Proceedings of the 4th International Conference on Machine Learning and Automation
DOI: 10.54254/2755-2721/2026.38142

Vision-Language-Action Intelligence for Traditional Industrial Robotic Arms

Weijian Wan

Faculty of Engineering, The University of New South Wales, Sydney, Australia
wanweijian2024@gmail.com

Abstract. Vision-language-action (VLA) intelligence provides a unified route for connecting visual perception, language understanding, and executable control in traditional industrial robotic arms. This paper examines the transition from modular perception-planning-control pipelines to generalist policies that predict robot actions from multimodal observations and instructions. It reviews representative architectures, including RT-1, RT-2, Octo, and OpenVLA, and compares three major action-representation strategies: discrete action tokens, continuous action prediction, and specialised policy heads. Particular attention is given to the role of large-scale robot datasets, vision-language pre-training, action tokenization, and embodiment-specific adaptation. The analysis further considers the progression from multimodal intelligence to embodied intelligence and distinguishes cross-task, cross-environment, and cross-embodiment generalization. Existing results indicate that shared backbones and heterogeneous trajectory data can reduce the cost of adapting policies to new tasks and robot platforms, but reliable industrial deployment still depends on control precision, inference latency, safety constraints, data consistency, and closed-loop feedback. The paper concludes that practical VLA systems for industrial arms will require coordinated advances in data standards, model efficiency, controller integration, and hardware-aware evaluation.

Keywords: Vision-language-action, industrial robotic arms, embodied intelligence, robot learning

1. Introduction

Industrial robotic arms offer high motion accuracy, stability, and repeatability, and are widely used for handling, welding, assembly, and sorting. Most existing systems follow a modular perception-decision-planning-control architecture. The function of each module is clear, which also makes debugging easier. However, these systems depend on predefined programs and fixed procedures. When object positions, task requirements, or production environments change, the recognition method, motion trajectory, and control parameters often need to be adjusted again.

In recent years, large language models (LLMs) and vision-language models (VLMs) have begun to be applied to robotics. LLMs can understand language and decompose tasks, while VLMs connect images with language so that a robot can identify objects and interpret instructions. Robot systems are therefore moving from separate modules toward multimodal information integration [1].

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2026.38142

However, these models usually provide only text descriptions or task steps. Actual motions still depend on a motion planner and a low-level controller, leaving a gap between task understanding and physical execution.

Vision-language-action (VLA) models add robot actions to visual and linguistic information. Based on scene images, language instructions, and robot states, a VLA model predicts the position and orientation of the end effector or the state of the gripper, thereby connecting multimodal information with control commands. VLA research also emphasizes generalization. This refers to whether a model can use previously learned knowledge to handle new objects, instructions, or task combinations, rather than merely repeating tasks seen during training.

VLA has now become an important direction in embodied intelligence and robot manipulation. RT-2 transfers vision-language knowledge to robot control and demonstrates the possibility of handling new objects and instructions [2]. OpenVLA releases its model weights and training methods, providing a basis for fine-tuning and evaluation on different robot platforms [3]. Related studies already cover grasping, placing, and multitask manipulation, although most experiments are still conducted in laboratory or tabletop settings.

From a theoretical perspective, VLA places visual perception, language understanding, and action generation in one framework, providing a new way to reduce information loss between modules. From an application perspective, if a robotic arm can adjust its operation according to visual information and language requirements, less repeated programming and reconfiguration may be needed when a task changes, improving adaptability in flexible manufacturing. At present, VLA can also work with existing motion planning, safety monitoring, and low-level control systems rather than completely replacing conventional control.

This paper first introduces the main concepts, technical foundations, and development of VLA. It then analyses major model architectures, training data, and action-generation methods. Next, it discusses representative studies and the application of VLA to grasping, handling, and assembly with six-degree-of-freedom industrial robotic arms. Finally, it summarises the remaining problems in generalization, control accuracy, real-time performance, and safety, and discusses possible future directions.

2. Evolution of VLA architectures and unified control mechanisms

2.1. From modular robot policies to end-to-end VLA architectures

Traditional robot control systems usually design perception, task planning, motion planning, and low-level control as separate modules. A camera first identifies and locates the target, a planner then generates a trajectory, and a controller finally drives the joints. This structure makes faults easier to locate and remains common on fixed production lines. However, interfaces between modules must be specified in advance, and visual errors may continue to affect grasp points and motion trajectories. When an object, robotic arm, or task changes, the recognition model, rule base, and control parameters often require adjustment. Early end-to-end imitation learning could predict actions directly from images, but it was generally trained on only a small number of tasks. VLA does not completely remove low-level control. Instead, it places vision, language, and action in the same learning process so that high-level semantics can directly influence action generation.

