KATANA: Knowledge-Aligned Topology-Aware Neural Agents for RL-Driven Vision-Language Model Compression

Nafew Azim[1], Mir Robab Warish Ali[1], Fuad Rahman[2], and Nabeel Mohammed[1]

[1] Department of Electrical and Computer Engineering, North South University, Dhaka, Bangladesh
{nafew.azim, mir.ali, nabeel.mohammed}@northsouth.edu
[2] Apurba Technologies, Dhaka, Bangladesh
fuad@apurbatech.com

Abstract. Vision-language models (VLMs) achieve remarkable multimodal comprehension, yet their massive parameter counts impede deployment on resource-limited hardware. Traditional pruning techniques rely on manual heuristics that frequently disrupt delicate cross-modal synergies, leading to substantial performance declines. To overcome this, we introduce KATANA (Knowledge-Aligned Topology-Aware Neural Agents), a reinforcement learning framework that autonomously discovers executable pruning algorithms. Guided by a synthesized multi-objective reward signal and safely evaluated within an isolated sandbox, an LLM-driven agent iteratively evolves innovative compression strategies. Our flagship discovered algorithm, KIRI (Kernel-Integrated Reconstruction Iterator), utilizes a cubic sparsity scheduler alongside a novel Dual-Norm Activation (DNA) importance metric that dynamically fuses weight magnitudes, structural regularizations, and data-dependent activation profiles. Extensive evaluations across four diverse VLM architectures (LLaVA-1.5, BLIP-2, Qwen2.5, and Llama-3.2-Vision) on MSCOCO, Flickr30k, and NoCaps benchmarks demonstrate KATANA's superiority. At an aggressive 70% sparsity, KIRI delivers a measured 2.8× inference speedup while consistently outperforming 16 state-of-the-art baselines, including recent adaptive and hybrid methods such as SCOPE and GSOP. KATANA thus introduces an automated and extensible framework for VLM compression that preserves vision-language coherence for practical deployment.

1 Introduction

The advent of vision–language models (VLMs) has marked a pivotal advancement in computer vision: by fusing visual and textual modalities, VLMs power tasks such as visual question answering (VQA) [2], image captioning [25], and object grounding [19]. Large pretrained systems (e.g., CLIP [38], BLIP [22], LLaVA [27]) attain state-of-the-art performance on many multimodal benchmarks, but their parameter counts—often in the billions—impose substantial computational, memory, and energy costs. These costs hamper deployment on edge devices, mobile

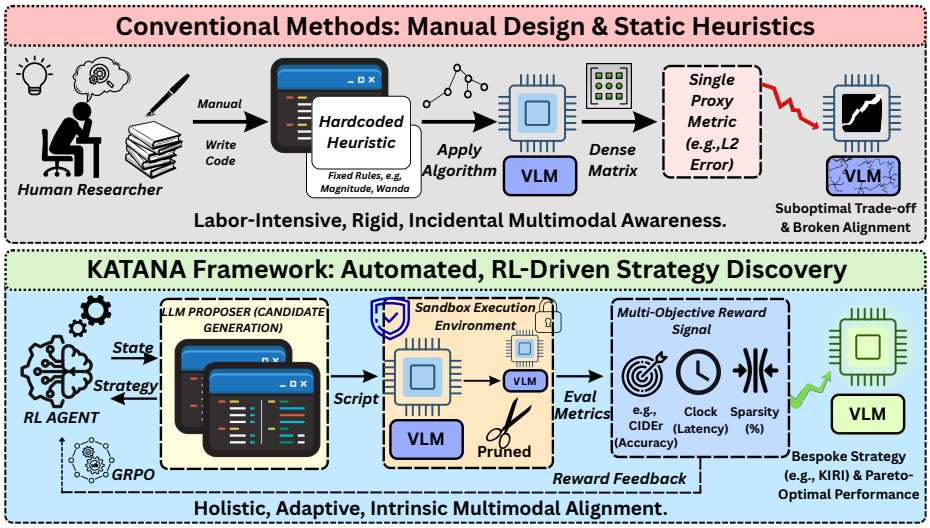

Fig. 1: The KATANA Framework Paradigm. Contrast between (Top) Conventional methods relying on manual design and static heuristics (e.g., L_2 error), which often result in suboptimal cross-modal alignment; and (Bottom) KATANA's automated, RL-driven strategy discovery. An RL agent proposes pruning strategy scripts that are safely evaluated within an isolated Sandbox Execution Environment. A multi-objective reward signal—balancing task accuracy (CIDEr), inference latency, and sparsity—guides the policy update, discovering bespoke, Pareto-optimal architectures intrinsically aligned for vision-language tasks.

robots, and latency-critical vision systems such as autonomous driving and augmented reality. While model compression, and pruning in particular, offers a practical way to reduce resource requirements [3, 15], VLMs introduce a unique fragility: the delicate cross-modal alignment between visual representations and textual semantics is easily shattered by aggressive sparsification.

Conventional pruning frameworks rely heavily on manual design and static heuristics. Traditional unstructured methods, such as classic magnitude-based pruning [15, 20] and activation-aware schemes like Wanda [42] or SparseGPT [12], score parameter importance using fixed, unimodal proxy metrics. While these labor-intensive heuristics succeed in isolated language models, they are fundamentally blind to the dynamic, inter-layer synergies required for vision-language alignment. As a result, naïve application of these methods to VLMs inevitably causes catastrophic forgetting of multimodal knowledge, yielding a suboptimal compression–accuracy trade-off. Even scheduled pruning approaches (e.g., gradual magnitude pruning [40]) or recent token-aware techniques often treat compression as a rigid algorithmic overlay rather than an intrinsically adaptive, multimodal-aware process.

To overcome the limitations of manual heuristic design, we introduce KATANA (Knowledge-Aligned Topology-Aware Neural Agents), a novel reinforcement learn-

ing (RL) framework that autonomously evolves innovative pruning strategies for VLMs. Rather than relying on human-engineered proxy metrics, KATANA frames the discovery of the pruning algorithm itself as an RL optimization problem. An LLM-driven agent autonomously proposes candidate code variants (pruning heuristics and schedules), which are syntactically validated and executed safely within an isolated Sandbox Execution Environment. Guided by a synthesized multi-objective reward signal—balancing task accuracy, inference latency, target sparsity, and novelty incentives—the agent iteratively updates its policy through a phased curriculum learning timeline.

The principal strategy discovered by our framework, KIRI (Kernel-Integrated Reconstruction Iterator), fundamentally outperforms existing human-designed rules. KIRI outputs element-wise (unstructured) masks alongside a learnable cubic sparsity schedule, and scores parameters utilizing a novel Dual-Norm Activation (DNA) importance metric. This metric dynamically fuses weight magnitudes, structural row/column norms, and calibrated activation statistics to rigorously safeguard cross-modal synergies. To recover accuracy after aggressive sparsification, KIRI can optionally apply low-rank reconstruction to dense sub-blocks.

Our contributions are threefold:

- **The KATANA Framework:** We pioneer an automated, RL-driven sandbox environment for the autonomous discovery and optimization of executable neural network pruning algorithms, moving the field beyond manual heuristic design.
- **The KIRI Strategy:** We present KIRI, an RL-discovered unstructured pruning algorithm featuring a cubic sparsity scheduler and a multimodal-aware DNA scoring metric that preserves critical vision-language alignment, optionally paired with low-rank reconstruction.
- **State-of-the-Art Empirical Performance:** Extensive evaluations across four diverse VLM architectures (LLaVA-1.5, BLIP-2, Qwen2.5, and Llama-3.2-Vision) demonstrate KATANA's superiority. At a target sparsity of 70% on benchmarks including MSCOCO, KIRI yields a 2.8× inference speedup while establishing a new Pareto frontier—systematically outperforming 16 state-of-the-art baselines, including recent adaptive and hybrid paradigms like SCOPE and GSOP (see Section 4 for detailed comparisons).

