CoDA: Cosmos-driven DCT-Adaptive Sparse Attention for Efficient Robotic Trajectory Generation

Anonymous CVPR submission

Paper ID *****

Abstract

Large-scale pretrained world models—most notably NVIDIA Cosmos Predict 2.5—provide powerful trajectory generation capabilities for robotic manipulation. Yet this power comes at a steep computational cost: applying full attention across all timesteps becomes a direct bottleneck in real-world robot deployment. We observe that a robot does not need to "think hard" at every moment equally. Recent work on action tokenization (e.g., the FAST framework) has already demonstrated that the high-frequency DCT components of an action sequence effectively encode action complexity. We repurpose this insight into dynamic attention control: at moments when DCT high-frequency energy is strong—when the robot demands precise, complex reasoning—we retain more visual tokens; in less demanding intervals, we aggressively reduce them. This modulation is applied per-chunk, leaving open the possibility of further acceleration in real-time deployment scenarios. Building on this, we propose ***CoDA*** *(**Co**smos-driven* ***D**CT-**A**daptive Sparse Attention), which applies a DCT-based dynamic token budget to the cross-attention layers of Cosmos Predict 2.5. We introduce two gating variants:* ***CoDA-S****, a supervised gate that mimics DCT-derived complexity scores, and* ***CoDA-A****, an autonomous gate that self-optimizes the accuracy–efficiency trade-off via a penalty-regularized objective. Both variants are fine-tuned from a full-attention model (the baseline) via warm-start, preserving the backbone's manipulation capability. On the MimicGen Square task, compared to this baseline, CoDA-S reduces visual tokens by* ***50.7%*** *without degrading trajectory accuracy, and CoDA-A reduces visual tokens by* ***25.8%*** *while* improving *GT MSE by* $\sim$***30%*** *(0.51 $\rightarrow$ 0.36). Eliminating unnecessary tokens suppresses attention noise, thereby improving trajectory accuracy itself. Our results demonstrate that adaptive sparsity can simultaneously achieve computational reduction and trajectory accuracy improvement in 7-DoF robotic manipulation.*

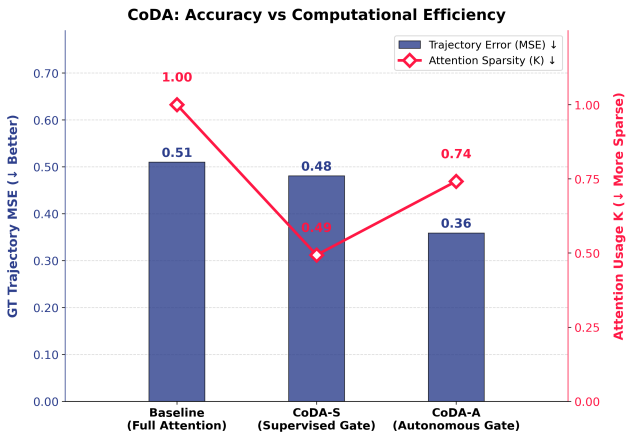

Figure 1. **CoDA achieves a Pareto improvement over the full-attention Baseline on MimicGen Square.** Blue bars (left axis, ↓) show GT Trajectory MSE; the red diamond line (right axis, ↓) shows average token usage K (lower = more sparse). CoDA-S halves the token budget (K=0.49) with negligible accuracy loss. CoDA-A simultaneously reduces MSE by $\approx$29% *and* cuts token usage by 25.8%—demonstrating that task-guided sparsity suppresses attention noise and improves trajectory accuracy rather than merely saving compute.

1. Introduction

Large-scale pretrained world models such as NVIDIA Cosmos Predict 2.5 [1] learn causal relationships of the world from video data, providing rich visual priors for visuomotor policy learning in robotic manipulation. Unlike Vision-Language-Action (VLA) models [6, 7, 18, 25] that rely on static image-text data and must implicitly infer physical dynamics, world models directly learn visual dynamics from video, enabling more generalizable manipulation capabilities. Our objective addresses a distinct bottleneck compared to large-scale distributed fine-tuning frameworks such as Cosmos Policy [19]: rather than scaling compute for offline policy generation, we focus on enabling single-GPU real-time deployment by efficiently leveraging the learned visual dynamics priors of world models. Yet this power carries an unresolved fundamental question: *when does a robot*

need more attention? No existing method has directly addressed this question.

In practice, only a small fraction of tokens in the attention matrix contribute meaningfully to the output [5, 21, 30]. A simple reaching motion and a precise grasping maneuver require fundamentally different levels of computation. Yet full attention ignores this distinction, forcing identical computation at every moment. More fundamentally, unnecessary tokens dilute attention scores, preventing the model from focusing on the moments that truly matter.

Existing sparse attention methods rely on fixed patterns or post-hoc token dropping, without considering the temporal complexity variations of robotic manipulation or leveraging domain-specific signals. They amount to reducing computation blindly, without knowing when or where to focus.

We answer this question by drawing inspiration from the FAST tokenizer [28], which demonstrated that high-frequency DCT components of action sequences serve as a reliable proxy for action complexity. We translate this frequency-domain insight into an architectural mechanism: a per-chunk dynamic token budget governing the cross-attention layers of Cosmos Predict 2.5. Crucially, CoDA-A goes beyond a fixed heuristic mapping—it treats DCT-derived spectral features as learnable inputs to a gating network optimized end-to-end against trajectory loss, enabling the model to distinguish genuine action complexity from high-frequency noise artifacts. We introduce two gating variants: CoDA-S, a supervised gate, and CoDA-A, an autonomous gate.

