2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

AdapGrasp: A Stiffness and Grasp Affordance Dataset with a Transformer-based Adaptive Grasp Model

Menghao Pu, *Student Member, IEEE,* Chaoqun Han, Zhiping Chai, Tiyong Zhao, Dunxuan Wu, Pu Wen, Xingxing Ke, *Member, IEEE*, Han Ding, *Senior Member, IEEE*, Zhigang Wu, *Member, IEEE*

***Abstract*— Robotic grasp has been employed in various industrial, household, and medical applications. However, neglecting the final grasping state and objects' stiffness and affordance, prevailing strategies predominantly emphasize the grippers' initial state upon reaching grasp positions and often fail due to damage or grasp slippage. Here, we propose an AdapGrasp strategy with a dataset named AdapGraspDataset and a corresponding model named AdapGraspNet. The dataset focuses on the object stiffness and grasp affordance. Specifically, for objects with different stiffness properties, the corresponding final grasp width (FGW) is annotated to ensure the object's intactness. For objects' grasp affordance properties, higher grasp affordance weight (GAW) is typically annotated closer to the centroid, increasing grasp stability. Meanwhile, to output the set of grasping configurations (initial and final grasp states) more accurately, a denoising principle is introduced to build a corresponding transformer-based model. It enables more accurate convergence of FGW and GAW, achieving a precision of 98.04% and a mean absolute final width error of 2.71 pixels. Finally, extensive real-world experiments are conducted, where the AdapGrasp strategy ensures the intactness of fragile objects and thus enhances grasping stability without any additional sensors. It achieves a grasping accuracy of 95% and yields a 19.5% improvement compared with those without FGW and GAW. The AdapGrasp strategy is publicly available at https://embodied-soft-intelligence.github.io/AdapGrasp/.**

I. Introduction

Robotic grasp has been employed in various scenarios, including industrial [1], household [2], and medical fields [3]. These researches primarily utilize deep learning technology to output initial grasp configurations, and then use a rigid gripper to grasp [4], [5], achieving relatively high performance in certain objects. For different scenarios, diverse datasets are employed, such as Cornell Dataset [6], Jacquard Dataset [7], Clutter Dataset [8], and Multi-Object Dataset [9]. They primarily contain common tools or food, annotated using rectangle boxes, denoting the initial configurations of the gripper, in Fig. 1b. However, in these datasets, where the gripper should stop closing depends on a predefined distance or force threshold value, which directly neglects objects' inherent characteristics, such as stiffness and the grasp affordance caused by the shape and mass of objects. Since each object has a different stiffness, a longer closing distance (from the initial width to the final width) may cause low-stiffness or lightweight objects collapsed or deformed, whereas a shorter closing distance may result in grasp failure for rigid and heavy objects [10]. Similarly, a fixed force threshold may not be universally applicable due to varying object tolerance to force. Therefore, the stiffness properties should be carefully considered.

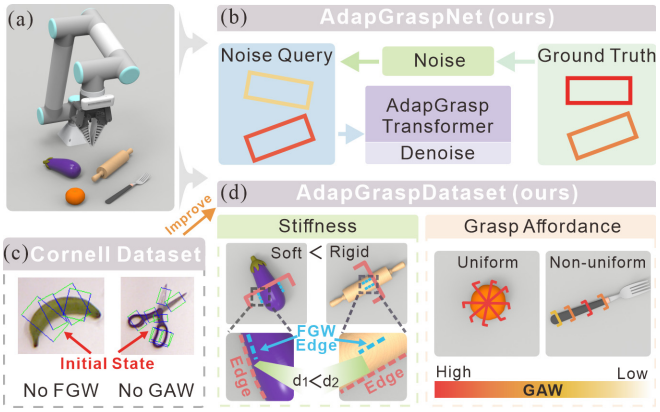

Figure 1. Overview of AdapGraspDataset, AdapGraspNet, and comparison with the Cornell dataset. (a) Schematic of grasping. (b) Training workflow of AdapGraspNet. (c) Annotation format of the Cornell dataset. (d) Annotation format of AdapGraspDataset.

Meanwhile, these datasets have uniformly distributed grasp boxes with identical affordance at each grasp position, in Fig. 1c. This representation does not align with the human's intuition. For instance, when humans generate an intention to grasp, several grasp positions are considered, each varying in optimality [11]. Supposing that all of the grasp positions are annotated with equal affordance, the robot may inaccurately prioritize suboptimal positions, such as the edge regions of objects, leading to slippage or an unstable grasp [12]. Therefore, the grasp affordance should be carefully considered accordingly.

To address the above limitations, we propose a feasible strategy, **AdapGrasp**, including AdapGraspDataset and AdapGraspNet. **Final grasp width (FGW)** and **grasp affordance weight (GAW)** are introduced to the dataset, considering the object's stiffness and grasp affordance in Fig. 1d. To account for the stiffness properties, FGW represents the

*The study is partly supported by National Natural Science Foundation of China (52575018, 52188102), National Key Research and Development Program of China Under Grant (2024YFB4707902), Cross-research Support Program of Huazhong University of Science and Technology Under Grant (2024JCYJ036), Open Projects of State Key Laboratory of Intelligent Manufacturing Equipment and Technology under Grant (IMETKF2025013).

