean Intersection Over Union): This segmentation metric evaluates spatial alignment between predictions and ground truth masks. Computed as the mean IoU across all test samples, mIoU provides a comprehensive measure of segmentation accuracy by quantifying the overlap ratio between predicted and actual regions of interest.

SIM (Similarity): This metric quantifies the alignment between the model's segmentation and the ground truth affordance region specified in the question, measuring the model's ability to interpret textual queries and localize corresponding spatial regions. It is computed as:

$$\text{SIM}(Y, M) = \sum_{i=1}^{n} \min(Y_i, M_i), \tag{18}$$

$$\sum_{i=1}^{n} Y_i = \sum_{i=1}^{n} M_i = 1, \tag{19}$$

where Y and M represent the ground truth and predicted segmentation masks, respectively, and n is the total number of segmentation points (pixels). Both masks are normalized to form probability distributions over the spatial domain.

MAE (Mean Absolute Error): MAE quantifies the total error magnitude between predicted and ground truth affordance segmentations, disregarding directional bias. This metric evaluates the model's pixel-level accuracy in segmenting object parts relevant to linguistic queries, measuring its ability to interpret affordance cues from natural language instructions:

Dataset	Head nums	Penetration Volume ↓	Simulation Displacement ↓	Contact Ratio ↑	Entropy ↑	Cluster Size ↑	ACC ↑
OakInk [31]	1	8.32	1.66	96	2.79	**4.14**	76.33%
	2	7.79	**1.38**	97	2.88	3.66	77.21%
	4	**7.32**	1.43	**98**	**2.94**	3.74	**80.08%**
GRAB [25]	1	5.71	1.25	98	2.79	**3.77**	**62.50%**
	2	5.11	**1.13**	**100**	2.87	3.75	**62.50%**
	4	**3.06**	1.66	94	**2.91**	3.53	**62.50%**
HO-3D [6]	1	8.73	2.5	94	2.81	**3.81**	70.00%
	2	10.99	**1.92**	96	**2.87**	3.66	**70.00%**
	4	**7.38**	2.33	**97**	2.85	3.70	72.00%
AffordPose [8]	1	19.77	**2.41**	96	2.92	**4.19**	63.83%
	2	25.31	2.71	**98**	**2.92**	3.96	63.78%
	4	**10.36**	3.59	91	**2.92**	3.93	**69.71%**

Table 9. **Parameter sensitivity experiment.** We performed parameter sensitivity experiments(Tab. 3) using the same setting

$$\mathsf{MAE}(Y, M) = \sum_{i}^{n} |Y_i - M_i|, \tag{20}$$

where n denotes the total number of points, and Y and M represent the ground truth and predicted segmentation masks respectively.

Visualization results in Figs. 10, 11 demonstrate our model's robustness. Notably, we intentionally introduced incorrect text instructions (Fig. 12, left) to test prediction consistency. Despite input discordance, the model adaptively suppresses spurious affordance predictions, exhibiting increased reliance on global point cloud features. This behavior suggests an inherent bias toward structural coherence over local text-instruction mismatches.

10. Parameter Sensitivity Analysis

We conduct sensitivity analysis on the cross-attention heads in our DAM module, evaluating how different configurations (1, 2, and 4 attention heads) impact grasp generation performance. As shown in Tab. 9.

11. Loss function

Based on Eq. 4, we derive the original output distribution of the diffusion model through:

$$\hat{h}_z = \frac{1}{\sqrt{\alpha_t}}(z^t - \sqrt{1-\alpha_t}\epsilon_\theta(z^t, f, t)), \tag{21}$$

where α_t denotes the noise scheduling parameter. The training objective encompasses both the reconstruction loss and physical constraints as follows [9, 28]:

$$h_p = Decoder(DAM(\hat{h}_z, f)), \tag{22}$$

$$h_m = ManoLayer(h_p), \tag{23}$$

$$\mathcal{L}_{recon} = \lambda_1 \mathcal{L}_{param} + \lambda_2 \mathcal{L}_{mesh}, \tag{24}$$

