Energy-Based Action Heads Know When They Don't Know

Abstract— **Vision-language-action (VLA) models have become popular for robot manipulation but still often fail at their tasks, particularly in out-of-distribution (OOD) scenarios. However, existing VLAs are very rigid, and require additional post-hoc methods for OOD-detection, failure prediction and some other test time behaviours. We propose the Energy-Based VLA (EB-VLA) to address this limitation. Instead of generating actions directly, EB-VLA learns a scalar energy landscape over the action space conditioned on multimodal context, generating action chunks via test-time optimization. Our experiments across different manipulation benchmarks demonstrate two benefits of EB-VLA: First, we achieve competitive manipulation performance, outperforming diffusion policies on contact-rich tasks and matching token-reasoning VLAs that are an order of magnitude larger. Second, the energy value serves as a learned confidence score, acting as a highly effective, zero-shot OOD detector against visual perturbations that requires no additional training or calibration. Our results highlight the potential of EB-VLA for self-monitoring robot policies that can tell when they are uncertain.**

I. INTRODUCTION AND RELATED WORK

Vision-language-action (VLA) models [3], [4], [13] map images, language, and proprioceptive state to continuous control sequences, achieving strong performance for generalist robotic manipulation. Most VLAs combine a vision-language model (VLM) backbone for semantic understanding with an action head that translates high-level intent into low-level robot actions. State-of-the-art VLAs mostly remain brittle and struggle in out-of-distribution situations. However, the predominant action heads directly generate actions without a score reflecting the model's confidence about their suitability for the given context. This lack of uncertainty-awareness in VLAs limits their interpretability and applicability to safety-critical scenarios.

There are two dominant designs for the action head. Earlier autoregressive approaches [3], [15] produce point estimates but cannot capture the multimodality inherent in human demonstrations. More recently, diffusion-based methods [1], [16], [17] have become prevalent due to their ability to model multimodal action distributions. Neither of the prevalent action head designs is inherently suited to obtain uncertainty estimates. Diffusion policies amortize a score function, i.e., the gradient of an implicit log-density [18], [19], but not the action likelihood itself, which could otherwise be used to measure model confidence. As a consequence, runtime monitoring of robot policies is mostly achieved through post-hoc methods. These include classical approaches, such as MC dropout [20] or deep ensembles [21], conformal prediction over actions [22], and recently FIPER [23], which combines random network distillation in the policy's embedding space with an action-chunk entropy score and conformal calibration on successful rollouts. While these approaches have demonstrated potential for detecting OOD situations and predicting failures, they either require additional auxiliary training or calibration procedures, or both. Ideally, VLAs should possess an inherent awareness of their own uncertainty, enabling built-in OOD detection and failure prediction.

Energy-based models, which learn a scalar energy function, offer a promising paradigm for more interpretable and uncertainty-aware robot policies. Recent work on energy-based transformers (EBTs) [2] has shown that energy-based models, which traditionally suffered from training-instability issues [25], [26], can be scaled to high-dimensional outputs. Recently, EBT-Policy [24] has applied energy-based models to robotic manipulation, outperforming diffusion policies on several manipulation benchmarks [30]. However, the applicability of energy-based models to language-conditioned, generalist manipulation remains an open question.

This work provides the first integration of an energy-based action head into a VLA, addressing the limitations in existing designs. The head learns a scalar energy function over the action space conditioned on the scene, with expert demonstrations placed at the minima of the resulting landscape. This has two useful properties. First, actions are produced by gradient descent on the energy, which recovers the multimodal expressiveness of diffusion without committing to a fixed denoising schedule [2], [24]. Second, the same scalar that drives action optimization also serves as a learned verifier of the resulting trajectory: the energy at convergence is a compatibility estimate between the scene and the produced action, available in the same forward pass as the action itself. As a consequence, instead of bolting OOD and failure detection onto a fixed generative policy, our energy-based action head supplies these signals intrinsically, requiring no auxiliary networks, calibration rollouts, or additional forward passes.

We instantiate this idea as the **Energy-Based VLA (EB-VLA)**. On the LIBERO benchmark [31], EB-VLA matches token-reasoning VLAs [6], [5] with up to 14× more parameters. On LIBERO-Plus with camera perturbations, the raw energy scalar, used as an OOD detector with a single threshold and no additional training, reaches 84.5% balanced accuracy. Our results demonstrate that the energy value in EB-VLA provides a valuable signal for runtime monitoring of VLAs, obtainable without any post-hoc machinery.

