See What Matters

Differentiable Grid Sample Pruning for Generalizable Vision-Language-Action (VLA) Models

ICML 2026

Yixu Feng · Zinan Zhao · Yanxiang Ma · Chenghao Xia · Chengbin Du · Yunke Wang · Chang Xu

The University of Sydney | City University of Hong Kong | StellarEdge Robotics

The Efficiency Bottleneck in VLA Models

High Computational Cost

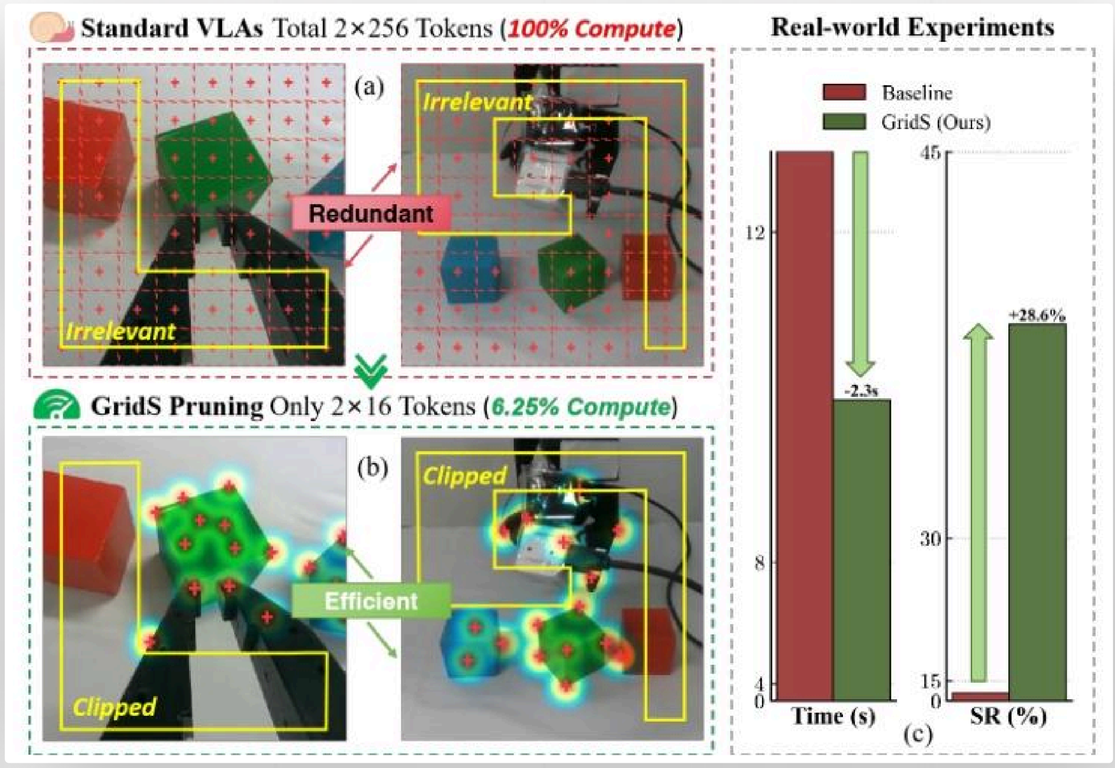
VLA models split images into hundreds of patches (e.g., 256 tokens). The Transformer's quadratic complexity ($O(N^2)$) creates a severe computational bottleneck.

The Speed-Accuracy Trade-off

- **Aggressive Pruning:** Discards too many tokens, losing critical details (e.g., grasp points).
- **Conservative Pruning:** Retains too many tokens, providing negligible speedup.

The Root Cause

Current methods rely on **discrete selection** over a fixed grid, introducing quantization errors and losing fine-grained spatial fidelity.



Dense Tokenization in Standard VLAs

A typical image is divided into a grid of small patches (tokens). Many tokens contain redundant information.

Our Solution: From Discrete Pruning to Continuous Resampling

Core Idea

We reformulate VLA token compression from a passive, discrete patch-dropping task into an **active, geometry-aware continuous resampling process**.

