AffordGrasp: Cross-Modal Diffusion for Affordance-Aware Grasp Synthesis

Supplementary Material

Method	Simulation Displacement ↓	Penetration Volume ↓	Penetration Distance ↓	Contact Ratio ↑
D-VQVAE [39]	**1.61**	0.94	5.17	98
FastGrasp [30]	1.83	**0.91**	**2.39**	**98**

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

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

Table 5. Different optimization method's inference efficiency.

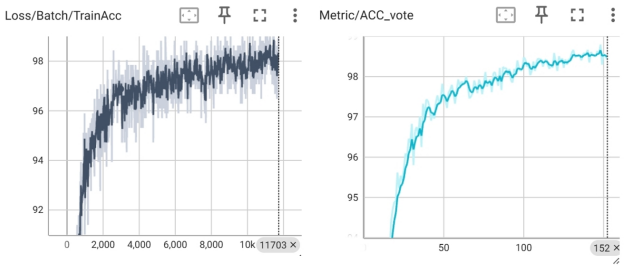

Figure 8. Training acc curve of the model in the afforPose dataset.

ACC	train set	val set	test set
classifier [39]	98.11%	97.90%	98.48%

Table 6. Classification accuracy experiment.

7. Overview of Material

The supplementary material offers a comprehensive overview of our experiments, results, and visualizations. Sec. 8 examines the impact of different autoencoder architectures on hand representation extraction. Sec. 10 details our automatic data annotation engine, while Sec. 10.1 describes the text generation pipeline and experimental setup. Sec. 10.2 presents the object affordance prediction pipeline, followed by an analysis of experimental results and visualizations. Finally, Sec. 11 explains why affordance cannot serve as training data. Sec. 12 presents the parameter sensitivity analysis, Sec. 13 outlines the training loss function, and Sec. 14 provides additional visualizations. Show more visualization object affordance.

8. Autoencoder Structure

We evaluated various autoencoder architectures (see Tab. 4) and found that FastGrasp outperformed the others. Additionally, FastGrasp achieved a significantly higher compression rate than D-VQVAE [39], leading us to select it as our autoencoder.

9. Inference Speed

We tried using test-time adaptation [11, 17] and gradient optimization [29, 34] during the diffusion model sampling process. We found that both methods significantly impacted the model's inference efficiency as show in Tab. 5.

10. Data engine

Our data engine mainly makes the best use of current data sets for additional annotation, such as OakInk [33]and GRAB[27]data sets, which only contain hand-object pair. We hope to enhance this data set with additional information through automated annotation methods. Finally, we get a dataset containing hand, object, text, and object affordance.

Our data pipeline comprises two sequential modules operating in strict dependency. The first module generates textual descriptions characterizing hand-object interactions, while the subsequent module deduces object affordance from these text descriptions. To streamline model training, we employ an offline methodology that directly produces affordance labels. The system enforces a unidirectional workflow where the second module's execution is contingent upon the successful completion of the first module, maintaining critical dependency between the text generation and affordance inference stages.

10.1. Text Instruction Generation

Language Affordance Predict. We observed that the AffordPose [10]dataset contains hand, object, and language affordance, such as annotations like "Handel-grasp, wrap-grasp". We consider training a classifier first, inputting the joint point cloud of the hand and the object, and predicting the language affordance corresponding to the grasping pose.

The model architecture builds on pre-trained Point-BERT [36] weights, processing combined hand-object point clouds through the following pipeline: 1) Point cloud unification through coordinate system alignment, 2) Uniform sampling to 4096 points via Farthest Point Sampling (FPS) for consistent input dimensions, and 3) Joint feature learning through the transformer backbone.

For the 9-class affordance prediction task, we employ standard cross-entropy loss:

$$\mathcal{L} = -\sum_{i=1}^{C} y_i \log(p_i) \tag{10}$$

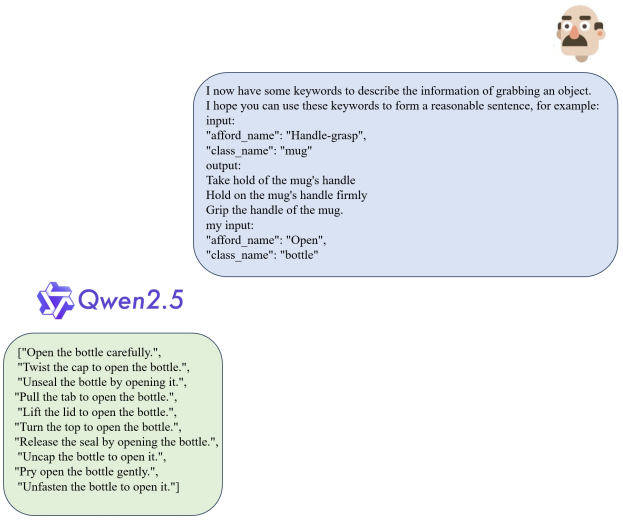

Figure 9. LLM generates the pipeline of language instructions.