II. METHODS

Our architecture has two components: a VLM backbone that encodes the language instruction and visual observations,

and an energy-based Transformer (EBT) action head that refines continuous actions by gradient descent on a learned energy landscape.

A. Vision-Language Backbone

The backbone pairs a fused DINOv2 [8] and SigLIP [9] vision encoder with a Qwen2.5 [10], [11] language model, following the Prismatic [12] recipe and the OpenVLA-OFT [13] architecture. The prompt is augmented with K_a learned latent embeddings that attend to the multimodal context during the forward pass. With two camera views (wrist and primary), the frozen vision encoder produces K_t tokens. The language model is fine-tuned with LoRA [27].

After the forward pass, we extract hidden states from all N_ℓ layers (including the embedding). We split each layer's states into the first K_t positions ($h_t^{(\ell)} \in \mathbb{R}^{K_t \times D}$, vision/task features) and the last K_a positions ($h_a^{(\ell)} \in \mathbb{R}^{K_a \times D}$, latent features), stacked across layers into $h_t \in \mathbb{R}^{N_\ell \times K_t \times D}$ and $h_a \in \mathbb{R}^{N_\ell \times K_a \times D}$. The proprioceptive state is linearly projected to $p \in \mathbb{R}^{1 \times D}$.

B. Energy-Based Transformer Action Head

The action head learns a scalar energy $E_\theta(a, C)$ over candidate action sequences $a \in \mathbb{R}^{H \times d_a}$ and VLM context C.

Candidate actions are projected to hidden dimension D and processed by L transformer blocks. Each block applies (i) bidirectional self-attention over action tokens with RoPE [14], (ii) cross-attention into a block-specific VLM context

$$C_i = [h_a^{(\ell_i)};\ p;\ h_t^{(\ell_i)}] \in \mathbb{R}^{(K_a+1+K_t)\times D}, \tag{1}$$

and (iii) a SwiGLU FFN [28]. Action tokens unidirectionally attend to the VLM context. Thus, MCMC refinement of a does not affect the conditioning input. After a final RMSNorm [29], a linear head produces per-timestep energies $e_t = E_\theta(z_t)$, and we can obtain the mean across the horizon as $E(a, C) = \frac{1}{H}\sum_t e_t$.

C. Gradient-Based Inference

At inference, action sequences are initialized from Gaussian noise $a^{(0)} \sim \mathcal{N}(0, \sigma_{\text{init}}^2 I)$ and refined by MCMC gradient descent on the learned energy landscape E_θ, with the model weights frozen. At each step k, we perturb the action sequence with Langevin noise $\epsilon^{(k)} \sim \mathcal{N}(0, \sigma_L^2 I)$ and take the gradient at the unperturbed actions,

$$g^{(k)} = \nabla_a E(a^{(k)} + \epsilon^{(k)}, C)\big|_{a=a^{(k)}}, \tag{2}$$

which smooths the landscape without accumulating noise across steps. Gradients are element-wise clipped to $[-c, c]$.

The update is energy-scaled per timestep with temperature τ:

$$a_t^{(k+1)} = a_t^{(k)} - \alpha \exp\left(\frac{e_t^{(k)}}{\tau}\right) g_t^{(k)}, \tag{3}$$

with $\alpha \sim \text{Uniform}[\alpha_{\min}, \alpha_{\max}]$. High-energy timesteps take larger steps, which become smaller as actions settle into expert-like regions. Actions are clamped to $[-a_{\max}, a_{\max}]$ after each update.

a) Training.: At each step we run $K_{\text{train}} \sim \text{Uniform}[K_{\min}, K_{\max}]$ MCMC updates from noise, detach the graph at every step except the last, and supervise with

$$\mathcal{L} = \|a^{(K_{\text{train}})} - a^*\|_1. \tag{4}$$

At test time, we run a fixed number of K_{inf} steps, without early stopping, and execute the resulting action sequence. The energy at the final iteration, $E(a^{(K_{\text{inf}})}, C)$, is a direct scalar confidence score, which we leverage for OOD detection in Section III-C.

D. Failure Prediction via Energy Landscape Analysis

Because the energy function E_θ is explicitly trained to assign low values to expert-like actions and high values to deviations, the energy magnitude at test time serves as a direct proxy for the model's confidence in its generated trajectory.