Menghao Pu and Chaoqun Han contributed equally to this work. Menghao Pu, Chaoqun Han, Zhiping Chai, Tiyong Zhao, Dunxuan Wu, Pu Wen, Han Ding, and Zhigang Wu are with the School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan, China (email: pumh@hust.edu.cn, chaoqun_han@hust.edu.cn, zhipingchai@hust.edu.cn, tiyongzhao@hust.edu.cn, dunxuanwu@hust.edu.cn, wenpu@hust.edu.cn, dinghan@hust.edu.cn, corresponding author e-mail: zgwu@hust.edu.cn). Xingxing Ke is with the School of Mechanical Engineering and Automation, Fuzhou University, Fuzhou, China (email: xxke@fzu.edu.cn). Zhigang Wu is also partly with Shenzhen Loop Area Institute, Shenzhen, China (email: Zhigangwu@slai.edu.cn).

final state of the gripper. For soft or fragile objects, FGW edges are positioned closer to the object's edges to prevent excessive deformation or damage. Conversely, for the rigid and heavy objects, FGW edges are positioned farther from the objects' edges to ensure a stable grasp. For the grasp affordance, each grasp box is assigned a GAW, which aligns with human intuition. Positions with a higher success rate, such as the object's centroid, receive higher GAW, while those with a low success rate receive lower GAW. These designs make the dataset more consistent with human grasp behavior. The dataset contains a total of 3,000 images, comprising 22 object categories and 28,547 grasp boxes. It includes soft or fragile objects, such as cakes and eggplants, as well as rigid objects, such as screwdrivers and peelers. It covers 22 distinct categories, providing a comprehensive benchmark for grasping diverse objects. To better predict FGW and GAW accurately, a denoise principle is introduced to the transformer-based model, which can output a set of initial and final grasp configurations, in Fig. 1b. Existing models primarily utilize hand-designed Non-Maximum Suppression (NMS) [13], [14] or the hand-designed pixel-wise label assignment strategies [4], [15], [16]. These methods heavily rely on hand-designed parameters, which can significantly impact model performance [17], while AdapGraspNet employs Hungarian Matching in the query part to achieve one-to-one matching, avoiding the need for NMS. In the denoise part, diverse noise perturbations are added to the ground truth, and the model is trained to denoise, accelerating transformer-based model training efficiency [18]. Moreover, it achieves high performance without pixel-wise-label conversion, significantly simplifying the learning process and enhancing the convergence of FGW and GAW. It is trained on AdapGraspDataset for 50 epochs, achieving 98.04% precision and a final width mean absolute error of 2.71 pixels. Finally, experiments are conducted using a fin-ray flexible gripper, where the AdapGrasp strategy ensures the intactness of fragile objects and grasping stability without any additional sensors, achieving a grasping accuracy of 95% with a 19.5% improvement over those without FGW and GAW.

The work's main achievements are summarized as follows.

1) Considering object stiffness and grasp affordance, we propose an adaptive grasp strategy (AdapGrasp) including a dataset (AdapGraspDataset) and a corresponding model (AdapGraspNet) to ensure the intactness of fragile objects and the stability of grasping.

2) The FGW and GAW, which consider object stiffness and grasp affordance, are introduced into AdapGraspDataset, containing a total of 3,000 images, comprising 22 object categories and 28,547 grasp boxes.

3) Leveraging the denoising principle, AdapGraspNet is able to consider the object stiffness and grasp affordance, leading to improved grasping performance.

4) With a fin-ray flexible gripper, AdapGrasp is proven to ensure the safe intactness of fragile objects and thus enhance grasping stability.

II. Related Work

A. Robotic Grasping Dataset

Jiang et al. [6] proposed the Cornell Dataset, introducing a grasping rectangle representation. The representation accounts for the initial grasp state, including initial grasp position, grasp angle, and initial grasp width. This dataset serves as a valuable benchmark. Depierre et al. [7] used a simulation environment to automate the sample labeling, then introduced the Jacquard Dataset and its corresponding criterion, which have been adopted in subsequent research. Chu et al. [19] introduced a dataset, named the Multi-Object Dataset, to address the limitation of the Cornell Dataset, which only contains single-object images. This dataset primarily focuses on multi-object scenarios. Wang et al. [8] introduced the Clutter Dataset, which focuses on grasping in cluttered environments. Each image contains one to ten stacked objects, providing a challenging benchmark. Guo et al. [20] proposed a dataset about fruits and vegetables, which utilizes the grasping rectangle representation to annotate fruits and vegetables.

Although the existing datasets have involved diverse objects, their representations are all based on the Cornell Dataset, which considers only the initial grasp state and assigns equal GAW to all grasp boxes. This limitation highly risks damage to soft objects and degrades the grasping performance in real-world scenarios.

B. Robotic Grasping Algorithms

Lenz et al. [13] presented a two-step cascaded network to output a robotic grasping configuration. The first network generates potential rectangles exhaustively, and the second network filters these rectangles. This approach lacks effectiveness and efficiency. Morrison et al. [15] proposed GG-CNN, which requires converting rectangle labels into pixel-wise labels, to achieve real-time grasp tasks. However, this approach relies on a hand-designed threshold, with the one-third central region manually specified as the graspable area. Kumra et al. [16] improved GG-CNN by incorporating residual blocks and proposed GR-CNN, which enhances the performance. Wang et al. [4] proposed TF-Grasp, which utilizes the transformer architecture to address grasping tasks based on pixel-wise-label conversion. Cheng et al. [14] proposed a single-stage anchor-free algorithm, which can achieve high grasp detection efficiency, but it needs to use the hand-designed NMS.

In brief, these existing robotic grasping algorithms have achieved high performance. However, they often incorporate hand-designed components in label assignment strategies or during model inference, which may require manual tuning across different scenarios or datasets and can impact the robustness of robotic grasping.

III. Proposed Approach

A. Data Collection of AdapGraspDataset

AdapGraspDataset primarily includes objects from the following categories: Tools/Hygiene, Kitchenware/Tableware, Foods, refer to Fig. 2. Tools/Hygiene include objects such as screwdrivers, brushes, toothbrushes, eyedroppers, and others. Kitchenware/Tableware include objects such as peeler, rolling pin, cup, and others. Foods include objects such as apple, cake, grape, egg, and others.