What is GridS?

⚡ **Plug-and-Play Module**
A lightweight, easily integrable component.

🎯 **Task-Aware**
Dynamically predicts salient spatial coordinates based on context.

🔄 **Continuous Sampling**
Queries continuous feature maps via differentiable bilinear interpolation.

Key Advantages

📐 **Preserves Geometry**
Extracts features at sub-patch precision.

🚀 **Drastic Compression**
Achieves high compression rates (e.g., <10% tokens).

🧩 **End-to-End Trainable**
Fully differentiable for joint optimization.

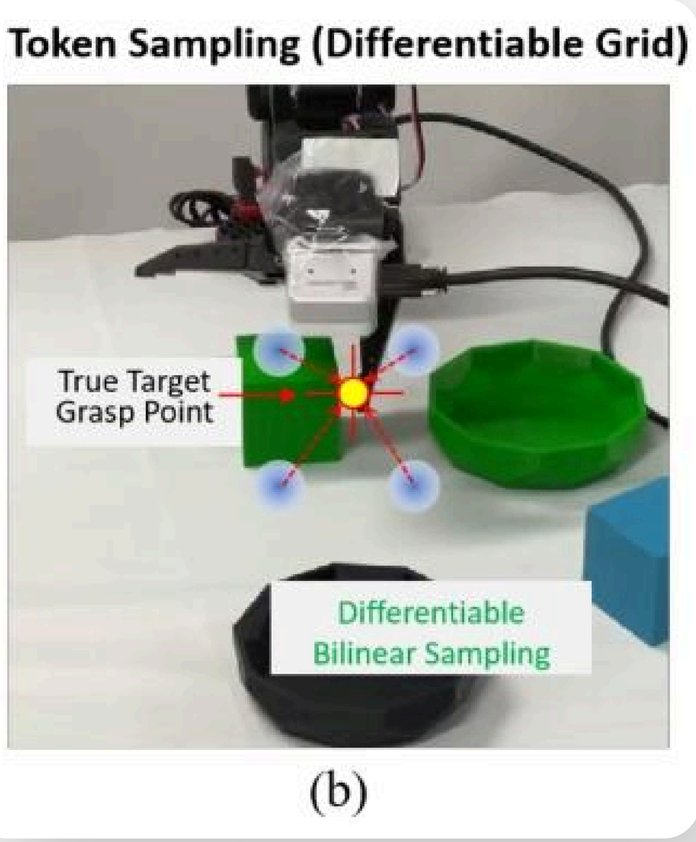


Figure: Discrete Patch Selection vs. Differentiable Continuous Resampling

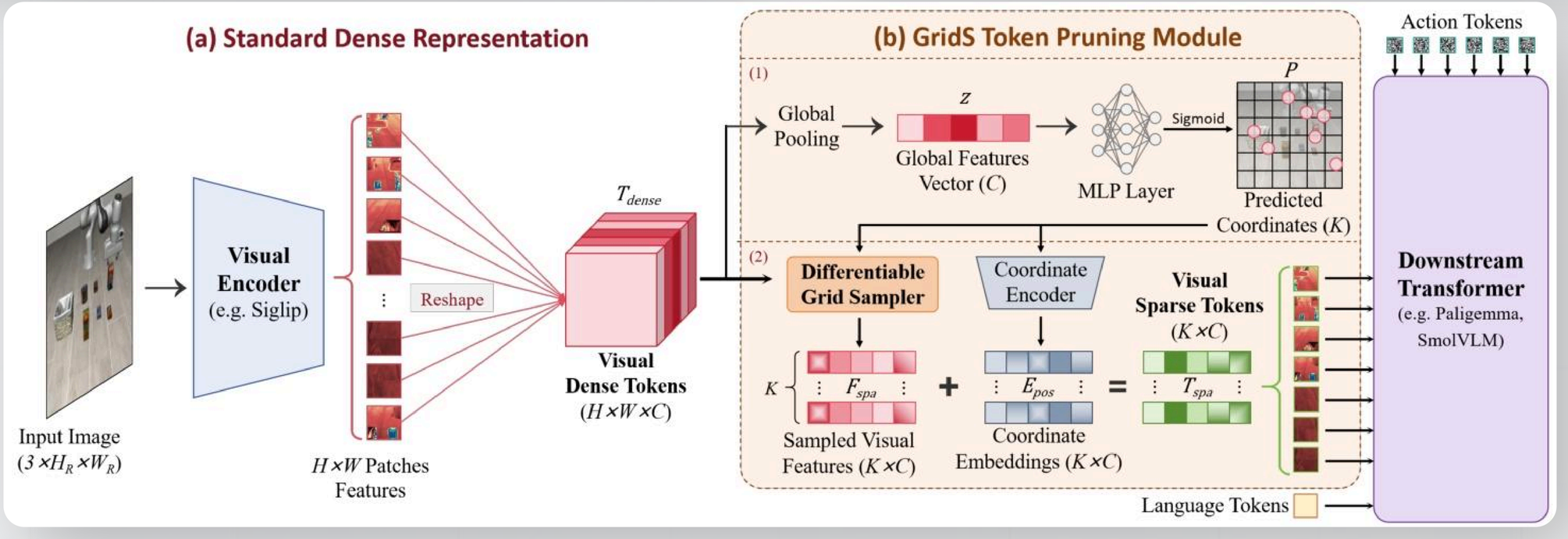
How GridS Works: The Two-Stage Pipeline

Stage 1: Global Coordinate Prediction

1. **Global Context:** Aggregate the entire feature map using Global Average Pooling to capture holistic visual information.
2. **Predict Coordinates:** A lightweight MLP outputs **K** sets of continuous, normalized coordinates (**P**) that represent the most task-relevant regions.

Stage 2: Grid Sampling with Geometry Injection

- **Differentiable Sampling:** Use bilinear interpolation to extract sparse features (**F_spa**) at the predicted coordinates **P** with sub-pixel precision.
- **Inject Geometry:** Encode the coordinates **P** into position embeddings (**E_pos**) to retain spatial awareness.