During task execution, if the MCMC refinement fails to drive the energy below a designated threshold, it indicates that the model cannot find an action plan compatible with the current observation and instruction. We leverage this property to predict imminent task failure during execution: by monitoring the energy trajectory during the early stages of inference, the system can identify out-of-distribution states or configurations that exceed its operational envelope.

This property represents an inherent, fundamental advantage of the energy-based formulation over both regression heads, which provide no confidence signal, and latent-convergence metrics used in recurrent architectures [7], which measure representational stability rather than action quality. The energy is trained end-to-end with a direct relationship to expert behavior, making it a more semantically meaningful indicator of whether a generated plan is likely to succeed.

III. EXPERIMENTS

We evaluate our approach along three axes. First, we isolate the EBT action head and validate it against Diffusion Policy on standard manipulation benchmarks without language conditioning (Section III-A). Second, we integrate the EBT head into a full VLA pipeline and compare against state-of-the-art VLAs on the LIBERO benchmark (Section III-B). Third, we analyze the energy signal as a zero-shot OOD detector (Section III-C) and failure predictor (Section III-D).

A. Action Head Validation on RoboMimic

Before integrating the EBT into a language-conditioned VLA, we first verify that the energy-based action head produces competitive imitation learning performance on its own. We evaluate against Diffusion Policy [1] across five manipulation tasks from the RoboMimic benchmark [30]: Lift, Can, Square, Transport, and Tool Hang, as well as the Push-T task from Diffusion Policy. Both methods use the same vision encoder and differ only in the action head. For each task, we report the maximum success rate over training, evaluated across 50 test environments.

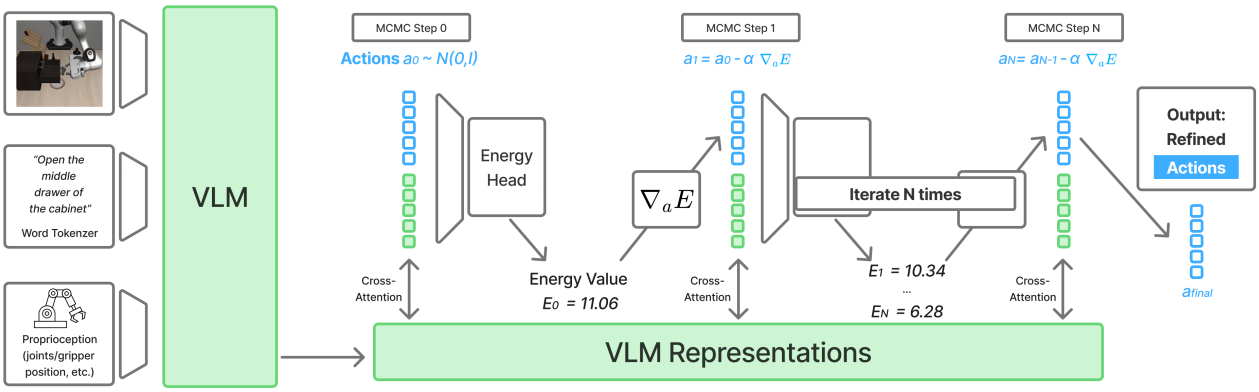

Fig. 1. Architecture of the Energy-Based VLA.

TABLE I

MAXIMUM SUCCESS RATE ON ROBOMIMIC (50 TEST ENVIRONMENTS). **BOLD** INDICATES THE BEST RESULT PER TASK.

	Lift	Can	Square	Transport	Tool Hang	Push-T
DP	**1.0**	**1.0**	**0.9**	**0.98**	0.76	**0.78**
EBT	**1.0**	**1.0**	**0.9**	0.96	**0.84**	0.74

TABLE II

COMPARISON ON THE LIBERO [31] BENCHMARK. **BOLD** INDICATES THE BEST PERFORMANCE, *Italics* THE SECOND BEST. "PARAMS" DENOTES BACKBONE SCALE IN BILLIONS.