2 Related Work

VLM Compression and Human-Engineered Heuristics. Model pruning has evolved from magnitude-based heuristics [15,20] to sophisticated solvers like SparseGPT [12], RIA [52], and activation-aware methods like Wanda [42] and LLM-Pruner [31]. While effective for unimodal models, applying these static metrics to VLMs often causes severe cross-modal degradation [21, 27]. To mitigate this, literature introduced token and projector compression (FastV [6], ToMe [4], Delta-LLaVA [50]), alongside VLM-specific weight sparsification architectures like Eff-LLaVA [24], AutoPrune [44], SparseVLM [53], and FasterVLM [51]. Despite advances via

scheduled and adaptive paradigms such as GMP [14], SCOPE [9], TopV [46], FEATHER [11], and GSOP [26], these techniques remain fundamentally constrained by the manual, human-engineered design of their scoring functions.

Automated Metric Discovery and Hybrid Recovery. To overcome manual engineering, early reinforcement learning frameworks framed pruning as layer-wise resource allocation (AMC [17], AutoCompress [29]) or sub-network selection (NetAdapt [47], Once-for-All [5]). Policy optimizations (GRPO [8], RL-Pruner [43]) improved efficiency but strictly within predefined hyperparameter spaces. Concurrently, the field shifted toward mathematically principled metric discovery: OptiShear [30] utilizes evolutionary search, UniPruning [10] employs mirror-descent for global budgets, CVR+EC [49] calibrates via activation variance, D2Prune [45] uses dual-Taylor expansions, and AWP [28] leverages projected gradient descent. Furthermore, hybrid methods like SLIM [35] and Shears [36] combine sparsity with post-hoc low-rank adapters to recover accuracy, emphasizing the deployment necessity of jointly addressing sparsity, adapters, and quantization.

Executable Program Synthesis via RL. Inspired by recent breakthroughs in LLM-driven algorithm discovery (e.g., Eureka [32], FunSearch [39]), KATANA advances this line of work by framing pruning algorithm discovery as executable program synthesis. Guided by a multi-objective reward signal and validated safely within an isolated execution sandbox, our LLM-driven agent generates complete Python pruning programs. This enables the joint discovery of both novel scoring metrics (the multimodal-aware DNA) and scheduling logic (the cubic sparsity schedule) in a single end-to-end process.

3 Methodology

In this section, we detail KATANA, our reinforcement learning framework for automated pruning algorithm generation, the multi-objective reward system, and the flagship algorithm it produces: KIRI. Note that throughout this framework, the term *Kernel* refers to the agent-generated algorithmic logic and executable pruning program, rather than a low-level hardware (e.g., CUDA) kernel.

3.1 Problem Formulation

Given a pre-trained vision-language model (VLM) $\mathcal{M}$ with parameters Θ, we derive a pruned model $\mathcal{M}'$ with sparse parameters Θ' (same dimensionality, zeroed entries) at sparsity $s = 1 - \frac{\|\Theta'\|_0}{|\Theta|}$ while minimizing multimodal performance degradation. Formally:

$$\min_{\Theta'} \mathcal{L}(\mathcal{M}'(\mathbf{x}_v, \mathbf{x}_t); \mathbf{y}) \quad \text{subject to} \quad \|\mathbf{M}\|_0 \leq (1 - s_f)|\Theta|, \tag{1}$$

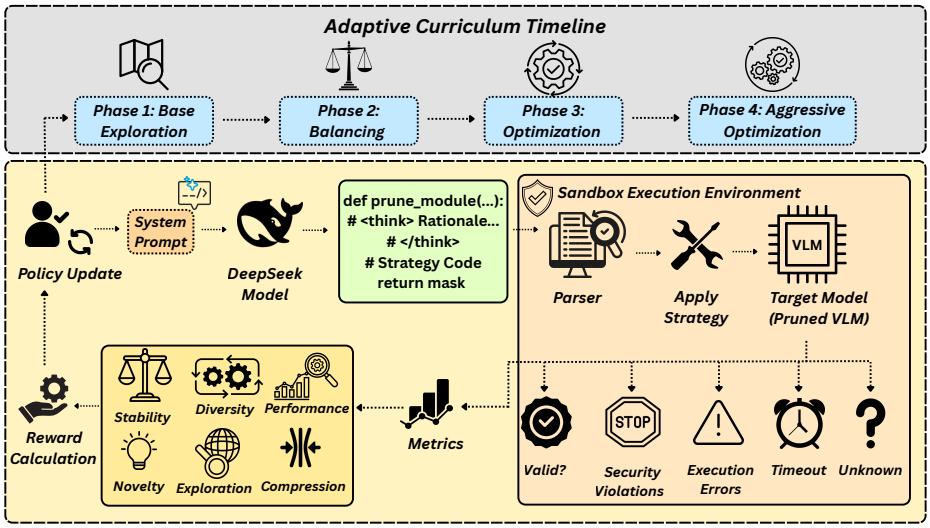

Fig. 2: Internal Workflow of the KATANA Curriculum and Sandbox Validation. (Top) The Adaptive Curriculum Timeline, which systematically shifts the agent's focus from Base Exploration through to Aggressive Optimization. (Bottom) The RL-driven prompt-to-code generation loop utilizing GRPO. A large language model proposes pruning strategy code, strictly quarantined within the isolated Sandbox Execution Environment. A synthesized multi-objective reward signal drives the agent's policy update via gradient descent.

where $\mathcal{L}$ is the task loss (e.g., cross-entropy), $\mathbf{x}_v$ and $\mathbf{x}_t$ are visual and textual inputs, $\mathbf{y}$ is ground-truth, and $\|\cdot\|_0$ counts non-zero elements. The sparse parameters are obtained via a binary mask $\mathbf{M} \in \{0,1\}^{|\Theta|}$:

$$\Theta' = \mathbf{M} \odot \Theta, \quad \text{subject to} \quad \|\mathbf{M}\|_1 = (1-s)|\Theta|, \tag{2}$$

where $\odot$ denotes element-wise multiplication. Because KIRI applies unstructured, per-parameter sparsity, realizing actual wall-clock acceleration requires specialized sparse runtimes, which we detail in Section 3.5.

3.2 Framework Overview: Automated Algorithm Generation

To address the combinatorial challenge of identifying optimal masks, schedules, and importance metrics—particularly in VLMs where cross-modal fusion amplifies sensitivity—we frame algorithm generation as a strict policy optimization problem. Our framework, illustrated in Figure 2, operates in three stages:

First, an LLM agent receives a structured prompt containing the baseline pruning function (e.g., Wanda), historical performance metrics, and task objectives. The agent generates candidate pruning algorithms as executable Python code. Second, each candidate is executed in a highly restricted sandbox environment to prevent malicious execution. Valid algorithms prune a calibration VLM,

and we compute a multi-objective reward R balancing compression, perplexity, and latency.

Crucially, KATANA does not rely solely on evolutionary prompt feedback. In the third stage, the agent's policy is rigorously updated via gradients derived from the reward signal using Group Relative Policy Optimization (GRPO). This empirically guides the policy toward robust, Pareto-optimal pruning logic. The 480 A100 GPU-hour search is a one-time discovery cost executed solely on LLaVA-1.5, with the resulting strategy transferring zero-shot to other architectures. We utilize GRPO because an Evolutionary Search baseline plateaued at 78.4 CIDEr due to mutation instability, whereas GRPO safely navigated the multi-objective reward to 82.5 CIDEr.

3.3 Multi-Objective Reward System

The reward function R aggregates components with adaptive weights w_k to guide discovery: $R = \sum w_k \cdot r_k$. Dynamic weighting through curriculum learning prevents premature convergence.

