HYPERVLA: EFFICIENT INFERENCE IN VISION-LANGUAGE-ACTION MODELS VIA HYPERNETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Built upon language and vision foundation models with strong generalization ability and trained on large-scale robotic data, Vision-Language-Action (VLA) models have recently emerged as a promising approach to learning generalist robotic policies. However, a key drawback of existing VLAs is their extremely high inference costs. In this paper, we propose HyperVLA to address this problem. Unlike existing monolithic VLAs that activate the whole model during both training and inference, HyperVLA uses a novel hypernetwork (HN)-based architecture that activates only a small task-specific policy during inference, while still retaining the high model capacity needed to accommodate diverse multi-task behaviors during training. Successfully training an HN-based VLA is nontrivial so HyperVLA contains several key algorithm design features that improve its performance, including properly utilizing the prior knowledge from existing vision foundation models, HN normalization, and an action generation strategy. Compared to monolithic VLAs, HyperVLA achieves a similar or even higher success rate for both zero-shot generalization and few-shot adaptation, while significantly reducing inference costs. Compared to OpenVLA, a state-of-the-art VLA model, HyperVLA reduces the number of activated parameters at test time by $90\times$, and accelerates inference speed by $120\times$.

1 INTRODUCTION

Motivated by the great success of foundation models in domains like NLP (GLM et al., 2024; Jiang et al., 2023; Yang et al., 2025; Bai et al., 2023a; DeepSeek-AI et al., 2025; xAI, 2025; Team et al., 2025a; OpenAI et al., 2024; Grattafiori et al., 2024) and CV (Dosovitskiy et al., 2021; Radford et al., 2021a; Yu et al., 2022; Kirillov et al., 2023; Oquab et al., 2024; Wang et al., 2023; Bai et al., 2023b; Chen et al., 2025; Team et al., 2025b) in recent years, robotic learning has been going through a paradigm shift from training moderate-size models on a narrow task distribution to training generalist control policies on large-scale robotic demonstration data collected from a diverse set of real-world scenarios (Firoozi et al., 2023; Hu et al., 2023). Vision-Language-Action (VLA) models (Brohan et al., 2022; 2023; O'Neill et al., 2024; Team et al., 2024; Kim et al., 2024; Black et al., 2024) are one important family of such models, which take language instructions and image observations as input and predict the robot's action output. They usually use existing language and vision foundation models as the backbone to improve generalization, and are further trained on large-scale robotic data to learn the complex mapping from multi-modal inputs to the robot's action output.

While VLAs have shown promising generalization, one key drawback of these models is their extremely high inference costs, e.g., OpenVLA (Kim et al., 2024), a state-of-the-art (SOTA) VLA model, has more than 7B parameters and can only infer at 6Hz even when equipped with an NVIDIA 4090 GPU. Such a high inference cost not only consumes significant memory, computation, and energy, but also makes it hard to solve dexterous tasks that require high-frequency manipulation.

By contrast, conventional methods for robotic learning from before the era of foundation models typically learn compact models that are much smaller than VLAs. Although such models cannot generalize across a diverse set of tasks, they can perform well on the specific task they are trained on given sufficient training data. Hence, the minimal model required to solve a specific task can be much smaller than a VLA with millions or billions of parameters. So a natural question arises:

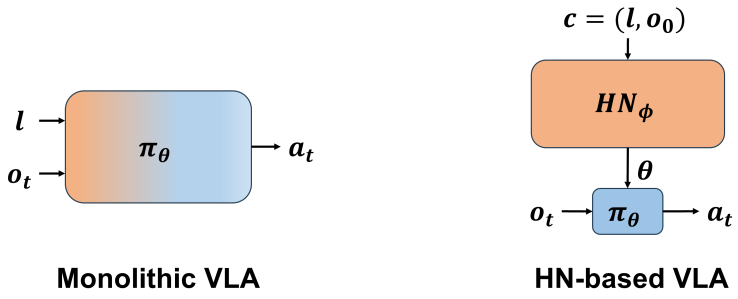

Figure 1: Comparison between the high-level framework of monolithic VLA (left) and HN-based VLA (right). We use orange to represent parameters activated during training, and blue to represent parameters activated at every timestep during inference. The monolithic VLA activates the whole model during both training and inference and is thus colored both orange and blue. By contrast, an HN-based VLA calls the HN at a low frequency only at the beginning of a new episode at test time, and calls a compact base network at every timestep for action prediction.

Can we learn a generalist policy that combines the best of both worlds: the strong generalization ability of VLAs, and the efficient inference of single-task policies? To achieve this, we need to learn a generalist policy with high model capacity to accommodate the diverse behaviors in multi-task data at training time, but only activate a small part of it at test time to keep inference efficient.

In this paper, we realize this goal via hypernetworks (HNs) (Ha et al., 2016). An HN is a network that generates the parameters of another base network conditioned on some context information. Its hierarchical architecture provides a natural way to decouple the skills required to solve different tasks (Xiong et al., 2024), so that we can learn an HN with high model capacity at training time, but only activate a compact HN-generated base network to solve a specific task efficiently at test time.

Therefore, we learn an HN-based VLA that generates policy parameters conditioned on task context c, which in our setting consists of both the language instruction l and the initial image o_0 of an episode (Figure 1). At training time, we train an HN with high model capacity to capture the complex mapping from the task context c to the corresponding policy parameters π_θ^c. At inference time, the large HN is called at a low frequency, only when the task context changes at the beginning of a new episode, while the compact generated policy is called at every timestep to process image observations and output action predictions, which significantly reduces inference cost compared to existing monolithic VLAs that activate the entire model at every timestep during inference.

However, HNs are known to be hard to optimize (Chang et al., 2020; Beck et al., 2023a; Xiong et al., 2024), and training an HN with millions of parameters on large-scale robotic data further aggravates this issue. We thus introduce several key algorithm design features to improve HN learning:

1. **Vision backbone:** While in principle we can generate the whole base policy with an HN, empirically we find it important to use existing vision foundation models as the backbone to improve generalization, as training an HN from scratch on existing robotic datasets, which are relatively small, is prone to overfitting.

2. **HN normalization:** Successful training of neural networks depends heavily on many optimization design choices, which are mainly tailored for training monolithic models and may not generalize to the different optimization dynamics of HNs (Chang et al., 2020; Beck et al., 2023a; Auddy et al., 2024). We thus investigate how parameter updates in HN training differ from standard training, and propose a simple yet effective solution by normalizing the context embedding in HNs, such that the base network parameters can be updated with similar dynamics as directly training the base network.

3. **Action generation strategy:** Unlike most existing VLAs that predict actions via autoregression (Brohan et al., 2022; 2023) or diffusion (Chi et al., 2023; Team et al., 2024), we find that learning a simple linear action head with MSE loss performs better when training an HN-based VLA and further accelerates inference.

We call our method HyperVLA, an HN-based VLA learned with the above algorithm design features. We train HyperVLA on the Open X-Embodiment (OXE) dataset (O'Neill et al., 2024), and evaluate it for both zero-shot generalization to seen and unseen tasks from the training scenarios, and few-shot adaptation to new domains. Compared to existing monolithic VLAs, HyperVLA achieves a similar or higher success rate during evaluation, while significantly improving inference efficiency by only activating a compact HN-generated policy at every timestep during inference, which validates the effectiveness of utilizing HNs for inference acceleration of VLAs. Compared to OpenVLA, a SOTA VLA with the best performance among the baselines, HyperVLA reduces the number of activated parameters at test time by $90\times$, and accelerates inference by $120\times$.

2 BACKGROUND

2.1 VISION-LANGUAGE-ACTION MODELS

Vision-Language-Action (VLA) models aim to learn a generalist robotic control policy, which takes in a language instruction l and image observations o_t at each timestep t and predicts the robot's action a_t. In this paper, we focus on image observations, though other input modalities can be easily integrated into our approach. To achieve good generalization, VLAs are usually built upon vision and language foundation models and trained on large-scale robotic data via behavior cloning (BC). The robotic dataset consists of expert demonstrations collected for different tasks in different scenarios. Each demonstration episode consists of a sequence of image observations and corresponding actions $(o_0, a_0, \ldots, o_{T-1}, a_{T-1}, o_T)$, and optionally a language instruction l if annotated. For each expert observation-action pair (o_t, a_t), the BC loss is defined as $L_{\text{BC}} = (\hat{a}_t - a_t)^2$, where $\hat{a}_t = \pi_\theta(o_t, l)$ is the action predicted by the policy.

2.2 HYPERNETWORKS

A hypernetwork (Ha et al., 2016) is a network that generates some or all of the parameters of a base network conditioned on some context c. This hierarchical architecture offers a powerful tool for multi-task robotic control, as we can generate different policies for different tasks conditioned on their task context. The parameters θ of the base network can be divided into $\theta^{\text{generated}}$ generated by the HN, and θ^{shared} which is not generated by the HN and shared across all the tasks. To generate $\theta^{\text{generated}}$, the HN first encodes the task context with a context encoder f to get a context embedding $e^{\text{context}} = f(c)$, then passes the context embedding through linear output heads to predict $\theta^{\text{generated}}$.

3 HYPERVLA

This section introduces the motivation, architecture, and algorithm design of HyperVLA. In Section 3.1, we analyze why existing monolithic VLAs have high inference costs and how an HN-based VLA can tackle this challenge. Then in Section 3.2 we introduce the architecture of HyperVLA. Finally, in Section 3.3, we propose several key algorithm design features to stabilize HyperVLA training and improve its performance.

3.1 FROM MONOLITHIC TO HN-BASED VLA

Existing VLAs usually have millions or billions of parameters. While such a high model capacity is necessary to accommodate diverse behaviors in multi-task data at training time, it introduces significant computational redundancy at test time as the minimal model required to solve a specific task is often much smaller than a large VLA (Yu et al., 2020; Kim et al., 2024).

Consequently, there is room for inference acceleration if we can only activate a small part of a huge VLA that is sufficient to solve the task at hand. However, since existing VLAs have monolithic architectures that require activating the whole model during both training and inference, they cannot decouple in parameter space the different skills required to solve different tasks (Xiong et al., 2024).

In this paper, we tackle this challenge by learning an HN-based VLA, as the hierarchical architecture of HNs provide a natural way to decouple inter-task and intra-task knowledge. Intuitively, the HN is a generalist that encodes inter-task knowledge about how to map from different task context to the

corresponding policy parameters, while the base network generated by the HN is a specialist that encodes intra-task knowledge about how to solve a specific task. At training time, the whole HN is activated to ensure sufficient model capacity to accommodate the diverse behaviors in multi-task data. However, at test time, we only need to call the HN once at the beginning of each episode to generate a compact task-specific policy that is used for the remainder of the episode.

