# SPEC-SPIKE: SPIKING DRAFTERS FOR VISION-LANGUAGE-ACTION SPECULATIVE DECODING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Spiking neural networks (SNNs), the native substrate of neuromorphic hardware, compute with sparse spike events and need far less synaptic arithmetic than dense networks. Speculative decoding of vision-language-action (VLA) policies relies on dense drafters whose action-token proposals a 7B-parameter verifier checks in one pass, and a spiking drafter should make these proposals far cheaper. Making only the drafter's core spiking, however, cuts the core's cost over $15\times$ but the drafter's only about $3\times$, because the dense projections that feed the core and read out its logits recur at every draft position and take $87\%$ of its analytical cost. We therefore design SPEC-SPIKE around these interfaces in two coupled steps. First, it restructures each interface by how often its input changes: context is cached per observation, token projections are tabulated, and the hidden-state and context projections are factored to low rank. Second, because this factorization costs a spiking drafter $14\%$ of its accepted prefix, against $5\%$ for a dense one, and more than halves its spike activity, SPEC-SPIKE recovers the factors on its own prediction histories. On LIBERO-Long, SPEC-SPIKE reaches mean task success not detectably different from a parameter-matched Spec-VLA-style dense drafter with the same interface optimizations ($49.93\%$ vs. $48.21\%$) at $11.35\times$ lower analytical drafter cost, and a second branch, still $6.35\times$ cheaper in drafter arithmetic than one dense branch, accepts longer prefixes than the single-branch dense drafter with fewer verifier passes. The released Spec-VLA drafter, run with official Spec-VLA and KERV code, spends $750$–$1{,}540\times$ more drafter arithmetic.

## 1 INTRODUCTION

Autoregressive vision-language-action (VLA) models map an image and an instruction to discrete action tokens (Brohan et al., 2023); OPENVLA spends one forward pass of a 7B-parameter model on each of the seven tokens of a control step (Kim et al., 2025b). Speculative decoding shortens this loop: a small drafter proposes several tokens, and the large model verifies them in one pass and keeps the longest acceptable prefix (Leviathan et al., 2023; Chen et al., 2023). For VLAs, relaxing the acceptance test to a tolerance on action bins raises the number of tokens accepted per pass (Wang et al., 2025a). The proposals, however, are not free. With a single LLaMA-block drafter on OPENVLA, pure neural speculation runs at $0.87$–$0.98\times$ the speed of autoregressive decoding on four LIBERO suites, and HeiSD names the draft model's inference overhead as the limitation of drafter-based speculation (Zheng et al., 2026b). Tree-structured candidates (Miao et al., 2024; Li et al., 2024b) add draft computation for every extra branch, so wherever proposals run on their own budget, the drafter's cost bounds how much candidate search speculation can afford.

A cheaper proposer is the natural remedy, and two routes suggest themselves. Proposers without a learned drafter retrieve demonstration actions from a database of training trajectories (Zheng et al., 2026b) or copy the previous step's action; they exploit resemblance to demonstrations or repetition in the action stream and can be combined with a learned drafter (Table 10). Spiking networks offer the second route. Their events are accumulations rather than multiply–accumulates, which spiking Transformers and language models use to reduce the estimated arithmetic cost of standalone models (Zhou et al., 2023; Yao et al., 2023; Zhu et al., 2024), and motor-control forward models, which predict the outcome of a command before slower feedback arrives, offer a design analogy for such a lightweight proposer (Wolpert & Kawato, 1998; Miall et al., 1993) (Figure 1). Placing a spiking



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

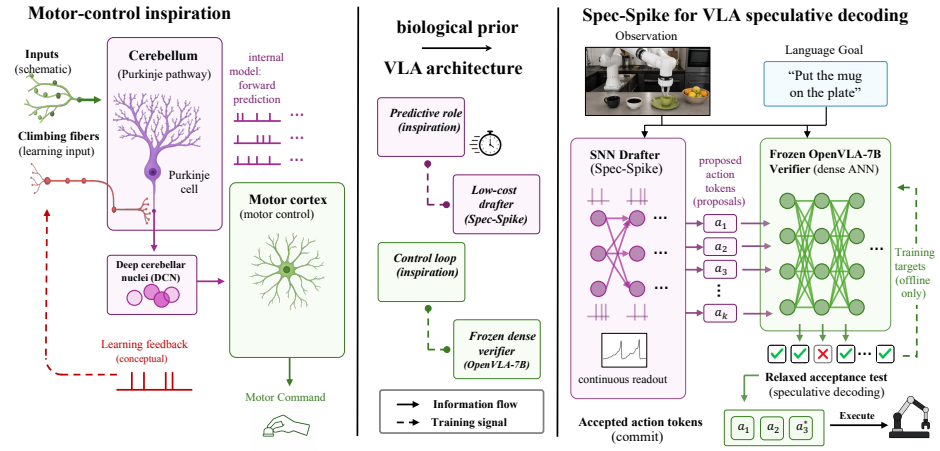



Figure 1: **A small predictor proposes; a dense verifier decides.** A lightweight spiking predictor supplies proposals to a dense verifier. Motor-control models motivate the analogy (Wolpert & Kawato, 1998; Miall et al., 1993); connections are schematic, not an anatomical circuit.

core inside the drafter, however, makes the complete drafter only about $3\times$ cheaper than a parameter-matched dense drafter in our accounting, although the core itself is more than $15\times$ cheaper.

The reason is that a drafter's cost does not lie in its core. At each of six draft positions the drafter projects the verifier's context, its previous hidden state, and the previous token into its input, and reads out logits through a dense head. In an unmodified spiking drafter these interface products account for about $87\%$ of the analytical synaptic cost, so the core's savings barely reach the drafter. Which of these dense computations must be repeated for every proposal, and does the answer differ between spiking and dense drafters?

Asking how often each interface input changes answers the first question and dictates the design (Figure 2). The verifier context is fixed within an observation, so its projection is computed once per observation; the previous token ranges over a finite action vocabulary, so its projection is read from a table; the hidden state changes with every call, so its projection is compressed to low rank, as is the once-per-observation context projection. The second question has a less expected answer. Low-rank compression that costs the dense drafter about $5\%$ of its accepted prefix costs the spiking drafter $14\%$ and more than halves its spike activity. We therefore recover the compressed factors on the original drafter's own prediction histories, the inputs it meets when it drafts, which restores both activity and acceptance. A bounded candidate tree then spends the saved budget on a second branch.

We compare SPEC-SPIKE with a parameter-matched ANN drafter in the style of Spec-VLA, reimplemented and given the same interface optimizations, so that the comparison isolates the core rather than the interfaces, and we run the official Spec-VLA system with its released drafter on the same inputs. We report the analytical synaptic cost of the drafter's own arithmetic separately from measured GPU time: the spiking operations run as dense GPU kernels, on which the spiking drafter is slower, the verifier is unchanged, and neuromorphic deployment (e.g., Loihi; Davies et al., 2018) motivates the design without being evaluated here.

**Contributions.**

- **Framework.** We propose SPEC-SPIKE, a spiking-drafter framework for speculative decoding of VLA policies that restructures every drafter interface by how often its input changes: context caching per observation, a token lookup table, and low-rank hidden-state and context projections (Section 3).
- **Analysis.** We derive an analytical cost decomposition of spiking drafters (Equation 4) and show that the dense interfaces, not the spiking core, dominate: they take $87\%$ of the cost, so a core over $15\times$ cheaper yields a drafter only $3\times$ cheaper, and compression removes $14\%$ of a spiking drafter's accepted prefix against $5\%$ of a dense one's (Sections 3.3 and 5.2).