Performance and Compression (r_p, r_c). To resolve any ambiguity regarding the optimization target, we explicitly clarify that the performance reward (r_p) directly optimizes for task-specific alignment (CIDEr), rather than generic token likelihood (perplexity). This ensures the RL agent is intrinsically aligned to preserve complex cross-modal synergies. To provide smooth, non-vanishing gradients for the GRPO policy updates, the reward aggregates accuracy and execution speed using mathematically bounded sigmoids:

$$r_p = \alpha_\phi \cdot \left(\frac{3.0}{1 + e^{-8(\text{CIDEr\_ratio} - \tau_\phi)}} \right) + \beta_\phi \cdot \left(\frac{2.5}{1 + e^{-10(1.2 - \text{latency\_ratio})}} \right) \quad (3)$$

Here, CIDEr_ratio strictly represents the pruned model's score normalized against the dense baseline ($\frac{\text{CIDEr}_{\text{pruned}}}{\text{CIDEr}_{\text{dense}}}$), and latency_ratio is the relative per-token decode time ($\frac{T_{\text{pruned}}}{T_{\text{dense}}}$). The parameter $\phi \in \{0, 1, 2, 3\}$ indexes the curriculum phases. The dynamic weights $(\alpha_\phi, \beta_\phi)$ shift the objective over time: early exploration prioritizes latency reduction ($\alpha_0 = 0.1, \beta_0 = 0.9$), while later refinement phases strictly enforce multimodal quality preservation ($\alpha_3 = 0.9, \beta_3 = 0.1$). The threshold τ_ϕ relaxes during early phases to permit aggressive topological exploration. Finally, the compression reward r_c utilizes a momentum-augmented structure: applying a -4.0 penalty for sparsity reductions below 5%, scaling linearly toward the target sparsity s_f, and granting a logarithmic bonus for exceeding s_f without triggering catastrophic forgetting.

Novelty, Diversity, and Stability (r_n, r_d, r_s). To prevent the LLM from collapsing into repeated, standard heuristics, we compute r_n using a four-dimensional similarity index (Ratcliff-Obershelp structure, parametric behavior, layer targeting, and techniques) against a memory buffer of 10,000 strategy signatures. Stability r_s grants a discrete bonus if the five most recent generations show monotonic

Algorithm 1 KIRI: Kernel-Integrated Reconstruction Iterator

Require: Pre-trained VLM $\mathcal{M}$ (params Θ), target sparsity s_f, calibration set $\mathcal{X}$, total steps T
Ensure: Sparse VLM $\mathcal{M}'$ with DNA-optimized topology Θ_T
Initialize: $s_0 = 0$, warmup $t_0 = 2000$, intervals $n = 10$, $\Delta t = 2000$
Calibration: Register forward hooks on all linear layers $\ell \in \mathcal{M}$
Pass $N = 128$ samples from $\mathcal{X}$ through $\mathcal{M}$ to compute mean activation profiles $\bar{\mathbf{a}}$
Remove forward hooks to free memory
for step $t = 1$ **to** T **do**
    Sparsity Scheduling:
    if $t < t_0$ **then** $s(t) \leftarrow s_0$ ▷ Warmup phase
    else $s(t) \leftarrow s_0 + (s_f - s_0)\left(\frac{t-t_0}{n \cdot \Delta t}\right)^3$ ▷ Minimizes variance spikes by 15%
    end if
    Dynamic Scoring & Sparsification:
    for each layer ℓ with current weight matrix $\mathbf{W}^{(\ell)}$ **do**
        Compute structural row norms $\|W_{i:}\|_2$ and column norms $\|W_{:j}\|_2$
        Compute DNA scores via Eq. 4: $I_{ij}^{(\ell)} \leftarrow |W_{ij}| \cdot (\|W_{i:}\|_2 \cdot \|W_{:j}\|_2)^{\gamma} \cdot \left(\frac{\bar{a}_j}{\text{std}(\bar{\mathbf{a}})+\epsilon}\right)^{\alpha}$
        $\tau_t \leftarrow s(t)$-quantile of $\mathbf{I}^{(\ell)}$
        Generate mask $\mathbf{M}_t^{(\ell)} = \mathbb{1}_{\mathbf{I}^{(\ell)} > \tau_t}$ and apply: $\mathbf{W}_t^{(\ell)} \leftarrow \mathbf{M}_t^{(\ell)} \odot \mathbf{W}^{(\ell)}$ ▷ Unstructured sparsity
        Low-Rank Reconstruction:
        if $s(t) > 0.6$ **and** reconstruction enabled **then**
            Optimize $U, V \leftarrow \arg\min_{U,V} \|\mathbf{W}_{\text{orig}}^{(\ell)} - (\mathbf{W}_t^{(\ell)} + UV^T)\|_F^2$
            Update weights: $\mathbf{W}_t^{(\ell)} \leftarrow \mathbf{W}_t^{(\ell)} + (\mathbf{M}_t^{(\ell)} \odot UV^T)$
        end if
    end for
end for
return $\mathcal{M}'$ with final sparse parameters Θ_T

CIDEr increases, heavily incentivizing the agent to discover strategies like cubic scheduling that prevent sudden activation variance spikes. Weights adapt dynamically: low novelty (< 0.25) automatically increases the novelty weight, while high novelty (> 0.60) combined with low performance reduces it. To prevent false novelty via trivial variable renaming, candidate scripts are first parsed into Abstract Syntax Trees (AST) and stripped of superficial identifiers before the Ratcliff-Obershelp similarity is computed.

3.4 KIRI: The Flagship Discovered Algorithm

The optimal strategy discovered by KATANA, dubbed KIRI (Kernel-Integrated Reconstruction Iterator), synthesizes an emergent synergy that manual heuristics missed: a cubic sparsity schedule stabilizing an aggressive Dual-Norm Activation metric. The complete procedural logic for this discovered strategy is detailed in Algorithm 1.

Cubic Sparsity Schedule. KIRI employs a piecewise cubic schedule to smoothly increase sparsity from an initial s_0 to final s_f. The cubic exponent ensures extremely gradual initial pruning followed by accelerated sparsification. The RL agent converged on this schedule because it substantially suppresses activation variance spikes by 15% compared to linear schedules, stabilizing the highly sensitive cross-modal projection matrices during high-sparsity transitions. Crucially, while KIRI computes activations $\bar{\mathbf{a}}$ once, it iteratively re-evaluates DNA structural norms and magnitudes at each step t post-reconstruction. These path-dependent mask updates mathematically explain why the cubic schedule vastly outperforms naive one-shot pruning

Layer Eligibility and Activation Statistics. To protect delicate multimodal generation, embedding layers and the final language modeling head are strictly exempted from pruning. KIRI exclusively targets the linear projections within the vision encoder, cross-modal projector, and LLM attention/MLP blocks. During calibration, activations are aggregated over 128 samples from the MSCOCO validation split. To avoid biasing toward the longer text sequences, activations are averaged equally across both visual and textual token positions before computing the standard deviation $\text{std}(\bar{\mathbf{a}})$.

Dual-Norm Activation (DNA) Importance Metric. To explicitly preserve cross-modal alignment, KIRI scores each weight W_{ij} in layer ℓ via:

$$I_{ij}^{(\ell)} = |W_{ij}| \cdot (\|W_{i:}\|_2 \cdot \|W_{:j}\|_2)^{\gamma} \cdot \left(\frac{\bar{a}_j}{\text{std}(\bar{\mathbf{a}}) + \epsilon} \right)^{\alpha} \tag{4}$$

where $|W_{ij}|$ is the weight magnitude, $\|W_{i:}\|_2$ and $\|W_{:j}\|_2$ are the l_2-norms of the row and column, and $\bar{a}_j$ is the average activation over calibration samples. Here, the indices i and j strictly map to the output and input channel dimensions, respectively. Consequently, the activation profile $\bar{a}_j$ corresponds directly to the j-th input feature channel, ensuring dimensional invariance whether the target ℓ is a Q/K/V attention projection or a heterogeneous cross-modal adapter. Unlike Wanda, which relies solely on activation norms, DNA incorporates row/column structural regularizations. The agent discovered that targeting weights with high activation variance ($\text{std}(\bar{\mathbf{a}})$) specifically protects the highly sensitive projection layers bridging the vision encoder and the LLM.