$$\begin{aligned}\mathcal{L}_{\text{mesh}} = & \frac{1}{|h_v|} \sum_{x \in h_v} \min_{y \in h_v^{\text{gt}}} \|x - y\|_2^2 \\ & + \frac{1}{|h_v^{\text{gt}}|} \sum_{y \in h_v^{\text{gt}}} \min_{x \in h_v} \|y - x\|_2^2.\end{aligned} \tag{25}$$

$$\mathcal{L}_{param} = \text{MSE}(h_p, h_p^{\text{gt}}), \tag{26}$$

where $\mathcal{L}_{param}$ indicates mean squared error loss between predicted h_p and GT hand MANO parameters h_p^{gt}, $\mathcal{L}_{mesh}$ measures chamfer distance between the predicted hand vertices h_v and the GT hand vertices h_v^{gt}, with h_v derived from the hand mesh h_m.

To enforce physically plausible hand representations, we implement three additional constraint losses [9, 28]:

$$\begin{aligned}\mathcal{L}_{\text{consist}} &= \text{Consist}(h_m, h_m^{\text{gt}}, P_g) \\ &= -\lambda_c \frac{\sum_{i=1}^{B} \sum_{p=1}^{N_o} \mathbb{I}\left[d_{\text{pred}}^{(i)}(p) < \tau\right] \cdot \mathbb{I}\left[d_{\text{gt}}^{(i)}(p) < \tau\right]}{\sum_{i=1}^{B} \sum_{p=1}^{N_o} \mathbb{I}\left[d_{\text{gt}}^{(i)}(p) < \tau\right]},\end{aligned} \tag{27}$$

$$\begin{aligned}\mathcal{L}_{\text{cmap}} &= \text{Contact}(h_m, P_g, M_c) \\ &= \frac{1}{B} \sum_{i=1}^{B} \min_{t \in \{1,\ldots,T\}} \frac{\sum_{p=1}^{N_o} M_c^{(i)}(p,t)\, d_{pred}^{(i)}(p)}{\sum_{p=1}^{N_o} M_c^{(i)}(p,t)},\end{aligned} \tag{28}$$

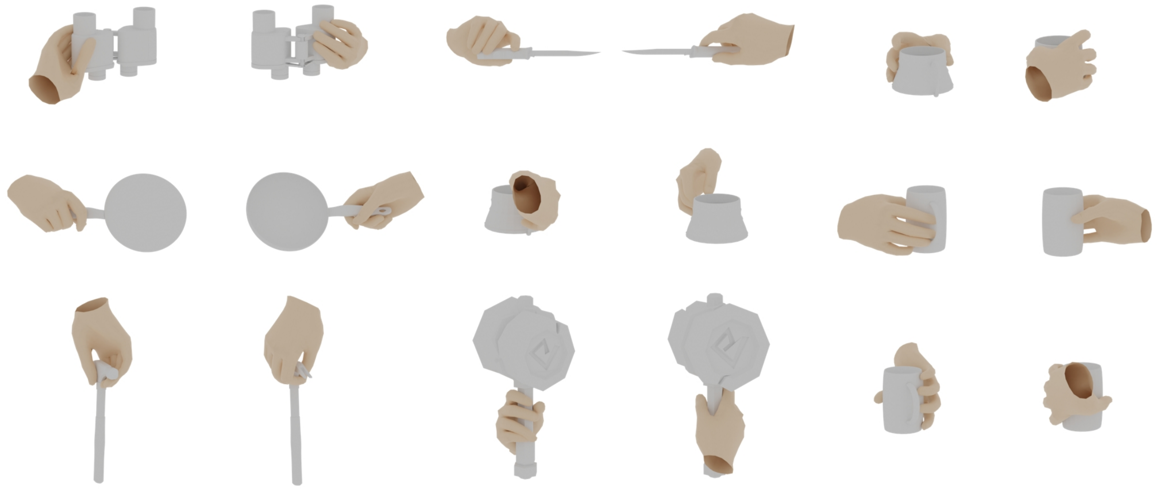


Figure 13. Visualization of randomly selected grasping poses across different objects, each displayed from two viewpoints.