- **Algorithm.** We introduce own-history projection recovery, which refits the compressed factors on the drafter's own prediction histories and restores the spike activity and acceptance that compression removes, and a bounded shared-prefix tree that spends the saved budget on a second candidate branch (Sections 3.4 and 3.5).
- **Experiments.** On 3 LIBERO suites with 3 training seeds each, SPEC-SPIKE lowers drafter cost $10.71$–$11.35\times$ against an equally optimized dense drafter, with lower task success detected in one of nine suite–seed pairs; a second branch accepts longer prefixes than one dense branch at $6.35\times$ lower cost; and the released Spec-VLA drafter, run with official Spec-VLA and KERV code, spends $750$–$1{,}540\times$ more drafter arithmetic (Section 5).

## 2 RELATED WORK

**Speculative decoding.** Exact speculative sampling preserves the target distribution through its acceptance test and correction distribution (Leviathan et al., 2023; Chen et al., 2023), a guarantee that relaxed acceptance gives up; blockwise parallel decoding already kept the longest prefix within a tolerance of the scoring model's greedy tokens (Stern et al., 2018). Tree-based methods verify many candidates in one pass (Miao et al., 2024; Cai et al., 2024), and drafters range from state-space models (Choi et al., 2025) and recurrent networks (Cheng et al., 2024) to backbone heads (Cai et al., 2024) and feature-space drafting (Li et al., 2024a), and speculation extends to vision-language models (Huang et al., 2025) and diffusion samplers (De Bortoli et al., 2025).

**VLA policies and their acceleration.** Parallel decoding of continuous action chunks (Kim et al., 2025a) and flow-matching action experts (Black et al., 2025; NVIDIA et al., 2025) remove the per-token loop; token-level speculation targets autoregressive action tokens, and our bin-distance rule assumes per-dimension bins rather than DCT–BPE tokens (Pertsch et al., 2025). Spec-VLA relaxes action-token matching by bin distance (Wang et al., 2025a), HeiSD combines neural and retrieval proposals (Zheng et al., 2026b), KERV adds kinematic compensation (Zheng et al., 2026a), WA-SpecDec adds world-model awareness to the prefill (Wen et al., 2026), and SV-VLA checks a heavy planner's chunks with a lightweight verifier (Wang et al., 2026b). Related ideas speculate over diffusion-policy denoising (Li et al., 2025) or cloud rollouts (Nguyen et al., 2026), prune visual tokens (Wang et al., 2026a), or distill discrete-diffusion VLAs (Song et al., 2026b). HeiSD released no official code, KERV no weights, and WA-SpecDec fine-tunes the verifier, so their published numbers are not a same-code comparison (Table 1); our control is a parameter-matched ANN, and we also run the official Spec-VLA and KERV code with the released Spec-VLA drafter (Section 5.4).

**Drafter training and history alignment.** DistillSpec finds on-policy, draft-generated data crucial for distilling drafters (Zhou et al., 2024b), and HASS aligns training and inference contexts (Zhang et al., 2025), a feature-level exposure bias (Ranzato et al., 2016; Bengio et al., 2015); our recovery instead preserves an already trained drafter after compression, using its own histories as fixed inputs.

**Spiking and low-rank computation.** Spiking Transformers replace attention products with spike-form or addition-only operations (Zhou et al., 2023; Yao et al., 2023; Zhou et al., 2024a); SpikeGPT and SpikingBrain are spiking language models of up to 216M and 76B parameters (Zhu et al., 2024; Pan et al., 2026); and SpikeVLA introduces spiking vision, language, and action-policy components for embodied navigation (Song et al., 2026a). In these works the spiking network is the task model, whereas ours only proposes to a frozen dense verifier. SpikeYOLO and NSLLM use integer-valued activations that expand into binary spikes over virtual steps (Luo et al., 2024; Xu et al., 2026); we execute the integer tensors in a GPU simulation and charge their amplitudes. Our low-rank initialization follows SVD-LLM (Wang et al., 2025b), and QP-SNN compresses SNNs (Wei et al., 2025).

## 3 SPEC-SPIKE

### 3.1 PROPOSAL AND VERIFICATION CONTRACT

A control step produces seven action tokens. The frozen OPENVLA-7B verifier supplies token $a_0$ and context $c$ after processing the image and instruction. A drafter proposes the remaining six

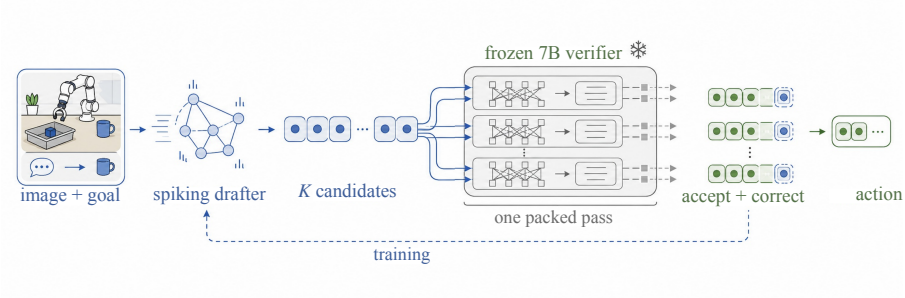



Figure 2: **One control step.** The frozen verifier encodes the image and instruction and supplies the first token and context; the SNN drafter proposes candidate action tokens, which the verifier checks in one packed pass before supplying the correction and decoding the remaining suffix. Several candidates are drawn ($K > 1$, Section 3.5); closed-loop results use $K = 1$ except the two-branch runs of Section 5.3. The dashed arrow marks offline training targets only.

tokens autoregressively. For a proposed token $\hat{a}_j$, let $b(\hat{a}_j)$ be its decoded action bin and let $a_j^\star$ be the verifier's greedy token conditioned on $a_0$ and the proposed tokens before position $j$. Following the bin-distance relaxation of Wang et al. (2025a), we accept the longest prefix satisfying

$$|b(\hat{a}_j) - b(a_j^\star)| \le r, \qquad r = 5. \tag{1}$$

At the first rejection, the verifier corrects and completes the suffix autoregressively (Figure 2). Equation 1 is approximate: it neither preserves the verifier's distribution nor guarantees task success.

The accepted-prefix length is the number of draft tokens accepted in a control step (0–6); acceptance length (AL) adds the verifier's own token $a_0$, so AL $\in [1, 7]$. Success rate (SR) is the fraction of complete robot episodes that achieve the task. AL measures local proposal agreement; SR measures closed-loop task completion. A higher AL from more candidates or relaxed acceptance is not by itself a better system, so we report AL with proposal cost, and with verifier calls and SR in rollouts.

### 3.2 Spiking Core and Continuous Readout

Both drafters use hidden width $H = 4096$, frozen verifier token embeddings, and a frozen 256-row action head. Before compression, both have 184,586,240 drafter parameters, excluding the shared frozen embedding and head. The ANN follows the one-layer LLaMA drafter of Spec-VLA (Wang et al., 2025a; Touvron et al., 2023), reimplemented with a 5461-wide gated MLP for parameter matching; unlike Spec-VLA's drafter, it conditions on one verifier context vector instead of attending over the multimodal prefix, reads out only the 256 action rows, and drafts one chain ($K = 1$) or the bounded two-branch tree of Section 3.5 ($K = 2$) per round rather than a dynamic multi-round tree (Appendix A). The SNN uses a $4096 \to 16384 \to 4096$ spike-driven MLP with $T = 4$ internal steps and integer spikes in $\{0, 1, 2, 3\}$.