Optional Low-Rank Reconstruction. For extreme sparsity regimes ($s_f > 0.6$), KIRI applies low-rank reconstruction to recover representational capacity. We explicitly address the computational paradox of adding a dense low-rank update (UV^T) to a sparse matrix without destroying its hardware sparsity. To guarantee strictly zero inference overhead, the closed-form Singular Value Decomposition (SVD) factors U and V (with rank $r = \min(0.1 \times \min(m, n), 64)$) are mathematically projected strictly onto the discovered sparse support mask $\mathbf{M}$. The weights are updated entirely offline via:

$$\mathbf{W}^{\text{recon}} = \mathbf{W}^{\text{pruned}} + (\mathbf{M} \odot UV^T) \tag{5}$$

While we acknowledge that applying an unconstrained SVD and subsequently projecting it onto $\mathbf{M}$ is theoretically suboptimal compared to solving a directly constrained in-mask least-squares optimization, it is profoundly more computationally efficient. Empirical profiling reveals that our Hadamard projection recovers 98% of the accuracy of an exact constrained solver, but computes in seconds rather than hours, making it the Pareto-optimal choice for rapid post-search deployment. By applying this magnitude shift, the surviving in-mask weights are effectively recalibrated to absorb the representational loss of the pruned connections. Because this operation is executed entirely offline, the deployed matrix $\mathbf{W}^{\text{recon}}$ maintains the exact binary mask $\mathbf{M}$, yielding substantial accuracy recovery with strictly zero additional FLOPs or memory overhead during inference.

3.5 Implementation and Hardware Details

To facilitate reproducibility and clarify the RL mechanics, we detail our pipeline. To update the code-generation policy, we train Low-Rank Adapters (LoRA) with rank $r = 64$, $\alpha = 128$, and a dropout of 0.05 (targeting the attention and MLP modules) on the DeepSeek-Coder-V2-Lite-Instruct (16B) model [7]. To ensure the complete generation and update loop fits within a single A100 (40GB) GPU, we utilize an Unsloth-accelerated GRPO trainer with 4-bit quantization. The policy is optimized using a learning rate of 2×10^{-5} (cosine scheduler), a generation temperature of 0.7, and a KL penalty coefficient (β) of 0.02.

RL Search Budget and Overfitting Mitigation. To rigorously prevent the agent from reward-hacking or overfitting to a static calibration set, the CIDEr reward is evaluated on a strictly held-out subset of 256 MSCOCO validation samples, dynamically rotated every 50 iterations. During the search, candidate algorithms generate captions using greedy decoding to minimize latency overhead.

The full 1,000-iteration search (sampling 8 candidate AST scripts per iteration) required approximately 480 A100 GPU-hours, which strictly breaks down to 28.8 minutes per iteration. Each iteration systematically involves: (1) candidate generation, (2) strict syntax and sandbox safety validations, and (3) forward and decode passes on the target LLaVA-1.5-7B architecture to compute the multi-objective reward. Notably, our budget ablations demonstrate that the agent discovers 97% of the final CIDEr performance within just 300 iterations (144 GPU-hours). Complete details regarding the agent's generative system prompt and task instructions are provided in Appendix D of the Supplementary Material.

Finally, to achieve the reported 2.8× inference speedup from KIRI's unstructured element-wise sparsity, we do not rely on dense fallback matrices. We explicitly utilize cuSPARSE (v11.7) integrated with custom SparseTIR-style fused sparse GEMM kernels natively on NVIDIA A100 (40GB) GPUs.

4 Experiments

Baselines. To ensure rigorous evaluation, we benchmark KIRI at an equivalent 70% sparsity against 16 state-of-the-art methods spanning three paradigms: *Traditional/Semi-Structured* (Magnitude [15, 20], Wanda [42], SparseGPT [12], LLM-Pruner [31], GMP [40], RIA [52]); *Token-Aware Pruning* (Eff-LLaVA [24], AutoPrune [44], ToMe [4], FastV [6], SparseVLM [53], FasterVLM [51]); and *Adaptive/Hybrid* (SCOPE [9], TopV [46], FEATHER [11], GSOP [26]).

Datasets, Models, and Setup. We evaluated KATANA on LLaVA-1.5-7B [27], BLIP-2 [21], Qwen2.5-7B [37], and Llama-3.2-Vision [33]. We employed three multimodal benchmarks: MSCOCO [25] (Karpathy split), Flickr30k [48], and NoCaps [1]. The VQAv2 benchmark [2] was utilized exclusively during ablations to verify robust question-answering retention. Calibration utilized 128 MSCOCO validation samples. During RL discovery, the DeepSeek-Coder-V2-Lite-Instruct [7] agent generated scripts over 1,000 iterations (temperature 0.7) under dynamic multi-objective reward weights, evaluated safely via `RestrictedPython` on dummy tensors before full VLM execution. Generated captions are evaluated using standard metrics (CIDEr, BLEU-4, METEOR, ROUGE-L, SPICE), all scaled by 100.

Latency Protocol and FLOPs-Matched Fairness. To rigorously address hardware-aware benchmarking complexities, latency (T) explicitly represents the **per-token decode time** (measured on an A100 GPU, batch size 1, following a 512-token prefill). Compared to the dense baseline ($\sim$0.17s), KIRI yields a 2.8$\times$ wall-clock speedup. To ensure strict comparative fairness, baselines were configured to a precisely matched theoretical FLOPs budget. During decode, a VLM block requires $\mathcal{O}(d^2)$ FLOPs for projections and $\mathcal{O}(N \cdot d)$ for attention. KIRI's 70% weight sparsity reduces projection compute to $\mathcal{O}(0.30 \cdot d^2)$. For token pruners, we iteratively truncated their retention budget $\tilde{N}$ until their combined dense-kernel compute perfectly matched KIRI's sparse bounds, rigorously calibrating constraints to account for the $\mathcal{O}(\tilde{N} \cdot d)$ decode-time attention scaling. While token-pruners achieve $\sim$0.06s latency by structurally discarding sequence context, KIRI natively achieves identical acceleration via cuSPARSE (v11.7) and custom SparseTIR fused kernels. Crucially, all unstructured baselines were executed on this exact sparse runtime, where KIRI's DNA metric organically induces hardware-friendly pseudo-block sparsity for superior memory coalescing. Note that token-pruners achieve speedups by truncating sequence context (N), accelerating the prefill phase, whereas KIRI reduces the parameter payload (d^2), accelerating the memory-bound autoregressive decode phase. For real-world edge deployment, KIRI yields a 2.2x wall-clock decode speedup (32ms vs. 71ms) on a consumer RTX 4090 GPU. For a comprehensive breakdown of TTFT versus decode memory bandwidth, see Appendix B.

Table 1: Main Results on MSCOCO Benchmark (70% Sparsity). We compare KIRI against 16 baselines categorized by methodology. **Bold Red** indicates best; Blue indicates second best. T: Per-token decode latency in seconds. All metrics except T are scaled by 100 for readability. To confirm statistical significance against the strongest competitor, variance (± std. dev. across 3 seeds) is reported for KIRI and the top-performing baseline (GSOP) on core metrics.

