applied sciences

MDPI

Article

Semantically Supervised SeDINO Encoder for Visual–Language–Action Model

Shen Tian [1,2,3], Dong Yu [1,3,*], Long Cui [2], Zhaoming Liu [2], Hongwei Wang [2], Zixuan Li [2] and Haotian Liu [2]

[1] Shenyang Institute of Computing Technology, Chinese Academy of Sciences, Shenyang 110168, China
[2] State Key Laboratory of Robotics And Intelligent Systems, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China
[3] University of Chinese Academy of Sciences, Beijing 100049, China
* Correspondence: yudong@sict.ac.cn

Abstract

With the rapid development of multi-modal large models, the Visual–Language–Action (VLA) model has gradually become a new paradigm for autonomous robot operations. The VLA model encodes experimental images and text instructions separately using an image encoder and a text encoder. The encoded multi-modal vector information is then fed into a large language model (LLM) to generate the next action. While they inherit the generalization capabilities of large language models, VLA models often struggle to ensure accuracy and reliability in complex scenes. Some studies have attempted to improve VLA performance by enhancing the fine-tuning process or introducing staged operations; however, these improvements often overlook the stable extraction of important visual features, which are crucial for VLA models. In typical VLA tasks, the instruction text inherently contains semantic information related to image elements. Research has shown that leveraging text supervision for visual feature extraction can enhance feature quality. In this paper, we propose a semantically supervised visual encoder called SeDINO (Semantically Supervised DINO), which efficiently fuses DINO's element localization capabilities with CLIP's semantic information. We further employ an MLP (Multi-Layer Perceptron) network to align the semantic vectors output by the CLIP text encoder with the image feature vectors derived from DINO, fully leveraging DINO's element localization and CLIP's semantic interaction capabilities. We validate SeDINO on six mainstream image datasets, and it demonstrates superior segmentation performance compared to current leading models. Additionally, we incorporate the proposed SeDINO into the VLA framework, using OpenVLA-7B and DINOv2-base as backbone models, and evaluate it on the LIBERO dataset and real-world scenarios.

Keywords: semantic supervision; DINO; VLA

Check for updates

Academic Editor: Andrea Prati

Received: 6 January 2026
Revised: 28 January 2026
Accepted: 28 January 2026
Published: 31 January 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

In recent years, with the accelerated advancement of multi-modal large models, the Visual–Language–Action (VLA) model has gradually become a new paradigm for autonomous robot operations. The VLA model fine-tunes large language models (LLMs) to learn the associations between images and robot actions under specific textual instructions and leverages large-scale multi-modal datasets such as OpenXEmbodyment to achieve scaling up [1]. During execution, the model encodes the current state's image information and instruction text and feeds the combined multi-modal vectors into the LLM to generate

the next action. The VLA inherits the generalization ability of large language models across different scenarios and tasks, enabling effective zero-shot performance in new environments [2,3]. However, in complex scenes or those with significant dataset discrepancies, the accuracy and reliability of task execution often become difficult to guarantee, mainly due to insufficient stability and semantic alignment of visual feature extraction, which is overlooked by existing optimization strategies such as reinforcement learning integration or task decomposition.

Visual encoders are the core of the VLA model's visual feature extraction. DINO (Deep Interactive Network for Object Detection), a mainstream self-supervised visual encoder based on ViT, is widely adopted in current VLA models for its strong part localization capabilities [4,5]. However, it lacks interaction with the semantic information inherent in the VLA model's instruction texts. In contrast, CLIP (Contrastive Language–Image Pre-Training), trained on large-scale image–text pairs, possesses robust semantic interaction abilities but weaker visual element localization [5]. The above situation holds back VLA model performance because reliably extracting visual features that match semantics is key for stable action generation. To address this, we propose enhancing visual encoders by fusing semantic information from instructions into image feature extraction, drawing insights from open-vocabulary segmentation research that pursues similar semantic–spatial integration goals.

Mainstream open-vocabulary segmentation methods verify that effective fusion of DINO's localization and CLIP's semantics is the key to balancing semantic alignment and spatial continuity [6–8]. Moreover, DINO's raw image vectors outperform CLIP's in quality [9]. Building on this, we propose SeDINO (Semantically Supervised DINO), a visual encoder that fuses DINO's element localization with CLIP's semantic information. Specifically, we use an MLP network to align CLIP's semantic vectors with DINO's image feature vectors [10] and supervise the latter accordingly. We further compare DINO's multi-head attention features, retaining only the most similar head weights to optimize spatial–semantic consistency, consistent with hybrid Vision Transformer design principles [11].

In summary, the core motivation of this study is to address the critical bottleneck in VLA systems: the lack of stable and semantically aligned visual feature extraction, which stems from the inherent limitations of existing visual encoders, i.e., DINO excels at spatial localization but lacks sufficient semantic interaction with instruction texts, while CLIP possesses robust semantic understanding but is weak in visual element localization. To tackle this gap, we develop SeDINO, aiming to enhance the reliability of VLA task execution (especially in complex or dataset-discrepant scenarios) by fusing the complementary strengths of DINO and CLIP with semantic supervision from instruction texts, ultimately advancing the practical applicability of VLA models for autonomous robot operations.

Our core contributions are threefold: (1) We propose SeDINO, a lightweight semantically supervised visual encoder that integrates DINO's spatial precision and CLIP's semantic interaction capability via a well-designed feature fusion strategy. (2) We conduct extensive empirical validation of SeDINO, verifying its superior segmentation performance across six benchmark datasets (Pascal VOC V20/V21, Pascal Context C59, COCO Stuff, Cityscapes, and ADE20K) compared to current mainstream models. (3) We demonstrate SeDINO's effectiveness as a visual encoder for a VLA system. It achieves notable performance improvements in evaluations on the LIBERO dataset and real-world scenes.

2. Materials and Methods

This section details the design of the Semantically Supervised DINO (SeDINO) visual encoder and its integration into the Visual–Language–Action (VLA) model, and the overall pipeline is illustrated in Figure 1. SeDINO fuses DINO's spatial localization and CLIP's

semantic interaction by training an MLP to align CLIP text vectors with DINO image features then selecting DINO's multi-head attention head with the highest image–text similarity to generate semantically supervised visual features. The MLP is trained on the COCO Stuff dataset with InfoNCE loss. For VLA integration, semantic elements are extracted from task instructions, SeDINO generates aligned image vectors for each element, and the fused vectors are used to fine-tune the OpenVLA-7B backbone (with DINOv2-base) on the LIBERO dataset and real-world scenarios. Subsequent sections elaborate on SeDINO's architectural design, MLP training, and the full VLA integration pipeline.