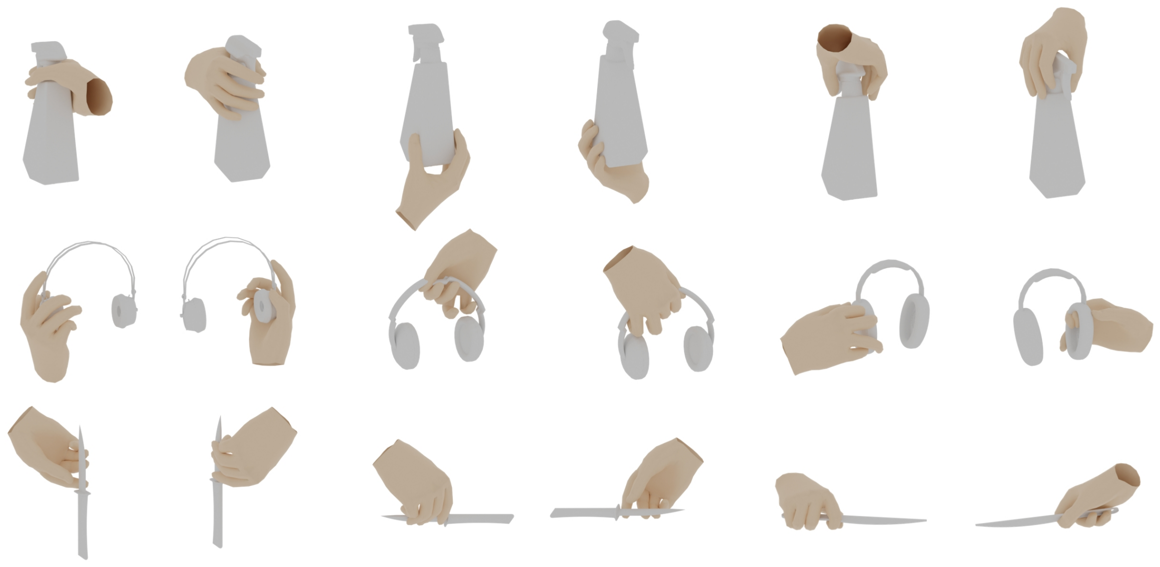


Figure 14. Visualization of three grasping poses for the same object, each shown from two viewpoints.

$$\begin{aligned}\mathcal{L}_{\text{penetr}} &= \text{Penetra}(h_m, P_g) \\ &= \lambda_p \frac{1}{B} \sum_{i=1}^{B} \sum_{p=1}^{N_o} \mathbb{I}\left[\text{Inside}\left(P_g^{(i)}(p), h_m^{(i)}\right)\right] \cdot d_{pred}^{(i)}(p).\end{aligned} \tag{29}$$

The predicted hand mesh's contact region with the object mesh, denoted as P_g, which we aim to grasp, is kept consistent with the ground-truth (GT) hand mesh's contact region through the consistency loss $\mathcal{L}_{\text{consist}}$ (Eq. 27). Specifically, h_m and h_m^{gt} represent the predicted and GT hand meshes, respectively, while P_g denotes the object mesh. B is the batch size, N_o the number of object points, τ the contact distance threshold, and λ_c a weighting coefficient. The distance term $d_{\text{pred}}^{(i)}(p)$ (or $d_{\text{gt}}^{(i)}(p)$) indicates the shortest distance from the p-th object point to the predicted (or GT) hand surface in the i-th sample.

