Appendix

1 FAST Tokenization

1.1 Mathematical Framework

The FAST compression pipeline consists of three stages:

Stage 1: Discrete Cosine Transform (DCT) Given an action chunk $\mathbf{a} = [a_0, a_1, ..., a_{T-1}] \in \mathbb{R}^{T \times D}$ where T is the temporal horizon and D is the action dimension, DCT is applied independently to each dimension:

$$C_k^{(d)} = \alpha_k \sum_{t=0}^{T-1} a_t^{(d)} \cos\left[\frac{\pi(2t+1)k}{2T}\right] \tag{1}$$

where $\alpha_k = \sqrt{1/T}$ for $k = 0$ and $\alpha_k = \sqrt{2/T}$ for $k > 0$. This transformation concentrates signal energy into low-frequency coefficients, with typically 90% of energy captured in the first 10-20% of coefficients.

Stage 2: Adaptive Quantization The DCT coefficients are quantized using a learnable, non-uniform scheme:

$$\hat{C}_k = \text{Quantize}(C_k/\sigma_k, B_k) \cdot \sigma_k \tag{2}$$

where σ_k is the empirical standard deviation of the k-th coefficient across the dataset, and $B_k \in \{2, 4, 8, 16\}$ represents adaptive bit allocation. Low-frequency coefficients receive $B_0 = 16$ bits while high frequencies use $B_{T-1} = 2$ bits.

Stage 3: Byte-Pair Encoding (BPE) The quantized coefficients are flattened and encoded using BPE with vocabulary size $V = 32,768$.

2 Action Chunking

Action chunking [1] is an inference strategy where a policy, for each inference call, outputs a sequence of multiple future actions, known as an "action chunk". This chunk represents the planned actions over a "prediction horizon," H.

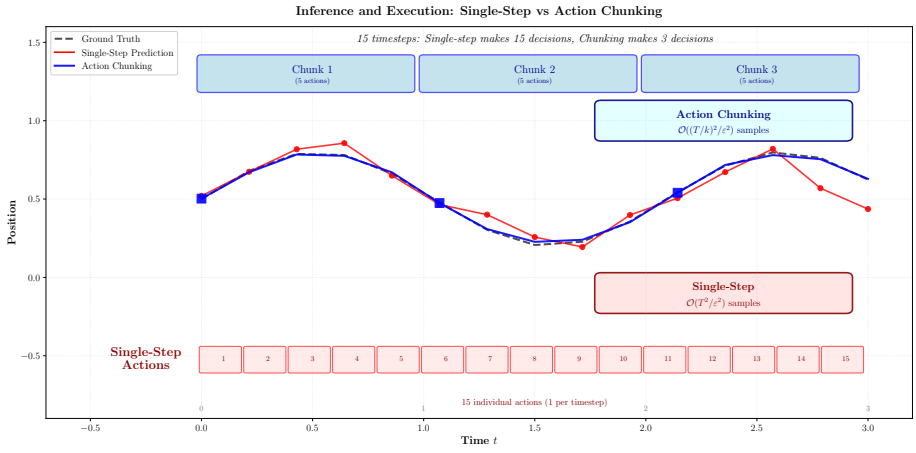

Figure 1: **Single-Step vs. Chunking** Comparison of error accumulation between single-step and chunked action prediction. Single-step prediction suffers from exponentially growing errors that compound over time, requiring $O(T^2/\epsilon^2)$ samples for convergence. Action chunking bounds errors within each chunk boundary, reducing sample complexity to $O((T/k)^2/\epsilon^2)$ where k is the chunk size, achieving $100\times$ better sample efficiency.

For CVAE-based policies, the chunking loss function is typically formulated as:

$$\mathcal{L} = \|a_{t:t+k} - \pi_\theta(s_t, z)\|_1 + \beta D_{KL}(q(z|a_{t:t+k}, s_t)\|p(z)) \tag{3}$$

where the first term is an action reconstruction loss and the KL divergence regularizes the latent space z. Temporal ensembling smooths execution by weighting overlapping predictions:

$$a_t^{exec} = \sum_{i=0}^{k-1} w_i \cdot a_t^{(t-i)} \quad \text{where} \quad w_i = \exp(-m \cdot i) \tag{4}$$

Instead of executing only the next immediate action, the robot executes the first s actions from this chunk, where s is the "execution horizon". This technique is the de facto standard in imitation learning for visuomotor control because it promotes temporal consistency in the robot's movements and reduces the frequency of "mode-jumping"—jerky, discontinuous behavior that can occur at the boundaries between consecutively generated chunks. While effective for dexterous manipulation, this approach can sacrifice reactivity to new observations, as the system is committed to executing a sequence of actions before processing new sensory input. Furthermore, when the latency of the model is high, synchronous execution of action chunks can lead to visible pauses between chunks, slowing down task completion and introducing a distribution shift in the robot's dynamics.