3.2 THE ARCHITECTURE OF HYPERVLA

The base policy We formulate the base policy as a Vision Transformer (ViT) (Dosovitskiy et al., 2020), which takes the image observation o_t as input to predict the robot's action a_t. Unlike existing VLAs, we do not feed the language instruction l into the base policy as it is already indirectly conditioned on the instruction via the HN that generated its parameters. The base policy consists of the following blocks in sequence (we omit the time index t for simplicity):

1. An image encoder, formulated as a ViT, encodes the image observation o into a sequence of token embeddings $\{e_i^{\text{image}}\}$ for the image patches;
2. A linear projection layer maps $\{e_i^{\text{image}}\}$ into a lower dimension for more efficient inference, represented as $\{e_i^{\text{proj}}\}$;
3. A policy head θ^{policy}, formulated as a small Transformer, takes $\{e_i^{\text{proj}}\}$ and a learnable action token e^{act} as inputs, and updates their token embeddings; and
4. An action head takes updated e^{act} as input to predict the robot's action $\hat{a}$.

The hypernetwork The HN consists of a context encoder parameterized as a Transformer with high model capacity, and linear output heads to generate base policy parameters. The context encoder takes in three inputs:

1. Pretrained instruction embeddings generated by a frozen T5 encoder (Raffel et al., 2020);
2. The class token embedding of the initial image generated by a frozen DINOv2 encoder. We find it helpful to condition the HN on the initial image, as the robot may see the same instruction in different scenarios, and a compact base policy may not have enough model capacity to solve the same task across scenarios with diverse visual appearance. By conditioning policy generation further on the initial image, the HN with high model capacity takes the responsibility of generalization across both instructions and scenarios, while the generated base policy only needs to solve a specific task in a specific scenario. Moreover, we only use the class token outputted by DINOv2 as HN input and discard the image patch tokens to avoid overfitting in the HN; and
3. A learnable task context token that integrates task context information.

The context encoder updates the embeddings of these tokens via self-attention, and the embedding of the task context token is fed into the HN output heads to generate base policy parameters.

3.3 ALGORITHM DESIGN FEATURES

HNs are known to be unstable and hard to optimize (Chang et al., 2020; Beck et al., 2023a; Xiong et al., 2024), and scaling them up to millions of parameters further aggravates this issue. We thus introduce several key algorithm design features that help stabilize and improve HyperVLA training.

3.3.1 VISION BACKBONE

While in principle we can generate the whole base policy by HN, empirically we find that training a large HN from scratch on robotic data alone is prone to overfitting due to the relatively small data size of existing robotic datasets. Instead, we use existing vision foundation models as the image encoder in the base network to improve generalization. Our method is agnostic to the choice of the vision encoder, and empirically we find that DINOv2 (Oquab et al., 2024) achieves the best performance and thus adopt it in HyperVLA.

Similar to previous work (Kim et al., 2024), we find it helpful to fine-tune this vision backbone instead of keeping it frozen when training on robotic data. Furthermore, we use a smaller learning rate for fine-tuning the vision backbone than for HN training because DINOv2 is already well pre-trained and only needs to be fine-tuned at a conservative rate to better align with robotic data.

3.3.2 CONTEXT EMBEDDING NORMALIZATION

Successful training of neural networks depends heavily on many optimization choices, such as network initialization, normalization layers, and gradient transformations. However, such choices are mainly tailored to monolithic models, and may need to be redesigned to fit the different optimization dynamics of HNs. For example, Chang et al. (2020) and Beck et al. (2023a) investigate how to initialize HNs properly so that the base network is initialized in the same way as commonly used initializers, an approach we adopt as well.

However, a proper initialization can only provide a good starting point for HN training and has no direct effect on the parameter update process during training. So in this paper, we further investigate how parameter updates in HN training differ from standard neural network training, and propose a simple yet effective solution by normalizing the context embedding in HNs, such that the base network parameters can be updated with similar dynamics as directly training the base network.

For simplicity, we use SGD in our derivation below. In standard neural network training, each parameter θ_i is updated by $\Delta\theta_i = -\alpha \cdot \frac{\partial L}{\partial \theta_i}$, where L is the loss function and α is the learning rate.

Now we generate θ with an HN. Let us denote the output head of the HN as ϕ, and the context embedding input to the output head as e, then we have $\theta_i = \sum_j e_j \phi_{ij}$. We omit the bias term as it has the same gradient as the base parameter. According to the chain rule, we have $\frac{\partial L}{\partial \phi_{ij}} = \frac{\partial L}{\partial \theta_i}\frac{\partial \theta_i}{\partial \phi_{ij}} = \frac{\partial L}{\partial \theta_i} e_j$, and the parameter update in the HN is $\Delta\phi_{ij} = -\alpha \cdot \frac{\partial L}{\partial \theta_i} e_j$. Then the parameter update in the base network is:

$$\Delta\theta_i = \sum_j (e_j + \Delta e_j)(\phi_{ij} + \Delta\phi_{ij}) - \sum_j e_j \phi_{ij}, \tag{1}$$

$$= \sum_j e_j \Delta\phi_{ij} + \Delta e_j \phi_{ij} + \Delta e_j \Delta\phi_{ij}, \tag{2}$$

$$\approx \sum_j e_j \Delta\phi_{ij} + \Delta e_j \phi_{ij}, \quad \text{(Omit the multiplication of two delta terms)} \tag{3}$$

$$= -\alpha \cdot \frac{\partial L}{\partial \theta_i} \sum_j e_j^2 + \sum_j \Delta e_j \phi_{ij}. \tag{4}$$

If we assume that both ϕ_{ij} and Δe_j are i.i.d. and follow a Gaussian distribution with zero mean, then $\mathbb{E}\left[\sum_j \Delta e_j \phi_{ij}\right] = 0$. Accordingly, we have $\Delta\theta_i \approx -\alpha \cdot \left(\sum_j e_j^2\right) \cdot \frac{\partial L}{\partial \theta_i}$, which indicates that when learning with HNs, the update on the base network parameters is scaled by a factor of $\sum_j e_j^2$ compared to directly optimizing the base network.

In HyperVLA, as the context embedding e is the output of a Transformer context encoder with layer normalization as its final layer, we have $\mathbb{E}\left[e_j^2\right] = 1$ and $\mathbb{E}\left[\sum_j e_j^2\right] = d_e$, where d_e is the dimension of e. Consequently, to keep the scale of parameter update in the base network unchanged, we can simply divide the context embedding by $\sqrt{d_e}$ before feeding it into the output head so that $\mathbb{E}\left[\sum_j e_j^2\right] = 1$ after normalization.

The derivation is different for more complex optimizers like Adam, which makes it much harder to theoretically keep the update scale unchanged like with the SGD optimizer. However, empirically we find that the same normalization operation of dividing the context embedding by $\sqrt{d_e}$ before feeding it into the HN output head still works well in practice.

3.3.3 ACTION GENERATION STRATEGY

Existing VLAs usually predict discretized actions autoregressively (Brohan et al., 2022; 2023; Kim et al., 2024), which requires multiple runs of the same model to generate different action dimensions

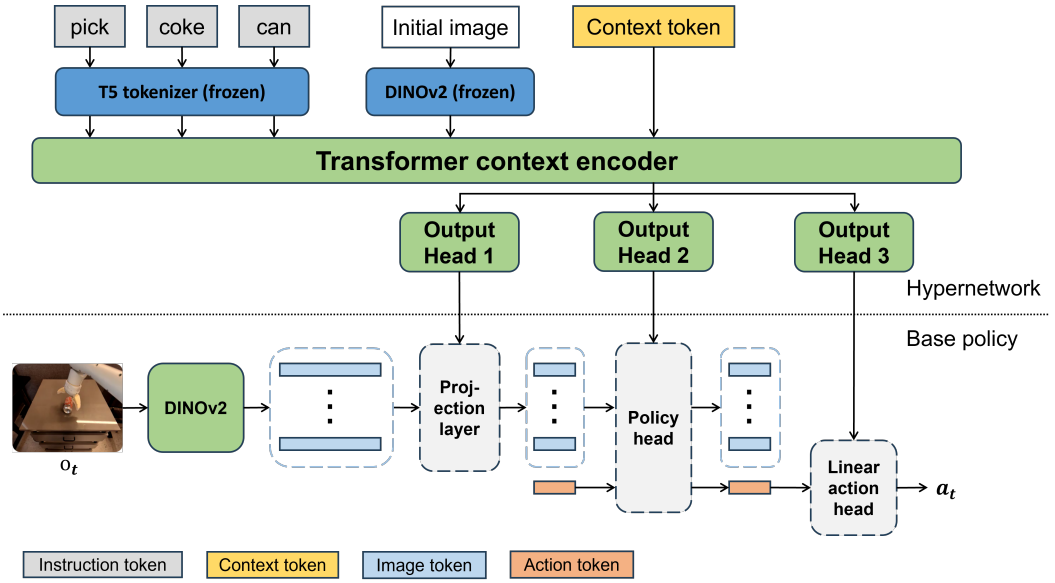

Figure 2: The framework of HyperVLA. The trainable parameters are marked as green blocks, while the HN-generated parameters are marked as light grey blocks with dashed edges.

sequentially, or learn a diffusion action head (Chi et al., 2023) that must be iteratively called to denoise actions (Team et al., 2024), both of which are time-consuming at training and test time. Instead, we find that training a simple linear action head with an MSE loss outperforms these more complicated action generation strategies in HyperVLA, while further reducing training and inference cost. This also agrees with the findings from some recent work (Kim et al., 2025).

Combining these features, the overall framework of HyperVLA is shown in Figure 2.

4 Experiments

Our experiments aim to answer the following questions:

- **Q1:** Can HyperVLA match the zero-shot generalization performance of existing monolithic VLAs on both seen and unseen tasks? (Section 4.2)
- **Q2:** Can HyperVLA adapt to new tasks by fine-tuning on only a few demonstrations, especially for long-horizon tasks? (Section 4.3)
- **Q3:** Can HyperVLA be more inference efficient than monolithic VLAs? (Section 4.4)
- **Q4:** How do the algorithm designs in HyperVLA influence its performance? (Section 4.5)

4.1 Experimental setup

Baselines We compare HyperVLA with the following monolithic VLAs as baselines: (1) **RT-1-X** (O'Neill et al., 2024): uses EfficientNet (Tan & Le, 2019) as the vision backbone, and conditions on the language instruction via FiLM (Perez et al., 2018). Each action dimension is discretized and predicted autoregressively. It has roughly 35M parameters. (2) **Octo** (Team et al., 2024): uses T5 (Raffel et al., 2020) as the language backbone and learns the visual encoder on robotic data alone. It predicts actions via policy diffusion (Chi et al., 2023). It has roughly 200M parameters. (3) **OpenVLA** (O'Neill et al., 2024): uses SigLIP (Zhai et al., 2023) and DINOv2 (Oquab et al., 2023) as the vision backbones, and Llama 2 (Touvron et al., 2023) as the language backbone. Llama 2 is further fine-tuned on robotic data to predict action tokens autoregressively like RT-1-X. It has about 7.6B parameters. All the baselines are trained on the Open X-Embodiment (OXE) dataset (O'Neill et al., 2024), which contains demonstrations collected from different robot embodiments.