The contact-map loss $\mathcal{L}_{\text{cmap}}$ (Eq. 28) encourages the generated hand mesh h_m to maintain meaningful contact with

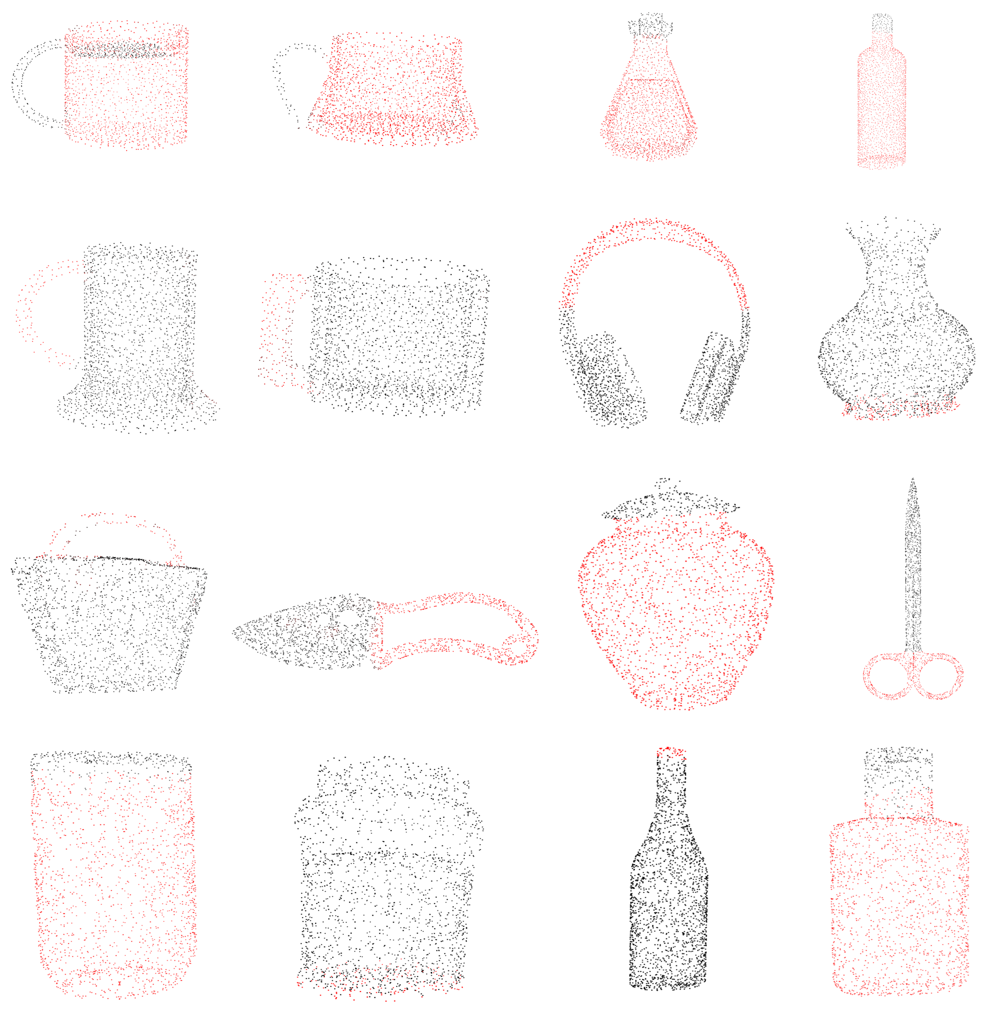


Figure 15. Visualizations of affordance generation for multiple objects.

the object mesh P_g. Here, $M_c^{(i)}(p,t)$ denotes the binary contact mask for the p-th object point at the t-th contact frame, and T is the total number of contact frames considered.

The penetration loss $\mathcal{L}_{\text{penetr}}$ (Eq. 29) serves as a physical constraint to penalize interpenetrations between the hand and object meshes. The indicator $\text{Inside}(P_g^{(i)}(p), h_m^{(i)})$ checks whether the p-th object point lies inside the predicted hand mesh. The coefficient λ_p balances the penalty term's contribution.