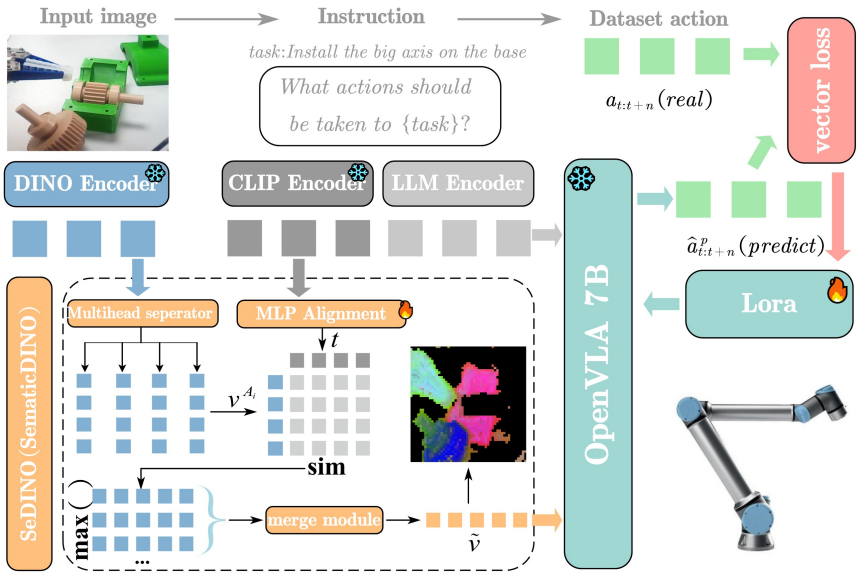

Figure 1. Schematic of SeDINO integration into the VLA model.

2.1. Design and Construction of the SeDINO Visual Perception Model

In this work, we propose a semantically supervised visual encoder called SeDINO (Semantic DINO), which efficiently integrates DINO's element localization capabilities with CLIP's semantic interaction abilities. We achieve this by training an MLP network to align the semantic vectors output by CLIP's text encoder with the image feature vectors derived from DINO. Furthermore, we compare the image elements obtained from DINO's multi-head attention with the aligned semantic vectors, retaining the weights of the attention head with the highest similarity to reconstruct the image vectors, as shown in Figure 2.

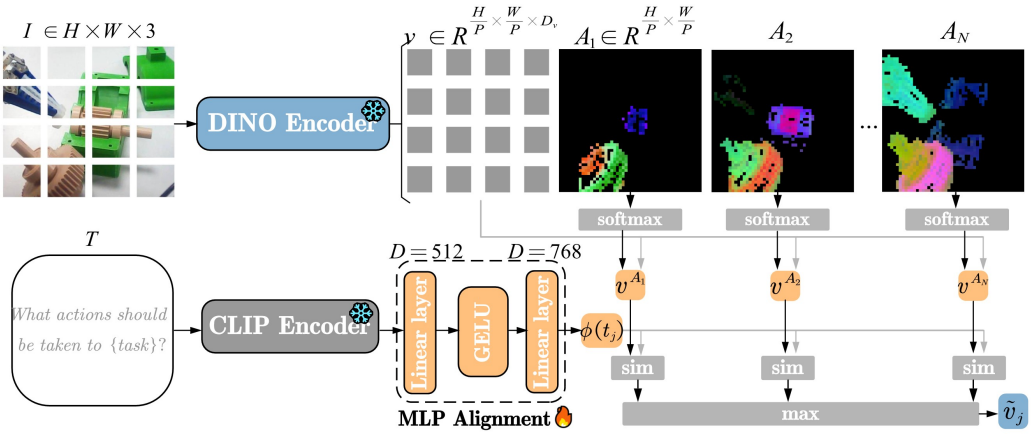

Figure 2. Architecture of SeDINO for semantically supervised visual feature extraction.

2.1.1. SeDINO Architecture Design

First, an image of size $(\mathcal{H}, \mathcal{W}, 3)$ is divided, with P as the patch size and encoded by the DINO encoder. This yields the image's feature vector and extracts the multi-head attention weights. Meanwhile, semantic information is encoded via CLIP's text encoder to obtain the text vector t. We aim to achieve semantic supervision of the image by comparing the similarity between the image and semantic vectors. However, due to the different dimensions of the CLIP and DINO encoders, direct similarity comparison is not feasible. To address this alignment issue, we introduce an MLP network to map CLIP's text vector to the same dimension as DINO's visual vector, facilitating subsequent similarity comparison. The specific construction and training process of the MLP will be detailed in the next section.

Next, we generate the weight vector v^{A_i} based on the feature vector v and the weight A_i of the i-th attention head using the formula below, which serves as the feature vector of the i-th attention head. The similarity between the re-weighted feature vector and the aligned text vector $\phi(t)$ is calculated using the formula below. In simple terms, the computed $\mathrm{sim}(v^{A_i}, t)$ refers to the average cosine similarity between the feature vector of each individual image patch and the dimension-aligned text vector $\phi(t)$, and this similarity is calculated under the weighting of a single selected i-th attention head. $[h, w]$ represents the row and column index of the feature vector corresponding to each image patch in the 2D feature map. The process is formulated as follows:

$$v^{A_i} = v \cdot softmax(A_i) \tag{1}$$

$$sim\left(v^{A_i}, t\right) = \frac{P^2}{H, W} \sum_{h,w} \frac{v^{A_i}_{[h,w]} \cdot \phi(t)}{\|v^{A_i}_{[h,w]}\| \|\phi(t)\|} \tag{2}$$

Next, taking the image–text pair (I_i, T_i) as an example, after calculating the similarity between each attention head and the text using the above method, we select the attention head with the highest similarity to the text vector as the feature vector $\tilde{v}_i$ after semantic supervision, which is expressed as follows:

$$\tilde{v}_i = v_i^{A_j} \mid j = \underset{k}{argmax} sim\left(v_i^{A_k}, t_i\right) \tag{3}$$

SeDINO's attention head selection mechanism shows variable robustness. Noisy or ambiguous instructions cause misselection of relevant attention heads and reduced accuracy. Minor textual paraphrases have negligible impact, while synonym substitutions reduce matching rates due to CLIP bias. Key failures occur with multi-attribute ambiguous instructions and lead to significant accuracy degradation. We retain only high-confidence attention head selections to avoid misalignment caused by noisy or ambiguous linguistic inputs.