We choose these models as the baselines as they are all trained on the OXE dataset and have the same input and output space for the model, which makes it easier to control variates and focus only on the influence of changing the model architecture from monolithic to HN-based. Many later VLAs are trained on larger and different datasets, which makes it hard to tell whether the performance difference is caused by the data or the use of HN. Nevertheless, future work can easily apply our HN-based architecture to other VLAs by adopting their original training recipe for a fair comparison.

Hyperparameters of HyperVLA In the base network, we use DINOv2 (Oquab et al., 2023) as the image encoder. The policy head is a Transformer with 4 layers, each with 4 attention heads. Its token embedding dimension and hidden layer dimension are set to 64 and 128 respectively. The base network takes only the current image observation as input, and predicts an action chunk of 4 steps. During evaluation, we further apply action ensemble (Zhao et al., 2023), i.e., averaging the last 4 steps' action predictions on the current step, to improve prediction accuracy. For the HN, we adopt T5 (Raffel et al., 2020) as the instruction encoder and DINOv2 as the image encoder, and freeze them during training. We learn a Transformer context encoder with 6 layers, with an embedding dimension of 128, MLP hidden dimension of 512 and 4 attention heads for each layer. The output heads are linear layers that map the context embedding to the base parameters.

Training setup For a fair comparison with the baselines, we also train HyperVLA on the OXE dataset. We train it for 100k steps with a batch size 256. See Appendix B.1 for the detailed training setup. We include the source code in the supplementary material to facilitate reproducibility.

4.2 Zero-shot generalization results

Robot	**Google Robot**				**WidowX**				
Task set	pick	move	close drawer	Avg.	spoon on towel	carrot on plate	stack cube	eggplant in basket	Avg.
ID or OOD	both	ID	ID		ID	ID	OOD	OOD	
RT-1-X	29	46	**65**	47	10	12	0	0	6
Octo	7	26	31	21	5	1	0	45	13
OpenVLA	10	**72**	54	45	25	**18**	**34**	**65**	**36**
HyperVLA	$\mathbf{58 \pm 3}$	$\mathbf{73 \pm 1}$	$\mathbf{58 \pm 7}$	$\mathbf{63 \pm 3}$	$\mathbf{48 \pm 3}$	$\mathbf{21 \pm 5}$	$\mathbf{39 \pm 8}$	$\mathbf{52 \pm 13}$	$\mathbf{40 \pm 5}$

Table 1: Evaluation success rates of different methods on SIMPLER. For Google Robot, each column represents a task set which contains multiple different instructions, and we report the average success rate over the whole task set. The "ID or OOD" row represents if the task set is in-distribution (ID) or out-of-distribution (OOD). For our method, we report the performance mean and standard error averaged over 5 random seeds. For the baselines, we cannot report the confidence interval, as only a single model checkpoint is publicly available for each baseline method.

To answer Q1 about zero-shot generalization performance, We evaluate on the SIMPLER benchmark (Li et al., 2024b), which reproduces some tasks from the OXE dataset in simulation and is specifically designed to align with real-world evaluation results, so that different VLAs can be compared in a reproducible way. SIMPLER includes two commonly used robot arms, Google Robot and WidowX, and defines a set of different tasks for each robot. SIMPLER evaluates on both tasks that have been seen during training with different demonstration number ranging from 1 to more than 2,000, and unseen tasks with new instructions (see Appendix B.2 for more details). For seen tasks, generalization across parametric variations is evaluated, such as object layout, position and orientation. For unseen tasks, generalization across instructions is further evaluated.

Table 1 compares the success rates of different methods on the SIMPLER benchmark. Among the baselines, OpenVLA performs the best overall, as it builds upon strong language and vision foundation models which facilitate generalization, but at the expense of a higher inference cost. Our method achieves similar performance to OpenVLA on most task sets, while significantly outperforming all baselines on the picking task set. This validates that HyperVLA can significantly reduce training and inference cost (Section 4.4) without sacrificing performance.

4.3 Few-shot adaptation results

To answer Q2 about few-shot adaptation performance, we evaluate on the LIBERO benchmark (Liu et al., 2023), which is commonly used to evaluate data-efficient fine-tuning of VLAs. Following

the same setup as in OpenVLA, we evaluate on four task suites in LIBERO, i.e., LIBERO-Spatial, LIBERO-Object, LIBERO-Goal and LIBERO-Long, each containing 10 tasks (instructions) with 50 demonstrations for each task (see Appendix B.2 for more details). For a fair comparison, we preprocess the demonstrations in the same way as OpenVLA. We fine-tune HyperVLA on the first three task suites for 10k steps, and LIBERO-Long for 60k steps as it constitutes of long-horizon tasks that are harder to solve. All the other hyperparameters are set in the same way as for pretraining.

As shown in Table 2, our method significantly outperforms Octo and OpenVLA on all the task suites after fine-tuning, validating the effectiveness of our method for few-shot adaptation to unseen tasks. The significant advantage of HyperVLA on LIBERO-Long further validates that it can also solve complicated long-horizon tasks by only activating a compact base policy at inference time.

	LIBERO-Spatial	LIBERO-Object	LIBERO-Goal	LIBERO-Long	Average
Octo	79	86	85	51	75
OpenVLA	85	88	79	54	77
HyperVLA	**95**	**94**	**92**	**74**	**89**

Table 2: Evaluation success rate of different methods on LIBERO after fine-tuning. We evaluate on each task for 50 episodes. The results of the baselines are taken from Kim et al. (2024).

4.4 Inference efficiency

To answer Q3 about inference efficiency, Table 3 compares the number of parameters activated during training and inference, the inference speed, and FLOPs of different methods. For the activated parameters at inference time, we exclude the instruction encoder in all the methods and the HN in our method, as they are only activated once at the beginning of each episode, and their computational costs are negligible compared to the total inference cost across the whole episode.

Method	**# params activated for training**	**# params activated for test**	**Time per inference step (ms)**	**FLOPs**
RT-1-X	35M	35M	88	-
Octo	200M	100M	96	5.6×10^{10}
OpenVLA	7.6B	7.6B	482	4.0×10^{12}
HyperVLA	86M (shared) + 216M (HN)	86M (shared) + 0.1M (generated)	**4**	$\mathbf{4.7 \times 10^{10}}$

Table 3: Number of parameters activated during training and test, inference speed, and FLOPs of different methods. Time per inference step is measured by running each model on an NVIDIA L4 GPU. We were unable to measure the FLOPs of RT-1-X as its model checkpoint is wrapped up.

While HyperVLA learns both a shared DINOv2 image encoder with 86M parameters and an HN with 216M parameters (100M for the frozen T5 encoder + 86M for the frozen DINOv2 encoder + 30M for the learned context encoder) during training, at test time it only activates the shared DINOv2 backbone and a compact base network with 0.1M parameters for each inference step, which leads to a significant acceleration. Compared to OpenVLA, the baseline with the best performance, HyperVLA reduces the model size at inference time 90-fold and accelerates the inference speed 120-fold. Although RT-1-X and Octo have a similar or smaller number of activated parameters during inference than HyperVLA, their inference is still much slower, as they use either autoregression or a diffusion policy to predict the action, both of which require more iterations over the model parameters than the simple linear action head used in HyperVLA.

Based on the above results, we conclude that HyperVLA not only achieves similar or better performance compared to the baselines, but also significantly reduces inference costs. Moreover, while our primary goal is to improve the inference efficiency of VLAs, our method also significantly reduces computational costs during training. Specifically, OpenVLA is trained on 64 A100 GPUs for 14 days (Kim et al., 2024), while HyperVLA can be trained on just 4 A5000 GPUs in a single day.

4.5 Ablation studies

To answer Q4 about the effectiveness of the algorithm designs in HyperVLA, we run ablation studies by removing each of the algorithm design features proposed in Section 3.3 from it. The ablation results in Table 4 validate that all the proposed designs contribute to the success of HyperVLA.

Vision backbone: When removing the DINOv2 backbone, we increase the number of training steps to 600k for a fair comparison, as training the whole model from scratch takes longer to converge. However, even with a larger training budget, it still significantly underperforms HyperVLA, illustrating the importance of utilizing the prior knowledge from vision foundation models.

HN normalization: To ablate HN normalization, we do not normalize the context embedding before feeding it into the HN output heads. In general, this variant performs slightly worse than HyperVLA on seen tasks, but significantly worse on the two OOD WidowX tasks, which validates the importance of stabilizing HN learning with context embedding normalization.

Action generation strategy: We replace the linear action head in HyperVLA with a diffusion action head like in Octo (Team et al., 2024), and increase the training steps to 400k. The diffusion-head variant underperforms HyperVLA, which illustrates that a simple linear action head trained with MSE loss is sufficient when training an HN-based VLA, and also improves training efficiency compared to more complicated action head designs like diffusion.

Please see Appendix C.1 for ablation results on more detailed design choices in HyperVLA.

Method	Google robot				WidowX				
	pick	move	close drawer	Avg	spoon on towel	carrot on plate	stack cube	eggplant in basket	Avg
HyperVLA (Full)	$\mathbf{58 \pm 3}$	$\mathbf{73 \pm 1}$	$\mathbf{58 \pm 7}$	$\mathbf{63 \pm 3}$	$\mathbf{48 \pm 3}$	$\mathbf{21 \pm 5}$	$\mathbf{39 \pm 8}$	$\mathbf{52 \pm 13}$	$\mathbf{40 \pm 5}$
- Vision backbone	24	30	38	31	21	0	0	0	5
- HN normalization	$\mathbf{53 \pm 4}$	$\mathbf{71 \pm 4}$	48 ± 1	57 ± 1	$\mathbf{48 \pm 3}$	$\mathbf{27 \pm 6}$	19 ± 3	31 ± 9	31 ± 4
- Linear action head	49	56	55	53	42	23	15	39	30

Table 4: Ablations on how different algorithm designs in HyperVLA influence its performance.

5 Related work

Using language and vision foundation models as backbone enables VLAs to generalize across a broad range of tasks at the expense of high inference cost. Using smaller backbone models is thus a straightforward way to accelerate VLAs (Belkhale & Sadigh, 2024; Wen et al., 2025; Shukor et al., 2025). Predicting action chunks (Zhao et al., 2023; Team et al., 2024; Black et al., 2024) instead of a single-step action is another common approach, so the VLA does not need to be called at every timestep. The idea of learning a large VLA but only partially activating it during inference has also been explored: DeeR-VLA (Yue et al., 2024) early exits from intermediate layers if the layer output is sufficient for action prediction. Closely related to the hierarchical architecture in our method, dual-system VLAs (Shentu et al., 2024; Han et al., 2024; Zhang et al., 2024; Bu et al., 2024; Cui et al., 2025) learn both a high-level planner that generates a latent goal and operates at a low frequency, and a low-level policy that conditions on this latent goal to generate per-step actions. Compared to existing methods, HyperVLA accelerates VLA inference in an orthogonal way by decoupling the skills required to solve different tasks via HNs and can be combined with existing approaches for further acceleration, such as parameterizing the high-level and low-level models in dual-system VLAs as HNs. Due to space limitation, see Appendix A for more related work on general VLAs and context-conditioned policy generation beyond the domain of VLAs.