Brohan et al. first proposed RT-1, which places images, natural-language instructions, and robot actions in a single Transformer policy. The model extracts image features, compresses them through TokenLearner, and then predicts discrete action tokens. The research team collected approximately 130,000 real-robot episodes, covering more than 700 tasks, and tested seen tasks, unseen tasks, and

DOI: 10.54254/2755-2721/2026.38142

background interference [4]. The experiments showed that one model can learn from diverse robot data without training a separate policy for every task. RT-1 also outperformed the comparison methods of its time on unseen instructions. However, its knowledge mainly came from robot-operation data, so its visual and language abilities remained limited by the training scenes. RT-1 therefore illustrates how robot policies began to move from single-task to multitask learning.

Building on RT-1, Zitkovich et al. proposed RT-2. It discretises continuous actions into text tokens and jointly fine-tunes robot trajectories with internet-scale data such as visual question answering. The same model can therefore process images and language while also producing executable robot actions. The authors conducted about 6,000 evaluations involving new objects, new backgrounds, new environments, and instructions requiring simple semantic reasoning [2]. For example, the robot could select a target according to its size, symbol, or function. The results indicate that some knowledge learned by a vision-language model from web data can be transferred to robot control. RT-2 is representative here because it does more than add language input to a robot: it converts actions into outputs that can be predicted together with words.

Research then expanded from one robot to multiple robot platforms. The Octo Model Team proposed the open-source Octo policy and pre-trained it on about 800,000 trajectories from Open X-Embodiment. The researchers evaluated zero-shot use and fine-tuning on nine robot platforms. Octo can accept a language instruction or a goal image, and its input modules and action head can be modified for new sensors and action spaces. Fine-tuning for a new platform can be completed within several hours on a consumer GPU [5]. Octo does not require every robot to use the same action interface. It keeps a shared model backbone while adapting part of the architecture to each robot. In this review, it mainly demonstrates how VLA models began to support different robot embodiments.

Kim et al. proposed OpenVLA, which applies an open-source VLM to robot action prediction. It uses Llama 2 7B as the language backbone and combines SigLIP and DINOv2 to extract different visual features. The model was trained on 970,000 real-robot trajectories. In comparisons across multiple robots and 29 tasks, OpenVLA achieved an absolute task-success rate 16.5% higher than the 55-billion-parameter RT-2-X, although it had only about one seventh as many parameters [3]. This result suggests that a larger model is not automatically better; visual features and training data also affect manipulation performance. From RT-1 to OpenVLA, researchers first enabled one model to learn many tasks, then introduced web-based vision-language knowledge and data from different robots. A generalist policy does not mean that a model can directly control every robot. Rather, part of its capability can be reused when the task or platform changes, reducing the amount of retraining and programming required.

2.2. Multimodal representation and unification of the action space

At the algorithmic level, VLA must process images, language, and actions, although these three data types are different. Images are usually divided into visual patches, language is represented by tokens, and robotic-arm actions are continuous values such as position, orientation, joint angles, or gripper states. Most models first extract image features and map them into a space that a language model can process, allowing images and instructions to participate in the same computation. Actions are more difficult to handle: coarse divisions may reduce robotic accuracy, whereas complex modules can slow training and inference. Common approaches include discrete action tokens, continuous action prediction, and a separate policy head.

Discrete action tokens are represented by RT-1, RT-2, and OpenVLA. They divide each dimension of a continuous action into intervals; RT-1, RT-2, and OpenVLA discretize each action dimension into 256 bins and predict the corresponding bin as a token [2-4]. This enables language

DOI: 10.54254/2755-2721/2026.38142

instructions and robot actions to use similar prediction methods and makes it easier to reuse existing VLM architectures. However, dividing continuous values into bins introduces quantisation error. If an action sequence contains many tokens, autoregressive generation may also be slow. To reduce the number of tokens in high-frequency action sequences, Pertsch et al. proposed FAST, which first applies a discrete cosine transform to the action sequence and then compresses the tokens using byte-pair encoding. The universal FAST+ action tokenizer was trained on one million real-robot action trajectories and achieved results close to diffusion-based VLA models in high-frequency manipulation while reducing training time by as much as five times [6]. This shows that the effectiveness of discrete actions also depends on how tokens are divided and compressed.