Notably, this attention head selection process is driven by explicit semantic supervision; only the attention head whose feature vector achieves the highest similarity with the instruction-aligned text vector is retained. This supervision mechanism ensures that the output visual features are strictly anchored to the semantic intent of the task instruction rather than CLIP/DINO's generic or unsupervised semantic associations.

2.1.2. MLP Alignment Network Training

The MLP (Multi-Layer Perceptron) is a fully connected neural network with strong fitting capabilities, meaning each neuron in one layer is connected to all neurons in the next layer. We use this MLP network to solve the dimension mismatch problem between CLIP text vectors and DINO image vectors. Specifically, the MLP takes 512-dimensional text vectors from the CLIP text encoder as input and outputs 768-dimensional vectors that match

the dimension of DINOv2-base image vectors. The output layer uses the GELU activation function to introduce non-linearity (helping the model to learn complex relationships) into the alignment process. To train the MLP, we adopt the InfoNCE loss function, which is a widely used contrastive loss for aligning image and text features. In the loss function, B simply represents the number of image–text pairs used in each training batch (batch size). The loss function is expressed as follows:

$$L_{InfoNCE} = -\frac{1}{2B}\sum_{i=1}^{B}\log\frac{\exp(sim(\tilde{v}_i, t_i))}{\sum_{i=1}^{B}\exp(sim(\tilde{v}_j, t_i))} - \frac{1}{2B}\sum_{i=1}^{B}\log\frac{\exp(sim(\tilde{v}_i, t_i))}{\sum_{i=1}^{B}\exp(sim(\tilde{v}_i, t_j))} \quad (4)$$

We select the COCO Stuff dataset with associated text captions to train the MLP. During the preparation phase, DINO extracts the image feature vectors from the dataset while CLIP extracts the text feature vectors. The MLP is trained using image batch sizes, and, after 100 training iterations, the model stabilizes. The loss function curves for the training and validation sets are shown in Figure 3 below.

We select the COCO Stuff dataset with associated text captions to train the MLP. Here, the semantic ground truth for CLIP-DINO alignment is defined as the text–image semantic correspondence between COCO Stuff's official object category labels and the dataset's visual content, which serves as the supervised signal for MLP training. The free text captions are preprocessed by filtering irrelevant descriptions, lexical normalization, and semantic disambiguation with the official category lexicon to mitigate inherent semantic ambiguity. During the preparation phase, DINO extracts the image feature vectors from the dataset while CLIP extracts the text feature vectors. The MLP is trained using image batch sizes, and, after 100 training iterations, the model stabilizes. The loss function curves for the training and validation sets are shown in Figure 3 below.

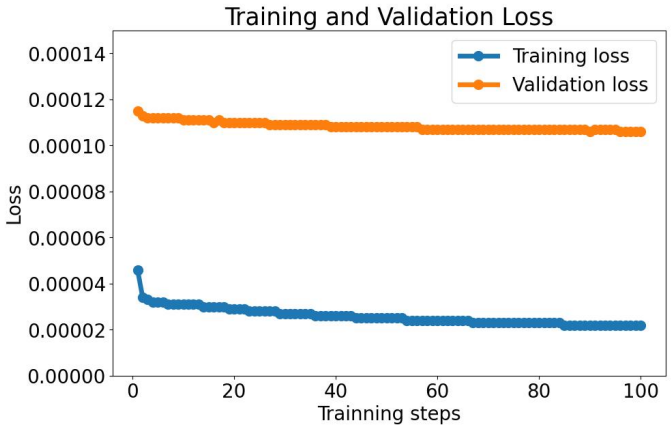

Figure 3. Training curve of the Multi-Layer Perceptron (MLP).

The InfoNCE loss serves as the core semantic supervision signal here; it ensures that the MLP-aligned text and image vectors of positive pairs are pulled closer. This supervised training guarantees that the dimension conversion is not merely a mathematical mapping, but a semantically consistent alignment tailored to the VLA model's instruction action demands.

2.2. Integration of SeDINO into the VLA Model

To apply the constructed encoder to the VLA model, we first extract the set of semantic elements from the instruction in the VLA task. Each element is represented by a semantically supervised image vector obtained using SeDINO. All these semantic element vectors are then combined to serve as the input to the VLA model, which is subsequently fine-tuned. We utilize OpenVLA-7B and DINOv2-base as the backbone models and conduct testing on

the LEBRO dataset as well as real-world scenarios, as shown in Figure 1. In the VLA task, we employ real-time experimental photos as the visual information and text instructions as the semantic information. However, instructions often contain multiple semantic labels. We extract nouns following verbs and prepositions in the instructions as semantic labels. Since the instructions used in the study are structured, simple sentences, almost no semantic extraction errors occur in the tests. The semantic extractor we use is similar to spacy, which can identify a set of semantic tags $\{T_1, T_2, \ldots, T_N\}$ from the instruction text, as shown in Figure 4 below.

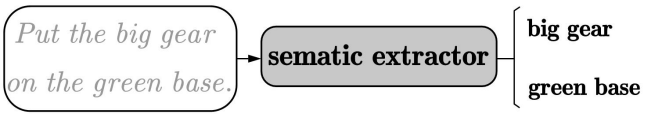

Figure 4. Extraction of semantic tags from instruction texts.

Based on the above content, each semantic label T_i can supervise SeDINO to obtain the corresponding image vector v^i. Finally, all image vectors corresponding to the semantic information extracted from the instruction are merged into $\tilde{v} = \sum_{i=1}^{N} \tilde{v}_i$, which serves as the final image vector submitted to the VLA model, where N is the number of semantic labels in the instruction. Utilizing the previously trained MLP model, we construct the SeDINO encoder and replace the original DINO encoder in the OpenVLA-7B model. We select the libero-spatial-no-loops subset from the LIBERO dataset for training, and the loss function curve of the fine-tuning process is shown in Figure 5.