Method	Params	Spat.	Obj.	Goal	Long	Avg.
SmolVLA [32]	2.2B	93.0	94.0	91.0	77.0	88.8
OpenVLA [3]	7B	84.7	88.4	79.2	53.7	76.5
OpenVLA-OFT [13]	7B	**97.6**	*98.4*	**97.9**	**94.5**	**97.1**
WorldVLA [33]	7B	87.6	96.2	83.4	60.0	81.8
π_0-FAST [4]	3B	*96.4*	96.8	88.6	60.2	85.5
CoT-VLA [6]	7B	87.5	91.6	87.6	69.0	81.1
FlowVLA [34]	8.5B	93.2	95.0	91.6	72.6	88.1
SpatialVLA [35]	4B	88.2	89.9	78.6	55.5	78.1
ThinkAct [36]	7B	88.3	91.4	87.1	70.9	84.4
Fast-ThinkAct [37]	3B	92.0	97.2	90.2	79.4	89.7
TraceVLA [38]	7B	84.6	85.2	75.1	54.1	74.8
MolmoAct [5]	7B	87.0	95.4	87.6	77.2	86.6
RD-VLA [7]	0.5B	92.0	**99.0**	*96.0*	84.8	93.0
EB-VLA (Ours)	0.5B	95.0	**99.0**	*96.0*	*90.2*	*95.1*

As shown in Table I, EBT matches Diffusion Policy on the simpler tasks (Lift, Can, Square) and achieves a notable improvement on Tool Hang (0.84 vs. 0.76), the most challenging contact-rich task in the benchmark. We attribute this to the gradient-based refinement procedure, which allows the energy head to more precisely resolve the fine-grained action sequences required for tool manipulation, where small errors in gripper pose compound quickly. On Transport, EBT performs slightly below DP (0.96 vs. 0.98), and on Push-T the gap is larger (0.74 vs. 0.78), suggesting that the coarse-to-fine denoising schedule of diffusion policies may be better suited to tasks with broad, unimodal action distributions. In summary, these results establish our EBT action head as a competitive drop-in replacement for diffusion-based action heads, motivating its integration into a full VLA pipeline.

B. Language-Conditioned Evaluation on LIBERO

We use the LIBERO benchmark to evaluate the performance of our energy-based action head in language-conditioned manipulation. As shown in Table II, our approach achieves strong results across different LIBERO suites, comparable to state-of-the-art VLAs. We emphasize that our objective in this work is not to build a VLA that beats all baselines in raw performance, but rather to highlight specific properties enabled by the energy-based action head, such as OOD and failure detection.

C. Out-of-Distribution (OOD) Detection

A critical advantage of the EBT is its ability to quantify uncertainty without additional training heads or sampling-based variance estimates. We define the **energy-based anomaly score** as the final scalar energy value $E(a^{(K)}, C)$ obtained after K iterations of MCMC refinement.

To evaluate the robustness of our uncertainty signal, we measure OOD detection performance using the LIBERO-Plus benchmark [39], specifically focusing on **Camera Perturbations**. These perturbations introduce out-of-distribution visual noise through varying degrees of severity in camera pose shifts and focal distortions that were not present during the original training distribution.

We establish a statistical baseline using a set of 120 "vanilla" (in-distribution) rollouts to compute the baseline mean (μ_v) and standard deviation (σ_v). We then define a classification threshold T:

$$T = \mu_v + k \cdot \sigma_v \tag{1}$$

where k is a multiplier controlling the sensitivity of the detector. Any rollout yielding an energy $E > T$ is flagged as an OOD event. We evaluate this detector across 825 OOD rollouts containing the aforementioned camera perturbations.

TABLE III

OOD DETECTION PERFORMANCE ON LIBERO-PLUS CAMERA PERTURBATIONS: EVALUATING STANDARD DEVIATION MULTIPLIERS FOR ENERGY-BASED THRESHOLDING ($\mu_v = 5.869$, $\sigma_v = 0.407$).

Multiplier	Threshold	Accuracy	Bal. Acc.	TPR (Catch)	FPR (False Alarm)	F1-Score	FP	FN
0.2σ	5.950	86.5%	74.8%	90.4%	40.8%	92.1%	49	79
0.5σ	6.073	85.4%	81.3%	86.8%	24.2%	91.2%	29	109
0.7σ	6.154	84.7%	83.7%	85.0%	17.5%	90.6%	21	124
0.8σ	6.195	83.1%	83.5%	82.9%	15.8%	89.5%	19	141
0.9σ	6.235	82.4%	84.2%	81.8%	13.3%	89.1%	16	150
1.0σ	6.276	81.6%	84.5%	80.6%	11.7%	88.4%	14	160
1.5σ	6.480	70.3%	80.1%	66.9%	6.7%	79.7%	8	273
2.0σ	6.683	52.9%	70.9%	46.8%	5.0%	63.4%	6	439
3.0σ	7.090	17.9%	53.0%	5.9%	0.0%	11.2%	0	776

As shown in Table III, the 1.0σ multiplier provides the most effective balance for general deployment, achieving a balanced accuracy of 84.5%. For safety-critical applications that require greater outlier sensitivity, the 0.2σ threshold successfully identifies 90.4% of all perturbations (TPR).