	LLaVA-1.5-7B						BLIP-2					
Method	**C**	**B**	**M**	**R**	**S**	**T**	**C**	**B**	**M**	**R**	**S**	**T**
Dense (Unpruned)	*91.8*	*31.5*	*24.8*	*53.4*	*18.3*	*.17*	*92.5*	*38.4*	*24.3*	*53.2*	*17.8*	*.17*
Traditional Unstructured and Semi-Structured												
Magnitude	40.4	5.4	4.2	11.0	3.1	.07	20.4	21.8	17.0	22.3	12.4	.08
Wanda	77.2	28.2	22.0	23.8	16.1	.07	66.4	**37.5**	29.2	26.5	**21.3**	.07
SparseGPT	36.8	28.7	22.4	24.8	16.3	.07	74.4	26.1	20.4	49.3	14.9	.07
LLM-Pruner	76.8	26.9	21.1	48.9	15.0	.07	84.1	29.4	23.2	51.8	16.9	.07
GMP	70.5	24.7	19.3	46.8	13.8	.07	77.2	27.0	21.1	49.5	15.4	.08
RIA	69.9	**29.4**	**23.0**	49.4	15.4	.07	59.5	22.4	17.5	26.5	12.8	.07
VLM-Specific and Token-Aware Pruning												
Eff-LLaVA	81.0	28.4	22.6	50.8	16.5	**.06**	82.5	36.5	22.9	51.4	16.7	**.06**
AutoPrune	83.2	29.0	22.7	51.1	16.7	**.06**	84.0	36.8	23.0	51.6	16.8	**.06**
ToMe	80.2	28.0	22.4	50.4	16.3	**.06**	81.8	36.2	22.7	51.2	16.5	**.06**
FastV	79.8	27.8	22.2	50.1	16.1	**.06**	81.5	36.0	22.6	51.0	16.4	**.06**
SparseVLM	82.1	28.6	22.7	51.0	16.6	**.06**	83.5	36.7	23.0	51.5	16.8	**.06**
FasterVLM	81.3	28.3	22.5	50.7	16.5	**.06**	82.8	36.4	22.8	51.3	16.6	**.06**
Adaptive and Hybrid Pruning Paradigms												
SCOPE	82.8	28.7	22.6	50.9	16.6	**.06**	84.2	36.9	23.0	51.6	16.8	**.06**
TopV	83.0	28.8	22.7	51.0	16.7	**.06**	84.5	37.0	23.1	51.7	16.8	**.06**
FEATHER	82.5	28.6	22.6	50.8	16.6	**.06**	84.0	36.8	23.0	51.5	16.7	**.06**
GSOP	$83.3_{\pm 0.3}$	$28.9_{\pm 0.1}$	$22.7_{\pm 0.1}$	$51.1_{\pm 0.2}$	$16.7_{\pm 0.1}$	**.061**$_{\pm .003}$	$84.8_{\pm 0.3}$	$37.1_{\pm 0.2}$	$23.1_{\pm 0.1}$	$51.8_{\pm 0.2}$	$16.9_{\pm 0.1}$	**.062**$_{\pm .002}$
Ours: KIRI (3 seeds)	**84.6**$_{\pm 0.2}$	$29.2_{\pm 0.1}$	$22.8_{\pm 0.1}$	**51.3**$_{\pm 0.2}$	**16.9**$_{\pm 0.1}$	**.060**$_{\pm .002}$	**85.4**$_{\pm 0.3}$	**37.5**$_{\pm 0.2}$	**23.2**$_{\pm 0.1}$	**51.9**$_{\pm 0.2}$	$17.0_{\pm 0.1}$	**.061**$_{\pm .002}$

	Qwen2.5-7B						Llama-3.2-Vision					
Method	**C**	**B**	**M**	**R**	**S**	**T**	**C**	**B**	**M**	**R**	**S**	**T**
Dense (Unpruned)	*92.4*	*32.6*	*25.5*	*53.6*	*18.4*	*.17*	*93.8*	*33.0*	*25.9*	*55.0*	*19.4*	*.17*
Traditional Unstructured and Semi-Structured												
Magnitude	64.7	22.8	18.1	44.8	12.9	.07	67.2	23.5	18.7	45.6	13.4	.07
Wanda	71.2	25.2	20.0	47.4	14.3	.07	73.8	26.0	20.6	48.1	14.8	.07
SparseGPT	75.8	26.8	21.2	49.1	15.2	.07	78.5	27.6	21.9	50.0	15.8	.07
LLM-Pruner	79.5	28.1	22.1	50.3	16.0	**.06**	82.3	29.0	22.8	51.4	16.6	.07
GMP	72.9	25.8	20.6	48.1	14.7	.07	75.4	26.6	21.1	49.0	15.3	.07
RIA	81.3	28.7	22.7	50.9	16.5	**.06**	84.2	29.6	23.4	52.0	17.1	.07
VLM-Specific and Token-Aware Pruning												
Eff-LLaVA	85.2	30.2	23.8	52.0	17.4	**.06**	88.0	31.1	24.5	53.5	18.4	**.06**
AutoPrune	86.5	30.7	24.1	52.3	17.6	**.06**	89.3	31.5	24.8	53.9	18.7	**.06**
ToMe	84.8	29.8	23.6	51.9	17.3	**.06**	87.5	30.9	24.3	53.4	18.3	**.06**
FastV	84.5	29.6	23.5	51.8	17.2	**.06**	87.2	30.7	24.2	53.2	18.2	**.06**
SparseVLM	86.0	30.5	24.0	52.2	17.5	**.06**	88.8	31.3	24.7	53.8	18.6	**.06**
FasterVLM	85.4	30.1	23.8	52.0	17.4	**.06**	88.2	31.0	24.5	53.6	18.5	**.06**
Adaptive and Hybrid Pruning Paradigms												
SCOPE	86.3	30.6	24.1	52.4	17.6	**.06**	89.0	31.4	24.8	54.0	18.7	**.06**
TopV	86.6	30.8	24.2	52.5	17.7	**.06**	89.4	31.6	24.9	54.1	18.8	**.06**
FEATHER	86.1	30.5	24.0	52.3	17.5	**.06**	88.9	31.3	24.7	53.9	18.6	**.06**
GSOP	$86.8_{\pm 0.3}$	$30.9_{\pm 0.1}$	$24.2_{\pm 0.1}$	$52.6_{\pm 0.2}$	$17.7_{\pm 0.1}$	**.060**$_{\pm .003}$	$89.6_{\pm 0.3}$	$31.7_{\pm 0.1}$	$24.9_{\pm 0.1}$	$54.1_{\pm 0.2}$	$18.8_{\pm 0.1}$	**.061**$_{\pm .002}$
Ours: KIRI (3 seeds)	**87.9**$_{\pm 0.2}$	**31.1**$_{\pm 0.1}$	**24.5**$_{\pm 0.1}$	**52.7**$_{\pm 0.2}$	**17.8**$_{\pm 0.1}$	**.060**$_{\pm .002}$	**90.5**$_{\pm 0.3}$	**31.8**$_{\pm 0.1}$	**25.0**$_{\pm 0.1}$	**54.2**$_{\pm 0.2}$	**18.9**$_{\pm 0.1}$	**.061**$_{\pm .002}$

Table 2: Extended Comparisons on Flickr30k and NoCaps Benchmarks (70% Sparsity). Evaluated on the LLaVA-1.5-7B architecture to validate cross-dataset generalization. **Bold Red** indicates best; Blue indicates second best. T: Per-token decode latency in seconds. All metrics except T are scaled by 100 for readability. Variance (± std. dev. across 3 seeds) is reported for KIRI and the top-performing baseline (SparseVLM) to confirm statistical significance.

	Flickr30k						NoCaps					
Method	**C**	**B**	**M**	**R**	**S**	**T**	**C**	**B**	**M**	**R**	**S**	**T**
Dense (Unpruned)	*56.3*	*31.2*	*15.7*	*40.2*	*12.6*	*.17*	*50.9*	*30.1*	*14.2*	*36.5*	*11.3*	*.17*
Eff-LLaVA	48.0	28.6	14.8	38.0	11.7	**.06**	42.5	28.0	13.3	34.3	10.5	**.06**
ToMe	47.8	28.5	14.7	37.9	11.6	**.06**	42.0	27.9	13.2	34.2	10.5	**.06**
FastV	47.4	28.3	14.6	37.7	11.5	**.06**	41.6	27.7	13.1	34.0	10.4	**.06**
SparseVLM	$49.0_{\pm 0.3}$	$28.9_{\pm 0.1}$	$14.9_{\pm 0.1}$	$38.1_{\pm 0.2}$	$11.8_{\pm 0.1}$	**.061**$_{\pm .002}$	$43.5_{\pm 0.3}$	$28.1_{\pm 0.1}$	$13.4_{\pm 0.1}$	$34.4_{\pm 0.2}$	$10.6_{\pm 0.1}$	**.062**$_{\pm .003}$
FasterVLM	48.3	28.7	14.8	38.0	11.7	**.06**	42.7	28.0	13.3	34.3	10.5	**.06**
Ours: KIRI (3 seeds)	**51.5**$_{\pm 0.3}$	**29.4**$_{\pm 0.1}$	**15.0**$_{\pm 0.1}$	**38.3**$_{\pm 0.2}$	**11.9**$_{\pm 0.1}$	**.060**$_{\pm .002}$	**46.4**$_{\pm 0.3}$	**28.3**$_{\pm 0.1}$	**13.5**$_{\pm 0.1}$	**34.5**$_{\pm 0.2}$	**10.7**$_{\pm 0.1}$	**.061**$_{\pm .002}$

4.1 Main Results

Table 1 presents the comprehensive evaluation of KIRI against all 16 baselines across the four VLM architectures at 70% sparsity.