At action position $j$, the input is

$$x_j = W_e e(a_{j-1}) + W_h h_{j-1} + W_c c + p_j. \tag{2}$$

Here $p_j$ is a learned position embedding and $h_{j-1}$ is the drafter's previous output. An integer leaky integrate-and-fire (I-LIF) neuron (Luo et al., 2024) encodes this input before the first large linear map; a second neuron separates the two maps. At each internal step, our variant integrates its input with leak $1/2$, emits a clipped rounded value, and subtracts that value from its state. During training, the backward pass treats rounding as the identity and passes gradients only inside the clipping range $[0, 3]$, a surrogate gradient for the non-differentiable spike (Neftci et al., 2019). All integer-neuron states are reset at every draft call, so cross-token history is carried by $h_{j-1}$ rather than by a membrane. The continuous readout is

$$h_j = \mathrm{LayerNorm}\left(\frac{1}{T}\sum_{\tau=1}^{T} W_2 s_{j,\tau}^{(1)}\right), \qquad q_j = \mathrm{softmax}(W_{\mathrm{head}} h_j). \tag{3}$$

This is the time-averaged output of the second linear map, taken *before* its final integer neuron rather than decoded from final spike counts; that neuron's activity does not enter the readout and is used only for diagnostics, including the early-stop probe of Appendix I.

### 3.3 Removing Repeated Dense Interface Work

Equation 2 separates three sources of dense work. First, $W_c c$ is identical across the six draft positions and all candidate branches for one observation, so we compute it once and invalidate this cache at the next observation. Second, the action vocabulary is finite: for unchanged weights and embeddings, we precompute $P[v] = W_e e(v)$ over the 256 action-token rows, a 2 MiB BF16 table. Changes to weights, embeddings, dtype, or device invalidate it, and an out-of-range token falls back to a dense projection whose executed work is counted.

Third, we approximate each of $W_h$ and $W_c$ by its own product $UD$, where $U \in \mathbb{R}^{H\times d}$ and $D \in \mathbb{R}^{d\times H}$, using $d = 256$. We initialize the factors with data-whitened SVD on a fixed calibration set and then recover them (Section 3.4). Neither the token projection nor the output head is low-rank approximated. Both architectures receive these changes, leaving 155,226,112 drafter parameters.

For six draft calls and vocabulary size $V = 256$, the interface MAC counts outside the core are

$$\begin{aligned} C_{\text{plain}} &= 18H^2 + 6HV, \\ C_{\text{cached}} &= 7H^2 + 6HV, \\ C_{\text{compressed}} &= 14Hd + 6HV. \end{aligned} \tag{4}$$

These expressions assume lookup hits; measured profiles include any fallback. We name drafters by these cases: *cached* applies the first two changes and *compressed* all three. None of this reduction comes from the spiking core (Section 5.2 prices each case).

### 3.4 Projection Recovery on Own-Drafter Histories

The initial drafters are trained with scheduled sampling (Bengio et al., 2015) to match verifier features and action bins, using hidden-state SmoothL1 plus hard-bin cross-entropy, and the SNN with an activity penalty.

After factorization, we freeze the original core, token projection, normalization, and output head. For each recovery observation, we run the original uncompressed drafter and record its own predicted tokens, hidden states, and output logits. The compressed drafter learns on those recorded inputs:

$$\mathcal{L}_{\text{rec}} = \frac{1}{6}\sum_{j=1}^{6} D_{\text{KL}}(q_j^0 \| q_j^d) + \text{SmoothL1}(h^d, h^0). \tag{5}$$

Superscripts 0 and $d$ denote original and compressed drafters. Only the four low-rank factor matrices are updated, for 20 epochs (14,260 updates), with the same observations, data order, and optimizer for both architectures and each architecture's own teacher histories. Because the inputs are the original drafter's histories, this is not on-policy training of the compressed drafter. The 20-epoch checkpoint was fixed before its closed-loop evaluation; Appendix A records the adaptive development history.

### 3.5 Spending Proposal Budget on Candidates

EAGLE-2 constructs context-dependent draft trees using draft-model confidence (Li et al., 2024b); here we use a bounded shared-prefix tree with the same candidate rule for both ANN and SNN. At each of the six depths, we keep the greedy path and up to $K - 1$ other paths with the highest cumulative drafter log-probability ($K$ is the tree width). Shared prefixes are generated once, and each retained parent is expanded once at the next depth. For $K = 1$ and $K = 2$ this gives 6 and 11 drafter calls. All executed parents count toward cost, even if their children are pruned.

The verifier packs unique prefix nodes into one call with an ancestor-only attention mask (Miao et al., 2024) and depth-based positions. It selects the longest accepted candidate prefix under Equation 1, with deterministic tie-breaking. The correction is read from the verification logits; the first suffix pass re-feeds the selected prefix, which is charged in time and positions. Extra candidates also add packed positions, and under Equation 6 one added OpenVLA-7B position costs more arithmetic than the whole drafter saving, so extra candidates pay off only for a proposer with its own budget.

# 4 EXPERIMENTAL SETUP

**Models and episodes.** The primary study uses a frozen OPENVLA-7B verifier on the ten LIBERO-Long tasks (Kim et al., 2025b; Liu et al., 2023). SPEC-SPIKE and the Spec-VLA-style ANN baseline each run cached and compressed for three training seeds (42–44; seed 44 is a fixed replication with no selection by outcome, Appendix A), one newly executed episode from each of 50 initial states per task: 6,000 episodes over the same 500 paired initial states. Verifier-only decoding and the released Spec-VLA system, with its official code, run on the same states (Appendices B and F). All runs use $r = 5$, BF16, and paired initial-state IDs (our drafters share one policy implementation), with $K = 1$ except the two-branch runs of Section 5.3, on seven NVIDIA RTX 5090 GPUs.

The primary analysis excludes initial-state IDs used to calibrate, recover, develop, or pilot the compressed drafters, leaving 413 *decision-excluded* episodes per configuration; earlier pre-compression models were evaluated on the same states. Both drafters were distilled from the verifier's own rollouts on all 500 initial states (Appendix A), so these masks remove decision exposure, not training exposure; absolute acceptance and SR, and comparisons with systems trained elsewhere, favor our drafters, and only the matched ANN/SNN contrast shares identical exposure. SR is averaged within each task and then equally across tasks; full-500 and pilot-excluded-455 sets are sensitivity analyses. Paired intervals resample episodes within tasks (5,000 replicates), conditional on the trained models.

**Cost and time are different measurements.** We estimate analytical synaptic cost from the numbers of executed MACs, $N_{\mathrm{MAC}}$, and accumulations, $N_{\mathrm{AC}}$,

$$E_{\mathrm{syn}} = (4.6N_{\mathrm{MAC}} + 0.9N_{\mathrm{AC}})/10^9 \quad \text{mJ}, \tag{6}$$

using the FP32, 45 nm arithmetic reference of Horowitz (2014) (0.9 pJ per addition, 3.7 pJ per multiplication), as spiking-Transformer studies do (Zhou et al., 2023; 2024a). An integer spike of amplitude $s$ contributes $s$ accumulations per outgoing synapse. The profiler counts executed drafter linear maps, dense interfaces, the action head, ANN attention products, and discarded tree-parent work (the *full drafter*); elementwise updates, normalization, and memory traffic are not priced, the verifier is priced separately with the same coefficients (Table 6), and none of this is a GPU power measurement.