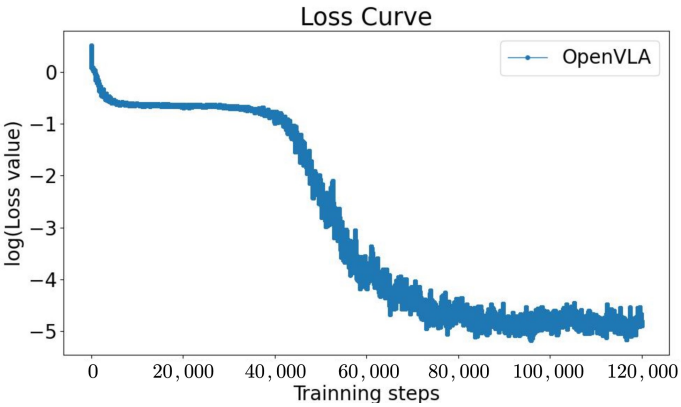

Figure 5. VLA model fine-tuning loss curve.

3. Results

To validate the effectiveness of our proposed SeDINO model in improving the performance of the VLA model, we evaluated SeDINO on six datasets—Pascal VOC (V20 and V21) [12], Pascal Context C59 [13], COCO Stuff [14], Cityscapes [15], and ADE20K [16]. Furthermore, we applied SeDINO to the VLA model and performed evaluation on the LIBERO dataset and real-world scenes, achieving notable performance improvements. The experimental results demonstrate that SeDINO's image segmentation ability surpasses that of most current mainstream models, and the application of SeDINO leads to a significant performance enhancement in the VLA model. All experiments involving segmentation and application of SeDINO in VLA tasks, as well as the model's training and deployment, were performed on a remote server with an NVIDIA GeForce RTX 4090 GPU. The experiments were implemented based on Python 3.10, with the deep learning framework PyTorch 2.1.2 and CUDA 11.8 for GPU acceleration to ensure efficient model training and inference.

3.1. Visual Dataset Validation Results

We conducted extensive experiments on six widely used and challenging datasets, including Pascal VOC (V20 and V21) with 20 classes, Pascal Context with 176 classes, COCO Stuff with 171 classes, Cityscapes with 30 classes, and ADE20K with 150 classes, to comprehensively evaluate the segmentation performance of our proposed SeDINO model. The IoU (Intersection over Union) results obtained from these assessments are summarized in Table 1. The experimental data clearly indicate that SeDINO consistently surpasses most current mainstream segmentation models across these diverse datasets, demonstrating its robustness and generalization ability. The improved performance can be attributed to the semantic supervision integrated into the model, which effectively guides the visual encoder to extract more meaningful and discriminative features. This validation confirms that leveraging semantic information during training significantly enhances the accuracy and reliability of image segmentation tasks. These results not only highlight the effectiveness of our SeDINO approach but also suggest its potential for broader applications in complex scene understanding and semantic reasoning in computer vision.

To clearly demonstrate the segmentation performance trends, three representative images were selected and segmented using both mainstream models and SeDINO, and the qualitative results are visualized in Figure 6. Rather than focusing on isolated numerical metrics (reported in full in Table 1), we emphasize the consistent pattern observed across all test cases: SeDINO consistently outperforms mainstream models in preserving the continuity of object boundaries and reducing missegmentation of fine-grained image elements. The visual comparison reveals a systematic advantage of SeDINO in both accuracy and continuity, which directly reflects the effectiveness of our semantic supervision mechanism in guiding targeted extraction of task-relevant image elements rather than generic visual features.

Table 1. Comparison with state-of-the-art segmentation models applied to image datasets.

Model	mIoU						
	V20	**V21**	**C59**	**Stuff**	**City**	**ADE20K**	**Avg**
GroupViT [17]	79.7	50.4	23.4	15.3	11.1	9.2	31.5
TCL [18]	77.5	51.2	30.3	19.6	23.1	14.9	36.1
MaskCLIP [19]	74.9	38.8	26.4	16.4	12.6	9.8	29.8
CLIP-DIY [20]	79.7	59.9	19.8	13.3	11.6	9.9	32.4
SCLIP [21]	80.4	59.1	34.2	22.4	32.2	16.1	40.7
CLIP-DINOiser [7]	80.9	62.1	35.9	24.6	31.1	20.0	42.4
ClearCLIP [22]	80.9	51.8	35.9	23.9	30.0	16.7	39.9
NACLiP [23]	79.7	58.9	35.2	23.3	35.5	17.4	41.7
FreeDA [24]	77.1	51.7	37.1	24.9	34.0	19.5	40.7
ProxyCLIP [8]	83.0	58.6	37.2	25.4	33.9	19.7	43.0
SeDINO (ours)	85.1	60.3	39.6	27.9	34.8	21.1	44.8

Evaluation experiments are conducted on the following image datasets: Pascal VOC (including V21 and V20 subsets), Pascal Context (C59), COCO Stuff, Cityscapes, and ADE20K. The best performance results across all datasets are underlined.

This consistent improvement in segmentation quality (evident across all visual examples in Figure 6) directly translates to enhanced perceptual accuracy for the VLA model. In subsequent validation experiments, we quantified how this upward trend in segmentation performance correlates with overall VLA model improvements. This verified that semantic supervision-induced segmentation gains are not isolated; instead, they systematically boost the model's capacity to perceive visual inputs and act upon them.

SeDINO exhibits a inference latency of 10 ms and a peak GPU VRAM footprint of approximately 3 GB for processing a single 224 × 244 image, with its lightweight architectural

design resulting in a negligible impact on the end-to-end inference efficiency of CLIP/DINOv2-based Vision–Language–Action models, suiting practical deployment requirements well.

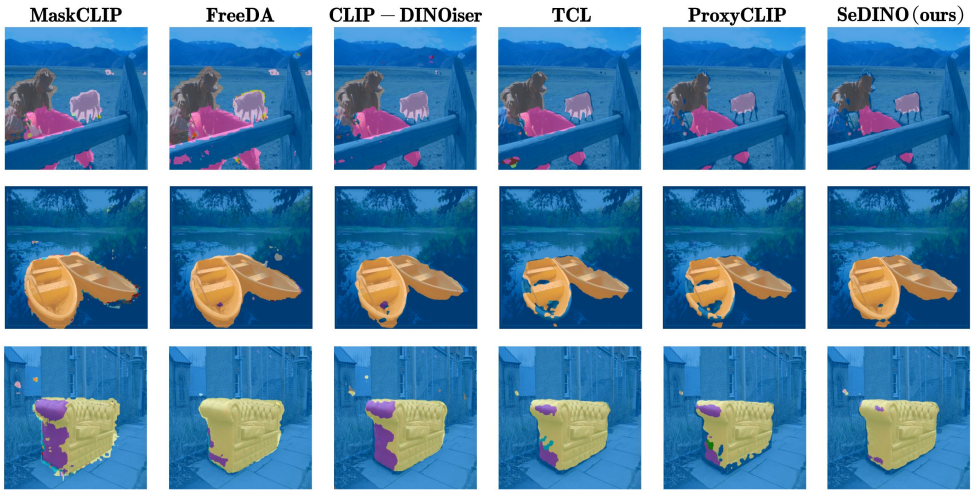

Figure 6. Qualitative comparison of segmentation results for state-of-the-art models.