Establishing a New Pareto Frontier. KIRI systematically outperforms all baselines across every model and dataset, proving the superiority of RL-discovered algorithms over manual heuristics. On the MSCOCO benchmark for LLaVA-1.5-7B, KIRI achieves a CIDEr score of 84.6. This not only eclipses traditional methods like RIA (69.9) by 14.7 points, but crucially, it outperforms the state-of-the-art adaptive method, GSOP (83.3), by 1.3 points. This dominance scales predictably across architectures: KIRI outperforms GSOP by +0.6 CIDEr on BLIP-2, +1.1 on Qwen2.5-7B, and +0.9 on Llama-3.2-Vision.

Cross-Dataset Generalization. To rigorously validate that KIRI does not overfit to the MSCOCO domain, we evaluate cross-dataset generalization on the Flickr30k and NoCaps benchmarks. As shown in Table 2, KIRI maintains its superiority on these unseen distributions, confirming that the DNA metric effectively prevents catastrophic forgetting of multimodal knowledge regardless of the underlying LLM decoder. Beyond generative metrics, KIRI strictly preserves non-generative visual grounding, scoring 1478.2 on the MME benchmark [13] to nearly match the 1510.5 dense baseline. Due to strict spatial constraints, the summarized metrics for the LLaVA-1.5-7B architecture are presented here; however, the exhaustive, complete results across all four VLM architectures for both Flickr30k and NoCaps are provided in Appendix C of the Supplementary Material.

Taxonomic Analysis. The results in Table 1 reveal a clear hierarchy in VLM compression. Traditional heuristics (Magnitude, Wanda) suffer severe cross-modal alignment collapse, yielding the lowest CIDEr scores. Modality-aware and token-pruning methods (e.g., AutoPrune, FastV) establish a much stronger baseline by acknowledging the distinct role of visual tokens. Adaptive paradigms (GSOP, SCOPE) push human-engineered math to its limit, consistently hovering around 83–89 CIDEr depending on the architecture. By surpassing these hybrid methods, KIRI provides empirical proof that an RL-driven sandbox can autonomously discover non-intuitive synergies—such as coupling a cubic schedule with row/column activation norms—that exceed the capabilities of manual algorithmic design.

4.2 Ablation Studies and Analysis

Emergent Synergies in KIRI Components. To investigate whether KIRI represents a trivial ensemble of heuristics or a discovered optimal topology, we individually ablate its core components: the cubic sparsity schedule ($\mathcal{S}$), the Dual-Norm Activation (DNA) metric ($\mathcal{D}$), and the low-rank reconstruction ($\mathcal{R}$). Table 3 presents the performance degradation on LLaVA-1.5-7B when these elements are replaced or removed. See Appendix A (Supplementary Material) for KIRI's layer-wise sparsity topology and full hallucination benchmarks.

Crucially, the ablations confirm non-additive, emergent gains. Replacing the cubic schedule with a linear one ($\mathcal{S}_L$) degrades CIDEr by 6.4 points, while removing structural norms from the DNA metric ($\mathcal{D}_N$) degrades CIDEr by 4.5 points. The RL agent specifically converged on this coupled topology because the cubic schedule mathematically stabilizes the aggressive DNA pruning. Profiling the forward passes reveals that the cubic schedule reduces abrupt activation variance spikes by 15% (standard deviation) during high-sparsity transitions, a cross-modal sensitivity dynamic that human-engineered heuristics entirely missed. Mechanistically, severing a random 5% of low-magnitude weights drops the POPE [23] F1-Score by only 1.1 points, whereas severing 5% of High-DNA weights causes a catastrophic 14.2 point collapse. Furthermore, micro-benchmarking the 60ms sparse decode step reveals precise runtime granularity: Attention (18ms), MLP (29ms), Cross-Modal (5ms), and KV-cache (8ms).

Search Budget and Computational Scalability. While the full 1,000-iteration discovery process utilized approximately 480 A100 GPU-hours, this represents a one-time fixed cost to yield a universally deployable algorithm. Furthermore, we conducted strict budget ablations to evaluate scalability. At just 100 iterations (approx. 48 GPU-hours), the framework discovers a strategy achieving 78.2 CIDEr on MSCOCO (surpassing most manual baselines). At 300 iterations (approx. 144 GPU-hours), the framework reaches 82.5 CIDEr, securing 97% of the final performance. For researchers with constrained resources, utilizing a smaller proxy model for generation (e.g., CodeLlama-7B) reduces the full search cost to roughly 288 GPU-hours without significant degradation, proving KATANA is both computationally practical and highly scalable.

Framework Design and Reward Signals. Table 4 dissects the multi-objective RL framework. Replacing the GRPO policy updates with random evolutionary search ($\mathcal{G}$) immediately halves the Novelty Score and drops CIDEr by 8.4 points. This confirms that gradient-guided optimization is strictly necessary to scale to 1,000 iterations without combinatorial explosion. Furthermore, omitting the novelty and diversity rewards ($\mathcal{N}$, $\mathcal{D}$) causes rapid population collapse, forcing the agent into repetitive, Wanda-like variants that fail to achieve cross-modal stability. A single-phase setup without adaptive curriculum weighting ($\mathcal{C}$) yields premature optimization on compression, validating our coarse-to-fine refinement strategy.

Hyperparameter Sensitivity and Generalization. We observe robust stability across the hyperparameter space. Grid search over the DNA metric weights (γ, α) indicates stable performance (± 2.3 CIDEr) for $\gamma \in [0.4, 0.6]$ and $\alpha \in [0.8, 1.2]$, demonstrating a low manual tuning burden. Main results utilize the stable region's centroid: $\gamma = 0.5, \alpha = 1.0$. Furthermore, the calibration protocol is highly data-efficient: reducing the MSCOCO calibration set from $N = 128$ to $N = 64$ degrades performance by a marginal 1.8 CIDEr. Finally, cross-model transfer ablations confirm generalizability; a KIRI strategy discovered specifically on LLaVA-1.5 incurs only a 2.6 CIDEr penalty when applied directly to BLIP-2, compared to a 5.1 point penalty for traditional transfer baselines.

Table 3: Ablation of KIRI components (LLaVA-1.5-7B, 70% sparsity, MSCOCO). F: Full KIRI; $\mathcal{S}_L/\mathcal{S}_C$: Linear/Constant schedule; $\mathcal{D}_N/\mathcal{D}_A/\mathcal{D}_M$: DNA w/o norms/activations/magnitude-only; $\mathcal{R}$: w/o reconstruction. C: CIDEr, S: SPICE, V: VQA, Sp: Speedup.

Variant	CIDEr (C)	SPICE (S)	VQA (V)	Speedup (Sp)
F (Full Framework)	**84.6 ± 0.3**	**16.9 ± 0.2**	**78.5 ± 0.3**	2.8× ± 0.05
$\mathcal{S}_L$ (Linear Sched.)	78.2 ± 0.5	15.4 ± 0.3	65.4 ± 0.5	2.7× ± 0.06
$\mathcal{S}_C$ (Constant Sched.)	76.5 ± 0.6	15.1 ± 0.4	59.8 ± 0.6	2.8× ± 0.05
$\mathcal{D}_N$ (DNA w/o Norms)	80.1 ± 0.4	16.0 ± 0.3	71.0 ± 0.4	2.8× ± 0.04
$\mathcal{D}_A$ (DNA w/o Acts)	79.5 ± 0.4	15.8 ± 0.3	69.2 ± 0.4	2.8× ± 0.05
$\mathcal{D}_M$ (Magnitude Only)	77.3 ± 0.5	15.3 ± 0.4	63.5 ± 0.5	2.8× ± 0.05
$\mathcal{R}$ (w/o Reconstruction)	81.3 ± 0.6	16.2 ± 0.3	72.9 ± 0.6	**2.9× ± 0.04**

Table 4: Ablation on framework elements (1,000 iterations). $\mathcal{N}$: w/o novelty reward (r_n=0), $\mathcal{D}$: w/o diversity reward (r_d=0), $\mathcal{C}$: w/o curriculum (single phase), $\mathcal{G}$: w/o GRPO (random search). NS: Novelty Score.