Our main contributions are summarized as follows:

- We propose the **CoDA** framework, which leverages DCT-based action complexity as a dynamic token budget to alleviate the computational bottleneck of world models in real-time robotic control.
- We introduce two gating variants—**CoDA-S** (supervised) and **CoDA-A** (autonomous)—and show that CoDA-A's data-driven optimization not only surpasses the supervised variant but also suppresses attention noise, improving trajectory accuracy beyond the full-attention baseline.
- We empirically demonstrate on the MimicGen Square task that adaptive sparsity can simultaneously reduce visual token usage and improve trajectory accuracy—a **Pareto improvement** over the full-attention baseline.

2. Related Work

2.1. Visuomotor Control via World Models

The paradigm of robotic manipulation has recently shifted toward large-scale generative architectures. Beyond traditional VLAs, generative policies such as Diffusion Policy [10] and video generation models [2, 14, 20] are increasingly repurposed as universal policies. Generative world models such as Genie [8] and large-scale video generators [26] further demonstrate that video generation can serve as an interactive simulator of physical dynamics, motivating their use as robot policy backbones. Flow-matching-based action policies such as π_0 [4] achieve remarkable generalization across dexterous, high-frequency tasks. World-model-based approaches such as Cosmos Policy [19], fine-tuned from the NVIDIA Cosmos Predict 2.5 foundation model [1], demonstrate powerful planning and manipulation capabilities by modeling causal visual dynamics. However, these large-scale models typically assume multi-GPU or cloud computing environments. For real-world deployment, especially closed-loop control on a single GPU, the inference latency caused by computing full attention across all timesteps remains a critical bottleneck. While recent works such as DiffuserLite [12] address inference speed by reducing diffusion sampling steps, and transformer-based policies such as Q-Transformer [9] highlight the computational demands of sequential attention in robotic control, the fundamental bottleneck of the cross-attention mechanism itself remains unaddressed. Our work seeks to bridge this gap, enabling the deployment of world-model-level intelligence within strict real-time constraints.

2.2. Sparse Attention and Dynamic Routing

Since the advent of Vision Transformers [13], the quadratic complexity of self-attention has spurred numerous efficiency methods, ranging from hardware-aware exact attention [11] to structural sparsity. Fixed pattern approaches like Longformer [3] and Swin Transformer [23] restrict attention to local windows. Token merging methods such as ToMe [5] and adaptive token sampling methods such as ATS [15] reduce FLOPs by combining or dropping tokens based on visual similarity or attention weights. While effective in static vision tasks, these methods rely on fixed patterns or visual similarities, critically ignoring the temporal complexity and action-specific demands inherent in robotic control, such as action chunking [16, 31].

Another line of research employs learnable sparsity via dynamic token routing and penalty regularization. Methods such as DynamicViT [30] and EViT [21] learn to progressively drop inattentive tokens by optimizing a computation-performance trade-off with sparsity penalties. However, these general-purpose penalties optimize for generic attention entropy rather than task-specific accuracy: tokens may be dropped precisely during the most demanding manipulation phases, where computational savings are most costly. CoDA overcomes this by introducing an accuracy-aware penalty directly tied to the trajectory loss $\mathcal{L}_{action}$, ensuring sparsity is pursued only when trajectory accuracy is sufficiently high.

2.3. Frequency Analysis for Action Representation

Analyzing physical actions in the frequency domain offers a robust way to disentangle structural trends from fine-grained details [17, 29]. The FAST [28] framework demonstrated that applying the Discrete Cosine Transform (DCT) to action sequences effectively encodes action complexity, proving that high-frequency components are crucial for dexterous control tasks. However, FAST uses this insight solely for offline tokenization during VLA training. In contrast, CoDA reinterprets DCT high-frequency energy as a dynamic per-chunk token budget for the cross-attention mechanism—directly linking action complexity to visual attention allocation at inference time. Critically, CoDA-A goes further: rather than using DCT features as a fixed heuristic, it treats them as learnable inputs optimized end-to-end against the trajectory loss, enabling the model to distinguish genuine action complexity from high-frequency noise artifacts.

3. Methodology

To address the computational bottleneck of applying full-attention world models to real-time robotic control, we propose **CoDA** (**Co**smos-driven **D**CT-**A**daptive Sparse Attention). Our core philosophy is that a robot's computational load should dynamically scale with the physical complexity of its actions. In this section, we first mathematically formulate action complexity using Discrete Cosine Transform (DCT) (Sec. 3.1), introduce the dynamic token budgeting mechanism in cross-attention (Sec. 3.2), and present our two adaptive gating variants: CoDA-S and CoDA-A (Sec. 3.3). Finally, we describe our progressive warm-start fine-tuning strategy (Sec. 3.4).

3.1. Quantifying Action Complexity via DCT

Not all segments of a robotic trajectory require the same level of visual reasoning. A steady translational motion represents a simple, low-frequency signal, whereas a precise contact-rich manipulation (e.g., insertion or grasping) exhibits rapid variations and high-frequency patterns. Drawing inspiration from FAST [28], we employ the unnormalized 1D Discrete Cosine Transform (DCT) to mathematically quantify this "action complexity."