D. Energy-Based Failure Prediction

We also analyze the temporal dynamics of the energy landscape to predict imminent task failure. To this end, we calculate the linear slope of energy values over a sliding window of past timesteps. In our setup, a "step" represents a sequential action generation checkpoint within the policy rollout (ranging from step 1 to 20), while the "window" defines the specific number of contiguous past steps used to calculate the linear regression of the energy metric. We introduce two primary predictive metrics: *Slope Start*, which tracks the rate of change of the initial unrefined energy across recent states, and *Slope Final*, which measures the trajectory of the converged energy after MCMC refinement. By evaluating these slopes at various standard deviation thresholds, we can assess the model's ability to detect degradation in action confidence prior to a catastrophic failure. We observe that a high *Slope Final* score is a strong indicator of failure. Intuitively, a rising final energy slope indicates that the optimization process is increasingly struggling to find expert-like actions, which often leads to task failure. Table IV summarizes the best predictive configurations for the final energy slope across all evaluation steps, highlighting the optimal window sizes and sensitivity multipliers that maximize balanced accuracy.

TABLE IV

BEST CONFIGURATIONS FOR FAILURE PREDICTION WITH OUR SLOPE FINAL SCORE.

Step	**Win.**	**Std**	**Acc.**	**Bal. Acc.**	**TPR**	**Spec.**
1	2	0.10	68.99%	50.00%	0.00%	100.00%
2	2	1.00	65.08%	52.89%	20.82%	84.97%
3	3	0.50	67.94%	56.84%	27.64%	86.04%
4	4	0.30	64.66%	53.24%	23.21%	83.28%
5	3	1.70	68.36%	50.67%	4.10%	97.24%
6	4	1.50	69.21%	51.19%	3.75%	98.62%
7	7	0.10	62.75%	52.71%	26.28%	79.14%
8	8	0.20	64.02%	53.16%	24.57%	81.75%
9	2	0.10	59.05%	55.19%	45.05%	65.34%
10	2	0.20	66.24%	60.97%	47.10%	74.85%
11	2	0.10	62.43%	64.69%	70.65%	58.74%
12	5	0.20	69.84%	65.36%	53.58%	77.15%
13	7	0.10	67.94%	64.64%	55.97%	73.31%
14	10	0.10	71.32%	65.69%	50.85%	80.52%
15	11	0.50	72.70%	64.24%	41.98%	86.50%
16	14	0.60	74.39%	63.12%	33.45%	92.79%
17	16	0.70	72.80%	60.37%	27.64%	93.10%
18	14	1.30	73.65%	60.70%	26.62%	94.78%
19	14	1.40	72.38%	60.16%	27.99%	92.33%
20	15	1.30	71.00%	60.38%	32.42%	88.34%

IV. CONCLUSIONS

In this work, we introduced the Energy-Based VLA (EB-VLA), presenting the first integration of an energy-based action head into a language-conditioned, generalist manipulation policy. By formulating action generation as MCMC refinement over a learned energy landscape, we demonstrated that EB-VLA matches the performance of state-of-the-art autoregressive and diffusion-based VLAs on standard benchmarks, including RoboMimic and LIBERO. In addition, we have shown properties of energy-based action heads such as OOD detection and failure prediction. However, further study is required into the expressivity of the energy head and the mentioned properties. We believe it is a promising direction for future action head design.

REFERENCES

[1] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song, "Diffusion Policy: Visuomotor Policy Learning via Action Diffusion," *The International Journal of Robotics Research*, vol. 44, no. 10–11, pp. 1684–1704, 2025.

[2] A. Gladstone, G. Nanduru, M. M. Islam, P. Han, H. Ha, A. Chadha, Y. Du, H. Ji, J. Li, and T. Iqbal, "Energy-Based Transformers are Scalable Learners and Thinkers," *arXiv preprint arXiv:2507.02092*, 2025.