where $C = 9$ denotes the number of affordance categories, y_i represents the ground-truth one-hot label, and p_i the predicted probability for class i.

We employ AffordPose's pre-trained classifier $Model_1$ (initialized with cold-start data) to generate pseudo-labels for the OakInk and GRAB datasets through a semi-supervised pipeline: 1) High-confidence pseudo-labeling: Using the 3σ principle (99.73% confidence interval) from the classifier's output distribution to select reliable predictions. 2) Human Validation: Manual verification of 100 randomly sampled instances to assess label quality. 3) Iterative refinement: Joint training on original AffordPose data and validated pseudo-labels across multiple cycles. This self-training paradigm progressively expands label coverage until achieving complete annotation of both target datasets.

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.

10.2. Object Affordance Generation

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 $v_i \in [0, 1]$, where vi 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:

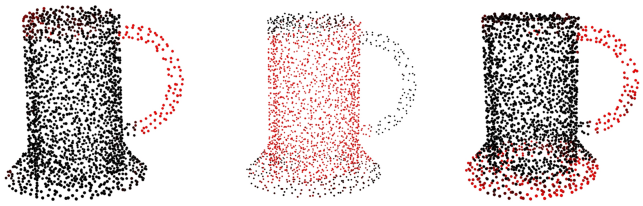

Figure 10. **Object Affordance Visualization.**
Left. Grip the handle of the mug securely.
Mid. Wrap your hand around the mug.
Right. Carefully support the mug to prevent spills.

Figure 11. **Object Affordance Visualization.**
Left. Carefully 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.

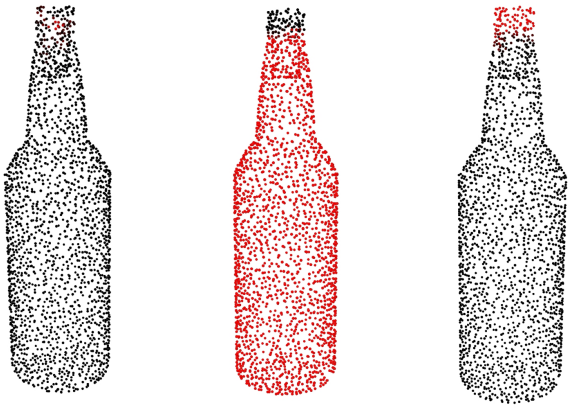

Figure 12. **Object Affordance Visualization.**
Left. Grip the handle of the bottle securely.
Mid. wrap-grasp the bottle to prevent split.
Right. twist the bottle to open it.

	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	0.000	0.030	0.013	0.029	0.006	0.009	0.019

Table 8. Assessment for different action.

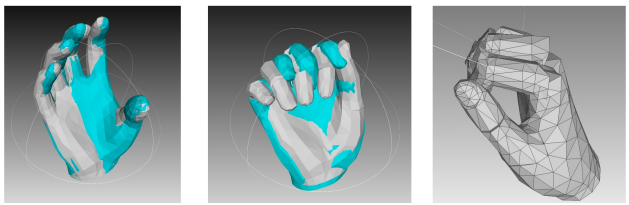

Figure 13. **Schematic diagram of fitting Mano parameters.** The white mesh represents the ground truth (GT), and the blue mesh indicates the fitted hand mesh.
Left. Fit the **Mano** hand mesh using Mano parameterized characterization.
Mid. Fit the **AffordPose** hand mesh using Mano parameterized characterization.
Right. AffordPose's hand mesh.

$$L_{focal} = -\alpha(1-p_t)^{\gamma} log(p_t) \tag{11}$$

$$L_{dice} = \sum_i mean(1.5 - dice_{pos} - dice_{neg}) \tag{12}$$

$$L = L_{focal} + L_{dice} \tag{13}$$

where p_t is the predicted probability for the positive class (pred) and the negative class ($1 -$ pred). The parameter α is a balancing factor that adjusts the relative weight of positive and negative samples, while γ is the focusing parameter, controlling the model's emphasis on hard-to-classify instances. dice_{pos} and dice_{neg} are the Dice coefficients for the positive and negative classes, respectively.