3.2. VLA Simulated and Real-World Scenario Validation Results

In this section, we apply semantic supervision to the SeDINO encoder based on the semantic information extracted from the textual instructions, as described in the previous methodology. The effect of semantic supervision on the image vectors is shown in the Figure 7 below. From the figure, it is evident that, under semantic supervision, image elements related to the instruction's semantic information are preserved, while unrelated elements are filtered out. This capability helps improve the stability of the VLA task execution. In subsequent work, we will use OpenVLA-7B as the base model for the VLA model and select DINOv2-base as the visual backbone for SeDINO. We plan to conduct further tests on the LEBRO dataset and real-world scenes, where we expect to see a significant performance improvement of the integrated model.

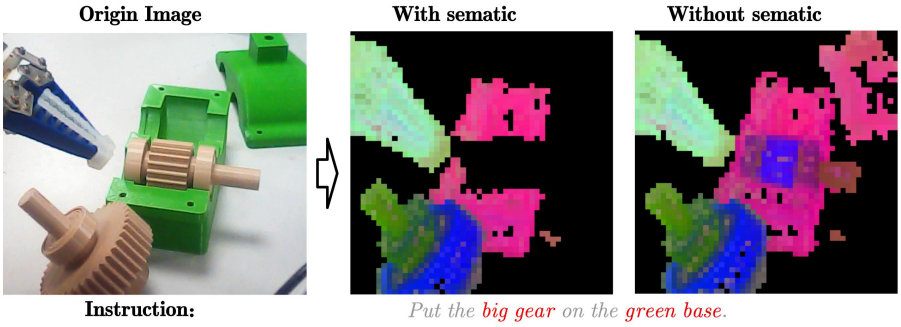

Figure 7. Impact of semantic supervision on image feature extraction.

To further analyze the impact of semantic supervision on visual encoders, three sets of grasping experiments were conducted using CLIP, DINO, and our proposed SeDINO, respectively. For each group, the average trajectory was calculated based on 20 successful experiments, as shown in Figure 8. From the results, it can be observed that, when SeDINO is adopted as the visual encoder, the average trajectory of the robotic arm is significantly smoother. In contrast, the average trajectories of the other two groups exhibit noticeable deviations. Combined with the analysis of image feature extraction results presented earlier, this deviation is attributed to the interference of image elements irrelevant to the instruction with the action generation of the VLA model. When SeDINO serves as the visual encoder,

irrelevant image elements are effectively filtered out, enabling the VLA model to generate smoother actions that directly move the robotic arm near the grasping target. It is worth noting that the calculated average trajectory points are three-dimensional coordinates in the robotic arm coordinate system, which need to be first transformed into the camera coordinate system using a pre-calibrated rotation matrix and then mapped to the image pixel coordinates based on the camera intrinsic parameters.

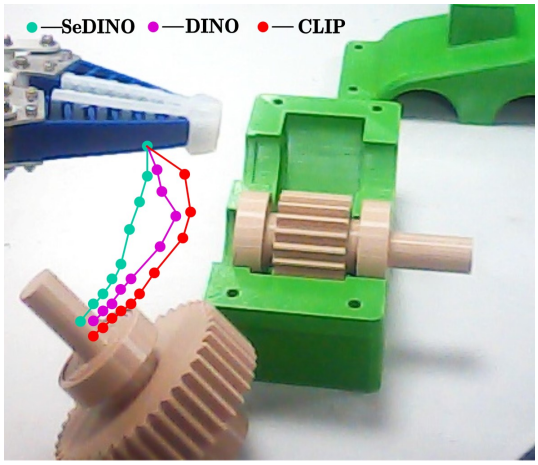

Figure 8. Impact of visual encoders on action command generation in VLA model.

To comprehensively validate the effectiveness of our proposed method, we conducted experimental evaluations in both simulated and real-world environments. For simulation validation, we utilized four datasets from the widely recognized LIBERO benchmark, covering various task categories. In real-world testing, we set up practical scenarios using a UR robotic arm equipped with an RGB camera, performing assembly tasks with a 3D-printed gear reducer as the target object. During both simulation and real-world experiments, we kept the same parameter settings for fine-tuning to ensure consistency.

To reduce memory consumption, we employed gradient accumulation, batching multiple forward and backward passes before updating the model weights. According to the LIBERO benchmark, tasks are categorized into four main types: spatial, object, goal, and long-term tasks. We applied a discrete label-based approach and a pre-action sequence optimization method to fine-tune the OpenVLA-7B model separately on each of these four datasets. The simulation results for CLIP-, DINO-, and SeDINO-fine-tuned models are visualized in Figure 9. While numerical performance values for each task category are detailed in our experimental tables, we emphasize the key trend observed: SeDINO drives consistent and meaningful performance improvements across all four LIBERO task categories (spatial, object, goal, and long-term), with the relative gain remaining stable rather than being confined to a single task type. This uniform upward trend in performance across diverse task categories validates that the semantic supervision mechanism in SeDINO confers broad-based improvements to VLA model capability rather than task-specific incremental gains.

To effectively validate the model's performance in real-world scenarios, we established a comprehensive testing system to conduct thorough evaluations of the improved model. The system utilizes a UR robotic arm equipped with a flexible gripper, enabling versatile object gripping and placement operations. A third-person RGB camera is selected to capture more comprehensive and clearer scene information. Prior to the experiments, we manually collected 90 sets of operational data, with each set comprising nearly 300 frames. Each frame corresponds to real-time robotic arm control data, as shown in Figure 10 below. To facilitate efficient data loading during fine-tuning, the collected data were converted

into the RLDS format, which is optimized for iterative model training and loading. This setup ensures that the model can be effectively tested and refined in practical, dynamic environments, providing valuable insights into its real-world applicability and robustness.