Policy time is synchronized over the whole control step; GPU kernels are dense even for spike-valued inputs, so the spiking drafter issues 3.9× the ANN's dense MACs (Table 6), analytical savings do not reduce GPU latency, and a device-power comparison would measure the simulation.

**Distinct evidence sets.** For matched-input attribution and candidate expansion we also use 400 fixed observations from 60 episodes over ten tasks, reused in development, not an unseen SR test.

# 5 RESULTS

## 5.1 SPEC-SPIKE AGAINST BASELINES

Table 1 sets SPEC-SPIKE beside every learned-drafter baseline we could run or find. On our verifier and initial states, compressed SPEC-SPIKE reaches mean SR $49.93 \pm 2.97\%$ with one branch and $50.22 \pm 1.30\%$ with two, against $48.21 \pm 2.64\%$ and $50.22 \pm 0.57\%$ for the parameter-matched Spec-VLA-style baseline (mean ± across-seed sample SD), at an $11.35\times$ lower analytical drafter cost with one branch (sample SD 0.06; Figure 3).

Comparable mean SR is not equivalence for every seed: SPEC-SPIKE minus the baseline is $+4.32$, $-4.67$, and $+5.51$ pp for seeds 42–44, only the last interval excluding zero, and a recorded no-decline criterion against SPEC-SPIKE's own cached control fails for seed 43 (Appendix B). SPEC-SPIKE cuts verifier passes from 7.00 per control step to 5.58 (baseline 5.50) at 3.7 ms more policy time on dense GPU kernels. On two further suites (Appendix C; seed 44 first, seeds 42 and 43 added afterwards under a protocol fixed before their runs), its analytical drafter cost is $10.99\times$ (LIBERO-Spatial) and $10.71\times$ (LIBERO-Goal) below the baseline's, three training seeds each. On the primary states no SR difference was detected on LIBERO-Spatial in any seed (SNN minus

Table 1: **SPEC-SPIKE and baselines.** Upper rows: our frozen OPENVLA-7B verifier, 413 LIBERO-Long initial states, $r = 5$, mean $\pm$ across-seed SD over three training seeds (per seed: Table 3, Appendix E). Reported rows are copied from S: Wang et al. (2025a), H: Zheng et al. (2026b), K: Zheng et al. (2026a), and W: Wen et al. (2026) with their own verifiers, acceptance in each paper's unit (Appendix F). $\Delta$SR: SR minus the same study's AR SR. Bold: best; underline: second best, per column among the speculative methods run on our verifier in a block. $\star$: Pareto-optimal in SR and drafter cost (no method of the block has both higher SR and lower cost). SPEC-SPIKE 's lower drafter cost holds in every seed (paired 95% intervals of the ratio exclude 1); SR differences are resolved only where Section 5.1 states. Calls: verifier passes per control step; Drafter: analytical synaptic cost per control step, not device energy.

| Method | SR (%) ↑ | $\Delta$SR | Accepted ↑ | Calls ↓ | Drafter (mJ) ↓ |
|---|---|---|---|---|---|
| LIBERO-*Long, run on our verifier and initial states* | | | | | |
| Verifier only (AR) | 53.56 | 0.00 | AL 1.00 | 7.00 | 0 |
| Spec-VLA, official[a⋆] | **53.45** | −0.12 | AL 2.06 | **4.55** | 515.4 |
| Spec-VLA-style[b], $K$=1 | $48.21 \pm 2.64$ | −5.35 | AL <u>2.64</u> | 5.50 | 3.802 |
| Spec-VLA-style[b], $K$=2 | <u>$50.22 \pm 0.57$</u> | −3.35 | AL **2.77** | <u>5.39</u> | 6.962 |
| **Spec-Spike (ours)**, $K$=1$^\star$ | $49.93 \pm 2.97$ | −3.63 | AL 2.56 | 5.58 | **0.335** |
| **Spec-Spike (ours)**, $K$=2$^\star$ | <u>$50.22 \pm 1.30$</u> | −3.35 | AL **2.77** | 5.41 | <u>0.598</u> |
| LIBERO-*Long, as reported (S/H/K/W: source table)* | | | | | |
| Spec-VLA, exact match (S T1) | 50.8 | −1.2 | L 1.67 | 4.90[†] | 535.3[†] |
| Spec-VLA, $r$=5 (S T1) | 55.0 | +3.0 | L 2.10 | 4.55[†] | 515.4[†] |
| HeiSD, neural only[c] (H T3) | 50.2 | −4.2 | AL 1.59 | n.r. | n.r. |
| HeiSD[d] (H T3) | 47.0 | −7.4 | AL 4.96 | n.r. | n.r. |
| KERV[d] (K T2) | 48.8 | −5.6 | AFEP 4.64 | 3.46[†] | 454.8[†] |
| Spec-VLA, $\rho$=10[e] (W T1) | 49.4 | −8.4 | L 2.22[†] | 4.30[†] | 502.2[†] |
| WA-SpecDec, $\rho$=10[f] (W T1) | 56.2 | −1.6 | n.r. | n.r. | n.r. |
| LIBERO-*Spatial, own verifier, three training seeds,* 440 *initial states; verifier only (AR)* 79.32% | | | | | |
| Spec-VLA-style[b], $K$=1 | $78.64 \pm 1.18$ | −0.68 | AL **2.02** | **6.04** | 3.802 |
| **Spec-Spike (ours)**, $K$=1$^\star$ | $\mathbf{80.45 \pm 0.99}$ | +1.14 | AL 1.90 | 6.14 | **0.346** |
| LIBERO-*Goal, own verifier, three training seeds,* 440 *initial states; verifier only (AR)* 77.95% | | | | | |
| Spec-VLA-style[b], $K$=1$^\star$ | $\mathbf{74.92 \pm 1.84}$ | −3.03 | AL **2.40** | **5.72** | 3.802 |
| **Spec-Spike (ours)**, $K$=1$^\star$ | $74.02 \pm 2.05$ | −3.94 | AL 2.31 | 5.80 | **0.355** |

[a]Official code and released drafter, one run; multi-round dynamic tree. [b]The parameter-matched ANN baseline: a Spec-VLA-style drafter we reimplement with the same interface optimizations (Section 3.2). [c]HeiSD's "Pure D-SD". [d]Executes some actions without verification. [e]Spec-VLA's rule as run by WA-SpecDec. [f]Verifier adapted with LoRA and a world-aware prefill bias. [†]Not reported by the source; measured by us with the official code and the released Spec-VLA drafter on our verifier (closed loop for $r$=5, the 400 fixed inputs otherwise). n.r.: not reported and not measurable without released code or weights.

ANN $+1.82 \pm 2.17$ pp across seeds); on LIBERO-Goal SPEC-SPIKE had lower SR in one of three seeds (seed 44, $-4.55$ pp, $[-8.86, -0.23]$), and SNN minus ANN is $-0.91 \pm 3.15$ pp across seeds.