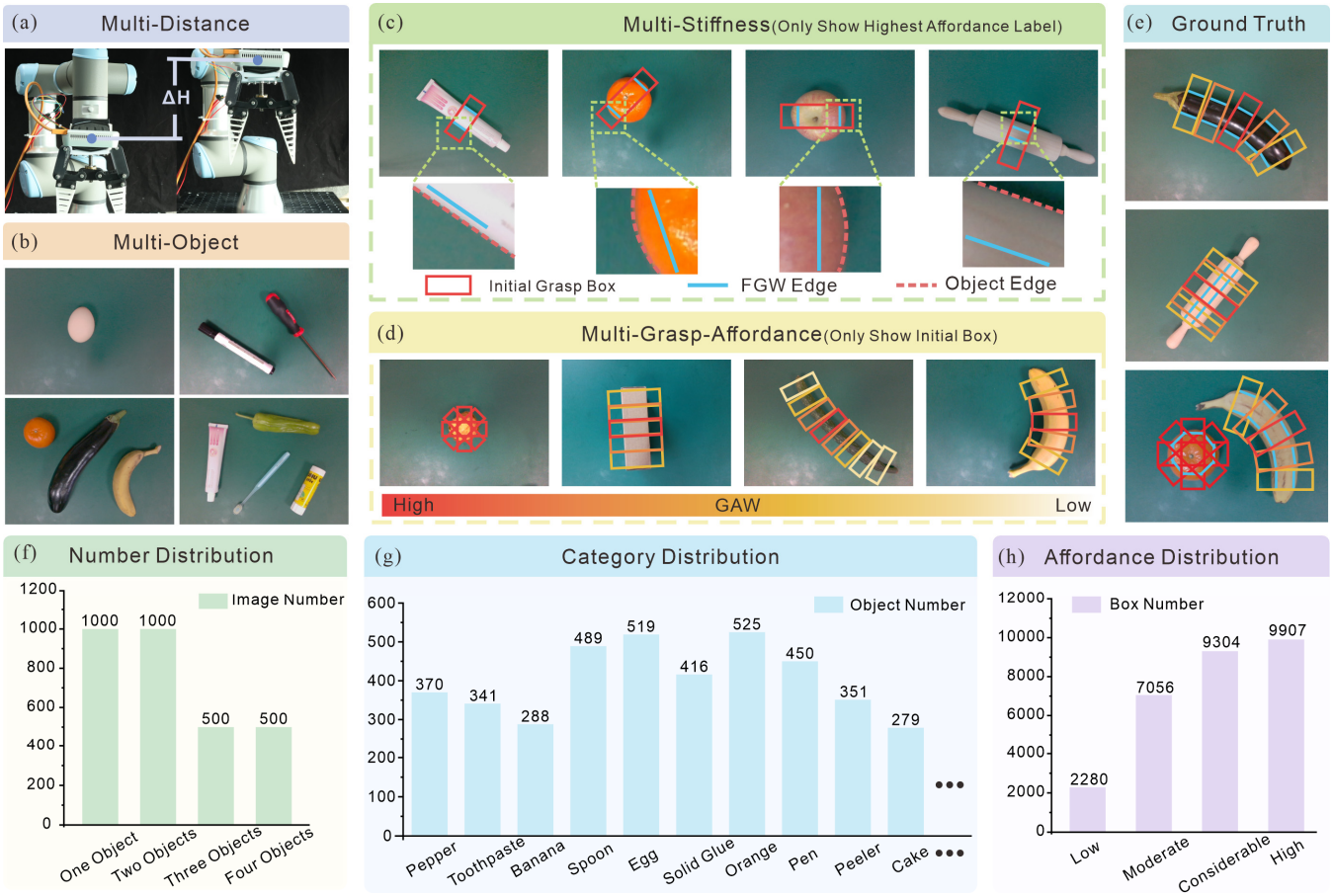

Figure 2. The annotation format and distribution of AdapGraspDataset. (a) Multi-distance, (b) Multi-object, (c) Multi-stiffness, and (d) Multi-grasp affordance annotations. (e) The examples of ground truth. (f–h) Distributions of object number, object category, and grasp affordance.

To enhance the diversity of the dataset, different distances between the camera and objects are employed, in Fig. 2a. There are various combinations when collecting data, resulting in not only images of individual objects but also images of multiple object categories, in Fig. 2b. It includes 22 object categories, with a total of 3,000 images, in Fig. 2f, 2g.

B. Data Annotation of AdapGraspDataset

To formally describe the grasp, we denote a grasp configuration as,

$$G = \left(x, y, w, h, \theta, FGW, GAW\right), \tag{1}$$

where $\left(x, y\right)$ represents the grasp center, w represents the initial grasp width, h represents the width of the gripper, θ represents the grasp orientation, and FGW and GAW correspond to the final grasp width and grasp affordance weight, respectively.

To incorporate the stiffness properties and the grasp affordance properties, we introduce FGW and GAW, as shown in Fig. 2c, 2d. To account for the stiffness properties, each object category should be assigned a different final grasp configuration. For soft or fragile objects, such as toothpastes and oranges, FGW edges are positioned closer to the object's edges to ensure their intactness. For rigid and heavy objects, such as rolling pins, FGW edges are positioned farther to the object's edges to ensure the stability of grasping, in Fig. 2c. Experiments are conducted to determine the optimal position of FGW, preventing damage to the object.

To account for the grasp affordance, each grasp position of an object is assigned a GAW. Following human intuition, we conducted a manual annotation. Five volunteers are invited to label grasp positions for each object and assign a corresponding GAW. We define GAW into four levels, including level 0 (0.125), level 1 (0.375), level 2 (0.625), and level 3 (0.875), where level 3 represents the strongest grasp affordance. The final GAW for each grasp position is obtained by averaging the annotations from all participants. To ensure the reliability of the annotations, we conduct experimental validation of the GAW labels for each object, ultimately confirming their effectiveness, in Fig. 2d. A total of 28547 grasp configurations are annotated, each assigned a corresponding GAW, in Fig. 2h.