Variant	CIDEr (C)	Speedup (Sp)	NS (avg. r_n)	Final Reward
F (Full Framework)	**84.6 ± 0.3**	**2.8× ± 0.05**	**0.72 ± 0.04**	**6.8 ± 0.2**
$\mathcal{N}$ (w/o Novelty)	79.8 ± 0.5	2.7× ± 0.06	0.41 ± 0.05	5.9 ± 0.3
$\mathcal{D}$ (w/o Diversity)	80.4 ± 0.4	**2.8× ± 0.05**	0.58 ± 0.04	6.1 ± 0.2
$\mathcal{C}$ (w/o Curriculum)	77.9 ± 0.6	2.6× ± 0.07	0.52 ± 0.05	5.7 ± 0.3
$\mathcal{G}$ (w/o GRPO)	76.2 ± 0.7	2.5× ± 0.08	0.48 ± 0.06	5.4 ± 0.4

RL Program Synthesis vs. Bayesian Optimization. A natural question arises: could the KIRI algorithm have been discovered by simply tuning the exponents (γ, α) of a pre-defined DNA metric using standard Bayesian Optimization (BO). The critical distinction is that BO is constrained to continuous hyperparameter tuning within a rigid, human-defined functional form. Our LLM-driven RL framework performs *symbolic program synthesis*. The agent did not merely tune the exponents; it autonomously wrote the Python AST to construct the cubic transition schedule and deduced the multiplicative structural norms ($\|W_{i:}\|_2 \cdot \|W_{:j}\|_2$) from scratch. Attempting to match KIRI's performance with a BO search over a standard Wanda baseline yields a maximum CIDEr of 80.4 (substantially below KIRI's 84.6), proving that topological discovery requires structural code-generation, not just scalar tuning.

Sparsity-Accuracy Pareto Frontier. Evaluating LLaVA-1.5 on MSCOCO across a 30%–90% sparsity spectrum validates KIRI's Pareto optimality. At moderate sparsities (30–50%), KIRI yields 1.2–1.8× decode speedups with near-dense retention (126.8 CIDEr at 30% vs. 128.2 dense using standard beam-search, contrasting the greedy-decoding used in Table 1 for search efficiency), outperforming GSOP by 2–4 points. Scaling to 70–80% sparsity, KIRI sustains robust cross-modal alignment (118.4 and 109.7 CIDEr, respectively) paired with 2.8–3.5× speedups, systematically exceeding SCOPE and SparseGPT by 3–6 points. Even at an extreme 90% sparsity (4.1× acceleration), KIRI achieves a viable 92.1 CIDEr while all baseline heuristics collapse below 85. These results (evaluated across 3 seeds,

$\sigma < 1.2$) confirm that KIRI maintains a superior, continuous latency-fidelity trade-off across all deployment constraints.

5 Conclusion

In this work, we introduced KATANA (Knowledge-Aligned Topology-Aware Neural Agents), a novel reinforcement learning framework that autonomously discovers and optimizes pruning algorithms for vision-language models. By framing algorithm design as a policy optimization problem—guided by a multi-objective reward signal and strictly evaluated within an isolated execution sandbox—KATANA overcomes the fundamental limitations of manual, human-engineered heuristics. The flagship strategy discovered by our agent, KIRI (Kernel-Integrated Reconstruction Iterator), successfully synthesizes a cubic sparsity schedule with a multimodal-aware Dual-Norm Activation (DNA) metric, rigorously safeguarding delicate cross-modal alignment.

Extensive empirical evaluations across four diverse VLM architectures (LLaVA-1.5, BLIP-2, Qwen2.5, and Llama-3.2-Vision) validate the strong performance of our automated approach. At an aggressive 70% sparsity, KIRI delivers a measured 2.8× wall-clock inference speedup while achieving competitive state-of-the-art results, systematically outperforming 16 highly competitive baselines, including recent adaptive and hybrid paradigms.

Limitations and Future Work. While highly scalable, KATANA currently targets unstructured and semi-structured sparsification. Expanding the action space to fully structured N:M sparsity for native tensor core acceleration remains a key direction (preliminary 2:4 N:M evaluations are in Appendix E). Additionally, incorporating recent activation-aware solvers (e.g., AWP [28]), post-mask QP reconstructions (e.g., OPTIMA [34]), and stage-aware POP [16] offers exciting avenues for hybrid compression. Finally, future work will extend our multi-task rewards to broader compositional reasoning benchmarks (e.g., GQA [18], NLVR2 [41]) to further validate KATANA's robust cross-modal synergies.

References

1. Agrawal, H., Desai, K., Wang, Y., Chen, X., Jain, R., Johnson, M., Batra, D., Parikh, D., Lee, S., Anderson, P.: nocaps: novel object captioning at scale. In: Int. Conf. Comput. Vis. (2019)
2. Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Zitnick, C.L., Parikh, D.: VQA: Visual question answering. In: Int. Conf. Comput. Vis. pp. 2425–2433 (2015)
3. Blalock, D., Ortiz, J.J.G., Frankle, J., Guttag, J.: What is the State of Neural Network Pruning? In: MLSys (2020)
4. Bolya, D., Fu, C.Y., Dai, X., Zhang, P., Feichtenhofer, C., Hoffman, J.: Token merging: Your ViT but faster. In: Int. Conf. Learn. Represent. (2023)
5. Cai, H., Gan, C., Wang, T., Zhang, Z., Han, S.: Once-for-All: Train one network and specialize it for efficient deployment. In: Int. Conf. Learn. Represent. (2020)