We validate our affordance prediction model through comprehensive benchmarking against state-of-the-art 3D affordance learning methods [2, 15], 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:

$$\begin{aligned} \text{SIM}(Y, M) &= \sum_{i=1}^{n} \min(Y_i, M_i), \\ \text{s.t.} \quad \sum_{i=1}^{n} Y_i &= \sum_{i=1}^{n} M_i = 1, \end{aligned} \tag{14}$$

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 Erro): 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 lan-

Dataset	Head nums	Penetration Volume ↓	Simulation Displacement ↓	Contact Ratio ↑	Entropy ↑	Cluster Size ↑	ACC ↑
OakInk [33]	1	6.71	1.72	96	2.78	4.04	83.67%
	2	5.29	**1.52**	96	2.87	4.13	83.89%
	4	**4.99**	1.54	**97**	**2.89**	**4.18**	**83.96%**
GRAB [27]	1	5.40	1.15	98	2.79	**3.78**	**67.50%**
	2	5.03	**0.85**	**100**	2.87	3.75	**67.50%**
	4	**2.91**	1.21	**100**	**2.88**	3.48	**67.50%**
HO-3D [9]	1	8.45	2.5	99	2.82	3.61	71.00%
	2	12.31	**1.82**	99	**2.87**	3.46	**74.00%**
	4	**8.04**	2.12	**99**	2.82	**3.94**	66.00%
AffordPose [10]	1	19.54	**2.33**	96	2.92	**4.29**	63.83%
	2	22.61	2.52	**98**	2.90	4.07	63.78%
	4	**12.23**	4.34	95	**2.92**	4.10	**64.19%**

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

guage instructions:

$$\mathsf{MAE}(Y, M) = \sum_{i}^{n} |Y_i - M_i|, \tag{15}$$

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.

11. Mesh fitting

The parametric modeling approach in AfforPose demonstrates fundamental incompatibility with our framework due to its non-differentiable rigid-body formulation. While MANO's differentiable statistical model permits direct gradient-based optimization through our loss function (Eq. 25), AfforPose's discrete articulation mechanics disrupt backpropagation gradients.

Our comparative analysis reveals irreducible representation conflicts: MANO's PCA-based shape space models continuous deformation, whereas AfforPose constructs hands through kinematic chains of rigid segments. This structural dichotomy introduces significant mesh artifacts in AfforPose data (Fig. 13, right), particularly near joint regions. Native MANO parameterization achieves superior surface reconstruction (Fig. 13, left), while AfforPose-to-MANO fitting produces geometrically inconsistent results (Fig. 13, mid) due to incompatible deformation paradigms.

This fundamental incompatibility between AfforPose's kinematic assembly and MANO's continuous shape space precludes effective parameter transfer, suggesting these representations occupy distinct manifold regions unsuitable for direct optimization.

12. 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.

13. Loss function

Based on the formulation 3, we derive the original output distribution of the diffusion model through:

$$h_c = h_z + \epsilon_t - \epsilon_\theta(z_0^t, t, f) \tag{16}$$

Here h_c signifies the diffusion model's output for the coarse-grained hand grasping pose. The training objective encompasses both the reconstruction loss and physical constraints as follows [11, 30]:

$$h_p = Decoder(DAM(h_c, f)) \tag{17}$$
$$h_m = ManoLayer(h_p) \tag{18}$$
$$\mathcal{L}_{recon} = \lambda_1 \mathcal{L}_{param} + \lambda_2 \mathcal{L}_{mesh} \tag{19}$$
$$\mathcal{L}_{mesh} = \text{Chamfer-Dis}(h_v, h_v^{\text{gt}}) \tag{20}$$
$$\mathcal{L}_{param} = \text{MSE}(h_p, h_p^{\text{gt}}) \tag{21}$$

$\mathcal{L}_{param}$ 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 being derived from h_m sample.