[3] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi, *et al.*, "OpenVLA: An Open-Source Vision-Language-Action Model," *arXiv preprint arXiv:2406.09246*, 2024.

[4] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, *et al.*, "π_0: A Vision-Language-Action Flow Model for General Robot Control," *arXiv preprint arXiv:2410.24164*, 2024.

[5] J. Lee, J. Duan, H. Fang, Y. Deng, S. Liu, B. Li, B. Fang, J. Zhang, Y. R. Wang, S. Lee, *et al.*, "MolmoAct: Action Reasoning Models that can Reason in Space," *arXiv preprint arXiv:2508.07917*, 2025.

[6] Q. Zhao, Y. Lu, M. J. Kim, Z. Fu, Z. Zhang, Y. Wu, Z. Li, Q. Ma, S. Han, C. Finn, *et al.*, "CoT-VLA: Visual Chain-of-Thought Reasoning for Vision-Language-Action Models," in *Proc. CVPR*, pp. 1702–1713, 2025.

[7] Y. Tur, J. Naghiyev, H. Fang, W.-C. Tsai, J. Duan, D. Fox, and R. Krishna, "Recurrent-Depth VLA: Implicit Test-Time Compute Scaling of Vision–Language–Action Models via Latent Iterative Reasoning," *arXiv preprint arXiv:2602.07845*, 2026.

[8] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, *et al.*, "DINOv2: Learning Robust Visual Features without Supervision," *arXiv preprint arXiv:2304.07193*, 2023.

[9] X. Zhai, B. Mustafa, A. Kolesnikov, and L. Beyer, "Sigmoid Loss for Language Image Pre-Training," in *Proc. ICCV*, pp. 11975–11986, 2023.

[10] A. Yang, B. Yang, B. Hui, *et al.*, "Qwen2 Technical Report," *arXiv preprint arXiv:2407.10671*, 2024.

[11] Qwen Team, "Qwen2.5: A Party of Foundation Models," September 2024.

[12] S. Karamcheti, S. Nair, A. Balakrishna, P. Liang, T. Kollar, and D. Sadigh, "Prismatic VLMs: Investigating the Design Space of Visually-Conditioned Language Models," in *International Conference on Machine Learning*, 2024.

[13] M. J. Kim, C. Finn, and P. Liang, "Fine-Tuning Vision-Language-Action Models: Optimizing Speed and Success," *arXiv preprint arXiv:2502.19645*, 2025.

[14] J. Su, M. Ahmed, Y. Lu, S. Pan, W. Bo, and Y. Liu, "RoFormer: Enhanced Transformer with Rotary Position Embedding," *Neurocomputing*, vol. 568, p. 127063, 2024.

[15] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn, *et al.*, "RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control," *arXiv preprint arXiv:2307.15818*, 2023.

[16] Y. Ze, G. Zhang, K. Zhang, C. Hu, M. Wang, and H. Xu, "3D Diffusion Policy: Generalizable Visuomotor Policy Learning via Simple 3D Representations," in *Proc. RSS*, 2024.

[17] S. Liu, L. Wu, B. Li, H. Tan, H. Chen, Z. Wang, K. Xu, H. Su, and J. Zhu, "RDT-1B: A Diffusion Foundation Model for Bimanual Manipulation," *arXiv preprint arXiv:2410.07864*, 2024.

[18] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-Based Generative Modeling through Stochastic Differential Equations," in *International Conference on Learning Representations (ICLR)*, 2021.

[19] J. Ho, A. Jain, and P. Abbeel, "Denoising Diffusion Probabilistic Models," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[20] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning," in *International Conference on Machine Learning*, 2016.

[21] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[22] A. Z. Ren, A. Dixit, A. Bodrova, S. Singh, S. Tu, N. Brown, P. Xu, L. Takayama, F. Xia, J. Varley, *et al.*, "Robots That Ask For Help: Uncertainty Alignment for Large Language Model Planners," in *Conference on Robot Learning (CoRL)*, 2023.

[23] R. Römer, A. Kobras, L. Worbis, and A. P. Schoellig, "Failure Prediction at Runtime for Generative Robot Policies," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[24] T. Davies, Y. Huang, A. Gladstone, Y. Liu, X. Chen, H. Ji, H. Liu, and L. Hu, "EBT-Policy: Energy Unlocks Emergent Physical Reasoning Capabilities," *arXiv preprint arXiv:2510.27545*, 2025.