### 5.2 WHY THE INTERFACE MATTERS

With plain interfaces (Table 7, one change per row) the ANN/SNN ratio is only 3.13×, because the 308 M interface and head MACs that both drafters execute outweigh the SNN core's 242 M accumulations. Caching the context and tabulating token projections raise the ratio to 5.43×, and the recovered rank-256 interfaces to 11.12× (11.35× in closed loop). Counted as operations the compressed ratio is 2.81×, and pricing an AC at 0.9 rather than 4.6 pJ supplies the remaining factor of about 4; since an amplitude-$s$ event is charged $s$ additions, a non-spiking core with the same integer activations would be charged identically, so the ratio credits sparse integer activity at this price without separating it from spiking dynamics; at the reference's 16-bit prices (1.5 pJ per MAC, 0.4 pJ per addition) the same counts give 8.8×. Rebuilding the lookup table every 150 distinct observations lowers the ratio to 8.30×, and every observation to 1.17× (Figure 5).

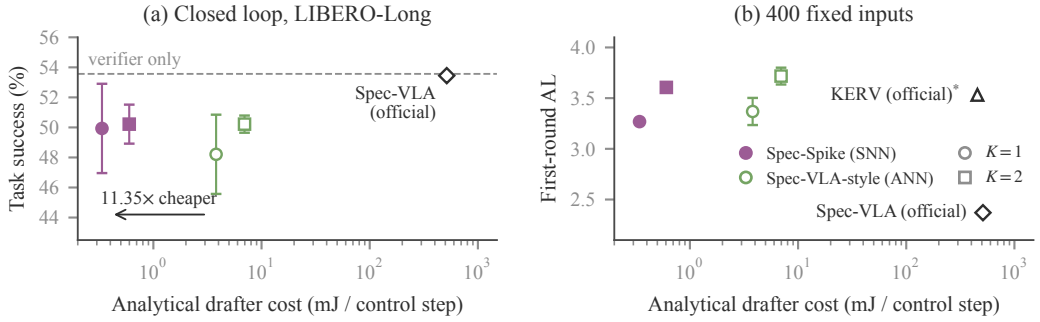



Figure 3: **Task success and AL against analytical drafter cost** (log scale). (a) Closed loop, 413 LIBERO-Long states, mean ± across-seed SD; dashed: verifier only. (b) First-round AL, 400 fixed inputs. Filled: spiking; open: dense; circles $K{=}1$, squares $K{=}2$; official Spec-VLA and KERV run released code and drafter. *Fills 13% of action dimensions from a Kalman filter without verification.

Table 2: **Interface ablation and rank on fixed inputs.** Rows two to four each add one change; row four is SPEC-SPIKE and the last two change only its rank. Retained: accepted prefix relative to the same seed's cached drafter. Mean ± across-seed SD, three seeds, 400 development inputs (not an SR test); bold: SPEC-SPIKE entries best among rows one to four.

| | Retained (%) ↑ | | Analytical cost | |
|---|---|---|---|---|
| Interface | ANN | SNN | SNN (mJ) ↓ | ANN/SNN ↑ |
| Cached, uncompressed | ref. | ref. | $0.787 \pm 0.006$ | $5.43 \pm 0.04$ |
| + rank 256, no recovery | $95.2 \pm 3.1$ | $86.2 \pm 2.7$ | $0.181 \pm 0.004$ | $21.01 \pm 0.49$ |
| + recovery, verifier histories | $102.4 \pm 2.6$ | $97.2 \pm 4.2$ | $0.322 \pm 0.007$ | $11.80 \pm 0.26$ |
| + recovery, own histories | $103.1 \pm 6.9$ | $\mathbf{102.6} \pm 2.1$ | $0.342 \pm 0.001$ | $11.11 \pm 0.04$ |
| rank 128, own histories | $99.1 \pm 2.9$ | $99.4 \pm 3.4$ | $0.305 \pm 0.002$ | $12.35 \pm 0.09$ |
| rank 512, own histories | $103.1 \pm 5.1$ | $101.1 \pm 4.2$ | $0.414 \pm 0.003$ | $9.35 \pm 0.06$ |

Rank-256 factorization alone (Table 2, three seeds) gives the highest cost ratio, 21.0×, but the SNN keeps only 86.2% of its accepted prefix against 95.2% for the ANN, and its amplitude-weighted accumulations fall from about 242 to 94 million per control step. Recovery restores both activity and acceptance, and recovery on the drafter's own histories retains more SNN acceptance than recovery on verifier histories in every seed (by 7.3, 5.9, and 2.9 points). Without recovery even rank 1024 keeps only 94.6% of the SNN prefix (seeds 42/43), so factorization and recovery form one design. Under INT8/INT4 core weights or feature noise both keep ≥95% of their prefix (Appendix H).

### 5.3 CANDIDATE EXPANSION UNDER THE SAVED BUDGET

With a second draft branch (the $K{=}2$ rows of Table 1), the compressed SNN reaches AL 2.771 per control step against 2.642 for the single-branch ANN with 5.41 rather than 5.50 verifier passes, at 6.35× lower analytical cost counting all eleven draft calls. The prefix gain over ANN $K = 1$ excludes zero for two of three seeds and over the SNN's own single branch for all three, while every per-seed SR interval against ANN $K = 1$ contains zero, so no closed-loop SR difference is detected.

At matched width no difference between the drafters is detected: with $K = 2$ the ANN and SNN reach AL 2.770 and 2.771 and both reach 50.22% mean SR, and every per-seed SR and prefix interval of their difference contains zero (Appendix E); the SNN does so at 11.64× lower analytical cost. The second branch costs the ANN 3.16 mJ per control step and the SNN 0.26 mJ, so the gain comes from affording the branch, not from a better SNN search algorithm; on fixed inputs, where the SNN trails slightly at matched width, the pattern is the same (Table 9). The extra branch grows each packed verification from 6 to 9.03 (SNN) and 9.30 (ANN) nodes and adds 1.4–2.5 ms of policy time.

Exact residual evaluation of the alternative calls, as in delta networks (O'Connor & Welling, 2017; Neil et al., 2017), raises the fixed-input ANN $K{=}1$ / SNN $K{=}2$ ratio from 6.3 to $7.8\times$ via integer activations, not spiking (Sharir & Anandkumar, 2023); see Appendix G.

### 5.4 System-Level Budget and Published Results

Per control step (Table 6), drafting takes 6.98 (ANN) and 8.53 ms (SNN) of about 128–130 ms, against 154.7 ms for autoregressive decoding in the same session. The verifier dominates analytical arithmetic: its prefill costs 8.8 J against 3.8 and 0.34 mJ for the two drafters, and speculation adds 0.13–0.14 J, so against the equally optimized baseline the drafter saving matters where proposals have their own budget, not as a cut in total arithmetic. On fixed inputs the released systems verify 174 (Spec-VLA) and 704 (KERV) nodes per step, adding 5.2 and 21.7 J from tree size, not spiking.