6 Conclusion

In this paper, we analyzed why existing VLAs have high inference cost due to their monolithic architectures, and proposed an HN-based solution that decouples the skills required to solve different tasks at test time for inference acceleration. To stabilize HN training and improve its performance, we further proposed several key algorithm design features, including how to properly integrate vision backbones, HN normalization, and a simple linear action head trained with MSE loss. Building upon HN's ability to decouple the skills to solve different tasks and this algorithm design, we proposed HyperVLA, which achieves performance similar to or even better than that of existing monolithic VLAs, while significantly improving inference efficiency.

Our paper opens many interesting directions for future work on HN-based VLAs, such as evaluating on real robots, scaling up the HN model size, and training on more recent and larger robotic datasets (Khazatsky et al., 2024; Bjorck et al., 2025) for further performance improvement, and integration with task planning to solve more complicated long-horizon tasks.

REPRODUCIBILITY STATEMENT

We have made the following efforts to ensure the reproducibility of our work:

1. We clearly describe our method in Section 3, including both the detailed architecture of HyperVLA in Section 3.2, and the key algorithm designs in Section 3.3.
2. We include the source code to reproduce both HyperVLA and the baseline results in the supplementary material.
3. We use publicly available datasets and benchmarks (OXE, SIMPLER, and LIBERO) for experiments, and follow the same data preprocessing pipeline as in previous work (Team et al., 2024; Kim et al., 2024).
4. We clearly describe the hyperparameters of experiments in Section 4.1 and Appendix B.1 to facilitate reproducibility.

REFERENCES

Sayantan Auddy, Sebastian Bergner, and Justus Piater. Effect of optimizer, initializer, and architecture of hypernetworks on continual learning from demonstration. In *European Robotics Forum*, pp. 315–320. Springer, 2024.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. Qwen technical report, 2023a. URL https://arxiv.org/abs/2309.16609.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond, 2023b. URL https://arxiv.org/abs/2308.12966.

Jacob Beck, Matthew Thomas Jackson, Risto Vuorio, and Shimon Whiteson. Hypernetworks in meta-reinforcement learning. In *6th Annual Conference on Robot Learning*, 2022.

Jacob Beck, Matthew Thomas Jackson, Risto Vuorio, and Shimon Whiteson. Hypernetworks in meta-reinforcement learning. In *Conference on Robot Learning*, pp. 1478–1487. PMLR, 2023a.

Jacob Beck, Risto Vuorio, Zheng Xiong, and Shimon Whiteson. Recurrent hypernetworks are surprisingly strong in meta-rl. *Advances in Neural Information Processing Systems*, 36:62121–62138, 2023b.

Suneel Belkhale and Dorsa Sadigh. Minivla: A better vla with a smaller footprint. *URL https://github. com/Stanford-ILIAD/openvla-mini*, 2024.

Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. pi0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

Qingwen Bu, Hongyang Li, Li Chen, Jisong Cai, Jia Zeng, Heming Cui, Maoqing Yao, and Yu Qiao. Towards synergistic, generalized, and efficient dual-system for robotic manipulation. *arXiv preprint arXiv:2410.08001*, 2024.

Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, Xindong He, Xu Huang, Shu Jiang, et al. Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. *arXiv preprint arXiv:2503.06669*, 2025.

Oscar Chang, Lampros Flokas, and Hod Lipson. Principled weight initialization for hypernetworks. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=H1lma24tPB.

Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, Lixin Gu, Xuehui Wang, Qingyun Li, Yimin Ren, Zixuan Chen, Jiapeng Luo, Jiahao Wang, Tan Jiang, Bo Wang, Conghui He, Botian Shi, Xingcheng Zhang, Han Lv, Yi Wang, Wenqi Shao, Pei Chu, Zhongying Tu, Tong He, Zhiyong Wu, Huipeng Deng, Jiaye Ge, Kai Chen, Kaipeng Zhang, Limin Wang, Min Dou, Lewei Lu, Xizhou Zhu, Tong Lu, Dahua Lin, Yu Qiao, Jifeng Dai, and Wenhai Wang. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling, 2025. URL https://arxiv.org/abs/2412.05271.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, pp. 02783649241273668, 2023.

Can Cui, Pengxiang Ding, Wenxuan Song, Shuanghao Bai, Xinyang Tong, Zirui Ge, Runze Suo, Wanqi Zhou, Yang Liu, Bofang Jia, et al. Openhelix: A short survey, empirical analysis, and open-source dual-system vla model for robotic manipulation. *arXiv preprint arXiv:2505.03912*, 2025.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

Jacopo Di Ventura, Dylan R Ashley, Vincent Herrmann, Francesco Faccio, and Jürgen Schmidhuber. Upside down reinforcement learning with policy generators. *arXiv preprint arXiv:2501.16288*, 2025.

Muhayy Ud Din, Waseem Akram, Lyes Saad Saoud, Jan Rosell, and Irfan Hussain. Vision language action models in robotic manipulation: A systematic review. *arXiv preprint arXiv:2507.10672*, 2025.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021. URL https://arxiv.org/abs/2010.11929.

Francesco Faccio, Vincent Herrmann, Aditya Ramesh, Louis Kirsch, and Jürgen Schmidhuber. Goal-conditioned generators of deep policies. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 7503–7511, 2023.

Roya Firoozi, Johnathan Tucker, Stephen Tian, Anirudha Majumdar, Jiankai Sun, Weiyu Liu, Yuke Zhu, Shuran Song, Ashish Kapoor, Karol Hausman, et al. Foundation models in robotics: Applications, challenges, and the future. *The International Journal of Robotics Research*, pp. 02783649241281508, 2023.

Team GLM, :, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Dan Zhang, Diego Rojas, Guanyu Feng, Hanlin Zhao, Hanyu Lai, Hao Yu, Hongning Wang, Jiadai Sun, Jiajie Zhang, Jiale Cheng, Jiayi Gui, Jie Tang, Jing Zhang, Jingyu Sun, Juanzi Li, Lei Zhao, Lindong Wu, Lucen Zhong, Mingdao Liu, Minlie Huang, Peng Zhang, Qinkai Zheng, Rui Lu, Shuaiqi Duan, Shudan Zhang, Shulin Cao, Shuxun Yang, Weng Lam Tam, Wenyi Zhao, Xiao Liu, Xiao Xia, Xiaohan Zhang, Xiaotao Gu, Xin Lv, Xinghan Liu, Xinyi Liu, Xinyue Yang, Xixuan Song, Xunkai Zhang, Yifan An, Yifan Xu, Yilin Niu, Yuantao Yang, Yueyan Li, Yushi Bai, Yuxiao Dong, Zehan Qi, Zhaoyu Wang, Zhen Yang, Zhengxiao Du, Zhenyu Hou, and Zihan Wang. Chatglm: A family of large language models from glm-130b to glm-4 all tools, 2024. URL https://arxiv.org/abs/2406.12793.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew

Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha,

Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

David Ha, Andrew Dai, and Quoc V Le. Hypernetworks. *arXiv preprint arXiv:1609.09106*, 2016.

ByungOk Han, Jaehong Kim, and Jinhyeok Jang. A dual process vla: Efficient robotic manipulation leveraging vlm. *arXiv preprint arXiv:2410.15549*, 2024.

Yafei Hu, Quanting Xie, Vidhi Jain, Jonathan Francis, Jay Patrikar, Nikhil Keetha, Seungchan Kim, Yaqi Xie, Tianyi Zhang, Hao-Shu Fang, et al. Toward general-purpose robots via foundation models: A survey and meta-analysis. *arXiv preprint arXiv:2312.08782*, 2023.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023. URL https://arxiv.org/abs/2310.06825.

Kento Kawaharazuka, Jihoon Oh, Jun Yamada, Ingmar Posner, and Yuke Zhu. Vision-language-action models for robotics: A review towards real-world applications. https://vla-survey.github.io, 2025.

Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. *arXiv preprint arXiv:2502.19645*, 2025.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything, 2023. URL https://arxiv.org/abs/2304.02643.

Xinghang Li, Peiyan Li, Minghuan Liu, Dong Wang, Jirong Liu, Bingyi Kang, Xiao Ma, Tao Kong, Hanbo Zhang, and Huaping Liu. Towards generalist robot policies: What matters in building vision-language-action models. *arXiv preprint arXiv:2412.14058*, 2024a.

Xuanlin Li, Kyle Hsu, Jiayuan Gu, Karl Pertsch, Oier Mees, Homer Rich Walke, Chuyuan Fu, Ishikaa Lunawat, Isabel Sieh, Sean Kirmani, et al. Evaluating real-world robot manipulation policies in simulation. *arXiv preprint arXiv:2405.05941*, 2024b.

Yongyuan Liang, Tingqiang Xu, Kaizhe Hu, Guangqi Jiang, Furong Huang, and Huazhe Xu. Make-an-agent: A generalizable policy network generator with behavior-prompted diffusion. *arXiv preprint arXiv:2407.10973*, 2024.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Yueen Ma, Zixing Song, Yuzheng Zhuang, Jianye Hao, and Irwin King. A survey on vision-language-action models for embodied ai. *arXiv preprint arXiv:2405.14093*, 2024.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report, 2024. URL https://arxiv.org/abs/2303.08774.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nico-

las Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2024. URL https://arxiv.org/abs/2304.07193.

Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6892–6903. IEEE, 2024.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021a. URL https://arxiv.org/abs/2103.00020.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021b.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

Sahand Rezaei-Shoshtari, Charlotte Morissette, Francois R Hogan, Gregory Dudek, and David Meger. Hypernetworks for zero-shot transfer in reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 9579–9587, 2023.

Elad Sarafian, Shai Keynan, and Sarit Kraus. Recomposing the reinforcement learning building blocks with hypernetworks. In *International Conference on Machine Learning*, pp. 9301–9312. Pmlr, 2021.

Yide Shentu, Philipp Wu, Aravind Rajeswaran, and Pieter Abbeel. From llms to actions: latent codes as bridges in hierarchical robot control. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 8539–8546. IEEE, 2024.

Mustafa Shukor, Dana Aubakirova, Francesco Capuano, Pepijn Kooijmans, Steven Palma, Adil Zouitine, Michel Aractingi, Caroline Pascal, Martino Russi, Andres Marafioti, et al. Smolvla: A vision-language-action model for affordable and efficient robotics. *arXiv preprint arXiv:2506.01844*, 2025.

Wenxuan Song, Jiayi Chen, Pengxiang Ding, Han Zhao, Wei Zhao, Zhide Zhong, Zongyuan Ge, Jun Ma, and Haoang Li. Accelerating vision-language-action model integrated with action chunking via parallel decoding. *arXiv preprint arXiv:2503.02310*, 2025.

Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pp. 6105–6114. PMLR, 2019.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy Lillicrap, Angeliki Lazaridou, Orhan Firat, James Molloy, Michael Isard, Paul R. Barham, Tom Hennigan, Benjamin Lee, Fabio Viola, Malcolm Reynolds, Yuanzhong Xu, Ryan Doherty, Eli Collins, Clemens Meyer, Eliza Rutherford, Erica Moreira, Kareem Ayoub, Megha Goel, Jack Krawczyk, Cosmo Du, Ed Chi, Heng-Tze Cheng, Eric Ni, Purvi Shah, Patrick Kane, Betty Chan, Manaal Faruqui, Aliaksei Severyn, Hanzhao Lin, YaGuang Li, Yong Cheng, Abe Ittycheriah, Mahdis Mahdieh, Mia Chen, Pei Sun, Dustin Tran, Sumit Bagri, Balaji Lakshminarayanan, Jeremiah Liu, Andras Orban, Fabian Güra, Hao Zhou, Xinying Song, Aurelien Boffy, Harish Ganapathy, Steven Zheng, HyunJeong Choe, Ágoston Weisz, Tao Zhu, Yifeng Lu, Siddharth Gopal,

Jarrod Kahn, Maciej Kula, Jeff Pitman, Rushin Shah, Emanuel Taropa, Majd Al Merey, Martin Baeuml, Zhifeng Chen, Laurent El Shafey, Yujing Zhang, Olcan Sercinoglu, George Tucker, Enrique Piqueras, Maxim Krikun, Iain Barr, Nikolay Savinov, Ivo Danihelka, Becca Roelofs, Anaïs White, Anders Andreassen, Tamara von Glehn, Lakshman Yagati, Mehran Kazemi, Lucas Gonzalez, Misha Khalman, Jakub Sygnowski, Alexandre Frechette, Charlotte Smith, Laura Culp, Lev Proleev, Yi Luan, Xi Chen, James Lottes, Nathan Schucher, Federico Lebron, Alban Rrustemi, Natalie Clay, Phil Crone, Tomas Kocisky, Jeffrey Zhao, Bartek Perz, Dian Yu, Heidi Howard, Adam Bloniarz, Jack W. Rae, Han Lu, Laurent Sifre, Marcello Maggioni, Fred Alcober, Dan Garrette, Megan Barnes, Shantanu Thakoor, Jacob Austin, Gabriel Barth-Maron, William Wong, Rishabh Joshi, Rahma Chaabouni, Deeni Fatiha, Arun Ahuja, Gaurav Singh Tomar, Evan Senter, Martin Chadwick, Ilya Kornakov, Nithya Attaluri, Iñaki Iturrate, Ruibo Liu, Yunxuan Li, Sarah Cogan, Jeremy Chen, Chao Jia, Chenjie Gu, Qiao Zhang, Jordan Grimstad, Ale Jakse Hartman, Xavier Garcia, Thanumalayan Sankaranarayana Pillai, Jacob Devlin, Michael Laskin, Diego de Las Casas, Dasha Valter, Connie Tao, Lorenzo Blanco, Adrià Puigdomènech Badia, David Reitter, Mianna Chen, Jenny Brennan, Clara Rivera, Sergey Brin, Shariq Iqbal, Gabriela Surita, Jane Labanowski, Abhi Rao, Stephanie Winkler, Emilio Parisotto, Yiming Gu, Kate Olszewska, Ravi Addanki, Antoine Miech, Annie Louis, Denis Teplyashin, Geoff Brown, Elliot Catt, Jan Balaguer, Jackie Xiang, Pidong Wang, Zoe Ashwood, Anton Briukhov, Albert Webson, Sanjay Ganapathy, Smit Sanghavi, Ajay Kannan, Ming-Wei Chang, Axel Stjerngren, Josip Djolonga, Yuting Sun, Ankur Bapna, Matthew Aitchison, Pedram Pejman, Henryk Michalewski, Tianhe Yu, Cindy Wang, Juliette Love, Junwhan Ahn, Dawn Bloxwich, Kehang Han, Peter Humphreys, Thibault Sellam, James Bradbury, Varun Godbole, Sina Samangooei, Bogdan Damoc, Alex Kaskasoli, Sébastien M. R. Arnold, Vijay Vasudevan, Shubham Agrawal, Jason Riesa, Dmitry Lepikhin, Richard Tanburn, Srivatsan Srinivasan, Hyeontaek Lim, Sarah Hodkinson, Pranav Shyam, Johan Ferret, Steven Hand, Ankush Garg, Tom Le Paine, Jian Li, Yujia Li, Minh Giang, Alexander Neitz, Zaheer Abbas, Sarah York, Machel Reid, Elizabeth Cole, Aakanksha Chowdhery, Dipanjan Das, Dominika Rogozińska, Vitaliy Nikolaev, Pablo Sprechmann, Zachary Nado, Lukas Zilka, Flavien Prost, Luheng He, Marianne Monteiro, Gaurav Mishra, Chris Welty, Josh Newlan, Dawei Jia, Miltiadis Allamanis, Clara Huiyi Hu, Raoul de Liedekerke, Justin Gilmer, Carl Saroufim, Shruti Rijhwani, Shaobo Hou, Disha Shrivastava, Anirudh Baddepudi, Alex Goldin, Adnan Ozturel, Albin Cassirer, Yunhan Xu, Daniel Sohn, Devendra Sachan, Reinald Kim Amplayo, Craig Swanson, Dessie Petrova, Shashi Narayan, Arthur Guez, Siddhartha Brahma, Jessica Landon, Miteyan Patel, Ruizhe Zhao, Kevin Villela, Luyu Wang, Wenhao Jia, Matthew Rahtz, Mai Giménez, Legg Yeung, James Keeling, Petko Georgiev, Diana Mincu, Boxi Wu, Salem Haykal, Rachel Saputro, Kiran Vodrahalli, James Qin, Zeynep Cankara, Abhanshu Sharma, Nick Fernando, Will Hawkins, Behnam Neyshabur, Solomon Kim, Adrian Hutter, Priyanka Agrawal, Alex Castro-Ros, George van den Driessche, Tao Wang, Fan Yang, Shuo yiin Chang, Paul Komarek, Ross McIlroy, Mario Lučić, Guodong Zhang, Wael Farhan, Michael Sharman, Paul Natsev, Paul Michel, Yamini Bansal, Siyuan Qiao, Kris Cao, Siamak Shakeri, Christina Butterfield, Justin Chung, Paul Kishan Rubenstein, Shivani Agrawal, Arthur Mensch, Kedar Soparkar, Karel Lenc, Timothy Chung, Aedan Pope, Loren Maggiore, Jackie Kay, Priya Jhakra, Shibo Wang, Joshua Maynez, Mary Phuong, Taylor Tobin, Andrea Tacchetti, Maja Trebacz, Kevin Robinson, Yash Katariya, Sebastian Riedel, Paige Bailey, Kefan Xiao, Nimesh Ghelani, Lora Aroyo, Ambrose Slone, Neil Houlsby, Xuehan Xiong, Zhen Yang, Elena Gribovskaya, Jonas Adler, Mateo Wirth, Lisa Lee, Music Li, Thais Kagohara, Jay Pavagadhi, Sophie Bridgers, Anna Bortsova, Sanjay Ghemawat, Zafarali Ahmed, Tianqi Liu, Richard Powell, Vijay Bolina, Mariko Iinuma, Polina Zablotskaia, James Besley, Da-Woon Chung, Timothy Dozat, Ramona Comanescu, Xiance Si, Jeremy Greer, Guolong Su, Martin Polacek, Raphaël Lopez Kaufman, Simon Tokumine, Hexiang Hu, Elena Buchatskaya, Yingjie Miao, Mohamed Elhawaty, Aditya Siddhant, Nenad Tomasev, Jinwei Xing, Christina Greer, Helen Miller, Shereen Ashraf, Aurko Roy, Zizhao Zhang, Ada Ma, Angelos Filos, Milos Besta, Rory Blevins, Ted Klimenko, Chih-Kuan Yeh, Soravit Changpinyo, Jiaqi Mu, Oscar Chang, Mantas Pajarskas, Carrie Muir, Vered Cohen, Charline Le Lan, Krishna Haridasan, Amit Marathe, Steven Hansen, Sholto Douglas, Rajkumar Samuel, Mingqiu Wang, Sophia Austin, Chang Lan, Jiepu Jiang, Justin Chiu, Jaime Alonso Lorenzo, Lars Lowe Sjösund, Sébastien Cevey, Zach Gleicher, Thi Avrahami, Anudhyan Boral, Hansa Srinivasan, Vittorio Selo, Rhys May, Konstantinos Aisopos, Léonard Hussenot, Livio Baldini Soares, Kate Baumli, Michael B. Chang, Adrià Recasens, Ben Caine, Alexander Pritzel, Filip Pavetic, Fabio Pardo, Anita Gergely, Justin Frye, Vinay Ramasesh, Dan Horgan, Kartikeya Badola, Nora Kassner, Subhrajit Roy, Ethan Dyer, Víctor Campos Campos, Alex Tomala, Yunhao Tang, Dalia El