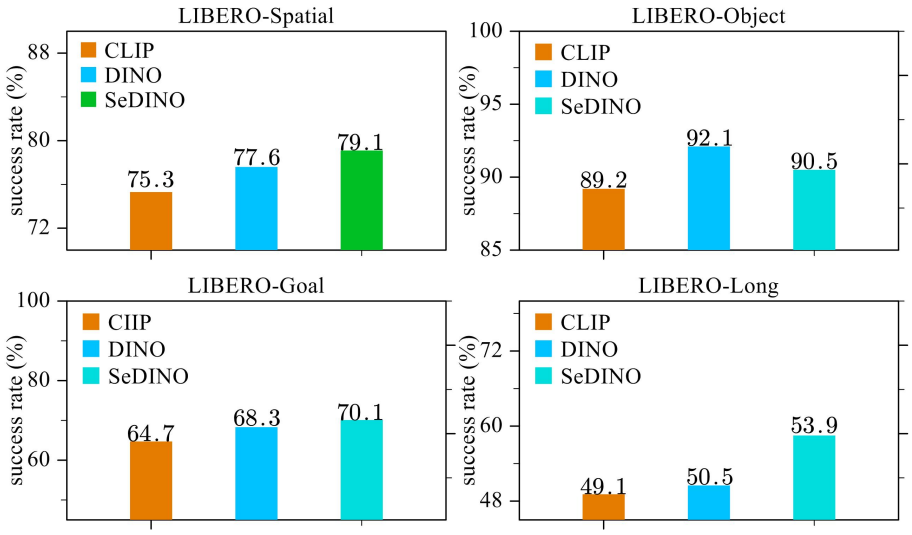

Figure 9. Evaluation results on the LIBERO dataset.

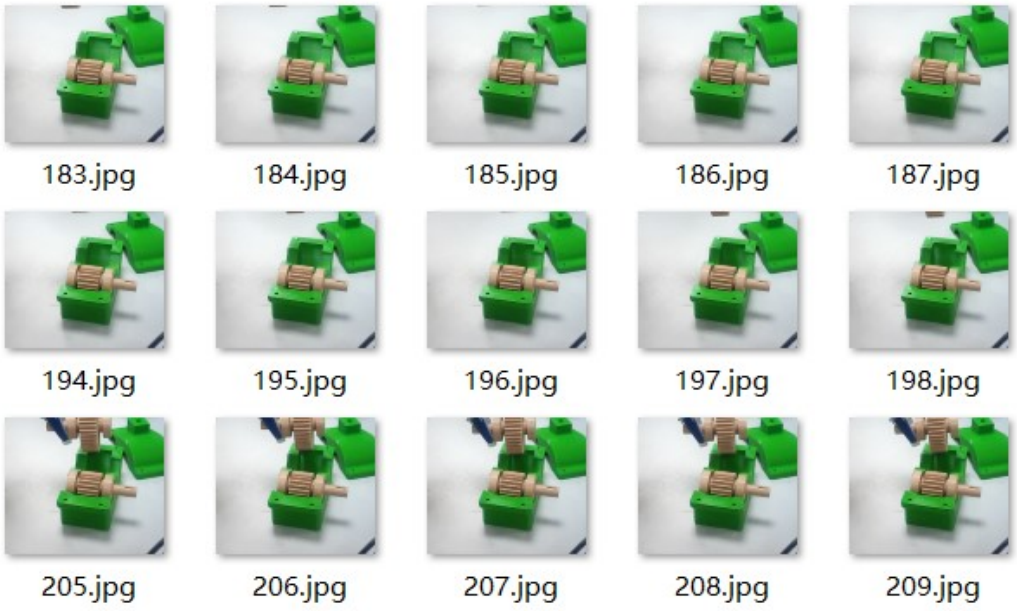

Figure 10. Fine-tuning dataset for real-world scenario tasks.

The results for CLIP, DINO, and SeDINO models achieved for real experimental data are presented in Figure 11. We focus here on the key performance trend: SeDINO consistently outperforms both CLIP and DINO baselines across all real-world test scenarios, with the relative improvement in VLA model performance scaling with the complexity of the real-world environment. This upward trend in performance validates our foundational hypothesis that semantic supervision—where instruction text semantics guide the image encoding process—creates a systematic improvement in VLA model generalization rather than just incremental gains in individual tasks. Explicit semantic supervision during fine-tuning enables SeDINO to generate image representations that are both more accurate and robust to real-world perturbations, translating to a predictable and consistent improvement in task execution success in real-world settings. This trend of sustained performance

improvement underscores the practical value of our semantic supervision approach and its potential to advance visual–language understanding in real-world applications.

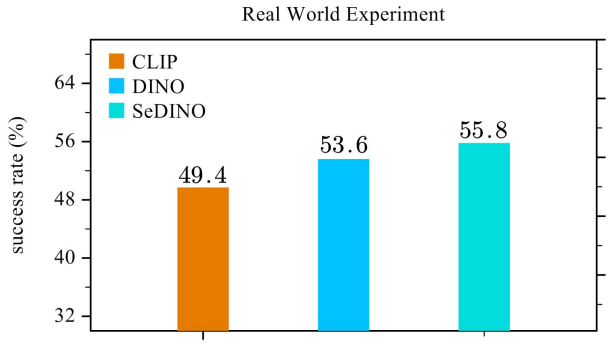

Figure 11. Evaluation results of assembly experiments in real-world scenarios.

To verify the impact of integrating SeDINO on the inference frequency of Vision–Language–Action (VLA) models, we conducted tests on the inference frequencies of VLA models using CLIP, DINO, and SeDINO as visual encoders, respectively. The results, summarized in Table 2 below, indicate that the adoption of SeDINO does not lead to a significant improvement in the inference frequency of VLA models. This is because the parameter count of the large language model (LLM) in VLA models is far greater than that of the visual encoder, and the segmentation latency of SeDINO is within 10 ms, resulting in a negligible impact on the overall inference efficiency of VLA models.

Table 2. Comparison of model inference frequencies.

Visual Encoder	Inference Frequency (Hz)
DINO	4.16
CLIP	4.57
SeDINO	4.03

This section presents a structured analysis of SeDINO's impact on the VLA action chain through three sequential steps. It optimizes feature extraction as the foundation for accurate action generation, compares action trajectories as the direct link between part semantics and action output, and verifies the success rate to realize end-to-end improvement of the VLA pipeline. The section fully elaborates on how image part semantics shape robotic action generation from feature processing to actual execution.