- **Final Tokens:** Combine **F_spa** and **E_pos** to produce the final sparse visual tokens (**T_spa**).



Datasets: LIBERO & LIBERO-PLUS

LIBERO: A Benchmark for Long-Horizon Manipulation

What is it?

A simulated benchmark for studying generalization in long-horizon, multi-stage manipulation tasks with language guidance.

Our Results on LIBERO

- **Massive Compression:** Tokens reduced from 256 to 16 (93.75% reduction).
- **Significant Speedup:** FLOPs ↓76%, Time ↓>2.1s/task.
- **No Performance Loss:** Success Rate improved by +1.6%.
- **Key Finding:** Even K=1 achieves ~96.6% success rate.

LIBERO-PLUS: Bridging to the Real World

What is it?

An extension of LIBERO that introduces real-world robotic experiments on a physical SO100 arm and more challenging OOD scenarios.

Our Results on LIBERO-PLUS

- **Inference Speedup:** Avg. speedup of **1.23x** on GPU.
- **Overall Performance:** Success rate improved by **+22.2%**.
- **OOD Robustness:** Success rate increased by **+28.6%** with distractors.

Study on the Number of Tokens (K)

We conducted a study to determine the optimal number of tokens (K) for our GridS method, balancing performance (success rate) and computational efficiency (FLOPs).

Performance Analysis

General Trend: Success rate increases with more tokens, but with diminishing returns beyond a certain point.