Given an action chunk $A \in \mathbb{R}^{T\times D}$ where T is the time horizon and D is the action dimension—computed from ground-truth action chunks during training and from the previous chunk's predicted actions (or teacher-forced targets) during inference-time gating—we transform the temporal sequence of each dimension d into the frequency domain:

$$\mathcal{C}_d(k) = \sum_{n=0}^{T-1} A_d(n) \cos\left(\frac{\pi(2n+1)k}{2T}\right), \qquad (1)$$

where $k \in \{0, \dots, T-1\}$ denotes the frequency index.

We hypothesize that the high-frequency components contain the critical variations that demand acute visual attention. Thus, we define the action complexity score ϕ as the ratio of high-frequency spectral magnitude to the total spectral magnitude, summed across all action dimensions:

$$\phi = \frac{\sum_{d=1}^{D}\sum_{k=\lfloor T/2\rfloor}^{T-1} |\mathcal{C}_d(k)|}{\sum_{d=1}^{D}\sum_{k=0}^{T-1} |\mathcal{C}_d(k)| + \epsilon}, \qquad (2)$$

Following common frequency-band partitioning in prior action-frequency analyses, we use the Nyquist midpoint as the boundary and treat indices $k \geq \lfloor T/2\rfloor$ as high-frequency components, which capture rapid temporal variations in action trajectories. Here, ϵ is a small constant for numerical stability. Since both numerator and denominator share the same per-dimension scale, the ratio remains invariant to the absolute magnitude of each action dimension. A higher ϕ indicates a complex maneuver requiring denser visual context, while a lower ϕ implies a predictable motion where visual attention can be safely sparsified. Although ϕ is computed from ground-truth actions during training and from model predictions at inference, this train–inference distribution shift is mitigated by CoDA-A's end-to-end optimization, which directly trains the gating network on predicted spectral features.

3.2. Dynamic Cross-Attention via Token Budgeting

We build our policy upon the Cosmos Predict 2.5 DiT architecture [1, 27]. The fundamental computational bottleneck lies in the cross-attention mechanism, where action query tokens Q attend to a dense set of visual key tokens K_v, V_v:

$$\text{Attn}(Q, K_v, V_v) = \text{softmax}\left(\frac{QK_v^\top}{\sqrt{d}}\right) V_v \qquad (3)$$

Instead of aggregating over all N visual tokens, CoDA introduces a dynamic *token budget* $K_{budget} \in [0, 1]$. After computing attention scores, we retain only the top-$\lceil K_{budget} \times N\rceil$ visual tokens by masking the rest to $-\infty$ before the softmax. This confines value aggregation to the most relevant visual tokens, reducing the effective context each action query must integrate. The degree of sparsity is determined per-chunk by the complexity score ϕ, directly linking action complexity to visual attention allocation. While this masking theoretically reduces the computation required for value aggregation, realizing strict hardware-level acceleration requires specialized sparse kernels; this limitation is further discussed in Sec. 5.

3.3. Adaptive Gating: CoDA-S and CoDA-A

To determine K_{budget} dynamically during inference, we design a lightweight **Gating Head** $\mathcal{G}$. It is a 3-layer MLP that extracts four DCT-derived spectral features from the