4. Discussion

4.1. Performance Advantages and Mechanisms

When compared with current mainstream models via image data comparisons on standard datasets, our experimental results demonstrate that the proposed SeDINO model outperforms other models in segmentation accuracy. By examining the segmentation results of specific images, SeDINO achieves significant improvements in both the positional accuracy of image elements and regional continuity. This superior segmentation performance stems from the core design of SeDINO, which fuses CLIP's text–image semantic alignment with DINO's robust visual feature extraction, and introduces instruction text-driven semantic supervision to guide the image-encoding process. Specifically, standalone CLIP lacks a fine-grained visual feature localization capability for robotic scenes, while DINO fails to leverage the semantic information inherent in VLA task instructions. SeDINO compensates for these inherent limitations of CLIP and DINO by establishing a semantic mapping between task instructions and visual features, which refines the localiza-

tion of key image elements and ensures the continuity of segmented regions for robotic manipulation-related objects.

In validation experiments combining the VLA model, initial tests with experimental photos verified the effectiveness of semantic supervision in extracting image features. Subsequently, experiments were conducted on four types of datasets within the LIBERO dataset and in real assembly scenarios, and the results indicate that our proposed SeDINO model can significantly enhance VLA model performance across all task categories and real-world scenarios. The performance gain in VLA tasks is attributed to the fact that SeDINO's semantically supervised image representations are more aligned with the semantic intent of task instructions; semantic supervision filters out irrelevant visual noise in complex robotic scenes and highlights task-critical image features, which enables the VLA model to more accurately understand the correspondence between visual inputs and language instructions, thus improving the stability and execution accuracy of the model in real assembly and LIBERO benchmark tasks.

Furthermore, the validation of the SeDINO model alone and in conjunction with the VLA model fully demonstrates the effectiveness of semantic supervision in visual feature extraction. Particularly in VLA tasks, where task instructions inherently contain semantic information, leveraging semantic supervision from instructions can improve model stability and further enhance overall performance—this validates that semantic supervision is a general and effective strategy to bridge the gap between visual feature extraction and language instruction understanding in robotic VLA models.

4.2. Limitations

This study presents a semantic supervision framework called SeDINO for robotic VLA models, and, while comprehensive experimental results validate its effectiveness in segmentation and VLA task performance improvement, several key limitations of the method should be explicitly acknowledged. First, semantic supervision in SeDINO may face failure in specific scenarios; the framework relies on strict semantic consistency between instruction texts and visual scenes and will lose effective supervision efficacy when instructions contain ambiguous spatial or object attribute descriptions, or when visual scenes suffer from severe occlusions, blurring, or low illumination. In such cases, the semantically supervised image-encoding process is disrupted, and SeDINO's performance may degrade to the level of the standalone CLIP/DINO baselines. Second, SeDINO's overall performance is highly dependent on the text–image alignment quality of the pre-trained CLIP model; for domain-specific robotic manipulation vocabulary and rare robotic scene descriptions where CLIP exhibits poor alignment performance, the semantic supervision signal extracted by SeDINO will be weak or distorted, which directly reduces the accuracy of mask generation and semantically supervised image feature extraction. Third, we currently cannot easily distinguish whether task execution failures are caused by perceptual errors or action control errors. Analysis of policy learning and semantic impact on final actions is also lacking, which will be improved in future research.

5. Conclusions

To fully leverage the semantic information embedded in the instruction text within the Visual–Language–Action (VLA) task and to supervise the stable extraction of visual features, we proposed a semantically supervised visual encoder named SeDINO (Semantic DINO). This encoder advances the integration of existing pretrained components by fusing DINO's element localization capabilities with CLIP's semantic interaction abilities. Additionally, we employed an MLP network to align the semantic vectors output by CLIP's text encoder

with the image feature vectors derived from DINO, thereby maximizing the complementary strengths of DINO's localization and CLIP's semantic understanding.

We further incorporated the constructed SeDINO into the VLA model, using OpenVLA-7B and DINOv2-base as backbone models. We first validated SeDINO across six mainstream image datasets, demonstrating that its segmentation performance surpasses that of current state-of-the-art models. Subsequently, we evaluated the improved VLA model on the LIBERO dataset and real-world scenarios, with results showing that the application of our enhanced encoder leads to significant performance improvements. These experimental results validate our initial hypothesis: that leveraging the inherently semantic information contained in instruction texts to supervise the image-encoding process can effectively enhance the performance of VLA models. While SeDINO exhibited clear advantages in aligning visual features with task instruction semantics, its effectiveness was contingent on strict text–visual semantic consistency and high-quality CLIP pre-training alignment, with scalability to larger VLA models and diverse robotic environments to be explored in future work.

Author Contributions: Conceptualization, S.T., D.Y., L.C., Z.L. (Zhaoming Liu) and H.W.; Methodology, S.T., L.C., Z.L. (Zhaoming Liu), H.W. and Z.L. (Zixuan Li); Software, S.T. and H.L.; Validation, S.T. and H.L.; Formal analysis, S.T., L.C., Z.L. (Zhaoming Liu), H.W. and Z.L. (Zixuan Li); Investigation, S.T.; Resources, S.T.; Data curation, S.T., L.C., Z.L. (Zhaoming Liu) and H.W.; Writing—original draft, S.T.; Writing—review & editing, S.T.; Visualization, S.T. and H.L.; Supervision, S.T. and D.Y.; Project administration, D.Y., Z.L. (Zixuan Li) and H.L.; Funding acquisition, D.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key Research and Development Program of China (no. 2023YFB4705100), Joint Fund Project (no. DICP&SIA UN202501), and Basic Research Project (no. 2023JC1K15).