C. Architecture of AdapGraspNet

In Fig. 3, AdapGraspNet comprises a backbone (ResNet-50), an encoder, and a decoder. RGB images serve as the input for the model, and are processed through the backbone to generate multiple feature layers. These feature layers are added with positional information and then processed by the encoder as tokens. The encoder utilizes the attention mechanism to capture the key information, and then produces memories as the key and the value of the decoder. In the original transformer architecture, queries of the decoder are randomly initialized or set to zero, slowing down model

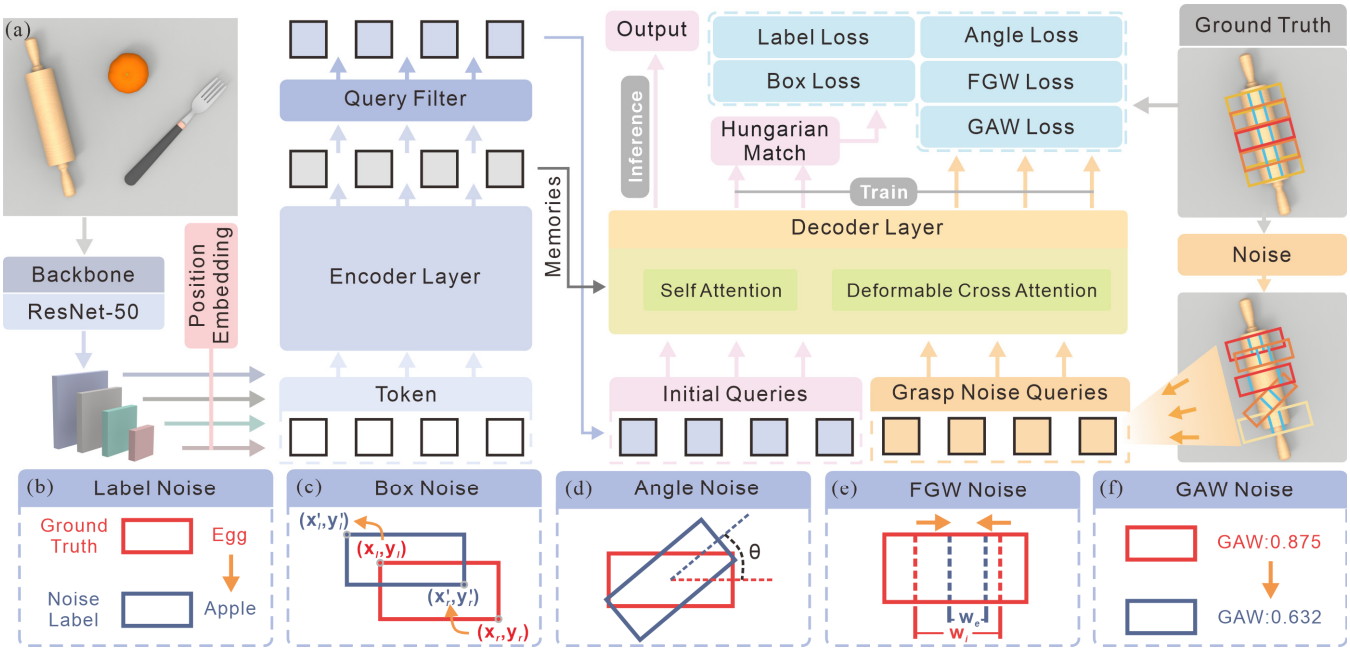

Figure 3. Schematics of AdapGraspNet. (a) The backbone, encoder, and decoder architecture of AdapGraspNet. (b-f) Adding noise to label, box, angle, FGW, and GAW.

convergence and degrading performance [21]. Here, a query filter is designed to select high-confidence outputs from the encoder as initial queries. These high-confidence outputs carry grasp box and finial grasp width information, which accelerates model convergence and improves accuracy.

The decoder training is composed of two parts: a query part and a denoise part. In the query part, the initial queries are processed by decoder layers. The outputs are matched one-to-one with the ground truth using Hungarian Matching. In the denoise part, the ground truth is added with noises as grasp noise queries, including the label noise, the box noise, the angle noise, the final width noise, and the affordance noise. In the label noise, a portion of the ground truth's labels are modified to other labels, in Fig. 3b. In the box noise, perturbations are introduced to the center coordinates, the heights, and the initial widths of the ground truth, in Fig. 3c. For the angle noise and the final width noise, they are similar to the box noise and are added to the corresponding ground truth, in Fig. 3d-e. For the affordance noise, it selects a noise value to replace the original one, in Fig. 3f. Referring to DINO [18], the added noise consists of the positive noise and the negative noise. Different from the label noise, noises in others are evenly distributed, with equal positive and negative parts. Positive noise causes minor perturbations, while negative noise allows for larger deviations. **Box noise**: The center coordinates must not exceed half of the original height or width for positive noise, whereas negative noise can exceed this limit. Similarly, the height and width noise for positive noise is constrained within half of the original values, while negative noise can exceed this threshold. **Angle noise**: Positive noise remains within ±30 degrees, whereas negative noise extends beyond 30 degrees. **Final width noise**: Follows the same rule as the box noise, the positive noise cannot exceed half of the original width, while the negative noise can. **Affordance noise**: For positive samples, perturbations remain within ±0.125, whereas for negative samples, perturbations extend beyond 0.125.

These grasp noise queries are processed through the decoder and then directly compared with the ground truth for loss computation. This denoising strategy reduces the convergence time while also enhancing its performance. AdapGraspNet uses the Focal Loss for label predictions, while the L1 loss is applied to box, angle, FGW, and GAW. During inference, the denoise part is disabled, and only the query part is executed.