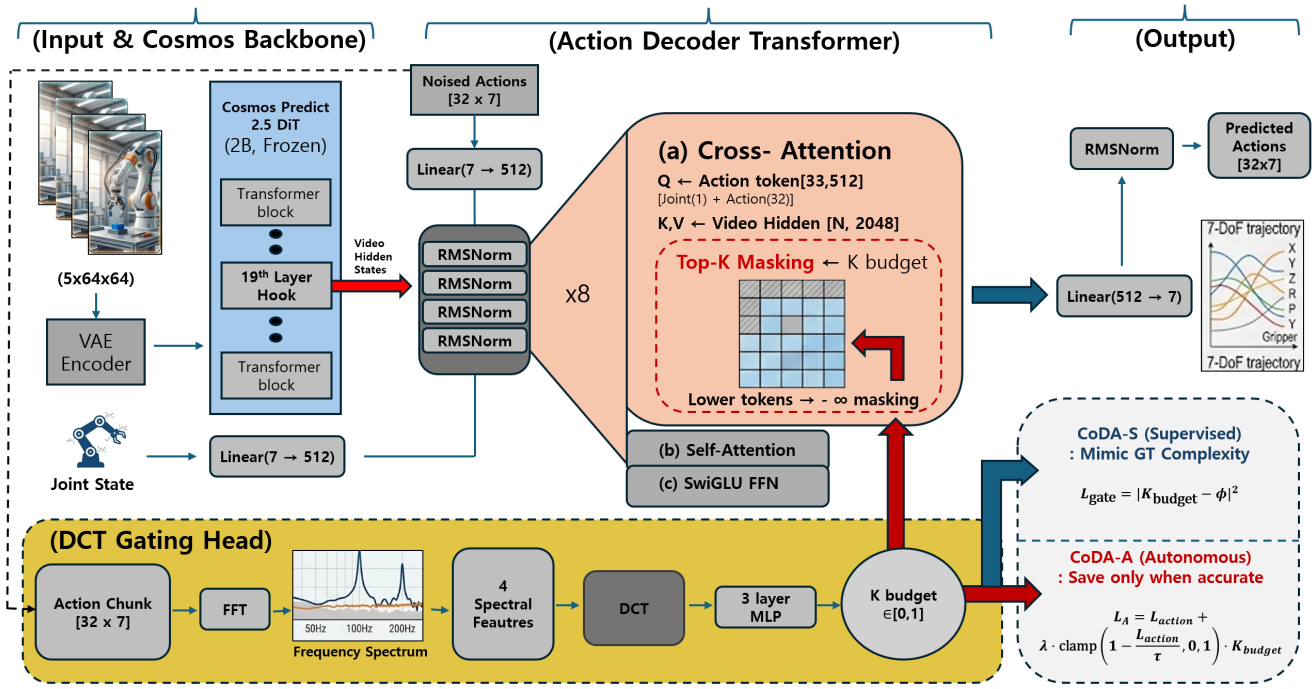

Figure 2. **Overview of the CoDA framework.** *(Left)* The frozen Cosmos Predict 2.5 backbone extracts video hidden states from 5-frame RGB observations via its 19th-layer hook. Joint state is embedded and concatenated with noised action tokens as queries to the Action Decoder. *(Center)* The Action Decoder Transformer (8 layers) applies (a) Cross-Attention with dynamic Top-K token masking, (b) Self-Attention, and (c) SwiGLU FFN at each layer. In cross-attention, action query tokens attend to visual key-value tokens; CoDA retains only the top-$\lceil K_{budget} \times N \rceil$ tokens by masking the rest to $-\infty$, concentrating attention on task-relevant visual features. *(Bottom)* The DCT Gating Head extracts four spectral features from the current action chunk via FFT and DCT, and predicts $K_{budget} \in [0, 1]$ through a 3-layer MLP. CoDA-S (supervised) trains the gate to mimic ground-truth DCT complexity ϕ; CoDA-A (autonomous) optimizes K_{budget} directly via an accuracy-aware sparsity penalty, reducing tokens only when trajectory accuracy is guaranteed. *(Right)* The decoder predicts a 7-DoF action trajectory (32×7).

available action chunk signal at each step (ground truth in training; model-generated chunk signal in inference)—total spectral magnitude, high-frequency ratio (i.e., ϕ), dominant frequency index, and spectral entropy—and predicts the token budget $K_{budget} = \mathcal{G}(f_{DCT}) \in [0, 1]$. We propose two distinct learning paradigms for this gating mechanism.

CoDA-S (Supervised Gating). In the supervised variant, we enforce the gating head to explicitly mimic the human-understandable physical complexity. Using the offline dataset, we pre-calculate the ground-truth complexity ϕ for every action chunk. The gate is then trained using a standard Mean Squared Error (MSE) objective:

$$\mathcal{L}_{gate} = \|K_{budget} - \phi\|^2 \quad (4)$$

This ensures the model reliably reduces computation during low-frequency motions, grounding the sparsity in pure physical kinematics. In practice, the predicted K_{budget} is discretized to one of three hardware-aligned token counts $\{16, 32, 64\}$ to satisfy the block-alignment requirements of the Triton sparse attention kernel (thresholds: $\phi < 0.15 \rightarrow 16$, $\phi < 0.25 \rightarrow 32$, otherwise $\rightarrow 64$). These thresholds were selected to partition the empirical ϕ distribution of the training set into three roughly equal-frequency bins, ensuring balanced coverage across low-, mid-, and high-complexity chunks.

CoDA-A (Autonomous Gating). While CoDA-S relies on fixed kinematic heuristics, CoDA-A optimizes the accuracy–efficiency trade-off autonomously. We introduce an *Accuracy-aware Sparsity Penalty* that forces the model to reduce visual tokens only when the trajectory accuracy is guaranteed:

$$\mathcal{L}_A = \mathcal{L}_{action} + \lambda \cdot \text{clamp}\left(1 - \frac{\mathcal{L}_{action}}{\tau}, 0, 1\right) \cdot K_{budget} \quad (5)$$

Here, $\mathcal{L}_{action}$ is the flow-matching trajectory loss [4, 22]. We set τ=0.3 based on the converged Stage 0 validation-loss scale, and use a small λ=0.01 so that the sparsity term regularizes rather than dominates the primary trajectory objective; both were selected empirically based on val-

idation stability and accuracy–efficiency trade-offs. When $\mathcal{L}_{action} > \tau$, the gate term vanishes and the model focuses solely on accuracy recovery. When $\mathcal{L}_{action} < \tau$, the penalty is activated and gradients flow directly through K_{budget} to the Gating Head, rewarding reduced token usage. The small λ and the clamped gate prevent abrupt oscillations, yielding stable convergence. As demonstrated in our experiments, this data-driven optimization not only saves computation but also suppresses attention noise, improving trajectory accuracy.

3.4. Progressive Warm-Start Fine-tuning

Training a 2B parameter foundation model with dynamic sparsity from scratch is highly susceptible to representation collapse. To stabilize the learning process, we propose a progressive warm-start strategy. In **Stage 0**, we fine-tune the pretrained Cosmos Predict 2.5 weights using full attention ($K_{budget} = 1.0$) to establish a robust visuomotor policy baseline. In **Stage 1**, we initialize from the Stage 0 checkpoint and attach the Gating Head. The Cosmos backbone remains frozen throughout; only the action decoder and Gating Head are fine-tuned. This curriculum ensures the model first acquires the dense visuomotor priors necessary for manipulation before learning to sparsify them.