Continuous action prediction retains numerical action values without placing every dimension into bins. Chi et al. proposed Diffusion Policy, which treats visually conditioned action sequences as a conditional denoising process. Across 15 tasks in four robot-manipulation benchmarks, it improved performance by an average of 46.9% over previous methods [7]. It can represent multiple possible action distributions and generate a relatively smooth action trajectory as a block, but normally requires several denoising steps during inference. Black et al. proposed π0, which adds an action expert beside a pre-trained VLM and uses flow matching to generate continuous action chunks. The model was pre-trained on data from seven robot configurations and 68 tasks, and can process action sequences at up to 50 Hz. Its experiments included dexterous tasks such as folding clothes, clearing a table, and assembling a box [8]. Continuous methods reduce quantisation error and are more suitable when trajectory smoothness and control frequency are important. Their disadvantage is greater sampling computation, which creates additional demands for deployment and real-time scheduling.

Another approach is to add a separate policy head. Vision and language are processed by a shared backbone, while a specialised module generates robot actions. Octo uses readout tokens to aggregate observation information and a smaller diffusion action head to predict continuous actions [5]. When the robot changes, the backbone can be retained while only the action head is adjusted. Li et al. proposed CogACT, which uses a VLM as a cognitive module and adds an action module based on a Diffusion Transformer. Tests with five robot embodiments in simulation and real environments showed that sequence diffusion improved task success and adaptability across robots [9]. This architecture does not require the VLM to complete all action prediction by itself, allowing the action module to focus more directly on robot control. However, the connection between the two modules and the allocation of training data still need to be adjusted for each task. The three approaches have different strengths: discrete tokens integrate easily with language models, continuous prediction better preserves action precision, and a separate action head is easier to adapt to a robot controller. For industrial robotic arms, the control cycle, allowable error, computing resources, and safety requirements must also be considered instead of comparing only average experimental success rates.

Table 1. Comparison of three action-representation methods

Action representation	Representative models	Main advantages	Main limitations
Discrete action tokens	RT-1, RT-2, OpenVLA, FAST	Unified semantics; convenient training	Quantisation error; autoregressive generation
Continuous action prediction	Diffusion Policy, π0	Higher precision; smooth trajectories	Iterative sampling; greater computation

DOI: 10.54254/2755-2721/2026.38142

Table 1. (continued)

Separate policy head	Octo, CogACT	Platform adaptation; control specialisation	Module connection; data allocation

3. Key development directions for real-world embodied manipulation

3.1. From multimodal intelligence to embodied intelligence

Embodied intelligence generally refers to an agent that relies on a physical body to continuously perceive and act in a real or simulated environment, adjusting its next action according to the result. Here, the embodiment is not limited to the external form of a robotic arm. It also includes cameras, force sensors, joints, grippers, and the control system that executes actions. Unlike models that process only text or images, the output of an embodied system changes the surrounding environment, and the resulting change becomes the next input. For example, after a robotic arm grasps a cup, the cup position, gripper force, and camera image all change. The model must use this feedback to judge whether the cup is secure and how to move next. Embodied intelligence therefore emphasises a continuous closed loop of perception, decision, action, and feedback rather than a single recognition result or answer.

Multimodal intelligence provides an important foundation for this process. Images describe objects, positions, and spatial relations; language expresses task goals; and robot states record joint positions and gripper conditions. Conventional multimodal models mainly study how different types of information correspond, for example, by answering a question about an image or locating an object from a text instruction. Such a model can explain what is present in a scene and what an instruction requires, but it usually does not bear the consequences of an action. Embodied intelligence further requires the model to use this information for physical operation and to handle changes such as occlusion, contact error, and target movement during execution. Multimodal understanding is therefore one part of embodied intelligence, but a model enters embodied operation only when it produces physical actions and receives environmental feedback.

Driess et al. proposed PaLM-E, which sends visual information, continuous state estimates, and text inputs to a pre-trained language model and jointly trains it on robot planning, visual question answering, and image captioning. The model can produce multi-step task plans from camera images and robot states, and it also shows positive transfer across tasks. PaLM-E is important because it connects web-based language knowledge with a robot's current observations, rather than reasoning from text alone. However, its robot output mainly consists of high-level plans, while concrete actions still require a low-level policy. This work is therefore more useful for explaining how a multimodal model obtains context related to the physical environment than for showing that end-to-end action control has already been achieved [10].