Badawy, Elspeth White, Basil Mustafa, Oran Lang, Abhishek Jindal, Sharad Vikram, Zhitao Gong, Sergi Caelles, Ross Hemsley, Gregory Thornton, Fangxiaoyu Feng, Wojciech Stokowiec, Ce Zheng, Phoebe Thacker, Çağlar Ünlü, Zhishuai Zhang, Mohammad Saleh, James Svensson, Max Bileschi, Piyush Patil, Ankesh Anand, Roman Ring, Katerina Tsihlas, Arpi Vezer, Marco Selvi, Toby Shevlane, Mikel Rodriguez, Tom Kwiatkowski, Samira Daruki, Keran Rong, Allan Dafoe, Nicholas FitzGerald, Keren Gu-Lemberg, Mina Khan, Lisa Anne Hendricks, Marie Pellat, Vladimir Feinberg, James Cobon-Kerr, Tara Sainath, Maribeth Rauh, Sayed Hadi Hashemi, Richard Ives, Yana Hasson, Eric Noland, Yuan Cao, Nathan Byrd, Le Hou, Qingze Wang, Thibault Sottiaux, Michela Paganini, Jean-Baptiste Lespiau, Alexandre Moufarek, Samer Hassan, Kaushik Shivakumar, Joost van Amersfoort, Amol Mandhane, Pratik Joshi, Anirudh Goyal, Matthew Tung, Andrew Brock, Hannah Sheahan, Vedant Misra, Cheng Li, Nemanja Rakićević, Mostafa Dehghani, Fangyu Liu, Sid Mittal, Junhyuk Oh, Seb Noury, Eren Sezener, Fantine Huot, Matthew Lamm, Nicola De Cao, Charlie Chen, Sidharth Mudgal, Romina Stella, Kevin Brooks, Gautam Vasudevan, Chenxi Liu, Mainak Chain, Nivedita Melinkeri, Aaron Cohen, Venus Wang, Kristie Seymore, Sergey Zubkov, Rahul Goel, Summer Yue, Sai Krishnakumaran, Brian Albert, Nate Hurley, Motoki Sano, Anhad Mohananey, Jonah Joughin, Egor Filonov, Tomasz Kępa, Yomna Eldawy, Jiawern Lim, Rahul Rishi, Shirin Badiezadegan, Taylor Bos, Jerry Chang, Sanil Jain, Sri Gayatri Sundara Padmanabhan, Subha Puttagunta, Kalpesh Krishna, Leslie Baker, Norbert Kalb, Vamsi Bedapudi, Adam Kurzrok, Shuntong Lei, Anthony Yu, Oren Litvin, Xiang Zhou, Zhichun Wu, Sam Sobell, Andrea Siciliano, Alan Papir, Robby Neale, Jonas Bragagnolo, Tej Toor, Tina Chen, Valentin Anklin, Feiran Wang, Richie Feng, Milad Gholami, Kevin Ling, Lijuan Liu, Jules Walter, Hamid Moghaddam, Arun Kishore, Jakub Adamek, Tyler Mercado, Jonathan Mallinson, Siddhinita Wandekar, Stephen Cagle, Eran Ofek, Guillermo Garrido, Clemens Lombriser, Maksim Mukha, Botu Sun, Hafeezul Rahman Mohammad, Josip Matak, Yadi Qian, Vikas Peswani, Pawel Janus, Quan Yuan, Leif Schelin, Oana David, Ankur Garg, Yifan He, Oleksii Duzhyi, Anton Älgmyr, Timothée Lottaz, Qi Li, Vikas Yadav, Luyao Xu, Alex Chinien, Rakesh Shivanna, Aleksandr Chuklin, Josie Li, Carrie Spadine, Travis Wolfe, Kareem Mohamed, Subhabrata Das, Zihang Dai, Kyle He, Daniel von Dincklage, Shyam Upadhyay, Akanksha Maurya, Luyan Chi, Sebastian Krause, Khalid Salama, Pam G Rabinovitch, Pavan Kumar Reddy M, Aarush Selvan, Mikhail Dektiarev, Golnaz Ghiasi, Erdem Guven, Himanshu Gupta, Boyi Liu, Deepak Sharma, Idan Heimlich Shtacher, Shachi Paul, Oscar Akerlund, François-Xavier Aubet, Terry Huang, Chen Zhu, Eric Zhu, Elico Teixeira, Matthew Fritze, Francesco Bertolini, Liana-Eleonora Marinescu, Martin Bölle, Dominik Paulus, Khyatti Gupta, Tejasi Latkar, Max Chang, Jason Sanders, Roopa Wilson, Xuewei Wu, Yi-Xuan Tan, Lam Nguyen Thiet, Tulsee Doshi, Sid Lall, Swaroop Mishra, Wanming Chen, Thang Luong, Seth Benjamin, Jasmine Lee, Ewa Andrejczuk, Dominik Rabiej, Vipul Ranjan, Krzysztof Styrc, Pengcheng Yin, Jon Simon, Malcolm Rose Harriott, Mudit Bansal, Alexei Robsky, Geoff Bacon, David Greene, Daniil Mirylenka, Chen Zhou, Obaid Sarvana, Abhimanyu Goyal, Samuel Andermatt, Patrick Siegler, Ben Horn, Assaf Israel, Francesco Pongetti, Chih-Wei "Louis" Chen, Marco Selvatici, Pedro Silva, Kathie Wang, Jackson Tolins, Kelvin Guu, Roey Yogev, Xiaochen Cai, Alessandro Agostini, Maulik Shah, Hung Nguyen, Noah Ó Donnaile, Sébastien Pereira, Linda Friso, Adam Stambler, Adam Kurzrok, Chenkai Kuang, Yan Romanikhin, Mark Geller, ZJ Yan, Kane Jang, Cheng-Chun Lee, Wojciech Fica, Eric Malmi, Qijun Tan, Dan Banica, Daniel Balle, Ryan Pham, Yanping Huang, Diana Avram, Hongzhi Shi, Jasjot Singh, Chris Hidey, Niharika Ahuja, Pranab Saxena, Dan Dooley, Srividya Pranavi Potharaju, Eileen O'Neill, Anand Gokulchandran, Ryan Foley, Kai Zhao, Mike Dusenberry, Yuan Liu, Pulkit Mehta, Ragha Kotikalapudi, Chalence Safranek-Shrader, Andrew Goodman, Joshua Kessinger, Eran Globen, Prateek Kolhar, Chris Gorgolewski, Ali Ibrahim, Yang Song, Ali Eichenbaum, Thomas Brovelli, Sahitya Potluri, Preethi Lahoti, Cip Baetu, Ali Ghorbani, Charles Chen, Andy Crawford, Shalini Pal, Mukund Sridhar, Petru Gurita, Asier Mujika, Igor Petrovski, Pierre-Louis Cedoz, Chenmei Li, Shiyuan Chen, Niccolò Dal Santo, Siddharth Goyal, Jitesh Punjabi, Karthik Kappaganthu, Chester Kwak, Pallavi LV, Sarmishta Velury, Himadri Choudhury, Jamie Hall, Premal Shah, Ricardo Figueira, Matt Thomas, Minjie Lu, Ting Zhou, Chintu Kumar, Thomas Jurdi, Sharat Chikkerur, Yenai Ma, Adams Yu, Soo Kwak, Victor Ähdel, Sujeevan Rajayogam, Travis Choma, Fei Liu, Aditya Barua, Colin Ji, Ji Ho Park, Vincent Hellendoorn, Alex Bailey, Taylan Bilal, Huanjie Zhou, Mehrdad Khatir, Charles Sutton, Wojciech Rzadkowski, Fiona Macintosh, Roopali Vij, Konstantin Shagin, Paul Medina, Chen Liang, Jinjing Zhou, Pararth Shah, Yingying Bi, Attila Dankovics, Shipra Banga, Sabine Lehmann, Marissa Bredesen, Zifan Lin, John Eric Hoffmann, Jonathan Lai, Raynald Chung, Kai Yang, Nihal Balani, Arthur Bražinskas, Andrei Sozanschi, Matthew Hayes, Héctor Fernández

Alcalde, Peter Makarov, Will Chen, Antonio Stella, Liselotte Snijders, Michael Mandl, Ante Kärrman, Paweł Nowak, Xinyi Wu, Alex Dyck, Krishnan Vaidyanathan, Raghavender R, Jessica Mallet, Mitch Rudominer, Eric Johnston, Sushil Mittal, Akhil Udathu, Janara Christensen, Vishal Verma, Zach Irving, Andreas Santucci, Gamaleldin Elsayed, Elnaz Davoodi, Marin Georgiev, Ian Tenney, Nan Hua, Geoffrey Cideron, Edouard Leurent, Mahmoud Alnahlawi, Ionut Georgescu, Nan Wei, Ivy Zheng, Dylan Scandinaro, Heinrich Jiang, Jasper Snoek, Mukund Sundararajan, Xuezhi Wang, Zack Ontiveros, Itay Karo, Jeremy Cole, Vinu Rajashekhar, Lara Tumeh, Eyal Ben-David, Rishub Jain, Jonathan Uesato, Romina Datta, Oskar Bunyan, Shimu Wu, John Zhang, Piotr Stanczyk, Ye Zhang, David Steiner, Subhajit Naskar, Michael Azzam, Matthew Johnson, Adam Paszke, Chung-Cheng Chiu, Jaume Sanchez Elias, Afroz Mohiuddin, Faizan Muhammad, Jin Miao, Andrew Lee, Nino Vieillard, Jane Park, Jiageng Zhang, Jeff Stanway, Drew Garmon, Abhijit Karmarkar, Zhe Dong, Jong Lee, Aviral Kumar, Luowei Zhou, Jonathan Evens, William Isaac, Geoffrey Irving, Edward Loper, Michael Fink, Isha Arkatkar, Nanxin Chen, Izhak Shafran, Ivan Petrychenko, Zhe Chen, Johnson Jia, Anselm Levskaya, Zhenkai Zhu, Peter Grabowski, Yu Mao, Alberto Magni, Kaisheng Yao, Javier Snaider, Norman Casagrande, Evan Palmer, Paul Suganthan, Alfonso Castaño, Irene Giannoumis, Wooyeol Kim, Mikołaj Rybiński, Ashwin Sreevatsa, Jennifer Prendki, David Soergel, Adrian Goedeckemeyer, Willi Gierke, Mohsen Jafari, Meenu Gaba, Jeremy Wiesner, Diana Gage Wright, Yawen Wei, Harsha Vashisht, Yana Kulizhskaya, Jay Hoover, Maigo Le, Lu Li, Chimezie Iwuanyanwu, Lu Liu, Kevin Ramirez, Andrey Khorlin, Albert Cui, Tian LIN, Marcus Wu, Ricardo Aguilar, Keith Pallo, Abhishek Chakladar, Ginger Perng, Elena Allica Abellan, Mingyang Zhang, Ishita Dasgupta, Nate Kushman, Ivo Penchev, Alena Repina, Xihui Wu, Tom van der Weide, Priya Ponnapalli, Caroline Kaplan, Jiri Simsa, Shuangfeng Li, Olivier Dousse, Fan Yang, Jeff Piper, Nathan Ie, Rama Pasumarthi, Nathan Lintz, Anitha Vijayakumar, Daniel Andor, Pedro Valenzuela, Minnie Lui, Cosmin Paduraru, Daiyi Peng, Katherine Lee, Shuyuan Zhang, Somer Greene, Duc Dung Nguyen, Paula Kurylowicz, Cassidy Hardin, Lucas Dixon, Lili Janzer, Kiam Choo, Ziqiang Feng, Biao Zhang, Achintya Singhal, Dayou Du, Dan McKinnon, Natasha Antropova, Tolga Bolukbasi, Orgad Keller, David Reid, Daniel Finchelstein, Maria Abi Raad, Remi Crocker, Peter Hawkins, Robert Dadashi, Colin Gaffney, Ken Franko, Anna Bulanova, Rémi Leblond, Shirley Chung, Harry Askham, Luis C. Cobo, Kelvin Xu, Felix Fischer, Jun Xu, Christina Sorokin, Chris Alberti, Chu-Cheng Lin, Colin Evans, Alek Dimitriev, Hannah Forbes, Dylan Banarse, Zora Tung, Mark Omernick, Colton Bishop, Rachel Sterneck, Rohan Jain, Jiawei Xia, Ehsan Amid, Francesco Piccinno, Xingyu Wang, Praseem Banzal, Daniel J. Mankowitz, Alex Polozov, Victoria Krakovna, Sasha Brown, MohammadHossein Bateni, Dennis Duan, Vlad Firoiu, Meghana Thotakuri, Tom Natan, Matthieu Geist, Ser tan Girgin, Hui Li, Jiayu Ye, Ofir Roval, Reiko Tojo, Michael Kwong, James Lee-Thorp, Christopher Yew, Danila Sinopalnikov, Sabela Ramos, John Mellor, Abhishek Sharma, Kathy Wu, David Miller, Nicolas Sonnerat, Denis Vnukov, Rory Greig, Jennifer Beattie, Emily Caveness, Libin Bai, Julian Eisenschlos, Alex Korchemniy, Tomy Tsai, Mimi Jasarevic, Weize Kong, Phuong Dao, Zeyu Zheng, Frederick Liu, Fan Yang, Rui Zhu, Tian Huey Teh, Jason Sanmiya, Evgeny Gladchenko, Nejc Trdin, Daniel Toyama, Evan Rosen, Sasan Tavakkol, Linting Xue, Chen Elkind, Oliver Woodman, John Carpenter, George Papamakarios, Rupert Kemp, Sushant Kafle, Tanya Grunina, Rishika Sinha, Alice Talbert, Diane Wu, Denese Owusu-Afriyie, Cosmo Du, Chloe Thornton, Jordi Pont-Tuset, Pradyumna Narayana, Jing Li, Saaber Fatehi, John Wieting, Omar Ajmeri, Benigno Uria, Yeongil Ko, Laura Knight, Amélie Héliou, Ning Niu, Shane Gu, Chenxi Pang, Yeqing Li, Nir Levine, Ariel Stolovich, Rebeca Santamaria-Fernandez, Sonam Goenka, Wenny Yustalim, Robin Strudel, Ali Elqursh, Charlie Deck, Hyo Lee, Zonglin Li, Kyle Levin, Raphael Hoffmann, Dan Holtmann-Rice, Olivier Bachem, Sho Arora, Christy Koh, Soheil Hassas Yeganeh, Siim Põder, Mukarram Tariq, Yanhua Sun, Lucian Ionita, Mojtaba Seyedhosseini, Pouya Tafti, Zhiyu Liu, Anmol Gulati, Jasmine Liu, Xinyu Ye, Bart Chrzaszcz, Lily Wang, Nikhil Sethi, Tianrun Li, Ben Brown, Shreya Singh, Wei Fan, Aaron Parisi, Joe Stanton, Vinod Koverkathu, Christopher A. Choquette-Choo, Yunjie Li, TJ Lu, Abe Ittycheriah, Prakash Shroff, Mani Varadarajan, Sanaz Bahargam, Rob Willoughby, David Gaddy, Guillaume Desjardins, Marco Cornero, Brona Robenek, Bhavishya Mittal, Ben Albrecht, Ashish Shenoy, Fedor Moiseev, Henrik Jacobsson, Alireza Ghaffarkhah, Morgane Rivière, Alanna Walton, Clément Crepy, Alicia Parrish, Zongwei Zhou, Clement Farabet, Carey Radebaugh, Praveen Srinivasan, Claudia van der Salm, Andreas Fidjeland, Salvatore Scellato, Eri Latorre-Chimoto, Hanna Klimczak-Plucińska, David Bridson, Dario de Cesare, Tom Hudson, Piermaria Mendolicchio, Lexi Walker, Alex Morris, Matthew Mauger, Alexey Guseynov, Alison Reid, Seth Odoom, Lucia Loher, Victor Cotruta, Madhavi Yenugula, Dominik Grewe, Anastasia Petrushkina, Tom Duerig,