4. Experiments

4.1. Experimental Setup

Task and Dataset. We evaluate CoDA on the MimicGen Square task [24], a 7-DoF robotic manipulation benchmark requiring a Franka Panda arm to pick up a square nut and place it onto a fixed peg. We use 3,000 automatically generated demonstrations for training and 100 held-out demonstrations for evaluation, with a 90/10 train/validation split.

Architecture and Training. Our backbone is Cosmos Predict 2.5 [1], a 2B-parameter Diffusion Transformer (DiT) pretrained on large-scale video data. In **Stage 0**, we fine-tune the pretrained backbone with full attention ($K_{budget} = 1.0$) for 200 epochs using a learning rate of 2×10^{-5} with AdamW optimizer and cosine annealing, establishing the *Baseline* (our Stage 0 full-attention model; see Sec. 3.4). In **Stage 1**, we initialize from the Stage 0 checkpoint, attach the Gating Head, freeze the Cosmos backbone, and fine-tune only the action decoder and Gating Head for an additional 200 epochs. All experiments are conducted on a single NVIDIA RTX 6000 GPU using bfloat16 precision.

Evaluation Metrics. We report four complementary metrics:

Table 1. Quantitative comparison on MimicGen Square. GT MSE measures trajectory accuracy against ground truth (↓). S0 Fidelity MSE measures deviation from the full-attention baseline (↓). Avg K indicates the average token budget ratio (↓ = more sparse). Token Reduction is computed as $(1 - \text{Avg K}) \times 100\%$.

Model	GT MSE ↓	S0 Fidelity ↓	Avg K ↓	Token Red.
Baseline	0.51	0.00	1.00	—
CoDA-S	0.48	0.21	0.49	50.7%
CoDA-A	**0.36**	0.20	0.74	25.8%

- **GT MSE** (↓): Mean Squared Error between the predicted and ground-truth action trajectories, measuring physical trajectory accuracy.
- **S0 Fidelity MSE** (↓): MSE between the sparse model's output and the full-attention baseline's output, quantifying how much intelligence is lost due to sparsification.
- **Avg K** (↓): Average token budget ratio across all evaluation chunks. Lower values indicate greater sparsity and computational savings.
- **Per-DoF MSE**: MSE decomposed across each of the 7 action dimensions (X, Y, Z, Roll, Pitch, Yaw, Gripper), enabling fine-grained error analysis.

All MSE values are reported in per-dimension Z-score normalized action space (zero mean, unit standard deviation computed from the training set). Reported values therefore reflect relative model comparisons rather than absolute physical error magnitudes.

4.2. Main Results

Table 1 and Figure 1 present our main quantitative results. We highlight three key findings.

CoDA-A achieves a Pareto improvement. CoDA-A reduces GT MSE from 0.51 to **0.36** while also lowering Avg K to 0.74 in this setting, indicating a favorable accuracy-efficiency trade-off. In Figure 1, both the bar and the line move downward relative to the Baseline, confirming that the accuracy gain is not achieved at the cost of efficiency. This demonstrates that adaptive sparsity is not merely a cost-saving measure: eliminating low-relevance tokens actively suppresses attention noise, enabling the model to concentrate on task-critical visual features and thereby improving trajectory accuracy.

CoDA-S maximizes token reduction. CoDA-S achieves the most aggressive sparsification, reducing the average token budget to K=0.49 (50.7% reduction) while maintaining GT MSE at 0.48—comparable to the full-attention Baseline (0.51). The supervised gate reliably maps low-frequency, kinematically simple motions to high sparsity, grounding token reduction in interpretable physical signals.

Sparse models preserve structural fidelity. Both CoDA-S and CoDA-A exhibit an S0 Fidelity MSE of ≈0.20, indicating that despite operating with fewer tokens, their output trajectories remain structurally consistent with the full-attention Baseline. The warm-start strategy (Sec. 3.4) is critical: by inheriting dense visuomotor priors from Stage 0, neither sparse model experiences representation collapse during Stage 1 fine-tuning.

4.3. Per-DoF Error Analysis

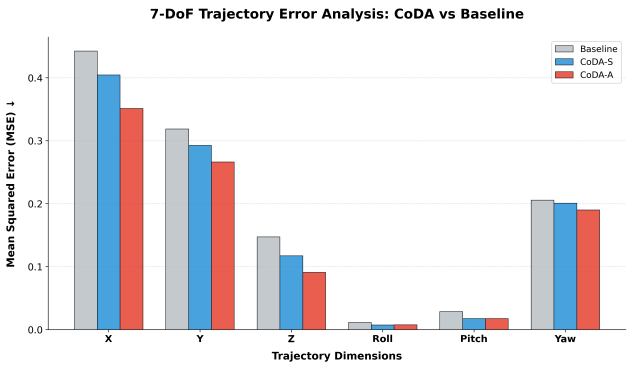

Figure 3. **Per-DoF MSE decomposition across 6 continuous action dimensions.** CoDA-A (red) reduces error over the Baseline (gray) across all translational (X, Y, Z) and rotational (Roll, Pitch, Yaw) axes. The improvement is most pronounced in the Z-axis (0.15→0.09, ≈40%), which governs vertical alignment during nut insertion—the most contact-rich phase. Roll and Pitch show smaller absolute gains consistent with their inherently low variance. The Gripper dimension is excluded: its binary open/close signal makes continuous MSE an uninformative metric.