To clarify the optimization objective, the detailed loss functions are given. The overall training loss includes the query loss and denoising loss. The query loss is used to supervise predictions based on matched ground-truth grasps. The denoising loss provides additional supervision on noisy queries. The specific equation is as follows,

$$\mathcal{L}_{total} = \frac{1}{n}\sum_{i=1}^{n}\left(\mathcal{L}_i^{query} + \mathcal{L}_i^{dn}\right). \tag{2}$$

Equation (2) defines the overall loss function, which is computed as the average over all training samples. For each image i, the loss consists of two components: the query loss and the denoising loss. The query loss is derived from randomly sampled queries that are matched to ground-truth grasps, but such matches may be unstable and provide relatively weak supervision. In contrast, the denoising loss constructs queries by perturbing ground-truth grasps, which ensures direct alignment with the annotations and yields stronger, more reliable supervision. By combining these two parts, the model benefits from complementary guidance, with denoising supervision in particular facilitating more stable and effective convergence. Notably, the introduction of the denoising query module leads to a better performance, as demonstrated by the ablation results in Table III.

$$\mathcal{L}_i^{query} = \frac{1}{p_i}\sum_{j=1}^{p_i}\left(\lambda_{\text{cls}}\,\text{FL}(\hat{p}_i^j, y_i^j) + \lambda_{\ell_1}\,\mathcal{L}_{\ell_1}(\hat{G}_i^j, G_i^j)\right), \tag{3}$$

Equation (3) specifies the query loss for the i^{th} image, averaged over its p_i matched ground-truth grasps. It combines the Focal Loss for classification and the L1 regression loss for grasp parameters, with weighting factors λ_{cls} and λ_{ℓ_1}.

$$\mathcal{L}_i^{\text{dn}} = \frac{1}{r_i}\sum_{k=1}^{r_i}\left(\lambda_{cls}\,\text{FL}(\hat{\tilde{p}}_i^k, \tilde{y}_i^k) + \lambda_{\ell_1}\,\mathcal{L}_{\ell_1}(\hat{\tilde{G}}_i^k, \tilde{G}_i^k)\right), \quad (4)$$

Equation (4) defines the denoising loss, which is computed over r_i noisy targets associated with the i^{th} image. Similar to the query loss, it consists of a Focal Loss for classification and an L1 regression loss for noisy ground-truth supervision. This denoising strategy improves convergence speed.

$$\mathcal{L}_{\ell_1}(\hat{G}, G) = \lambda_{box}\mathcal{L}_{\ell_1}^{box} + \lambda_\theta \mathcal{L}_\theta + \lambda_{FGW}\mathcal{L}_{\ell_1}^{FGW} + \lambda_{GAW}\mathcal{L}_{\ell_1}^{GAW}, \quad (5)$$

Equation (5) defines the overall regression loss $\mathcal{L}_{\ell_1}(\hat{G}, G)$, which is a weighted combination of different components corresponding to the grasp parameters. Specifically, $\mathcal{L}_{\ell_1}^{box}$ denotes the L1 loss for the bounding box parameters (x, y, w, h), $\mathcal{L}_\theta$ represents the angle loss that accounts for the periodicity of grasp orientation, while $\mathcal{L}_{\ell_1}^{FGW}$ and $\mathcal{L}_{\ell_1}^{GAW}$ are the L1 losses for FGW and GAW, respectively. The coefficients $\lambda_{box}, \lambda_\theta, \lambda_{FGW}, \lambda_{GAW}$ are balancing factors to control the contribution of each term.

$$\mathcal{L}_\theta = \begin{cases} \Delta\theta, & \text{if } \Delta\theta \le 0.5, \\ 1-\Delta\theta, & \text{if } \Delta\theta > 0.5, \end{cases} \quad \Delta\theta = |\hat{\theta} - \theta|, \quad (6)$$

Equation (6) defines the angle regression loss $\mathcal{L}_\theta$, This loss accounts for the periodicity of the grasping orientation. The normalized angle difference $\Delta\theta = |\hat{\theta} - \theta|$ is used. If $\Delta\theta \le 0.5$, the direct difference is applied; otherwise, $1-\Delta\theta$ is used, ensuring that opposite directions are treated as similar. This design is motivated by the periodic nature of grasp rotation angles, where a large numerical difference may correspond to a small actual deviation (e.g., –90° and 90° represent the same orientation). Therefore, the proposed loss effectively handles such cases and yields more accurate angle regression.

IV. Experiments

A. Training Details

The AdapGraspNet is trained on the AdapGraspDataset with an NVIDIA GeForce GTX 4090 GPU. The implementation framework is Pytorch with CUDA 11.8. The batch size is set as 16, and the optimizer is AdamW. The epoch number for training is 50. We use a two-stage learning rate of 10^{-4} before round 20 and 10^{-5} after round 20.

B. Evaluation Metrics

Based on the metric proposed in [6], the evaluation metric for grasp detection considers a predicted grasp rectangle to be correct if it satisfies: 1) the Jaccard index is greater than 0.25, and 2) the angle difference between the prediction and the ground truth is less than 30°. Building upon this, previous grasp evaluation methods typically assess only the correctness of the optimal grasp rectangle for each object. To comprehensively reflect model performance, we propose a more rigorous standard: evaluating the correctness of multiple predicted grasp rectangles that have GAW close to the ground truth for each object, and computing overall precision. We define this metric as grasp precision (GP). To evaluate the model's performance in predicting the FGW, we propose two metrics: Final Width Mean Absolute Error (FW-MAE) and Final Width Mean Relative Error (FW-MRE). These metrics calculate the average absolute and relative errors between the FGWs of all correctly predicted grasp rectangles and the corresponding ground truth.