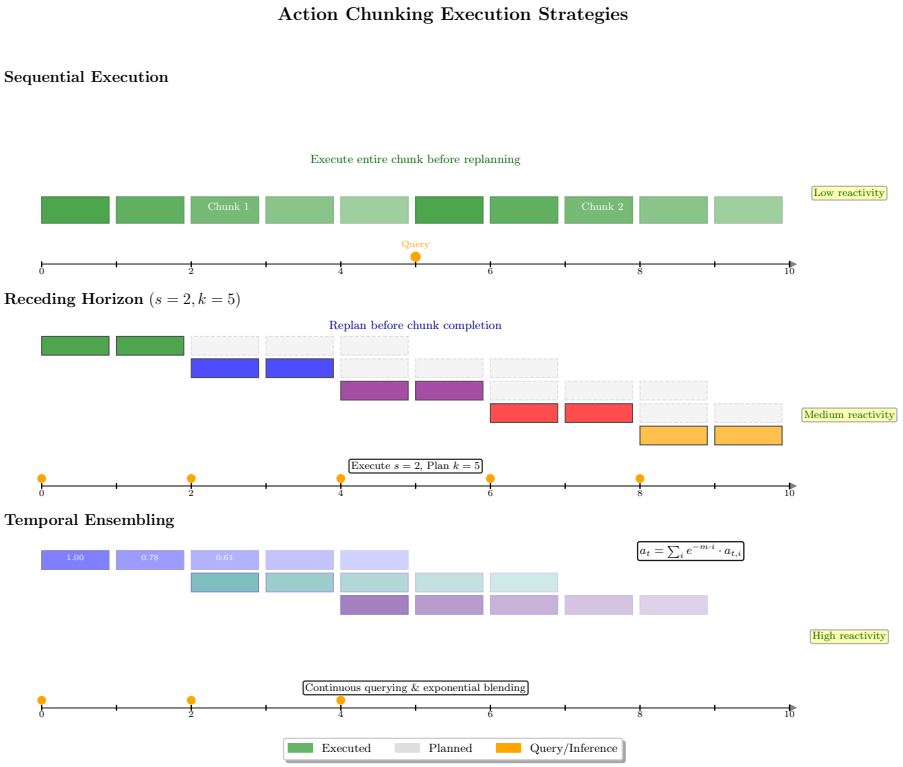

Figure 2: **Execution Strategies** Three execution strategies for action chunking with varying reactivity-smoothness trade-offs.

3 Metrics Catogories

3.1 Task Performance Metrics

Success Rate (SR) Success Rate (SR) [2] serves as the fundamental metric for VLA evaluation, measuring the percentage of successfully completed tasks. Research shows significant variation in success rates across different VLA models and tasks

$$SR = \frac{N_{\text{success}}}{N_{\text{total}}} \tag{5}$$

Current models achievie 12.4% average success on pick-up tasks, 6.0% on move-near tasks, 1.2% on put-on tasks, and 0.5% on put-in tasks. The RT-1-400k model demonstrates the highest performance with 34.4% success on pick-up tasks.

Task Completion Rate This evaluates the ability to execute multi-step task sequences, revealing critical limitations in current VLA models[2]. Analysis shows that success rates drop significantly between sequential steps, indicating challenges in interpreting complex natural language instructions that require multiple actions.

$$TCR = \frac{N_{\text{completed tasks}}}{N_{\text{task chains}}} \tag{6}$$

Action Accuracy This measures precision of predicted actions against ground truth trajectories, typically evaluated using Mean Squared Error (MSE) and its variations including Average MSE(AMSE) and Normalized AMSE (NAMSE)[3][4]. This metric provides direct assessment of model per-formance in offline evaluation scenarios where physical robot deployment is not feasible.

$$MSE = \frac{1}{T}\sum_{t=1}^{T}\|\mathbf{a}_t - \hat{\mathbf{a}}_t\|_2^2 \tag{7}$$

$$AMSE = \frac{1}{K}\sum_{k=1}^{K} MSE_k \tag{8}$$

$$NAMSE = \frac{AMSE}{\sigma^2_{\text{action}}} \tag{9}$$

3.2 Trajectory Quality Metrics

Path Length Path Length quantifies the total distance traveled to complete a task[5], serving as a crucial metric for efficiency evaluation. The metric can be calculated for both actual robot trajectories and virtual reference paths, providing insights into path optimization performance. Shorter paths generally indicate more efficient task execution, though this must be balanced against safety considerations.

$$PL = \sum_{i=1}^{L-1}\|\mathbf{p}_{i+1} - \mathbf{p}_i\|_2 \tag{10}$$

Path Smoothness Path Smoothness[3] evaluates the rate of change in trajectory direction, detecting oscillations that may arise from velocity changes or directional adjustments[6]. The metric is calculated as the absolute distance between displacement vectors in subsequent trajectory points, normalized by total path length. Smooth trajectories are essential for safe robot operation and reduced mechanical wear.