Figure 3 decomposes trajectory error across the six continuous action dimensions. Two patterns stand out.

Consistent improvement across all axes. CoDA-A reduces MSE over the Baseline in every dimension without exception (X: 0.44→0.35, Y: 0.32→0.27, Z: 0.15→0.09, Roll: 0.011→0.008, Pitch: 0.030→0.017, Yaw: 0.21→0.19). This consistency rules out the possibility that the aggregate MSE gain in Sec. 4.2 is driven by a single outlier dimension.

Largest gains in contact-rich axes. The Z-axis—which controls vertical positioning during the critical nut-insertion phase—shows the largest relative improvement (≈40%). We attribute this to CoDA-A dynamically allocating more tokens during high-frequency, contact-rich moments (as further evidenced by the attention maps in Sec. 4.6), where precise depth control matters most. Roll and Pitch exhibit smaller absolute errors overall, and their improvements, while consistent in direction, are accordingly smaller in magnitude. The Gripper channel is omitted from this figure: its binary open/close control signal renders continuous MSE an uninformative proxy for grasping performance.

Table 2. Ablation on gating strategy and temporal granularity under a fixed cosine LR schedule for fair comparison (Table 1 reports best-performing checkpoints). **Fixed-K**: uniform static sparsity at K=0.5 with no dynamic adjustment. **Episode-dynamic**: a single token budget assigned per full episode based on episode-level DCT complexity. **Chunk-dynamic**: per-chunk dynamic budget using the pre-computed GT DCT k-score directly as K (no learned gating head). **CoDA-S/A**: our proposed DCT-guided gating variants (Sec. 3.3).

Model	GT MSE ↓	Avg K ↓
Baseline (Full Attention)	0.652	1.000
Fixed-K (K=0.5, static)	0.800	0.500
Episode-dynamic	1.059	0.172
Chunk-dynamic (no learned gate)	0.747	0.172
CoDA-S (Ours)	**0.443**	0.501
CoDA-A (Ours)	0.492	0.698

4.4. Ablation Study: Architectural Choices for Dynamic Gating

While Table 1 reports best-performing checkpoints to demonstrate peak capability, this ablation evaluates all variants under a fixed, identical training schedule to isolate the effect of each architectural choice. Table 2 reveals a clear three-step progression.

Step 1: Necessity of dynamic gating. Applying static sparsification (Fixed-K, K=0.5) severely degrades trajectory accuracy (GT MSE=0.800) relative to the full-attention Baseline (0.652), despite using the same token budget as CoDA-S. Uniformly dropping tokens destroys task-critical visual priors at precisely the moments demanding precise reasoning.

Step 2: Necessity of chunk-level temporal granularity. Episode-dynamic gating—which assigns a single token budget per full episode—performs worst of all (GT MSE=1.059, exceeding even the Baseline), because a globally underestimated budget forces the model to operate with severe sparsity throughout contact-rich insertion phases. Finer-grained per-chunk budgeting (Chunk-dynamic) reduces this error to 0.747, confirming that action complexity fluctuates rapidly and requires localized temporal resolution.

Step 3: Effectiveness of DCT-guided gating signals. Even with chunk-level granularity, the naive Chunk-dynamic variant remains inferior to the Baseline. Replacing the naive signal with our DCT-derived complexity score (CoDA-S) dramatically improves accuracy to GT MSE=0.443—below the Baseline—at a comparable token budget (Avg K≈0.50). This confirms that the high-frequency DCT signal is the critical factor enabling the

model to retain tokens precisely when and where precise reasoning is required.

4.5. Training Dynamics

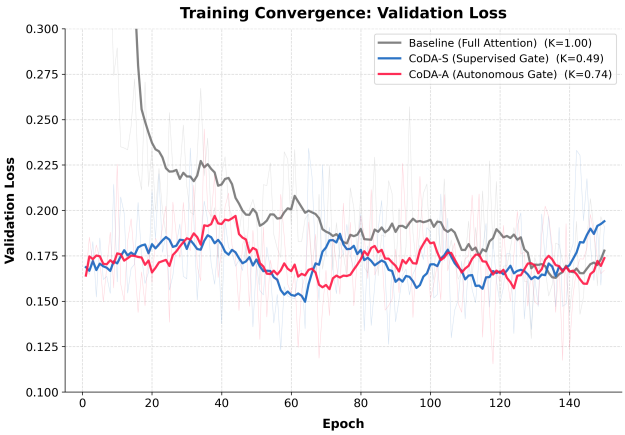

Figure 4. **Validation loss convergence for all three models.** The Baseline (gray) starts from pretrained Cosmos weights at loss ≈0.30 and requires ∼60 epochs to settle. CoDA-S (blue) and CoDA-A (red) inherit the Stage 0 checkpoint and begin Stage 1 already near loss ≈0.17—a direct consequence of the warm-start strategy. Both sparse models converge stably without loss spikes, confirming that freezing the Cosmos backbone and fine-tuning only the action decoder and Gating Head prevents representation collapse.