TABLE I. FW-MAE and FW-MRE Comparison

Method	FW-MAE (pixel)	FW-MRE (%)
GG-CNN-M	18.79	47.14
GR-CNN-M	11.02	25.36
TF-Grasp-M	12.96	35.92
AdapGraspNet (ours)	**2.72**	**7.07**

TABLE II. GP Comparison

Method	GP (%)
GG-CNN-M	63.19
GR-CNN-M	87.29
TF-Grasp-M	91.36
AdapGraspNet (ours)	**98.04**

TABLE III. Comparison of Ablation Experiment Results

Epochs	Without Grasp Noise Query		With Grasp Noise Query(**ours**)	
	FW-MRE (%)	GP (%)	FW-MRE (%)	GP (%)
10	10.71	89.47	9.29	91.56
20	9.79	93.05	8.79	95.52
30	8.09	96.31	7.37	97.84
40	7.92	96.53	7.25	97.73
50	7.57	96.71	7.07	98.04

TABLE IV. Generalization Experiments in Real World

Method	Success	Failure	Accuracy (%)
Single Object	92	8	92 (92/100)
Multiple Objects	88	12	88 (88/100)

C. Evaluation Metrics Comparison

We compare the performance of AdapGraspNet and GG-CNN [15], GR-CNN [16], TF-Grasp [4] on the AdapGraspDataset. These existing models cannot output FGW and GAW in their original architectures. Therefore, we add a head similar to the box head to output FGW. The GAW is used as a quality input; the results are obtained under the same training conditions as AdapGraspNet. The modified model architectures are denoted with the -M suffix to distinguish them from the original models (e.g., GR-CNN-M).

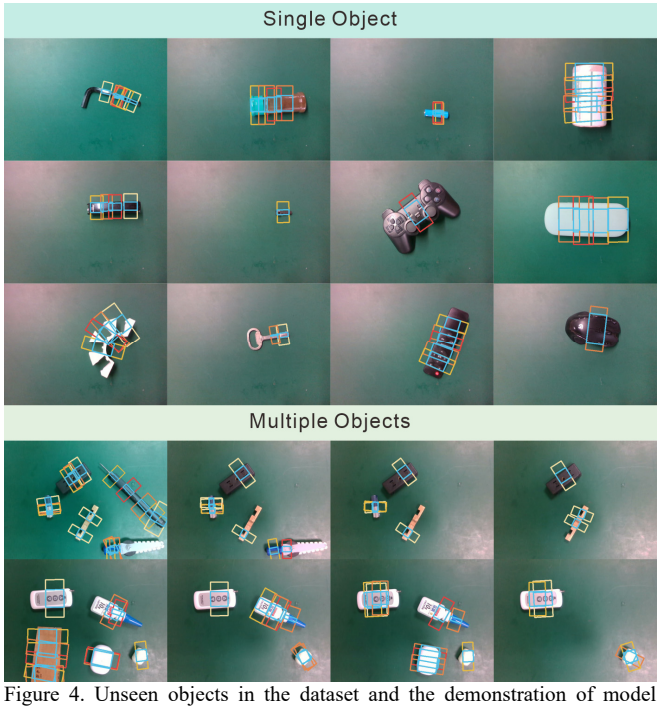

Figure 4. Unseen objects in the dataset and the demonstration of model performance. (a) Single object scenarios. (b) Multiple object scenarios.

Table I compares FW-MAE and FW-MRS, which indicate the performance of FGW. Higher values imply an increased risk of grasp failure or damage. FW-MAE of our model is only 2.72 pixels, which is significantly lower than GR-CNN, reducing its error to 25%. Moreover, FW-MRE is 7.07%, only 0.28 times that of GR-CNN.

Table II compares the GP between AdapGraspNet and other models. It proves that our model achieves a significantly higher GP compared to the other models.

D. Ablation Experiments and Generalization Experiments

To evaluate the effect of grasp noise queries in the AdapGraspNet model, we conduct ablation studies. As shown in Table III, incorporating grasp noise queries leads to superior FW-MER and GP metrics under the same number of training epochs, indicating that the grasp noise query module contributes positively to the convergence of the model.

To demonstrate the generalization ability of the model, we conduct multiple experiments to verify its ability to grasp unseen objects that are not included in the training dataset, in Fig. 4 and Table IV. For these unseen objects, the model still outputs reliable grasping boxes. In terms of grasp weight generation, the model assigns higher weights to regions closer to the center of mass; in terms of closing distance generation, the model adaptively determines suitable closing distances for different objects. These results demonstrate that AdaGraspNet exhibits reliable generalization capability.

TABLE V. GOMPARATIVE EXPERIMENTS IN REAL WORLD

Method	Unstable	Damaged	Success	Accuracy (%)
Without FGW&GAW	30	24	166	75.5(166/220)
With FGW&GAW **(ours)**	9	2	209	**95.0**(209/220)

Algorithm 1: ADAPGRASP EXECUTION

Require: *RGB Image* I_{RGB}, *Depth Image* I_D, *Camera Intrinsics K, Camera-to-Robot Transform* T_c^r

function ADAPGRASP_EXECUTION (I_{RGB}, I_D, K, T_c^r)
  while True do
    GraspCandidates ← *ADAPGRASP* (I_{RGB})
    if GraspCandidates is empty then
      break
    end if
    Sort GraspCandidates by GAW in descending order
    Grasp_best ← *Grasp_Candidates[0]* ▷ *(x, y, w, h, θ, FGW, GAW)*
    Depth ← *ExtractDepth (x, y, w, h,* I_D*)*
    (X_c, Y_c, Z_c) ← *PixelToCamera(x, y, Depth, K)*
    GraspPositionRobot ← $T_c^r \cdot [X_c, Y_c, Z_c, 1]^T$
    InitGraspWidth ← *ProjectWidthToRobot (w, Depth, θ, K)*
    FinalGraspWidth ← *ProjectWidthToRobot (FGW, Depth, θ, K)*
    Execute_Grasp (GraspPositionRobot, θ, InitGraspWidth, FinalGraspWidth)
  end while
end function

E. Experiments and Comparisons in Real-World Scenarios

Experiments are conducted to demonstrate the effectiveness of AdapGrasp strategy. We utilize a UR3 robot with a fin-ray flexible gripper and a Realsense D435 depth camera. The AdapGraspNet is tested on an Nvidia GeForce RTX 2060M, achieving 12.9 FPS. The detailed experimental procedure is presented in Algorithm 1.