The released Spec-VLA system, run with its official code on our states and verifier, reaches 2.13 tokens per verification round (it reports 2.10) and 53.45% SR with 4.55 verifier passes; its drafter, counting prefix encoding, every tree round, and the full-vocabulary head, costs 515 mJ per control step, about $1{,}540\times$ Spec-Spike with one branch and $860\times$ with two, of which about $135\times$ comes from its larger drafter and tree and $11.35\times$ is the part the matched comparison attributes to the core. Paired per seed, its SR differs from two-branch Spec-Spike in no seed and exceeds one-branch Spec-Spike in one (seed 43) and the reimplemented baseline in two (Appendix F). KERV's official code with the same drafter accepts a number of tokens not detectably different from two-branch Spec-Spike on the fixed inputs with 1.2 fewer verifier passes, but spends about $750\times$ its drafter arithmetic and fills 13% of action dimensions from a Kalman filter without verification. Against verifier-only decoding (53.56%), no SR difference was detected for two-branch Spec-Spike in any seed; with one branch it had lower SR in seed 43 ($-7.06$ pp, $[-12.93, -1.25]$), as did the baseline in seeds 42 and 44 (Appendix B). Copying the previous step's bins (Table 10) and KERV's Kalman compensation cost less per query than any drafter; HeiSD's retrieval also embeds two camera views per query and stores a 2.16 GB LIBERO-Long trajectory database, about $7\times$ Spec-Spike's 0.31 GB of parameters, and HeiSD and KERV execute part of their actions without verification (Appendix F).

## 6 Discussion and Conclusion

For a spiking drafter, arithmetic efficiency is decided at the scale of the complete proposer: caching and tabulating the dense interfaces raise the ANN/SNN ratio from about $3\times$ to $5.4\times$, and compressing them with recovery to $11.35\times$ at a LIBERO-Long mean task success not detectably different from the baseline's; a second branch costs an order of magnitude less than in the ANN, with no closed-loop difference detected.

Evidence covers LIBERO-Long, LIBERO-Spatial, and LIBERO-Goal (three training seeds each), and on LIBERO-Goal Spec-Spike's SR is below the baseline's for one of the three seeds (Section 5.1); the drafters were distilled from verifier rollouts on the evaluation initial states, and the recovery recipe was selected through adaptive development. Parameter matching does not isolate spiking from architecture, activation precision, or the SNN-only activity penalty, and we did not train a compute-matched or activation-quantized ANN drafter, which could recover part or all of the ratio; the released Spec-VLA system reaches higher mean SR than Spec-Spike and the reimplemented baseline. The cost is analytical: it prices the drafter's own arithmetic with reference coefficients, excludes the unchanged verifier (Table 6) and memory traffic, which often dominates SNN inference energy in analytical models that include it (Yan et al., 2026), and requires persistent lookup-table reuse. Both drafters store the same parameters, so a substrate that re-reads every weight per call realizes far less of the ratio; a CPU event kernel drafted $2.52\times$ faster at 16 threads, $1.22\times$ slower on one (Appendix D).

## Reproducibility Statement

Section 4 and Appendices A–D and F give the architectures, parameter counts, training and recovery objectives, optimizer settings, update counts, seeds, analysis masks, bootstrap procedure, and the operation-counting rules behind every cost. Every reported run has a recorded configuration,

checkpoint and code hash, paired initial-state hashes, per-episode task outputs, and cost and timing traces; the quantitative tables and figures are regenerated from these records by scripts that check each value against the recorded summaries.

## ETHICS STATEMENT

Evaluation uses simulated manipulation tasks. We make no safety guarantee for physical robots. Relaxed acceptance changes actions; on LIBERO-Long and LIBERO-Goal both learned drafters had lower mean SR than the verifier alone (Sections 5.1 and 5.4, Appendix C), and average SR is not a deployment-specific safety assessment.

## STATEMENT ON THE USE OF AI

In this work, we used generative AI tools to design and provide feedback on research methodology and experiments, to implement methods, to interpret results, to propose and refine hypotheses, to help develop the conceptual framework, and to assist with translation of the authors' notes into English. We have not used generative AI tools to generate synthetic datasets, to develop theoretical models, to formulate mathematical claims, or to provide ingredients for or write proofs; dataset cleaning and qualitative or thematic analysis are not applicable to this work. Additionally, we used generative AI tools to identify, summarize, and check literature and references, to create and edit software code, to draft and edit parts of the paper, to create and modify figures and illustrative artwork, to format references, and to suggest structure, title, and keywords. We have reviewed all AI-assisted work: every reference was verified by the authors against its original source, AI-assisted code was tested and its outputs were audited against recorded run artifacts, and quantitative plots were generated from recorded experimental measurements rather than image-model output. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

## REFERENCES

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*, volume 28, pp. 1171–1179, 2015. URL https://proceedings.neurips.cc/paper/2015/hash/e995f98d56967d946471af29d7bf99f1-Abstract.html.

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, Laura Smith, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. $\pi_0$: A vision-language-action flow model for general robot control. In *Proc. RSS*, 2025. doi: 10.15607/RSS.2025.XXI.010. arXiv:2410.24164.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Marcin Choromanski, Tianli Ding, Danny Driess, Kumar Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montserrat Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alexander Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil J. Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael S. Ryoo, Grecia Salazar, Pannag R. Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricut, Huong Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *Proc. CoRL*, volume 229 of *Proceedings of Machine Learning Research*, pp. 2165–2183, 2023. URL https://proceedings.mlr.press/v229/zitkovich23a.html. arXiv:2307.15818.

Tianle Cai, Yuhong Li, Zhengyang Geng, Hongwu Peng, Jason D. Lee, Deming Chen, and Tri Dao. Medusa: Simple LLM inference acceleration framework with multiple decoding heads. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings*

*of Machine Learning Research*, pp. 5209–5235, 2024. URL https://proceedings.mlr.press/v235/cai24b.html. arXiv:2401.10774.

Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. Accelerating large language model decoding with speculative sampling. arXiv:2302.01318, 2023. URL https://arxiv.org/abs/2302.01318.

Yunfei Cheng, Aonan Zhang, Xuanyu Zhang, Chong Wang, and Yi Wang. Recurrent drafter for fast speculative decoding in large language models. arXiv:2403.09919, 2024.

Daewon Choi, Seunghyuk Oh, Saket Dingliwal, Jihoon Tack, Kyuyoung Kim, Woomin Song, Seojin Kim, Insu Han, Jinwoo Shin, Aram Galstyan, Shubham Katiyar, and Sravan Babu Bodapati. Mamba drafters for speculative decoding. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 21404–21418, 2025. doi: 10.18653/v1/2025.findings-emnlp.1167. URL https://aclanthology.org/2025.findings-emnlp.1167/. arXiv:2506.01206.

Mike Davies, Narayan Srinivasa, Tsung-Han Lin, Gautham Chinya, Yongqiang Cao, Sri Harsha Choday, Georgios Dimou, Prasad Joshi, Nabil Imam, Shweta Jain, Yuyun Liao, Chit-Kwan Lin, Andrew Lines, Ruokun Liu, Deepak Mathaikutty, Steven McCoy, Arnab Paul, Jonathan Tse, Guruguhanathan Venkataramanan, Yi-Hsin Weng, Andreas Wild, Yoonseok Yang, and Hong Wang. Loihi: A neuromorphic manycore processor with on-chip learning. *IEEE Micro*, 38(1):82–99, 2018. doi: 10.1109/MM.2018.112130359.

Valentin De Bortoli, Alexandre Galashov, Arthur Gretton, and Arnaud Doucet. Accelerated diffusion models via speculative sampling. In *Proc. ICML*, volume 267 of *Proceedings of Machine Learning Research*, pp. 12590–12631, 2025. arXiv:2501.05370.

Mark Horowitz. Computing's energy problem (and what we can do about it). In *Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC)*, pp. 10–14, 2014.