Our total loss function for training the DAM (Fig. 3) can be written as:

$$L = L_{recon} + \lambda_3 L_{consist} + \lambda_4 L_{cmap} + \lambda_5 L_{penetr}. \quad (30)$$

The weight balancing coefficients λ_1, λ_2, λ_3, λ_4, λ_5 are used to balance these factors.

Through joint optimization of physical constraints and reconstruction objectives, our model effectively learns the physical interactions between hand meshes and objects, generating grasping poses that conform to natural physical principles. The two-stage training framework ultimately enables single-stage inference without requiring test-time adaptation (TTA), while still producing high-quality grasping configurations.

12. Visualization Result

We provide additional visualizations in Fig. 13, Fig. 14, and Fig. 15. Fig. 13 presents randomly sampled grasping poses generated for different objects, demonstrating the generalization capability of our method. Fig. 14 shows how varying textual prompts for the same object lead to distinct grasping configurations, indicating that textual guidance effectively modulates the generated poses. Finally, Fig. 15 visualizes

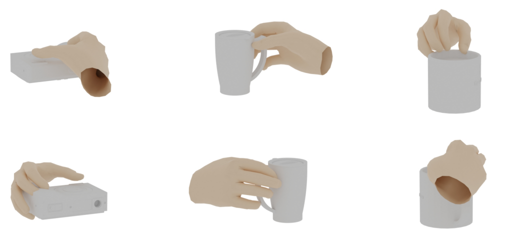


Figure 16. **Failure cases of our method.** Each pair displays a sample from two views.

the outputs of our affordance generator, providing intuitive evidence of how object-level affordance cues guide grasp synthesis.

13. Semantic Accuracy Assessment (ACC)

To evaluate the semantic consistency between the generated grasps and the input instructions, we adopt the taxonomy from AffordPose [8], which defines ten distinct affordance classes: $\{$*Handle-grasp, No-grasp, Press, Lift, Wrap-grasp, Twist, Support, Pull, Lever, Null*$\}$.

We utilize the semantic classifier (described in Sec. 9.1) to assess whether the generated hand pose semantically matches the target object affordance. The prediction process is formally defined as:

$$\mathbf{p} = \text{classifier}(h_v, P_g), \tag{31}$$

where classifier denotes the pre-trained classifier, h_v represents the hand vertices, and P_g is the object point cloud. The output $\mathbf{p} \in \mathbb{R}^{10}$ represents the confidence distribution over the ten affordance categories. The final predicted class $\hat{y}$ is determined by:

$$\hat{y} = \underset{k}{\text{argmax}}(\mathbf{p}), \tag{32}$$

where k denotes the affordance category index. We compute the Semantic Accuracy (ACC) as the percentage of samples for which the predicted class $\hat{y}$ matches the ground-truth text instruction.

14. Simulation Experiment

To evaluate the performance of **AffordGrasp** in simulation, we conduct controlled experiments in a physics-based environment. Our pipeline first applies **CrossDex** to map MANO-based grasp predictions generated by AffordGrasp into the ShadowHand joint space. This remapping ensures kinematic feasibility and prevents implausible joint configurations.

The adapted ShadowHand grasps are then used as *goal rewards* within the CrossDex reinforcement learning framework. During training, the policy is guided toward these target grasps, enabling AffordGrasp's grasp priors to be executed as physically consistent behaviors in simulation.

We report grasp success rates, MPJDR, and FOL metrics, along with qualitative visualizations to demonstrate the effectiveness of our approach. As shown in Tab. 10 and Fig. 17, AffordGrasp achieves a success rate comparable to CrossDex. Our method reaches an MPJDR of 0.161 and an FOL of 0.235, indicating that the RL policy produces grasp trajectories that are more physically consistent and stable under the guidance of the static reference. Visualizations further confirm that RL performs effective grasps with our affordance guidance.