For each type of object, we perform 10 grasping trials, is in Table V. In the model trained without the FGW and GAW varied in each trial, and the detailed results are presented in the datasets, the frequency of unstable grasps and object damage increases significantly. The unstable grasps are caused by deviations of the grasp position from the object's center of mass, while the destructive grasps result from excessive closing distances. Thus, this real-world experiment demonstrates the effectiveness of the FGW and GAW datasets.

To better illustrate the effectiveness of each module of AdapGrasp, we present the three comparative demonstrations as follows.

Comparative Demonstration of FGW: To demonstrate the importance of FGW, we conducted experiments using the rigid gripper Robotiq 2F-85 and the fin-ray flexible gripper, in Fig. 5. The left and middle columns represent cases without FGW, while the right column with FGW. We conducted three sets of experiments on an apple, a plastic cup, and a toothpaste. The results indicate that the rigid gripper causes damage to all objects, whereas the flexible gripper, without FGW, damages the plastic cup and toothpaste. However, when FGW is incorporated, the gripper successfully achieved damage-free grasping for all objects, refer to Mov. 1. These results confirm the effectiveness of FGW and the advantages of the flexible gripper.

Comparative Demonstration of GAW: To validate the role of GAW, we conduct a series of experiments using objects such as an eggplant, a banana, a screwdriver, and a spoon. In Fig. 6, the left column presents the grasping results

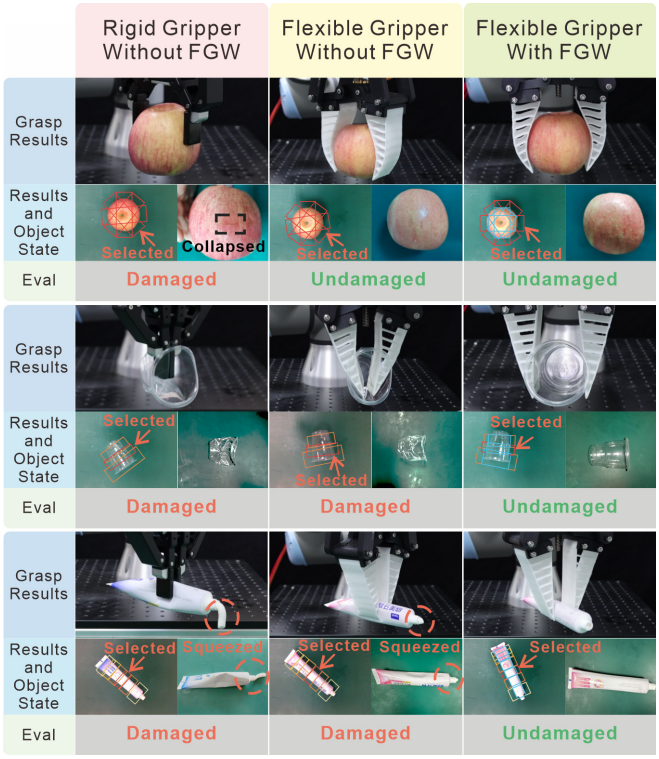

Figure 5. Experimental comparison on FGW. The first column represents a rigid gripper with FGW. The second column shows a flexible gripper without FGW. The third column illustrates a flexible gripper with FGW.

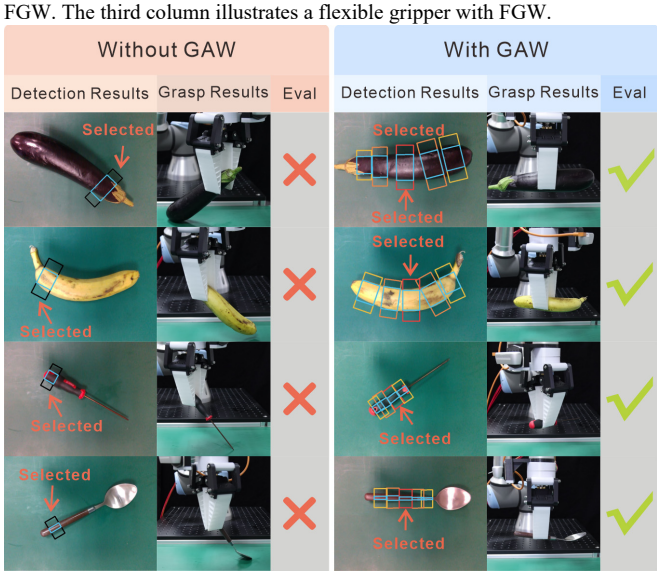

Figure 6. Experimental comparison on grasping with and without GAW. The first column represents the model without GAW, while the second column represents the model with GAW.

of the model trained with GAW, while the right column shows the results of the model trained without it. It is evident that the left column grasps align more closely with human intuition, tending toward the object's centroid, whereas the right column grasps tend to be closer to object edges, increasing the risk of grasp failure, refer to Mov. 1. These results confirm the necessity of incorporating GAW.

Demonstration of Multi-Object Grasping: Experiments are conducted in the multi-object scenarios, in Fig. 7 and Mov. 1. In this environment, the gripper with the AdapGrasp strategy can stably grasp objects with high reliability while selecting grasp positions that align with human intuition,

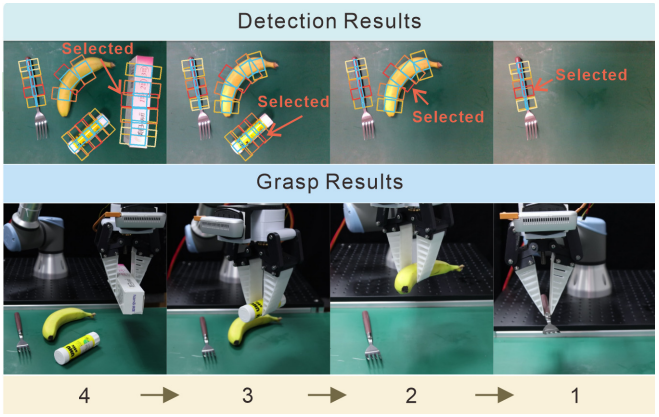

Figure 7. Experiment with multi-object in real-world scenarios

favoring locations closer to the object's centroid. Further, by incorporating FGW, the model can prevent damage.