Figure 4 compares the validation loss trajectories of all three models.

Warm-start eliminates cold-start cost. The Baseline (Stage 0) must build visuomotor priors from scratch atop the video-pretrained Cosmos weights, starting at a high initial loss (≈0.30) and requiring roughly 60 epochs before settling. CoDA-S and CoDA-A bypass this cold-start phase entirely: inheriting the Stage 0 checkpoint, both models enter Stage 1 with an already-low validation loss (≈0.17) and maintain stable convergence throughout.

No representation collapse under sparsification. Despite operating with reduced token budgets from the very first epoch of Stage 1, neither sparse model exhibits loss spikes or divergence. This stability arises from our architectural choice to freeze the Cosmos backbone and fine-tune only the lightweight action decoder and Gating Head, ensuring that the dense visuomotor priors established in Stage 0 are preserved intact while the gating mechanism is learned on top.

4.6. Qualitative Analysis: Attention Maps

Figure 5 shows Layer 3 cross-attention heatmaps across key manipulation stages.

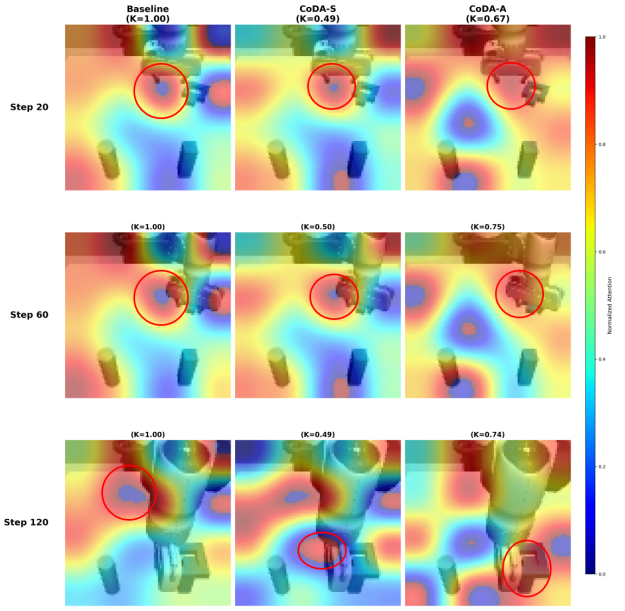

Figure 5. **Cross-attention heatmaps (Layer 3) at three manipulation stages.** Each row corresponds to a timestep (Step 20: approach, Step 60: contact, Step 120: insertion). Each column shows the attention distribution for Baseline (K=1.0), CoDA-S (K=0.49), and CoDA-A (K=0.74). Red circles mark the task-critical nut-peg contact zone. **Left (Baseline):** Attention is broadly diffused across the visual field throughout all stages, including task-irrelevant background regions. **Center (CoDA-S):** Attention is more concentrated near the manipulation workspace, though some diffuse patterns remain. **Right (CoDA-A):** At Step 120 (insertion), attention sharpens most strongly onto the nut-peg contact zone, while K rises dynamically to accommodate the increased complexity—demonstrating that CoDA-A allocates more tokens precisely when and where they matter.

Baseline exhibits attention diffusion. With full token access (K=1.0), the Baseline distributes attention broadly across the scene throughout all stages, including background regions irrelevant to the task. This attention dilution is the mechanism behind its higher trajectory error: softmax normalization forces the model to aggregate signal from noisy, task-irrelevant tokens alongside meaningful ones.

CoDA-A acts as an implicit attention regularizer. By pruning low-relevance tokens, CoDA-A constrains the attention distribution to focus on the task-critical zone. Critically, the dynamic nature of CoDA-A's gating is visible here: at the demanding insertion step (Step 120), the gate raises K to retain more tokens, producing the sharpest heatmap concentration of all three models. This is the mechanism underlying the Pareto improvement observed in Sec. 4.2: removing noisy tokens does not merely save compute—it forces the attention mechanism to find sharper, more informative alignments, directly improving trajectory

prediction.

5. Conclusion and Future Work

We presented **CoDA**, a DCT-adaptive sparse attention framework that dynamically allocates visual token budgets in proportion to robotic action complexity. By repurposing DCT high-frequency energy—originally proposed by FAST for offline tokenization—as a per-chunk gating signal in the cross-attention layers of Cosmos Predict 2.5, CoDA bridges the gap between world-model-level intelligence and real-time deployment constraints. Our two gating variants reveal a fundamental insight: sparsity, when guided by task-relevant signals, is not merely a computational shortcut but an implicit regularizer. CoDA-S reduces token usage by 50.7% while preserving trajectory accuracy, and CoDA-A achieves a Pareto improvement—simultaneously cutting tokens by 25.8% and improving GT MSE by $\sim$30%—by learning to suppress attention noise from task-irrelevant visual regions.

Limitations. Current experiments are limited to a single simulated manipulation task (MimicGen Square). Due to the substantial computational cost of fine-tuning the 2B-parameter backbone, our evaluation relies on a single training run; multi-task generalization and multi-seed variance are left for future work. Furthermore, while CoDA significantly reduces visual tokens, strict wall-clock inference speedup has not yet been demonstrated. Although CoDA-S already discretizes token budgets into hardware-aligned block sizes ($\{16, 32, 64\}$) to satisfy Triton kernel requirements, end-to-end kernel integration and wall-clock latency validation remain an immediate next step.