$$PS = \frac{1}{PL} \sum_{i=1}^{L-2} \|(\mathbf{p}_{i+2} - \mathbf{p}_{i+1}) - (\mathbf{p}_{i+1} - \mathbf{p}_i)\|_2 \tag{11}$$

Curvature Change Curvature Change[7] provides specialized evaluation for mobile robots, measuring trajectory smoothness while accounting for robot orientation. This metric proves particularly valuable for ar-like mobile robots where curvature directly relates to turning radius constraints. Unlike path smoothness, curvature change incorporates angular velocity, providing more comprehensive trajectory assessment.

$$CC = \frac{1}{L-2} \sum_{i=1}^{L-2} \left|\kappa_{i+1} - \kappa_i\right|, \ \kappa_i = \frac{\theta_{i+1} - \theta_i}{\|\mathbf{p}_{i+1} - \mathbf{p}_i\|_2} \tag{12}$$

Trajectory Errors Trajectory Error encompasses Absolute Trajectory Error (ATE) and Relative Trajectory Error (RTE), measuring deviations between predicted and reference trajectories[8]. ATE evaluates global consistency while RTE assesses local accuracy, both critical for navigation and manipulation tasks requiring precise positioning[9].

$$ATE = \frac{1}{L} \sum_{i=1}^{L} \|\mathbf{p}_i - \mathbf{p}_i^*\|_2 \tag{13}$$

$$RTE = \frac{1}{L-\Delta} \sum_{i=1}^{L-\Delta} \left\|(\mathbf{p}_{i+\Delta} - \mathbf{p}_i) - (\mathbf{p}_{i+\Delta}^* - \mathbf{p}_i^*)\right\|_2 \tag{14}$$

3.3 Vision-Language Alignment Metrics

BLEU Score BLEU Score measures n-gram overlap between generated and reference captions, providing quantitative assessment of text generation quality[10]. BLEU-4 specifically evaluates 4-gram overlap, commonly used in image captioning tasks where VLA models must describe observed scenes.

$$BLEU_n = BP \, \exp\Big(\tfrac{1}{n} \sum_{k=1}^{n} \ln p_k\Big) \tag{15}$$

CIDEr Score CIDEr Score employs consensus-based evaluation, weighting n-grams using TF-IDF to prioritize informative content[11]. This metric proves particularly effective for evaluating descriptive capabilities of VLA models in explaining their reasoning and observations.

$$CIDEr = \frac{1}{M} \sum_{m=1}^{M} \frac{\sum_g w_g^{(c)} w_g^{(m)}}{\|w^{(c)}\|_2 \, \|w^{(m)}\|_2} \tag{16}$$

METEOR METEOR assesses semantic similarity between generated and reference text, accounting for synonyms and paraphrasing[12]. This metric provides more nuanced evaluation than simple n-gram matching, particularly valuable for natural language interaction capabilities.

$$METEOR = F_{\text{mean}}\big(1 - \gamma\, P_{\text{penalty}}\big) \tag{17}$$

IoU Intersection over Union measures object detection accuracy by evaluating overlap between predicted and ground truth bounding boxes[13]. This metric is fundamental for assessing visual perception capabilities that underpin VLA model performance.

$$IoU = \frac{|B_{\text{pred}} \cap B_{\text{gt}}|}{|B_{\text{pred}} \cup B_{\text{gt}}|} \tag{18}$$

CLIP Score CLIPScore utilizes pre-trained vision-language models to compute similarity between images and text in a shared embedding space. This metric effectively evaluates cross-modal alignment, crucial for VLA models that must bridge visual observations and linguistic instructions[14].

$$CLIPScore = \cos\big(\mathbf{e}_{\text{img}}, \mathbf{e}_{\text{text}}\big) \tag{19}$$

3.4 Safety and Robustness Metrics

Collision Rate Collision Rate quantifies the frequency of collisions during task execution, serving as a primary safety indicator[15]. This metric is particularly critical for mobile robots and humanoids operating in human environments where safety is paramount.

$$CR = \frac{N_{\text{collisions}}}{T_{\text{steps}}} \tag{20}$$

Obstacle Proximity Obstacle Proximity measures the minimum distance between the robot and environmental obstacles throughout task execution. This metric provides insights into safety margins and risk assessment capabilities of VLA models in cluttered environments[16].

$$OP = \min_{t}\, d_t^{\text{robot}\to\text{obstacle}} \tag{21}$$

Risk Factor Risk Factor offers comprehensive safety evaluation[17] by integrating proximity measurements throughout the route. Calculated as the average of the reciprocal distances from obstacles, this metric provides a holistic assessment of safety-conscious behavior.