V. Conclusion

In this work, we propose an AdapGrasp strategy with AdapGraspDataset and AdapGraspNet, by explicitly encoding object stiffness and grasp affordance. Specifically, we introduce two new metrics into the dataset: FGW and GAW. The former preserves the integrity of objects by adapting to their stiffness, while the latter improves grasp stability by prioritizing grasps closer to the object centroid. In real-world experiments, the AdapGrasp strategy delivers a grasping accuracy of 95%, yielding a 19.5% improvement compared with baselines without FGW and GAW. Simultaneously, to predict FGW and GAW more accurately, we adopt a denoising principle to build a transformer-based model, enabling more stable convergence. It ensures AdapGrasp can intactly grasp fragile objects, and enhance grasping stability without any additional sensors. Our current approach provides a practical solution by encoding stiffness- and affordance-aware priors.

Since the primary goal of this work is to propose a novel grasp representation rather than curate an exhaustive dataset, the resulting data contains several limitations. First, the object diversity and scale are relatively limited, and data collection adopts a top-down strategy, which somehow limits its applicability. Meanwhile, the provided annotations only represent a subset of human-preferred grasps, not all feasible grasp configurations. Furthermore, the FGW-based representation may also be less suitable for certain object types, such as objects that are both soft and heavy, where grasp stability may be affected. In the future, we will expand AdapGraspDataset to encompass more diverse and challenging objects, such as multi-stiffness cases, soft but heavy objects, and rigid yet lightweight ones, which pose additional challenges for grasp adaptation. We will also add real-time feedback to enable the system to adjust after failures, which further enhances its adaptability.

References

[1] X. Zhang et al., "Learning to Dexterously Pick or Separate Tangled-Prone Objects for Industrial Bin Picking," in IEEE Robotics and Automation Letters, vol. 8, no. 8, pp. 4919-4926, Aug. 2023.

[2] H.-S. Fang et al., "GraspNet-1Billion: A Large-Scale Benchmark for General Object Grasping," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 2020, pp. 11441-11450.

[3] F. Zhang et al., "Learning garment manipulation policies toward robot-assisted dressing," in Science Robotics, vol. 7, no. 65, Apr. 2022.
[4] S. Wang et al., "When Transformer Meets Robotic Grasping: Exploits Context for Efficient Grasp Detection," in IEEE Robotics and Automation Letters, vol. 7, no. 3, pp. 8170-8177, Jul. 2022.
[5] S. Yu et al., "SKGNet: Robotic Grasp Detection With Selective Kernel Convolution," in IEEE Transactions on Automation Science and Engineering, vol. 20, no. 4, pp. 2241-2252, Oct. 2023.
[6] Y. Jiang et al., "Efficient grasping from RGBD images: Learning using a new rectangle representation," 2011 IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China, 2011, pp. 3304-3311.
[7] A. Depierre et al., "Jacquard: A Large Scale Dataset for Robotic Grasp Detection," 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 2018, pp. 3511-3516.
[8] D. Wang et al., "High-Performance Pixel-Level Grasp Detection Based on Adaptive Grasping and Grasp-Aware Network," in IEEE Transactions on Industrial Electronics, vol. 69, no. 11, pp. 11611-11621, Nov. 2022.
[9] H. Nie et al., "Smaller and Faster Robotic Grasp Detection Model via Knowledge Distillation and Unequal Feature Encoding," in IEEE Robotics and Automation Letters, vol. 9, no. 8, pp. 7206-7213, Aug. 2024.
[10] S. Cui et al., "Grasp State Assessment of Deformable Objects Using Visual-Tactile Fusion Perception," 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 2020, pp. 538-544.
[11] G. Maiello et al., "Humans Can Visually Judge Grasp Quality and Refine Their Judgments Through Visual and Haptic Feedback," in Frontiers in Neuroscience, vol. 14, p. 591898, Jan. 2021.
[12] P. Hegemann et al., "Learning Symbolic Failure Detection for Grasping and Mobile Manipulation Tasks," 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Kyoto, Japan, 2022, pp. 4302-4309.
[13] I. Lenz et al., "Deep Learning for Detecting Robotic Grasps," in The International Journal of Robotics Research, Vol 34, Issue 4-5, pp. 705-724, Apr. 2015.
[14] H. Cheng et al., "A Robot Grasping System With Single-Stage Anchor-Free Deep Grasp Detector," in IEEE Transactions on Instrumentation and Measurement, vol. 71, pp. 1-12, 2022.
[15] D. Morrison et al., "Closing the Loop for Robotic Grasping: A Real-time Generative Grasp Synthesis Approach," in Robotics: Science and Systems XIV, Robotics: Science and Systems Foundation, Jun. 2018.
[16] S. Kumra et al., "Antipodal Robotic Grasping using Generative Residual Convolutional Neural Network," 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 2020, pp. 9626-9633.
[17] N. Carion et al., "End-to-end object detection with transformers," in European conference on computer vision (ECCV), Springer, 2020, pp. 213-229.
[18] H. Zhang et al., "DINO: DETR with Improved DeNoising Anchor Boxes for End-to-End Object Detection," in ICLR 2023: The Eleventh International Conference on Learning Representations, 2023.
[19] F.-J. Chu et al., "Real-World Multiobject, Multigrasp Detection," in IEEE Robotics and Automation Letters, vol. 3, no. 4, pp. 3355-3362, Oct. 2018.
[20] C. Guo et al., "End-to-End lightweight Transformer-Based neural network for grasp detection towards fruit robotic handling," in Computers and Electronics in Agriculture, vol. 221, p. 109014, Jun. 2024.
[21] X. Zhu et al., "Deformable DETR: Deformable Transformers for End-to-End Object Detection," in ICLR 2021: The Ninth International Conference on Learning Representations, 2021.