Haiduo Huang, Fuwei Yang, Zhenhua Liu, Xuanwu Yin, Dong Li, Pengju Ren, and Emad Barsoum. SpecVLM: Fast speculative decoding in vision-language models. arXiv:2509.11815, 2025.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. In *Proc. Robotics: Science and Systems (RSS) XXI*, Los Angeles, CA, USA, June 2025a. doi: 10.15607/RSS.2025.XXI.017. URL https://www.roboticsproceedings.org/rss21/p017.html. arXiv:2502.19645.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P. Foster, Pannag R. Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An open-source vision-language-action model. In *Proc. CoRL*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2679–2713, 2025b. URL https://proceedings.mlr.press/v270/kim25c.html. arXiv:2406.09246.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *Proc. ICML*, volume 202 of *Proceedings of Machine Learning Research*, pp. 19274–19286, 2023. URL https://proceedings.mlr.press/v202/leviathan23a.html.

Ye Li, Jiahe Feng, Yuan Meng, Kangye Ji, Chen Tang, Xinwan Wen, Shutao Xia, Zhi Wang, and Wenwu Zhu. TS-DP: Reinforcement speculative decoding for temporal adaptive diffusion policy acceleration. arXiv:2512.15773, 2025.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. EAGLE: Speculative sampling requires rethinking feature uncertainty. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 28935–28948, 2024a. URL https://proceedings.mlr.press/v235/li24bt.html.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. EAGLE-2: Faster inference of language models with dynamic draft trees. In *Proceedings of the 2024 Conference on Empirical*

*Methods in Natural Language Processing*, pp. 7421–7432. Association for Computational Linguistics, 2024b. doi: 10.18653/v1/2024.emnlp-main.422. URL https://aclanthology.org/2024.emnlp-main.422/.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 44776–44791, 2023. doi: 10.52202/075280-1939. URL https://papers.nips.cc/paper_files/paper/2023/hash/8c3c666820ea055a77726d66fc7d447f-Abstract-Datasets_and_Benchmarks.html.

Xinhao Luo, Man Yao, Yuhong Chou, Bo Xu, and Guoqi Li. Integer-valued training and spike-driven inference spiking neural network for high-performance and energy-efficient object detection. In *Proc. ECCV*, 2024. URL https://www.ecva.net/papers/eccv_2024/papers_ECCV/html/4704_ECCV_2024_paper.php. arXiv:2407.20708.

R. C. Miall, D. J. Weir, D. M. Wolpert, and J. F. Stein. Is the cerebellum a Smith predictor? *Journal of Motor Behavior*, 25(3):203–216, 1993. doi: 10.1080/00222895.1993.9942050.

Xupeng Miao, Gabriele Oliaro, Zhihao Zhang, Xinhao Cheng, Zeyu Wang, Zhengxin Zhang, Rae Ying Yee Wong, Alan Zhu, Lijie Yang, Xiaoxiang Shi, Chunan Shi, Zhuoming Chen, Daiyaan Arfeen, Reyna Abhyankar, and Zhihao Jia. SpecInfer: Accelerating large language model serving with tree-based speculative inference and verification. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 3*, ASPLOS '24, pp. 932–949, New York, NY, USA, 2024. Association for Computing Machinery. doi: 10.1145/3620666.3651335. URL https://doi.org/10.1145/3620666.3651335.

Yu Nasu. NNUE: Efficiently updatable neural-network-based evaluation functions for computer Shogi. Appeal document, The 28th World Computer Shogi Championship (WCSC28), April 2018. URL https://www.apply.computer-shogi.org/wcsc28/appeal/the_end_of_genesis_T.N.K.evolution_turbo_type_D/nnue.pdf. Ziosoft Computer Shogi Club. In Japanese with English abstract.

Emre O. Neftci, Hesham Mostafa, and Friedemann Zenke. Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks. *IEEE Signal Processing Magazine*, 36(6):51–63, 2019. doi: 10.1109/MSP.2019.2931595. arXiv:1901.09948.

Daniel Neil, Jun Haeng Lee, Tobi Delbruck, and Shih-Chii Liu. Delta networks for optimized recurrent network computation. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 2584–2593, Sydney, Australia, 2017. PMLR. URL https://proceedings.mlr.press/v70/neil17a.html.

Chanh Nguyen, Shutong Jin, Florian T. Pokorny, and Erik Elmroth. Speculative policy orchestration: A latency-resilient framework for cloud-robotic manipulation. arXiv:2603.19418, 2026.

NVIDIA, Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Zhenyu Jiang, Jan Kautz, Kaushil Kundalia, Lawrence Lao, Zhiqi Li, Zongyu Lin, Kevin Lin, Guilin Liu, Edith Llontop, Loic Magne, Ajay Mandlekar, Avnish Narayan, Soroush Nasiriany, Scott Reed, You Liang Tan, Guanzhi Wang, Zu Wang, Jing Wang, Qi Wang, Jiannan Xiang, Yuqi Xie, Yinzhen Xu, Zhenjia Xu, Seonghyeon Ye, Zhiding Yu, Ao Zhang, Hao Zhang, Yizhou Zhao, Ruijie Zheng, and Yuke Zhu. GR00T N1: An open foundation model for generalist humanoid robots. arXiv:2503.14734, 2025.

Peter O'Connor and Max Welling. Sigma delta quantized networks. In *International Conference on Learning Representations (ICLR)*, 2017. URL https://openreview.net/forum?id=HkNRsU5ge.

Yuqi Pan, Yupeng Feng, Jinghao Zhuang, Siyu Ding, Han Xu, Zehao Liu, Bohan Sun, Yuhong Chou, Xuerui Qiu, Anlin Deng, Anjie Hu, Shurong Wang, Peng Zhou, Man Yao, Jibin Wu, Jian Yang, Guoliang Sun, Bo Xu, and Guoqi Li. SpikingBrain: Spiking brain-inspired large models. *Transactions on Machine Learning Research*, 2026. arXiv:2509.05276.

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. In *Proc. Robotics: Science and Systems (RSS XXI)*, Los Angeles, CA, USA, 2025. doi: 10.15607/RSS.2025.XXI.012. URL https://www.roboticsproceedings.org/rss21/p012.html. arXiv:2501.09747.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. In *Proc. ICLR*, 2016. arXiv:1511.06732.

Or Sharir and Anima Anandkumar. Incrementally-computable neural networks: Efficient inference for dynamic inputs, 2023. Presented at the ICML 2023 Workshop on Efficient Systems for Foundation Models (ES-FoMo).

Ruiqi Song, Dujun Nie, Siyu Teng, Baiyong Ding, Xiaotong Zhang, Dong Li, Chenming Zhang, Yuchen Li, Hangbin Wu, and Long Chen. SpikeVLA: Vision-language-action models with spiking neural networks. In *Proc. ICML*, 2026a. URL https://arxiv.org/abs/2606.27807v1. arXiv:2606.27807.

Wenxuan Song, Jiayi Chen, Shuai Chen, Jingbo Wang, Pengxiang Ding, Han Zhao, Yikai Qin, Xinhu Zheng, Donglin Wang, Yan Wang, and Haoang Li. Fast-dVLA: Accelerating discrete diffusion VLA to real-time performance. In *Computer Vision – ECCV 2026*, Lecture Notes in Computer Science, pp. 361–378. Springer Nature Switzerland, 2026b. doi: 10.1007/978-3-032-37013-6_20. arXiv:2603.25661.