$$RF = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{d_t} \tag{22}$$

3.5 Efficiency Metrics

Inference Latency Inference Latency measures the time required to generate actions from visual observations and language instructions. This metric is crucial for real-time applications where responsive behavior is essential for effective human-robot interaction[**?**].

$$IL = t_{\text{infer\_end}} - t_{\text{infer\_start}} \tag{23}$$

Computation Time Computation Time evaluates processing requirements per decision cycle[18], informing deploy- ment considerations for resource-constrained environments. This metric helps determine feasibility for edge computing applications and consumer hardware deployment.

$$CT = \frac{\text{CPU/GPU cycles}}{\text{decision step}} \tag{24}$$

Memory Usage Memory Usage assesses resource consumption during operation[19], particularly relevant for compact VLA models like SmolVLA-450M designed for consumer hardware deployment. Efficient memory usage enables broader accessibility and real-world deployment scenarios.

$$MU = \max_t \big(\text{RAM}_t + \text{VRAM}_t\big) \tag{25}$$

References

[1] K. Black, M. Y. Galliker, and S. Levine, "Real-time execution of action chunking flow policies," 2025.

[2] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, *et al.*, "Rt-1: Robotics transformer for real-world control at scale," *arXiv preprint arXiv:2212.06817*, 2022.

[3] M. Dobiš, M. Dekan, P. Beňo, F. Duchoň, and A. Babinec, "Evaluation criteria for trajectories of robotic arms," *Robotics*, vol. 11, p. 29, Feb. 2022.

[4] K. K. A. Farag, H. H. Shehata, and H. M. El-Batsh, "Mobile robot obstacle avoidance based on neural network with a standardization technique," *J. Robot.*, vol. 2021, pp. 1–14, Nov. 2021.

[5] P. Fankhauser, M. Bloesch, D. Rodriguez, R. Kaestner, M. Hutter, and R. Siegwart, "Kinect v2 for mobile robot navigation: Evaluation and modeling," in *2015 International Conference on Advanced Robotics (ICAR)*, IEEE, July 2015.

[6] S. Guillén Ruiz, L. V. Calderita, A. Hidalgo-Paniagua, and J. P. Bandera Rubio, "Measuring smoothness as a factor for efficient and socially accepted robot motion," *Sensors (Basel)*, vol. 20, p. 6822, Nov. 2020.

[7] J.-H. Hwang, R. C. Arkin, and D.-S. Kwon, "Mobile robots at your fingertip: Bezier curve on-line trajectory generation for supervisory control," in *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*, IEEE, 2004.

[8] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, Oct. 2012.

[9] F. Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers, and W. Burgard, "An evaluation of the RGB-D SLAM system," in *2012 IEEE International Conference on Robotics and Automation*, IEEE, May 2012.

[10] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, (USA), p. 311–318, Association for Computational Linguistics, 2002.

[11] R. Vedantam, C. Lawrence Zitnick, and D. Parikh, "Cider: Consensus-based image description evaluation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[12] S. Banerjee and A. Lavie, "METEOR: An automatic metric for MT evaluation with improved correlation with human judgments," in *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization* (J. Goldstein, A. Lavie, C.-Y. Lin, and C. Voss, eds.), (Ann Arbor, Michigan), pp. 65–72, Association for Computational Linguistics, June 2005.

[13] M. Berman, A. R. Triki, and M. B. Blaschko, "The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[14] J. Hessel, A. Holtzman, M. Forbes, R. Le Bras, and Y. Choi, "CLIPScore: A reference-free evaluation metric for image captioning," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing* (M.-F. Moens, X. Huang, L. Specia, and S. W.-t. Yih, eds.), (Online and Punta Cana, Dominican Republic), pp. 7514–7528, Association for Computational Linguistics, Nov. 2021.

[15] M. Hoy, A. S. Matveev, and A. V. Savkin, "Algorithms for collision-free navigation of mobile robots in complex cluttered environments: a survey," *Robotica*, vol. 33, pp. 463–497, Mar. 2015.

[16] N. Blunder, M. Thiel, M. Schrick, J. Hinckeldeyn, and J. Kreutzfeldt, "Integration and evaluation of a close proximity obstacle detection for mobile robots in public space," 2022.

[17] A. Majumdar and M. Pavone, "How should a robot assess risk? towards an axiomatic theory of risk in robotics," in *Springer Proceedings in Advanced Robotics*, pp. 75–84, Cham: Springer International Publishing, 2020.

[18] J. Hartmanis and R. E. Stearns, "On the computational complexity of algorithms," *Trans. Am. Math. Soc.*, vol. 117, p. 285, May 1965.

[19] X.-H. Sun and D. Wang, "APC," *Perform. Eval. Rev.*, vol. 40, pp. 125–130, Oct. 2012.