Data Availability Statement: The datasets presented in this article are not available because the data are part of an ongoing study, and we are unable to provide access to them at this stage. No requests for the datasets will be accepted temporarily.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Kim, M.J.; Pertsch, K.; Karamcheti, S.; Xiao, T.; Balakrishna, A.; Nair, S.; Rafailov, R.; Foster, E.; Lam, G.; Sanketi, P. Openvla: An open-source vision-language-action model. *arXiv* **2024**, arXiv:2406.09246.
2. Lu, G.; Guo, W.; Zhang, C.; Zhou, Y.; Jiang, H.; Gao, Z.; Tang, Y.; Wang, Z. VLA-RL: Towards Masterful and General Robotic Manipulation with Scalable Reinforcement Learning. *arXiv* **2025**, arXiv:2505.18719. [CrossRef]
3. Pan, C.; Junge, K.; Hughes, J. Vision-language-action model and diffusion policy switching enables dexterous control of an anthropomorphic hand. *arXiv* **2024**, arXiv:2410.14022.
4. Liang, Z.; Li, Y.; Yang, T.; Wu, C.; Mao, S.; Pei, L.; Yang, X.; Pang, J.; Mu, Y.; Luo, P. Discrete Diffusion VLA: Bringing Discrete Diffusion to Action Decoding in Vision-Language-Action Policies. *arXiv* **2025**, arXiv:2508.20072.
5. Oquab, M.; Darcet, T.; Moutakanni, T.; Vo, H.; Szafraniec, M.; Khalidov, V.; Fernandez, P.; Haziza, D.; Massa, F.; El-Nouby, A. Dinov2: Learning robust visual features without supervision. *arXiv* **2023**, arXiv:2304.07193.
6. Dong, X.; Bao, J.; Zheng, Y.; Zhang, T.; Chen, D.; Yang, H.; Zeng, M.; Zhang, W.; Yuan, L.; Chen, D. Maskclip: Masked self-distillation advances contrastive language-image pretraining. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 10995–11005.
7. Wysoczańska, M.; Siméoni, O.; Ramamonjisoa, M.; Bursuc, A.; Trzciński, T.; Pérez, P. CLIP-DINOiser: Teaching CLIP a Few DINO Tricks for Open-Vocabulary Semantic Segmentation. *Lect. Notes Comput. Sci.* **2023**, *13752*, 320–337.
8. Lan, M.; Chen, C.; Ke, Y.; Wang, X.; Feng, L.; Zhang, W. Proxyclip: Proxy Attention Improves CLIP for Open-Vocabulary Segmentation. In *Computer Vision—ECCV 2024*; Lecture Notes in Computer Science; Springer Nature: Cham, Switzerland, 2023; pp. 70–88.
9. Wang, Y.; Shen, X.; Yuan, Y.; Du, Y.; Li, M.; Hu, S.X.; Crowley, J.L.; Vaufreydaz, D. Tokencut: Segmenting Objects in Images and Videos with Self-supervised Transformer and Normalized Cut. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 15790–15801. [CrossRef] [PubMed]

10. Rodriguez, D.S.; Gomez, A.E.R.; Serrezuela, R.R. Development of an embedded diagnostic tool for visual misalignment screening. *HardwareX* **2025**, *2025*, e00692. [CrossRef] [PubMed]
11. Rodriguez Serrezuela, R.; Zamora, R.S.; Hermosilla, D.M.; Gomez, A.E.R.; Reyes, E.M. Hybrid Convolutional Vision Transformer for Robust Low-Channel sEMG Hand Gesture Recognition: A Comparative Study with CNNs. *Biomimetics* **2025**, *10*, 806. [CrossRef] [PubMed]
12. Everingham, M.; Van Gool, L.; Williams, C.K.I.; Winn, J.; Zisserman, A. The Pascal Visual Object Classes Challenge 2012 (VOC2012) Results (2012). *J. Vis. Comput.* **2012**, *10*, 142–149.
13. Mottaghi, R.; Chen, X.; Liu, X.; Cho, N.-G.; Lee, S.-W.; Fidler, S.; Urtasun, R.; Yuille, A. The Role of Context for Object Detection and Semantic Segmentation in the Wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: New York, NY, USA, 2014; Volume 2014, pp. 891–898.
14. Caesar, H.; Uijlings, J.; Ferrari, V. Coco-stuff: Thing and Stuff Classes in Context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*; IEEE: New York, NY, USA, 2018; Volume 10, pp. 1209–1218.
15. Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; Schiele, B. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*; IEEE: New York, NY, USA, 2016; Volume 2016, pp. 3213–3223.
16. Zhou, B.; Zhao, H.; Puig, X.; Fidler, S.; Barriuso, A.; Torralba, A. Scene parsing through ADE20K dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*; IEEE: New York, NY, USA, 2017; Volume 2017, pp. 633–641.
17. Xu, J.; De Mello, S.; Liu, S.; Byeon, W.; Breuel, T.; Kautz, J.; Wang, X. Groupvit: Semantic Segmentation Emerges from Text Supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: New York, NY, USA, 2022; Volume 1, pp. 18134–18144.
18. Cha, J.; Mun, J.; Roh, B. Learning to Generate Text-Grounded Mask for Open-World Semantic Segmentation from Only Image-Text Pairs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: New York, NY, USA, 2023; pp. 11165–11174.
19. Zhou, C.; Loy, C.C.; Dai, B. Extract free dense labels from clip. In *European Conference on Computer Vision*; IEEE: New York, NY, USA, 2022; pp. 696–712.
20. Wysoczańska, M.; Ramamonjisoa, M.; Trzciński, T.; Siméoni, O. Clip-diy: Clip Dense Inference Yields Open-Vocabulary Semantic Segmentation for-Free. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*; IEEE: New York, NY, USA, 2024; Volume 1, pp. 1403–1413.
21. Wang, F.; Mei, J.; Yuille, A. Sclip: Rethinking Self-Attention for Dense Vision-Language Inference. In *European Conference on Computer Vision*; Springer Nature: Cham, Switzerland, 2024; Volume 1, pp. 315–332.
22. Lan, M.; Chen, C.; Ke, Y.; Wang, X.; Feng, L.; Zhang, W. Clearclip: Decomposing clip representations for dense vision-language inference. In *European Conference on Computer Vision*; Springer Nature: Cham, Switzerland, 2024; Volume 1, pp. 143–160.
23. Hajimiri, S.; Ayed, I.B.; Dolz, J. Pay Attention to Your Neighbours: Training-Free Open-Vocabulary Semantic Segmentation. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*; IEEE: New York, NY, USA, 2025; pp. 5061–5071.
24. Barsellotti, L.; Amoroso, R.; Cornia, M.; Baraldi, L.; Cucchiara, R. Training-Free Open-Vocabulary Segmentation with Offline Diffusion-Augmented Prototype Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: New York, NY, USA, 2024; Volume 1, pp. 3689–3698.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.