Antonio Sanchez, Steve Yadlowsky, Amy Shen, Amir Globerson, Lynette Webb, Sahil Dua, Dong Li, Surya Bhupatiraju, Dan Hurt, Haroon Qureshi, Ananth Agarwal, Tomer Shani, Matan Eyal, Anuj Khare, Shreyas Rammohan Belle, Lei Wang, Chetan Tekur, Mihir Sanjay Kale, Jinliang Wei, Ruoxin Sang, Brennan Saeta, Tyler Liechty, Yi Sun, Yao Zhao, Stephan Lee, Pandu Nayak, Doug Fritz, Manish Reddy Vuyyuru, John Aslanides, Nidhi Vyas, Martin Wicke, Xiao Ma, Evgenii Eltyshev, Nina Martin, Hardie Cate, James Manyika, Keyvan Amiri, Yelin Kim, Xi Xiong, Kai Kang, Florian Luisier, Nilesh Tripuraneni, David Madras, Mandy Guo, Austin Waters, Oliver Wang, Joshua Ainslie, Jason Baldridge, Han Zhang, Garima Pruthi, Jakob Bauer, Feng Yang, Riham Mansour, Jason Gelman, Yang Xu, George Polovets, Ji Liu, Honglong Cai, Warren Chen, XiangHai Sheng, Emily Xue, Sherjil Ozair, Christof Angermueller, Xiaowei Li, Anoop Sinha, Weiren Wang, Julia Wiesinger, Emmanouil Koukoumidis, Yuan Tian, Anand Iyer, Madhu Gurumurthy, Mark Goldenson, Parashar Shah, MK Blake, Hongkun Yu, Anthony Urbanowicz, Jennimaria Palomaki, Chrisantha Fernando, Ken Durden, Harsh Mehta, Nikola Momchev, Elahe Rahimtoroghi, Maria Georgaki, Amit Raul, Sebastian Ruder, Morgan Redshaw, Jinhyuk Lee, Denny Zhou, Komal Jalan, Dinghua Li, Blake Hechtman, Parker Schuh, Milad Nasr, Kieran Milan, Vladimir Mikulik, Juliana Franco, Tim Green, Nam Nguyen, Joe Kelley, Aroma Mahendru, Andrea Hu, Joshua Howland, Ben Vargas, Jeffrey Hui, Kshitij Bansal, Vikram Rao, Rakesh Ghiya, Emma Wang, Ke Ye, Jean Michel Sarr, Melanie Moranski Preston, Madeleine Elish, Steve Li, Aakash Kaku, Jigar Gupta, Ice Pasupat, Da-Cheng Juan, Milan Someswar, Tejvi M., Xinyun Chen, Aida Amini, Alex Fabrikant, Eric Chu, Xuanyi Dong, Amruta Muthal, Senaka Buthpitiya, Sarthak Jauhari, Nan Hua, Urvashi Khandelwal, Ayal Hitron, Jie Ren, Larissa Rinaldi, Shahar Drath, Avigail Dabush, Nan-Jiang Jiang, Harshal Godhia, Uli Sachs, Anthony Chen, Yicheng Fan, Hagai Taitelbaum, Hila Noga, Zhuyun Dai, James Wang, Chen Liang, Jenny Hamer, Chun-Sung Ferng, Chenel Elkind, Aviel Atias, Paulina Lee, Vít Listík, Mathias Carlen, Jan van de Kerkhof, Marcin Pikus, Krunoslav Zaher, Paul Müller, Sasha Zykova, Richard Stefanec, Vitaly Gatsko, Christoph Hirnschall, Ashwin Sethi, Xingyu Federico Xu, Chetan Ahuja, Beth Tsai, Anca Stefanoiu, Bo Feng, Keshav Dhandhania, Manish Katyal, Akshay Gupta, Atharva Parulekar, Divya Pitta, Jing Zhao, Vivaan Bhatia, Yashodha Bhavnani, Omar Alhadlaq, Xiaolin Li, Peter Danenberg, Dennis Tu, Alex Pine, Vera Filippova, Abhipso Ghosh, Ben Limonchik, Bhargava Urala, Chaitanya Krishna Lanka, Derik Clive, Yi Sun, Edward Li, Hao Wu, Kevin Hongtongsak, Ianna Li, Kalind Thakkar, Kuanysh Omarov, Kushal Majmundar, Michael Alverson, Michael Kucharski, Mohak Patel, Mudit Jain, Maksim Zabelin, Paolo Pelagatti, Rohan Kohli, Saurabh Kumar, Joseph Kim, Swetha Sankar, Vineet Shah, Lakshmi Ramachandruni, Xiangkai Zeng, Ben Bariach, Laura Weidinger, Tu Vu, Alek Andreev, Antoine He, Kevin Hui, Sheleem Kashem, Amar Subramanya, Sissie Hsiao, Demis Hassabis, Koray Kavukcuoglu, Adam Sadovsky, Quoc Le, Trevor Strohman, Yonghui Wu, Slav Petrov, Jeffrey Dean, and Oriol Vinyals. Gemini: A family of highly capable multimodal models, 2025a. URL https://arxiv.org/abs/2312.11805.

Kimi Team, Angang Du, Bohong Yin, Bowei Xing, Bowen Qu, Bowen Wang, Cheng Chen, Chenlin Zhang, Chenzhuang Du, Chu Wei, Congcong Wang, Dehao Zhang, Dikang Du, Dongliang Wang, Enming Yuan, Enzhe Lu, Fang Li, Flood Sung, Guangda Wei, Guokun Lai, Han Zhu, Hao Ding, Hao Hu, Hao Yang, Hao Zhang, Haoning Wu, Haotian Yao, Haoyu Lu, Heng Wang, Hongcheng Gao, Huabin Zheng, Jiaming Li, Jianlin Su, Jianzhou Wang, Jiaqi Deng, Jiezhong Qiu, Jin Xie, Jinhong Wang, Jingyuan Liu, Junjie Yan, Kun Ouyang, Liang Chen, Lin Sui, Longhui Yu, Mengfan Dong, Mengnan Dong, Nuo Xu, Pengyu Cheng, Qizheng Gu, Runjie Zhou, Shaowei Liu, Sihan Cao, Tao Yu, Tianhui Song, Tongtong Bai, Wei Song, Weiran He, Weixiao Huang, Weixin Xu, Xiaokun Yuan, Xingcheng Yao, Xingzhe Wu, Xinxing Zu, Xinyu Zhou, Xinyuan Wang, Y. Charles, Yan Zhong, Yang Li, Yangyang Hu, Yanru Chen, Yejie Wang, Yibo Liu, Yibo Miao, Yidao Qin, Yimin Chen, Yiping Bao, Yiqin Wang, Yongsheng Kang, Yuanxin Liu, Yulun Du, Yuxin Wu, Yuzhi Wang, Yuzi Yan, Zaida Zhou, Zhaowei Li, Zhejun Jiang, Zheng Zhang, Zhilin Yang, Zhiqi Huang, Zihao Huang, Zijia Zhao, Ziwei Chen, and Zongyu Lin. Kimi-vl technical report, 2025b. URL https://arxiv.org/abs/2504.07491.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Wenhai Wang, Jifeng Dai, Zhe Chen, Zhenhang Huang, Zhiqi Li, Xizhou Zhu, Xiaowei Hu, Tong Lu, Lewei Lu, Hongsheng Li, Xiaogang Wang, and Yu Qiao. Internimage: Exploring large-scale vision foundation models with deformable convolutions, 2023. URL https://arxiv.org/abs/2211.05778.