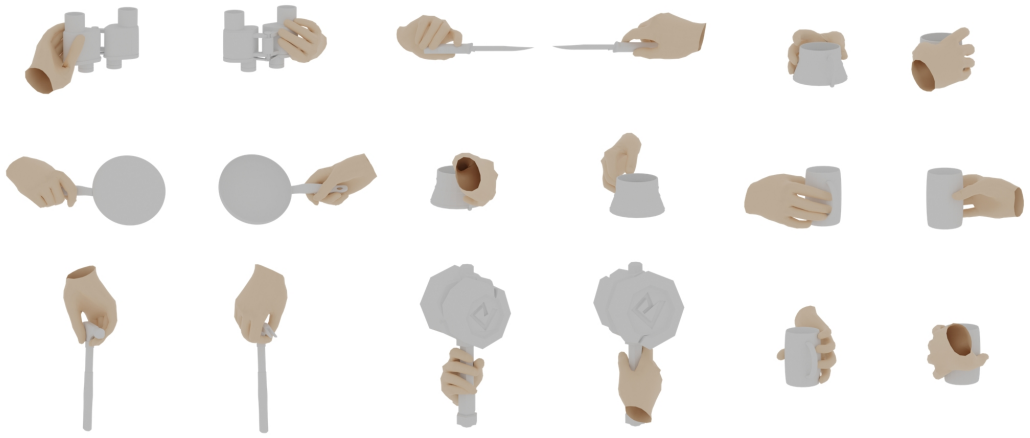

Figure 14. The visualization results, we randomly selected grasping poses for different objects, each shown from two different perspectives.

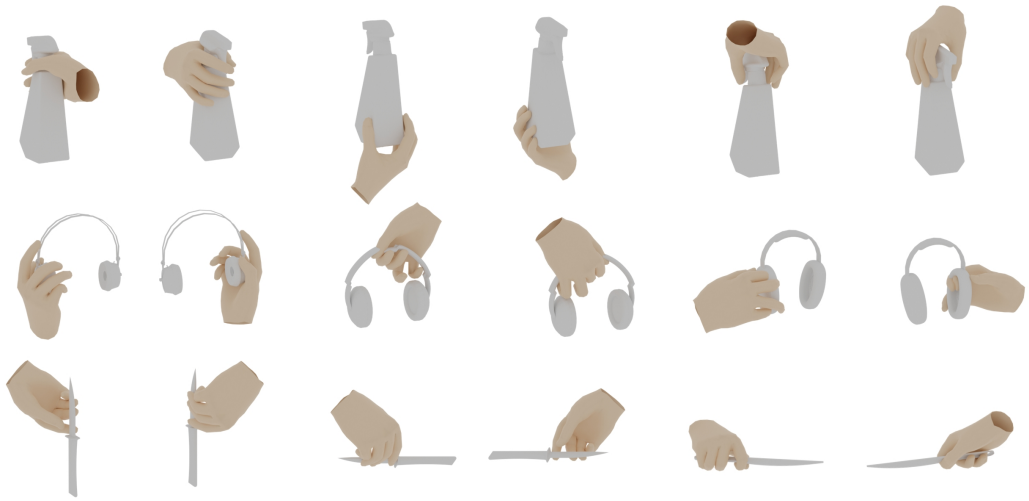

Figure 15. The visualization results, we selected three grasping poses for the same object, each shown from two different perspectives.

To enforce physically plausible hand representations, we implement three additional constraint losses [11, 30]:

$$\mathcal{L}_{consist} = \text{Consist}(h_m, h_m^{gt}, o_m) \quad (22)$$

$$\mathcal{L}_{cmap} = \text{Contact}(h_m, o_m) \quad (23)$$

$$L_{penetr} = \text{Penetra}(h_m, o_m) \quad (24)$$

The predicted hand mesh's contact region with the object mesh, denoted as o_m, that we aim to grasp is kept consistent with the GT hand mesh's contact region thanks to the $\mathcal{L}_{consist}$ loss function. The $\mathcal{L}_{cmap}$ the hand mesh, generated by the model, maintains contact with the object. $\mathcal{L}_{penetr}$ acts as a deterrent to physical volume penetration between hand mesh and the objects.

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

$$L = L_{recon} + \lambda_3 L_{consist} + \lambda_4 L_{cmap} + \lambda_5 L_{penetr} \quad (25)$$

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.

14. Visulization Result

We provide additional visualizations in Fig. 14 and Fig. 16. Fig. 14 shows randomly sampled grasping poses generated for different objects, demonstrating the method's generalization capabilities. Fig. 16 illustrates how varying textual prompts for the same object yield distinct grasping configurations. Our visual analysis reveals that modifying textual input significantly alters the generated poses, confirming our method's responsiveness to language guidance.

Figure 16. The visualization object affordance.