Key Observations:

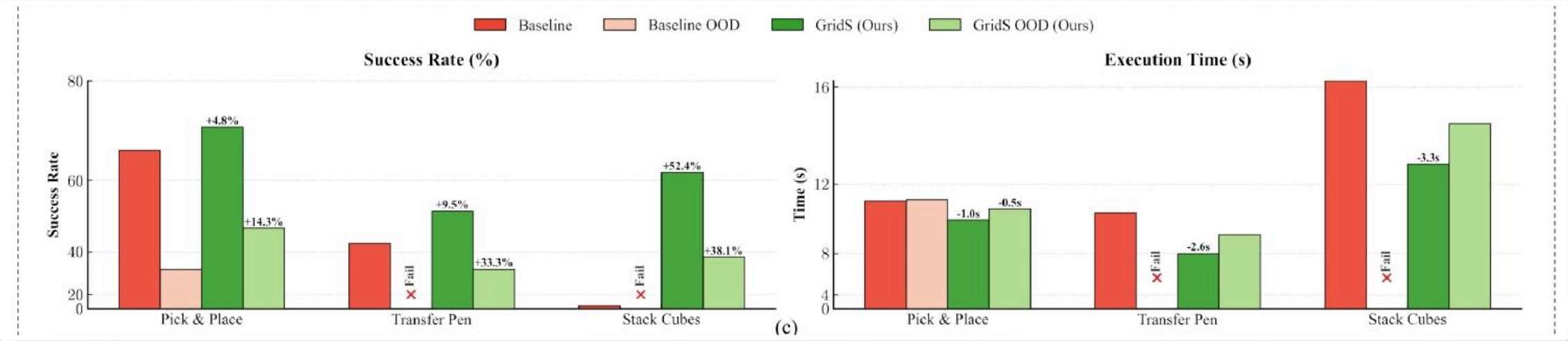
- **K=1:** Even a single token achieves a high success rate of **~96.6%**, capturing critical information efficiently.
- **K=16 (Optimal):** Reaches the **highest success rate (97.7%)** with a high compression ratio (93.75%).
- **K=64:** Only marginal performance gains, but at a much higher cost.

Efficiency Analysis

General Trend: Computational cost (FLOPs) increases linearly with the number of tokens (K). More tokens = higher cost.

Final Conclusion: Using **K=16** represents the **best trade-off** between accuracy and computational cost, offering near-perfect performance with minimal overhead.

Real-World Results: Performance & Robustness on SO100 Robot



Inference Speedup

1.23x on consumer-grade GPU

Significant speed improvement enables real-time interaction and responsiveness in dynamic physical environments.

Overall Performance Gain

+22.2% task success rate

Demonstrated superior task execution capabilities compared to the baseline model across the entire benchmark.

OOD Robustness

+28.6% in distracting scenarios

Showed strong generalization to unseen, Out-of-Distribution (OOD) environments with visual distractors.

Ablation Study: Understanding GridS's Components

We conducted an ablation study on the real robot with a fixed number of tokens (K=16) to validate the necessity of each key component in our GridS framework.

Coordinate Prediction
(Global Context)

What it does: Predicts task-relevant spatial coordinates using global image context.

Impact: Removal (random coords) caused a **−15.3%** drop in success rate, proving the value of data-driven region selection.

Geometry Injection
(Position Encoding)

What it does: Encodes continuous coordinates into position embeddings and adds them to the sampled features.

Impact: Without it, performance fell by **−8.7%**, showing that Transformers need explicit spatial information.

Differentiable Sampling
(vs. Discrete)

What it does: Uses bilinear interpolation to sample features at continuous sub-patch coordinates.

Impact: Switching to discrete selection led to a **−6.2%** drop, confirming the superiority of continuous sampling.

Conclusion: All three components—**coordinate prediction**, **geometry injection**, and **differentiable sampling**—are critical and contribute significantly to the overall performance of GridS, highlighting the value of our full pipeline.

Conclusion & Future Work

01. Introduced GridS

A novel differentiable grid sampling method breaking the performance-efficiency trade-off.

02. New Pruning

Demonstrated complex tasks with as few as **16 visual tokens**.

03. Enhanced Robustness

Significant gains in OOD generalization by effectively filtering out visual noise.

04. Practical Impact

Reduced computational costs by **76%**, enabling efficient real-time deployment on edge devices.

Code: https://github.com/Fediory/Grid-Sampler

Paper: https://arxiv.org/abs/2605.11817

Project Page & Demo: https://fediory.github.io/Grid-Sampler/

Thank You! | Q & A