Junjie Wen, Yichen Zhu, Jinming Li, Minjie Zhu, Zhibin Tang, Kun Wu, Zhiyuan Xu, Ning Liu, Ran Cheng, Chaomin Shen, et al. Tinyvla: Towards fast, data-efficient vision-language-action models for robotic manipulation. *IEEE Robotics and Automation Letters*, 2025.

xAI. Grok 3 beta: The age of reasoning agents. https://x.ai/news/grok-3, February 2025. Accessed: 2025-09-25.

Zheng Xiong, Risto Vuorio, Jacob Beck, Matthieu Zimmer, Kun Shao, and Shimon Whiteson. Distilling morphology-conditioned hypernetworks for efficient universal morphology control. In *International Conference on Machine Learning*, pp. 54777–54791. PMLR, 2024.

Aiyuan Yang, Bin Xiao, Bingning Wang, Borong Zhang, Ce Bian, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan, Fan Yang, Fei Deng, Feng Wang, Feng Liu, Guangwei Ai, Guosheng Dong, Haizhou Zhao, Hang Xu, Haoze Sun, Hongda Zhang, Hui Liu, Jiaming Ji, Jian Xie, JunTao Dai, Kun Fang, Lei Su, Liang Song, Lifeng Liu, Liyun Ru, Luyao Ma, Mang Wang, Mickel Liu, MingAn Lin, Nuolan Nie, Peidong Guo, Ruiyang Sun, Tao Zhang, Tianpeng Li, Tianyu Li, Wei Cheng, Weipeng Chen, Xiangrong Zeng, Xiaochuan Wang, Xiaoxi Chen, Xin Men, Xin Yu, Xuehai Pan, Yanjun Shen, Yiding Wang, Yiyu Li, Youxin Jiang, Yuchen Gao, Yupeng Zhang, Zenan Zhou, and Zhiying Wu. Baichuan 2: Open large-scale language models, 2025. URL https://arxiv.org/abs/2309.10305.

Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models, 2022. URL https://arxiv.org/abs/2205.01917.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Multi-task reinforcement learning without interference. In *Proc. Optim. Found. Reinforcement Learn. Workshop NeurIPS*, 2019.

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, pp. 1094–1100. PMLR, 2020.

Yang Yue, Yulin Wang, Bingyi Kang, Yizeng Han, Shenzhi Wang, Shiji Song, Jiashi Feng, and Gao Huang. Deer-vla: Dynamic inference of multimodal large language models for efficient robot execution. *Advances in Neural Information Processing Systems*, 37:56619–56643, 2024.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 11975–11986, 2023.

Jianke Zhang, Yanjiang Guo, Xiaoyu Chen, Yen-Jen Wang, Yucheng Hu, Chengming Shi, and Jianyu Chen. Hirt: Enhancing robotic control with hierarchical robot transformers. *arXiv preprint arXiv:2410.05273*, 2024.

Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.

Yifan Zhong, Fengshuo Bai, Shaofei Cai, Xuchuan Huang, Zhang Chen, Xiaowei Zhang, Yuanfei Wang, Shaoyang Guo, Tianrui Guan, Ka Nam Lui, et al. A survey on vision-language-action models: An action tokenization perspective. *arXiv preprint arXiv:2507.01925*, 2025.

A Further related work

VLAs Inspired by the success of foundation models in NLP and CV, RT-1 (Brohan et al., 2022) is a pioneering VLA work that validates the effectiveness of pretraining foundation models on large-scale robotic data. RT-2 (Brohan et al., 2023) further builds VLAs upon existing language and vision backbones to utilize their strong generalization ability, instead of learning a generalist controller from scratch on robotic data alone. O'Neill et al. (2024) propose the OXE dataset which validates the effectiveness of learning from cross-embodiment robotic datasets. Built upon these key ideas, more recent work investigates the design choices in VLAs in more detail, yielding further improvement from scaling up training data and learning from unlabeled videos (Black et al., 2024; Bu et al., 2025; Bjorck et al., 2025), the choice of backbone models (Kim et al., 2024; Li et al., 2024a), action representation and generation strategy (Team et al., 2024; Black et al., 2024; Song et al., 2025), fine-tuning on downstream tasks (Kim et al., 2024; 2025), etc. Please see Ma et al. (2024); Din et al. (2025); Kawaharazuka et al. (2025); Zhong et al. (2025) for more detailed reviews on VLAs.

Context-conditioned policy generation Faccio et al. (2023) and Di Ventura et al. (2025) adopt HNs to generate policies that can achieve different amounts of expected return in a single environment, and achieve promising performance on relatively simple continuous control tasks. Generating task-conditioned policies via HNs has been investigated in multi-task and meta-RL (Yu et al., 2019; Sarafian et al., 2021; Beck et al., 2022; 2023b; Rezaei-Shoshtari et al., 2023), but such work mainly focuses on learning lightweight models on narrow task distributions, and do not use HNs for inference acceleration. Make-An-Agent (Liang et al., 2024) treats parameter generation as a denoising process in the parameter space, and generates policy parameters via diffusion conditioned on demonstration trajectories, while our method generates the policy via HNs conditioned on the task context and can thus zero-shot generalize to a new task without additional demonstration trajectories from the new task. Closely related to our work, HyperDistill (Xiong et al., 2024) uses HNs to generate compact locomotion policies for efficient inference on different robot embodiments. Our method shares a similar motivation of accelerating inference via HNs but investigates a much more challenging setting of language-conditioned control with image observations and tackles the instability and generalization challenges in HN training at a much larger model scale.

B Additional experimental setup

B.1 HyperVLA training setup

To stabilize the performance of the learned model, we apply exponential moving average to the model parameters with a smoothing factor of 0.999, and save the smoothed parameters instead of the latest parameters in the model checkpoints for evaluation. We optimize with AdamW (Loshchilov & Hutter, 2017), with a weight decay coefficient of 0.05 on HN output heads. Following the setup in Octo (Team et al., 2024), we set the peak learning rate as 3e-4, and apply learning rate warmup for 2k steps, then anneal it with an inverse square root schedule. The learning rate for the DINOv2 image encoder in the base network shares the same schedule, but uses a much lower peak value of 3e-5. To enable better generalization, we augment both the language instruction by rephrasing, and the image observation by image augmentation as done in Octo. Other training hyperparameters follow the same setup as in Octo.

B.2 Evaluation benchmarks

SIMPLER Table 5 shows more detailed information about the evaluation tasks included in the SIMPLER benchmark.

LIBERO We evaluate few-shot adaptation of HyperVLA on the same four task suites as used in OpenVLA:

1. **LIBERO-Spatial** evaluates generalization to different layouts of the same set of objects;
2. **LIBERO-Object** evaluates generalization to different object types with the same scene layout;

Robot	Task suite	# Instruction	Seen during training	# Demonstration in OXE	# Eval
Google Robot	pick	13	6 seen, 7 unseen	~600 per seen object	150
	move	30	Yes	40 to 100 per instruction	180
	close drawer	3	Yes	$>$ 2000 per instruction	180
WidowX	spoon on towel	1	Yes	1	60
	carrot on plate	1	Yes	332	60
	stack cube	1	No	0	60
	eggplant in basket	1	No	0	60

Table 5: Summary of evaluation tasks in SIMPLER.

3. **LIBERO-Goal** evaluates generalization to different goals (instructions) with the same set of objects and layout; and
4. **LIBERO-Long** evaluates performance on long-horizon tasks with diverse objects, layouts, and tasks.

C Additional experimental results

C.1 Further ablation studies

Method	Google robot				WidowX				
	pick	move	close drawer	Avg	spoon on towel	carrot on plate	stack cube	eggplant in basket	Avg
HyperVLA (Full)	58 ± 3	73 ± 1	58 ± 7	63 ± 3	48 ± 3	21 ± 5	39 ± 8	52 ± 13	40 ± 5
Larger learning rate for DINOv2	3	13	17	11	0	0	0	0	0
Frozen DINOv2	56	47	58	54	18	40	0	3	15
Fine-tuned CLIP	58	61	59	59	63	30	13	0	27
Frozen SigLIP	20	34	49	34	3	0	0	0	1
Train base net alone	5	11	12	9	3	0	0	0	1

Table 6: Ablation results on how different algorithm designs in HyperVLA influence its performance.

We report further ablation results in Table 6 to validate the importance of the following design choices in HyperVLA. To reduce computational cost, we run each ablation experiment with only one seed, while the performance gap is significant enough to draw conclusions with high confidence.

Smaller learning rate for DINOv2 fine-tuning To ablate the importance of fine-tuning DINOv2 with a smaller learning rate as introduced in Section 3.3, we increase the learning rate for DINOv2 by 10 times, and use the same learning rate of 0.0003 for both HN training and DINOv2 fine-tuning. This variant performs poorly, which validates the importance of fine-tuning DINOv2 with a smaller learning rate to maintain its strong prior knowledge.

Fine-tuning versus freezing DINOv2 To validate the importance of fine-tuning DINOv2 in the base network, we ablate by freezing it while keeping the remaining settings unchanged. This variant underperforms HyperVLA, which validates the importance of fine-tuning DINOv2 during HyperVLA pretraining.

Choice of the image encoder As introduced in Section 3.3, HyperVLA can support different image encoders, and empirically we find DINOv2 to perform best. We ablate by using either CLIP (Radford et al., 2021b) or SigLIP (Zhai et al., 2023) as the image encoder in the base network. As our code is implemented in JAX and we can only find a PyTorch version of SigLIP, our code does not support fine-tuning SigLIP during training. The ablation results show that using fine-tuned DINOv2 outperforms fine-tuned CLIP, while frozen DINOv2 outperforms frozen SigLIP.

Importance of the HN We run this ablation experiment to validate that inference acceleration can not be achieved by training a small base network alone, and the HN in our method is essential to maintain high model capacity and achieve good performance. We experiment by removing the HN in HyperVLA and train the base network alone. This variant performs poorly, validating the importance of using HN.

C.2 QUALITATIVE ANALYSIS

We include example videos of rolling out different methods on different tasks in the supplemental material, and qualitatively analyze the common failure patterns of different methods as follows:

The main failure reason of our method is inaccurate grasping of the object to manipulate, e.g., the policy sometimes may close the gripper when the end-effector is still slightly above the target object, which makes the robot fail to pick up the object. In general, the action error of HyperVLA is small and may be mitigated by integrating more camera views or further fine-tuning.

The OpenVLA baseline significantly underperforms our method on the picking task of Google Robot, while performing similarly on the other tasks. Its main failure reason is similar to HyperVLA due to inaccurate grasping. However, it also makes some other obvious mistakes, such as the robot arm getting stuck in the air, and not picking the target object up as expected after successfully grasping it. We also find that the robot arm movement controlled by OpenVLA is less smooth than HyperVLA, possibly due to its autoregressive way of predicting discretized action tokens.

For the other two weaker baselines RT-1-X and Octo, in addition to the grasping error, they sometimes even can not correctly locate the object to manipulate, or misunderstand the language instruction semantically, such as moving a wrong object to a wrong target object in the moving tasks, possibly due to that they are not built upon language and vision foundation models with strong generalization ability.