On this basis, Zitkovich et al. proposed RT-2, representing robot actions as tokens that the model can predict and jointly training on internet vision-language data and robot trajectories. About 6,000 evaluations covered new objects, new backgrounds, and instructions not directly included in the robot data. The robot could not only recognise a target but also select it according to its purpose, size, or symbol. Compared with PaLM-E, RT-2 places semantic understanding and action output more directly within one model, so that understanding a task can influence how the robot acts. It demonstrates a key change from multimodal to embodied intelligence: evaluation must consider not only whether the answer is correct, but also whether the physical action completes the task in the real environment [2].

DOI: 10.54254/2755-2721/2026.38142

However, real embodied manipulation cannot depend only on a model's interpretation of a single image. As the robotic arm approaches an object, it may block the camera, and the object may slip when the gripper closes. The system must continuously receive visual, position, or force feedback and correct the action within the control cycle. In industrial robotic arms, grasping accuracy, operating speed, and collision risk directly affect production. The development of VLA therefore involves more than scaling up a model. The model must run reliably on a physical embodiment and cooperate with state estimation, safety limits, and low-level control.

3.2. Generalization across tasks and embodiments

Generalization in embodied systems can be considered at three levels. The first is cross-task generalization, in which a model uses learned operations to complete a new instruction or task combination. The second is cross-environment generalization, in which the policy remains effective when lighting, backgrounds, object positions, or disturbances change. The third is cross-embodiment generalization, in which a model is transferred from one robot to another. The third level is the most difficult because robotic arms differ in degrees of freedom, workspace, camera position, gripper structure, and action representation. An end-effector action learned on a seven-degree-of-freedom arm cannot directly serve as a joint command for another arm. Cross-embodiment transfer is therefore not a simple copy of model parameters; it requires the system to identify visual, linguistic, and operational knowledge that different robots can share.

The Open X-Embodiment Collaboration explored this problem from the data perspective. The team consolidated more than one million real-robot trajectories from 22 robot embodiments and 21 institutions into a common format, covering 527 skills and 160,266 tasks, and trained the RT-X models. The experiments showed that multi-robot data can provide positive transfer for some platforms, allowing experience from other robots to improve performance on the target robot. The contribution is not only a larger dataset, but also evidence that cross-embodiment research must first address data interfaces. If datasets use completely different camera names, action coordinates, and task descriptions, they remain difficult to train together even when the amount of data is large. Open X-Embodiment therefore provides a shared data foundation for later generalist policies [11].

Using this unified data, the Octo Model Team proposed Octo and pre-trained it on about 800,000 trajectories from Open X-Embodiment. Octo retains a shared Transformer backbone while allowing researchers to modify the input modules and action head for new sensors and action spaces. The authors evaluated it on nine robot platforms and showed that fine-tuning for a new platform can be completed within several hours using a consumer GPU. Octo does not claim that one action head can directly control every robot. Instead, it uses the pre-trained model as a stronger starting point and adapts it with a small amount of embodiment-specific data. This design reflects the current reality of cross-embodiment generalization and shows that generality often means reducing retraining costs rather than eliminating training [5].

Kim et al. proposed OpenVLA, combining an open-source vision-language model with robot action prediction and training a 7-billion-parameter model on 970,000 real-robot trajectories. The team evaluated the model on multiple robots and 29 tasks. OpenVLA achieved better average performance than several closed-source comparison models and released both its weights and fine-tuning methods. For cross-task and cross-environment generalization, it can use vision-language pre-training to recognise new objects and instructions. For cross-embodiment generalization, it still requires further fine-tuning on demonstrations from the target robot. OpenVLA is representative because it connects generalization with a practical adaptation process, allowing researchers to test

DOI: 10.54254/2755-2721/2026.38142

whether the model is actually effective on their own robots rather than relying only on results reported in the original paper [3].

Existing studies suggest that vision-language pre-training and multitask robot data can improve cross-task generalization, while greater scene diversity can strengthen cross-environment generalization. Cross-embodiment generalization still depends on unified data, action conversion, and fine-tuning on the target platform. For a six-degree-of-freedom industrial robotic arm, a practical approach is to retain VLA's shared understanding of vision and language, adapt the action module to the workspace, gripper, and control interface, and continue to use collision detection and a low-level controller. A future general embodied system may therefore consist of a shared multimodal backbone, replaceable embodiment interfaces, and reliable hardware control.