Method	Success Rate (%)	MPJPR ↓	FOL ↓
CrossDex[34]	92.90	-	0.260
AffordGrasp(Ours)	**92.96**	0.161	**0.235**

Table 10. Quantitative evaluation of AffordGrasp in the simulation environment.

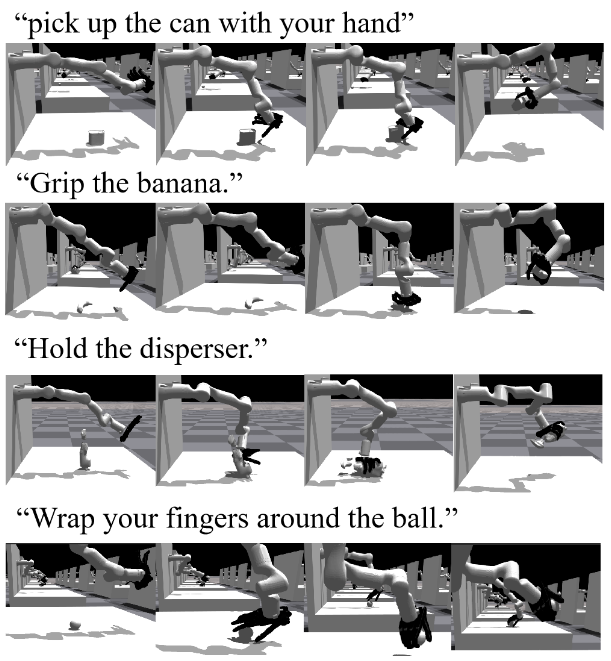
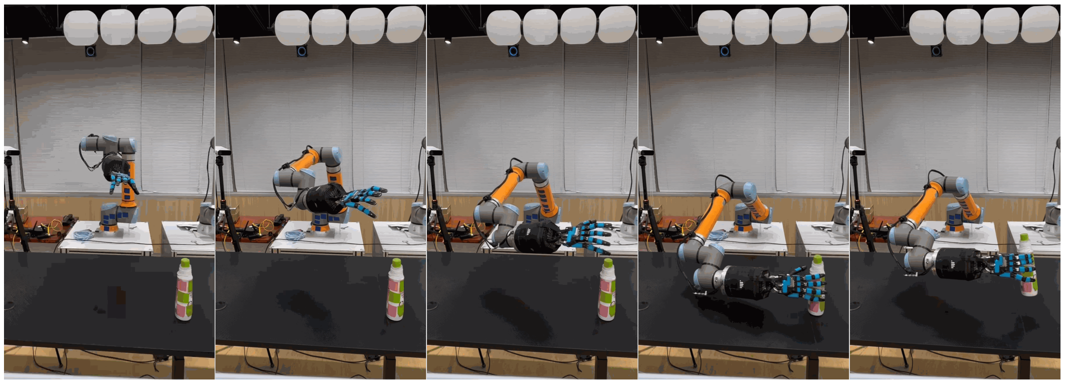
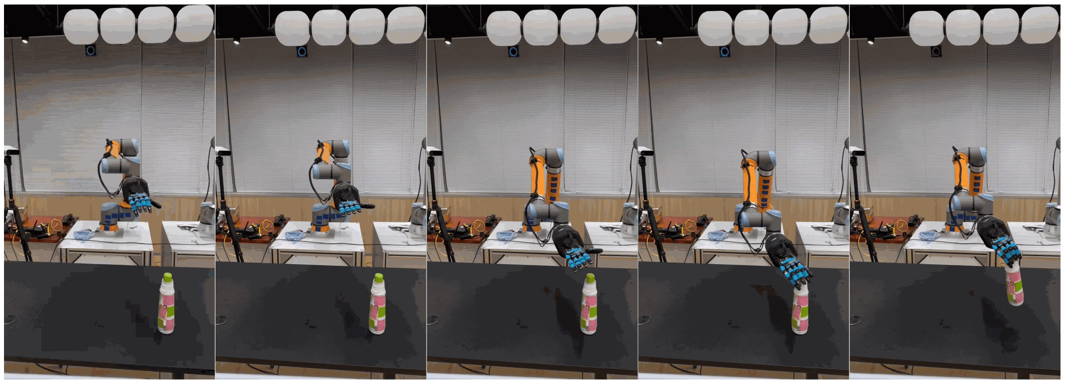

"pick up the can with your hand"
"Grip the banana."
"Hold the disperser."
"Wrap your fingers around the ball."