[25] Y. Du and I. Mordatch, "Implicit Generation and Modeling with Energy Based Models," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[26] Y. Song and D. P. Kingma, "How to Train Your Energy-Based Models," *arXiv preprint arXiv:2101.03288*, 2021.

[27] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-Rank Adaptation of Large Language Models," in *International Conference on Learning Representations (ICLR)*, 2022.

[28] N. Shazeer, "GLU Variants Improve Transformer," *arXiv preprint arXiv:2002.05202*, 2020.

[29] B. Zhang and R. Sennrich, "Root Mean Square Layer Normalization," *arXiv preprint arXiv:1910.07467*, 2019.

[30] A. Mandlekar, D. Xu, J. Wong, S. Nasiriany, C. Wang, R. Kulkarni, L. Fei-Fei, S. Savarese, Y. Zhu, and R. Martín-Martín, "What Matters in Learning from Offline Human Demonstrations for Robot Manipulation," in *Conference on Robot Learning (CoRL)*, 2021.

[31] B. Liu, Y. Zhu, C. Gao, Y. Feng, Q. Liu, Y. Zhu, and P. Stone, "LIBERO: Benchmarking Knowledge Transfer for Lifelong Robot Learning," *arXiv preprint arXiv:2306.03310*, 2023.

[32] M. Shukor, D. Aubakirova, F. Capuano, *et al.*, "SmolVLA: A Vision-Language-Action Model for Affordable and Efficient Robotics," *arXiv preprint arXiv:2506.01844*, 2025.

[33] J. Cen, C. Yu, H. Yuan, *et al.*, "WorldVLA: Towards Autoregressive Action World Model," *arXiv preprint arXiv:2506.21539*, 2025.

[34] Z. Zhong, H. Yan, J. Li, *et al.*, "FlowVLA: Visual Chain of Thought-Based Motion Reasoning for Vision-Language-Action Models," *arXiv preprint arXiv:2508.18269*, 2025.

[35] D. Qu, H. Song, Q. Chen, *et al.*, "SpatialVLA: Exploring Spatial Representations for Visual-Language-Action Model," *arXiv preprint arXiv:2501.15830*, 2025.

[36] C.-P. Huang, Y.-H. Wu, M.-H. Chen, Y.-C. F. Wang, and F.-E. Yang, "ThinkAct: Vision-Language-Action Reasoning via Reinforced Visual Latent Planning," *arXiv preprint arXiv:2507.16815*, 2025.

[37] C.-P. Huang, Y. Man, Z. Yu, M.-H. Chen, J. Kautz, Y.-C. F. Wang, and F.-E. Yang, "Fast-ThinkAct: Efficient Vision-Language-Action Reasoning via Verbalizable Latent Planning," *arXiv preprint arXiv:2601.09708*, 2026.

[38] R. Zheng, Y. Liang, S. Huang, J. Gao, H. Daumé III, A. Kolobov, F. Huang, and J. Yang, "TraceVLA: Visual Trace Prompting Enhances Spatial-Temporal Awareness for Generalist Robotic Policies," *arXiv preprint arXiv:2412.10345*, 2024.

[39] S. Fei, S. Wang, J. Shi, Z. Dai, J. Cai, P. Qian, L. Ji, X. He, S. Zhang, Z. Fei, J. Fu, J. Gong, and X. Qiu, "LIBERO-Plus: In-depth Robustness Analysis of Vision-Language-Action Models," *arXiv preprint arXiv:2510.13626*, 2025.

APPENDIX: IMPLEMENTATION DETAILS

TABLE V

CONCRETE VALUES FOR ALL HYPERPARAMETERS INTRODUCED IN SECTION II.

Description	Symbol	Value
Vision-Language Backbone		
Language model	—	Qwen2.5-0.5B
VLM transformer layers	—	24
Layers extracted (w/ embedding)	N_ℓ	25
LoRA rank	—	64
Vision encoder	—	DINOv2 + SigLIP (fused)
Tokens per camera view	—	256
Number of camera views	—	2 (wrist, primary)
Total vision tokens	K_t	512
Learned latent tokens	K_a	64
EBT Action Head		
EBT transformer blocks	L	4
FFN inner dimension (SwiGLU)	—	4D