4. Conclusion

This study has examined vision-language-action intelligence for traditional industrial robotic arms. Conventional robotic arms generally use a modular architecture in which perception, decision-making, planning, and control are separated. This approach is easy to debug and stable for fixed production tasks, but changes in objects, environments, or requirements often require new recognition rules, trajectories, and control parameters. VLA places vision, language, and action in one learning framework and generates actions from the scene and task instruction, offering a route from task-specific control toward more general robot policies. The development from RT-1 and RT-2 to Octo and OpenVLA shows a gradual shift from multitask learning on one robot to using vision-language knowledge and multi-robot data for improved adaptability. For action generation, discrete action tokens are easy to unify with a language model but may introduce quantisation error and slower generation; continuous action prediction can preserve precision and trajectory smoothness but requires more inference computation; and a separate policy head is easier to adapt to different robots but still requires a suitable connection between its modules. Generalization in real embodied manipulation involves not only new tasks and environments but also transfer between different robotic arms. Multitask data, vision-language pre-training, and unified data formats can improve generalization, but embodiment differences still limit direct transfer. VLA therefore cannot yet completely replace conventional control. A more suitable approach is to let VLA interpret tasks and adapt to change while motion planning, collision detection, and low-level control maintain flexibility, accuracy, and safety.

Future development of VLA for industrial robotic arms requires simultaneous improvements in data, models, control, and hardware. First, training data should cover more production environments and use more consistent formats for images, states, and actions across robots. In addition to successful demonstrations, datasets can retain grasping deviations and human corrections so that a model learns to recognise risk. Second, model architecture must balance general capability with real-time control. Large models can interpret more complex instructions, but industrial operation also requires a stable control cycle. In the short term, the more practical goal is not one model that directly controls every robotic arm, but a system that shares visual and language knowledge and can be adapted quickly to a particular robot. Future studies should also use common evaluation tasks and report success rate, control error, inference time, collisions, and operating stability. VLA is likely to move gradually from laboratories into industrial production only when it meets production requirements and abnormal actions can be detected and constrained.

DOI: 10.54254/2755-2721/2026.38142

References

[1] Zhang, H., Liang, S. T., Li, M. X., Tian, Y. L., Ge, J. W., Yu, H., Li, L. X., & Wang, F. Y. (2025). Vision-language-action models: From the early foundations to the state-of-the-art. *Acta Automatica Sinica, 51*(9), 1922–1950. (In Chinese).

[2] Zitkovich, B., Yu, T., Xu, S., et al. (2023). RT-2: Vision-language-action models transfer web knowledge to robotic control. *Proceedings of the 7th Conference on Robot Learning, PMLR 229*, 2165–2183.

[3] Kim, M. J., Pertsch, K., Karamcheti, S., et al. (2025). OpenVLA: An open-source vision-language-action model. *Proceedings of the 8th Conference on Robot Learning, PMLR 270*, 2679–2713.

[4] Brohan, A., Brown, N., Carbajal, J., et al. (2023). RT-1: Robotics Transformer for real-world control at scale. *Proceedings of Robotics: Science and Systems XIX*. https://doi.org/10.15607/RSS.2023.XIX.025

[5] Ghosh, D., Walke, H. R., Pertsch, K., et al. (2024). Octo: An open-source generalist robot policy. *Proceedings of Robotics: Science and Systems XX*. https://doi.org/10.15607/RSS.2024.XX.090

[6] Pertsch, K., Stachowicz, K., Ichter, B., et al. (2025). FAST: Efficient action tokenization for vision-language-action models. *Proceedings of Robotics: Science and Systems XXI*. https://doi.org/10.15607/RSS.2025.XXI.012

[7] Chi, C., Xu, Z., Feng, S., et al. (2025). Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research, 44*(10–11), 1684–1704. https://doi.org/10.1177/02783649241273668

[8] Black, K., Brown, N., Driess, D., et al. (2025). π0: A vision-language-action flow model for general robot control. *Proceedings of Robotics: Science and Systems XXI*. https://doi.org/10.15607/RSS.2025.XXI.010

[9] Li, Q., Liang, Y., Wang, Z., et al. (2024). CogACT: A foundational vision-language-action model for synergizing cognition and action in robotic manipulation. arXiv preprint arXiv:2411.19650.

[10] Driess, D., Xia, F., Sajjadi, M. S. M., et al. (2023). PaLM-E: An embodied multimodal language model. *Proceedings of the 40th International Conference on Machine Learning, PMLR 202*, 8469–8488.

[11] Open X-Embodiment Collaboration, O'Neill, A., Rehman, A., et al. (2024). Open X-Embodiment: Robotic learning datasets and RT-X models. *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 6892–6903. https://doi.org/10.1109/ICRA57147.2024.10611477