Future Work. Immediate next steps include validating Triton kernel integration on RTX 6000 Ada (SM_89) to demonstrate concrete latency reduction, and extending CoDA to multi-task and real-robot settings. Longer-term directions include exploring per-query token budgets for finer-grained adaptive attention, and applying the CoDA framework to other large-scale world models beyond Cosmos Predict 2.5.

References

[1] Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, et al. Cosmos world foundation model platform for physical AI. *arXiv preprint arXiv:2501.03575*, 2025. 1, 2, 3, 5

[2] Anurag Ajay et al. Is conditional video generation a good object-centric policy? In *ICLR*, 2024. 2

[3] Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020. 2

[4] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024. 2, 4

[5] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your ViT but faster. In *The Eleventh International Conference on Learning Representations*, 2023. 2

[6] 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. In *Robotics: Science and Systems*, 2022. 1

[7] 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. In *Conference on Robot Learning*, 2023. 1

[8] Jake Bruce, Michael Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, et al. Genie: Generative interactive environments. In *Proceedings of the 41st International Conference on Machine Learning*, 2024. 2

[9] Yevgen Chebotar, Quan Vuong, Karol Hausman, Fei Xia, Yao Lu, Alex Irpan, Aviral Kumar, Tianhe Yu, Alexander Herzog, Karl Pertsch, Keerthana Gopalakrishnan, Julian Ibarz, Sean Kirmani, Kanishka Rao, Paul Xu, Dmitry Kalashnikov, Chelsea Finn, and Sergey Levine. Q-Transformer: Scalable offline reinforcement learning via autoregressive Q-functions. In *Conference on Robot Learning*, 2023. 2

[10] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Robotics: Science and Systems*, 2023. 2

[11] Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems*, pages 16344–16359, 2022. 2

[12] Zibin Dong, Jianye Hao, Yifu Yuan, Fei Ni, Yitian Mu, Yan Li, and Yujing Zheng. Diffuserlite: Towards real-time diffusion planning. In *Advances in Neural Information Processing Systems*, 2024. 2

[13] 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. In *International Conference on Learning Representations*, 2021. 2

[14] Y Du et al. Learning universal policies via text-guided video generation. In *Advances in Neural Information Processing Systems*, 2023. 2

[15] Mohsen Fayyaz, Soroush Abbasi Koohpayegani, Farnoush Rezaei Jafari, Sunando Sengupta, Hamid Reza Vaezi Joze, Eric Sommerlade, Hamed Pirsiavash, and Juergen Gall. ATS: Adaptive token sampling for efficient vision transformers. In *European Conference on Computer Vision*, 2022. 2

[16] Zipeng Fu, Tony Z. Zhao, and Chelsea Finn. Mobile ALOHA: Learning bimanual mobile manipulation with low-cost whole-body teleoperation. In *Conference on Robot Learning*, 2024. 2

[17] Lionel Guéguen, Alex Sergeev, Ben Kadlec, Rosanne Liu, and Jason Yosinski. Faster neural networks straight from JPEG. In *Advances in Neural Information Processing Systems*, 2018. 3

[18] Moo Jin Kim, Karl Fang, Ted Xiao, et al. OpenVLA: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024. 1

[19] Moo Jin Kim, Yihuai Gao, Tsung-Yi Lin, Yen-Chen Lin, Yunhao Ge, Grace Lam, Percy Liang, Shuran Song, Ming-Yu Liu, Chelsea Finn, and Jinwei Gu. Cosmos policy: Fine-tuning video models for visuomotor control and planning. *arXiv preprint arXiv:2601.16163*, 2026. 1, 2

[20] Junbang Liang, Pavel Tokmakov, Ruoshi Liu, Sruthi Sudhakar, Paarth Shah, Rares Ambrus, and Carl Vondrick. Video generators are robot policies. *arXiv preprint arXiv:2508.00795*, 2025. 2

[21] Youwei Liang, Chongjian Ge, Zhan Tong, Yang Song, Jianping Wang, and Pengtao Xie. Not all patches are what you need: Expediting vision transformers via token reorganizations. In *International Conference on Learning Representations*, 2022. 2

[22] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022. 4

[23] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 2

[24] Ajay Mandlekar, Soroush Nasiriany, Bowen Wen, Iretiayo Akinola, Yashraj Narang, Linxi Fan, Yuke Zhu, and Dieter Fox. MimicGen: A data generation system for scalable robot learning using human demonstrations. In *Proceedings of The 7th Conference on Robot Learning*, pages 1820–1864, 2023. 5

[25] Octo Model Team et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024. 1

[26] OpenAI. Video generation models as world simulators. *OpenAI Technical Report*, 2024. 2

[27] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4199–4209, 2023. 3

[28] Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. FAST: Efficient action tokenization for vision-language-action models. *arXiv preprint arXiv:2501.09747*, 2025. 2, 3

[29] Zequn Qin, Pengyi Zhang, Fei Wu, and Xi Li. FcaNet: Frequency channel attention networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 783–792, 2021. 3

[30] Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. DynamicViT: Efficient vision transformers with dynamic token sparsification. In *Advances in Neural Information Processing Systems*, pages 13937–13949, 2021. 2

[31] Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *Robotics: Science and Systems*, 2023. 2