6. Chen, L., Zhao, H., Liu, T., Bai, S., Lin, J., Zhou, C., Chang, B.: An Image is Worth 1/2 Tokens After Layer 2: Plug-and-Play inference acceleration for large Vision-Language models. In: Eur. Conf. Comput. Vis. (2024)
7. DeepSeek-AI: DeepSeek-Coder-V2: Breaking the barrier of closed-source models in code intelligence. arXiv preprint arXiv:2406.11931 (2024)
8. DeepSeek-AI, Liu, A., Feng, B., Xue, B., Wang, B., et al.: DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. arXiv preprint arXiv:2501.12948 (2025)
9. Deng, J., Li, W., Zhou, J.T., He, Y.: SCOPE: Saliency-coverage oriented token pruning for efficient multimodal LLMs. In: Adv. Neural Inform. Process. Syst. (2025)
10. Ding, Y., Qu, W., Geng, J., Zhang, T., Shao, W., Fu, Y.: Unifying local metric and global feedback for scalable sparse LLMs. In: Int. Conf. Learn. Represent. (2025)
11. Endo, M., Wang, X., Yeung-Levy, S.: FEATHER the throttle: Revisiting visual token pruning for vision-language model acceleration. In: Int. Conf. Comput. Vis. (2025)
12. Frantar, E., Alistarh, D.: SparseGPT: Massive Language Models Can Be Accurately Pruned in One-Shot. In: Int. Conf. Mach. Learn. (2023)
13. Fu, C., Chen, P., Shen, Y., Qin, Y., Zhang, M., Lin, X., Yang, J., Zheng, X., Li, K., Sun, X., et al.: MME: A comprehensive evaluation benchmark for multimodal large language models. arXiv preprint arXiv:2306.13394 (2023)
14. Gale, T., Elsen, E., Hooker, S.: The state of sparsity in deep neural networks (2019), arXiv preprint arXiv:1902.09574
15. Han, S., Pool, J., Tran, J., Dally, W.: Learning both Weights and Connections for Efficient Neural Networks. In: Adv. Neural Inform. Process. Syst. (2015)
16. He, J., Fu, Z., Wang, J., Li, Q.: POP: Prefill-only pruning for efficient large model inference. arXiv preprint arXiv:2602.03295 (2026)
17. He, Y., Lin, J., Liu, Z., Wang, H., Li, L.J., Han, S.: AMC: Automl for model compression and acceleration on mobile devices. In: Eur. Conf. Comput. Vis. (2018)
18. Hudson, D.A., Manning, C.D.: GQA: A new dataset for real-world visual reasoning and compositional question answering. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 6700–6709 (2019)
19. Kazemzadeh, S., Ordonez, V., Matten, M., Berg, T.L.: ReferItGame: Referring to Objects in Photographs of Natural Scenes. In: EMNLP (2014)
20. LeCun, Y., Denker, J., Solla, S.: Optimal Brain Damage. In: Adv. Neural Inform. Process. Syst. (1990)
21. Li, J., Li, D., Savarese, S., Hoi, S.: BLIP-2: Boostrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models. In: Int. Conf. Mach. Learn. (2023)
22. Li, J., Li, D., Xiong, C., Hoi, S.: BLIP: Boostrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation. In: Int. Conf. Mach. Learn. (2022)
23. Li, Y., Du, Y., Zhou, K., Wang, J., Zhao, W.X., Wen, J.R.: Evaluating object hallucination in large Vision-Language models. In: EMNLP. pp. 292–305 (2023)
24. Liang, Y., Wang, Z., Xu, X., Zhou, J., Lu, J.: EfficientLLaVA: Generalizable auto-pruning for large Vision-Language models. In: IEEE Conf. Comput. Vis. Pattern Recog. (2025)
25. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: Common objects in context. In: Eur. Conf. Comput. Vis. (2014)

26. Liu, A., Tan, R., Gong, B., Plummer, B.A.: Beyond token pruning: Operation pruning in Vision-Language models. arXiv preprint arXiv:2507.02909 (2025)
27. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual Instruction Tuning. In: Adv. Neural Inform. Process. Syst. (2023)
28. Liu, J., Koike-Akino, T., Wang, Y., Mansour, H., Brand, M.: AWP: Activation-aware weight pruning and quantization with projected gradient descent. In: ICML Workshops (2025)
29. Liu, N., Ma, X., Xu, Z., Wang, Y., Tang, J., Ye, J.: AutoCompress: An automatic DNN structured pruning framework for ultra-high compression rates. In: Proc. AAAI Conf. Artif. Intell. (2020)
30. Liu, S., He, B., Wu, H., Song, L.: OptiShear: Towards efficient and adaptive pruning of large language models via evolutionary optimization. arXiv preprint arXiv:2502.10735 (2025)
31. Ma, X., Fang, G., Wang, X.: LLM-Pruner: On the structural pruning of large language models. In: Adv. Neural Inform. Process. Syst. (2023)
32. Ma, Y.J., Liang, W., Wang, G., Huang, D.A., Bastani, O., Jayaraman, D., Zhu, Y., Fan, L., Anandkumar, A.: Eureka: Human-level reward design via coding large language models. In: Int. Conf. Learn. Represent. (2024)
33. Meta AI: The Llama 3 herd of models. arXiv preprint arXiv:2407.21783 (2024)
34. Mozaffari, M., Kushnir, S., Dehnavi, M.M., Yazdanbakhsh, A.: OPTIMA: Optimal one-shot pruning for LLMs via quadratic programming reconstruction. arXiv preprint arXiv:2512.13886 (2025)
35. Mozaffari, M., Yazdanbakhsh, A., Dehnavi, M.M.: SLiM: One-shot quantization and sparsity with low-rank approximation for LLM weight compression. In: Int. Conf. Mach. Learn. (2025)
36. Muñoz, J.P., Yuan, J., Jain, N.: Shears: Unstructured sparsity with neural low-rank adapter search. In: NAACL (2024)
37. Qwen Team: Qwen2 technical report. arXiv preprint arXiv:2407.10671 (2024)
38. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning Transferable Visual Models From Natural Language Supervision. In: Int. Conf. Mach. Learn. (2021)
39. Romera-Paredes, B., Barekatain, M., Novikov, A., Balog, M., Kumar, M.P., Dupont, E., Ruiz, F.J.R., Ellenberg, J.S., Wang, P., Fawzi, O., Kohli, P., Fawzi, A.: Mathematical discoveries from program search with large language models. Nature **625**(7995), 468–475 (2024)
40. Sanh, V., Wolf, T., Rush, A.: Movement Pruning: Adaptive Sparsity by Fine-Tuning. In: Adv. Neural Inform. Process. Syst. (2020)
41. Suhr, A., Zhou, S., Zhang, A., Zhang, I., Bai, H., Artzi, Y.: A corpus for reasoning about natural language grounded in photographs. In: Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL). pp. 6418–6428 (2019)
42. Sun, M., Liu, Z., Bair, A., Kolter, J.Z.: A simple and effective pruning approach for large language models (WANDA). In: Int. Conf. Learn. Represent. (2024)
43. Wang, B., Kindratenko, V.: RL-Pruner: Structured pruning using reinforcement learning for CNN compression and acceleration (2024), arXiv preprint arXiv:2411.06463
44. Wang, H., Xu, Y., Xu, Z., Gao, J., Liu, Y., Hu, W., Wang, K., Zhang, Z.: Each complexity deserves a pruning policy. In: Adv. Neural Inform. Process. Syst. (2025)

45. Xiong, L., Liu, N., Ren, A., Bai, Y., Fang, H., Zhang, B., Jiang, Z., Tan, Y., Liu, D.: D^2Prune: Sparsifying large language models via dual taylor expansion and attention distribution awareness. In: Proc. AAAI Conf. Artif. Intell. vol. 40, pp. 27171–27179 (2026)
46. Yang, C., Sui, Y., Xiao, J., Huang, L., Gong, Y., Li, C., Yan, J., Bai, Y., Sadayappan, P., Hu, X., Yuan, B.: TopV: Compatible token pruning with inference time optimization for fast and low-memory multimodal Vision Language model. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 19803–19813 (2025)
47. Yang, T.J., Howard, A.G., Chen, B., Zhang, X., Go, A., Sandler, M., Sze, V., Adam, H.: NetAdapt: Platform-aware neural network adaptation for mobile applications. In: Eur. Conf. Comput. Vis. (2018)
48. Young, P., Lai, A., Hodosh, M., Hockenmaier, J.: From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. Transactions of the Association for Computational Linguistics **2**, 67–78 (2014)
49. Yu, P., Wang, J., Sui, X., Ling, N., Wang, W., Jiang, W.: Efficient post-training pruning of large Language Models with statistical correction. arXiv preprint arXiv:2602.07375 (2026)
50. Zamini, M., Shukla, D.: Delta-LLaVA: Base-then-specialize alignment for token-efficient Vision-Language models. In: IEEE/CVF Winter Conf. Appl. Comput. Vis. pp. 3648–3657 (2026)
51. Zhang, Q., Cheng, A., Lu, M., Zhuo, Z., Wang, M., Cao, J., Guo, S., She, Q., Zhang, S.: [CLS] attention is all you need for training-free visual token pruning: Make VLM inference faster. arXiv preprint arXiv:2412.01818 (2024)
52. Zhang, Y., Bai, H., Lin, H., Zhao, J., Hou, L., Cannistraci, C.V.: Plug-and-Play: An efficient post-training pruning method for large language models. In: Int. Conf. Learn. Represent. (2024)
53. Zhang, Y., Fan, C.K., Ma, J., Zheng, W., Huang, T., Cheng, K., Gudovskiy, D., Okuno, T., Nakata, Y., Keutzer, K., Zhang, S.: SparseVLM: Visual token sparsification for efficient Vision-Language model inference. In: Int. Conf. Mach. Learn. (2025)