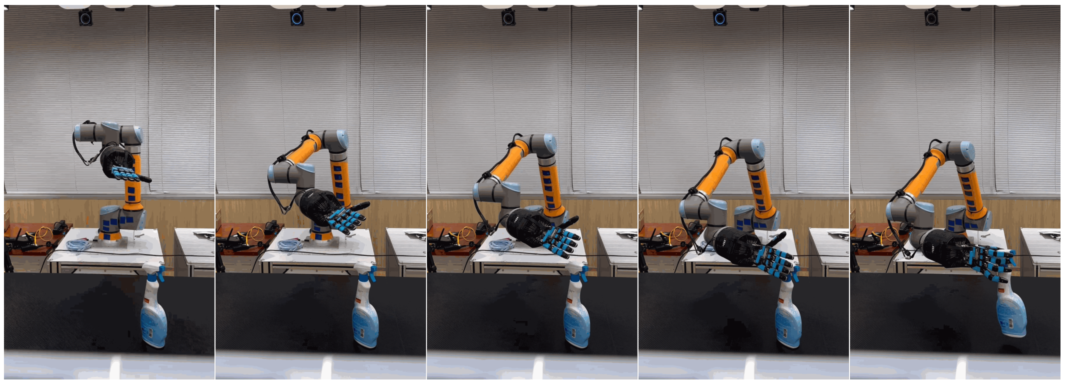
Figure 17. **Grasp execution results guided by AffordGrasp in the simulation environment.**

15. Test on Real Robot

We further evaluate AffordGrasp on a real robot. As shown in Fig. 18, we execute the generated motions using the ShadowHand platform. For the same object, different language instructions lead to diverse execution outcomes. Throughout the entire process, the hand maintains behavior that is consistent with the input semantics, which meets the essential requirements for successful grasping.


(a) Instruction: Wrap your hand around the bottle.


(b) Instruction: Twist the top of the bottle.


(c) Instruction: Press the dispenser to pour.

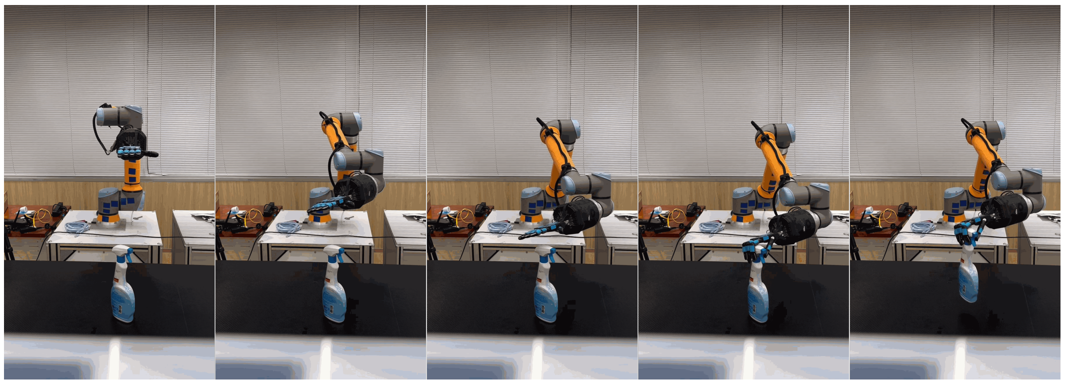

(d) Instruction: Twist the top of the dispenser to open it.

Figure 18. Real robot visualization.