Mitchell Stern, Noam Shazeer, and Jakob Uszkoreit. Blockwise parallel decoding for deep autoregressive models. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/hash/c4127b9194fe8562c64dc0f5bf2c93bc-Abstract.html. arXiv:1811.03115.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. arXiv:2307.09288, 2023. URL https://arxiv.org/abs/2307.09288.

Hanzhen Wang, Jiaming Xu, Yushun Xiang, Jiayi Pan, Yongkang Zhou, Yong-Lu Li, and Guohao Dai. SpecPrune-VLA: Accelerating vision-language-action models via action-aware self-speculative pruning. In *Proc. ICML*, 2026a. arXiv:2509.05614.

Songsheng Wang, Rucheng Yu, Zhihang Yuan, Chao Yu, Feng Gao, Yu Wang, and Derek F. Wong. Spec-VLA: Speculative decoding for vision-language-action models with relaxed acceptance. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 26928–26940, 2025a. doi: 10.18653/v1/2025.emnlp-main.1367. URL https://aclanthology.org/2025.emnlp-main.1367/. arXiv:2507.22424.

Xin Wang, Yu Zheng, Zhongwei Wan, and Mi Zhang. SVD-LLM: Truncation-aware singular value decomposition for large language model compression. In *International Conference on Learning Representations*, 2025b. URL https://mi-zhang.github.io/papers/2025_ICLR_SVD-LLM.pdf.

Zihua Wang, Zhitao Lin, Ruibo Li, Yu Zhang, Xu Yang, Siya Mi, and Xiu-Shen Wei. Open-loop planning, closed-loop verification: Speculative verification for VLA. arXiv:2604.02965, 2026b.

Wenjie Wei, Malu Zhang, Zijian Zhou, Ammar Belatreche, Yimeng Shan, Yu Liang, Honglin Cao, Jieyuan Zhang, and Yang Yang. QP-SNN: Quantized and pruned spiking neural networks. In *International Conference on Learning Representations*, pp. 96470–96495, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/f04aef8513b788a69229cede8c8cb5a8-Paper-Conference.pdf. arXiv:2502.05905.

Zikang Wen, Yuning Zhang, and Dong Yuan. WA-SpecDec: World-aware speculative decoding for vision-language-action models. arXiv:2608.08725v1, 2026. URL https://arxiv.org/abs/2608.08725v1.

Daniel M. Wolpert and Mitsuo Kawato. Multiple paired forward and inverse models for motor control. *Neural Networks*, 11(7–8):1317–1329, 1998. doi: 10.1016/S0893-6080(98)00066-5.

Han Xu, Xuerui Qiu, Yunhui Xu, Mohammed E. Elbtity, Peng Zhou, Yang Tian, Rui-Jie Zhu, Jiahong Zhang, Shaowei Gu, Yuqi Pan, Yuhong Chou, Qinghao Wen, Man Yao, Jiangbo Qian, Yonghong Tian, Lei Ma, Tiejun Huang, Jason K. Eshraghian, Bo Xu, and Guoqi Li. Neuromorphic spike-based large language model. *National Science Review*, 13(4):nwaf551, 2026. doi: 10.1093/nsr/nwaf551.

Zhanglu Yan, Zhenyu Bai, Kaiwen Tang, and Weng-Fai Wong. Reconsidering the energy efficiency of spiking neural networks inference from analytical perspectives. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2026. doi: 10.1109/TCAD.2026.3718799. Early access.

Man Yao, JiaKui Hu, Zhaokun Zhou, Li Yuan, Yonghong Tian, Bo Xu, and Guoqi Li. Spike-driven Transformer. In *Proc. NeurIPS*, 2023.

Lefan Zhang, Xiaodan Wang, Yanhua Huang, and Ruiwen Xu. Learning harmonized representations for speculative sampling. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=T9u56s7mbk. arXiv:2408.15766.

Zihao Zheng, Zhihao Mao, Maoliang Li, Jiayu Chen, Xinhao Sun, Zhaobo Zhang, Donggang Cao, Hong Mei, and Xiang Chen. KERV: Kinematic-rectified speculative decoding for embodied VLA models. In *Proc. 63rd ACM/IEEE Design Automation Conference (DAC)*, 2026a. URL https://arxiv.org/abs/2603.01581v2. arXiv:2603.01581.

Zihao Zheng, Zhihao Mao, Sicheng Tian, Jiayu Chen, Maoliang Li, Xinhao Sun, Zhaobo Zhang, Xuanzhe Liu, Donggang Cao, Hong Mei, and Xiang Chen. HeiSD: Hybrid speculative decoding for embodied vision-language-action models with kinematic awareness. arXiv:2603.17573v2, 2026b. URL https://arxiv.org/abs/2603.17573v2.

Chenlin Zhou, Han Zhang, Zhaokun Zhou, Liutao Yu, Liwei Huang, Xiaopeng Fan, Li Yuan, Zhengyu Ma, Huihui Zhou, and Yonghong Tian. QKFormer: Hierarchical spiking transformer using Q-K attention. In *Proc. NeurIPS*, 2024a. arXiv:2403.16552.

Yongchao Zhou, Kaifeng Lyu, Ankit Singh Rawat, Aditya Krishna Menon, Afshin Rostamizadeh, Sanjiv Kumar, Jean-François Kagy, and Rishabh Agarwal. DistillSpec: Improving speculative decoding via knowledge distillation. In *International Conference on Learning Representations*, pp. 32011–32050, 2024b. URL https://openreview.net/forum?id=rsY6J3ZaTF. arXiv:2310.08461.

Zhaokun Zhou, Yuesheng Zhu, Chao He, Yaowei Wang, Shuicheng Yan, Yonghong Tian, and Li Yuan. Spikformer: When spiking neural network meets transformer. In *Proc. ICLR*, 2023.

Rui-Jie Zhu, Qihang Zhao, Guoqi Li, and Jason K. Eshraghian. SpikeGPT: Generative pre-trained language model with spiking neural networks. *Transactions on Machine Learning Research*, 2024. arXiv:2302.13939.

# A TRAINING, RECOVERY, AND SELECTION HISTORY

**Initial drafters.** Both matched starting models have $184{,}586{,}240$ parameters and fixed $60{,}000$-update checkpoints for seeds $42$, $43$, and $44$. ANN has one LLaMA-style layer, $32$ heads, and intermediate width $5461$; SNN has hidden width $4096$, MLP width $16384$, four internal steps, and integer level count four. Shared embeddings and the action head are frozen. Training data are the verifier's greedy closed-loop rollouts from all $500$ initial states of the ten tasks ($198{,}005$ steps), recorded with last-layer hidden states and action tokens. Relative to the Spec-VLA drafter (Wang et al., 2025a), the ANN keeps one LLaMA decoder layer of width $4096$ with $32$ heads, feature SmoothL1 plus $0.1$-weighted token loss, and the same learning rate, batch size, warmup, and gradient clipping, but it does not attend over the multimodal prefix, uses a $5461$-wide MLP, a $256$-row action head with hard-bin targets, verifier rollouts instead of relabeled demonstrations, scheduled sampling instead of feature noise, and one chain ($K = 1$) or the bounded two-branch tree ($K = 2$) per round instead of a dynamic multi-round tree. The initial trainer uses hidden SmoothL1, hard-bin cross-entropy, and scheduled sampling, plus the SNN's uniform activity penalty of $1$. The internal-step count, the four integer levels, and the penalty weight were set during earlier development, the weight from a sweep that traded spike density against acceptance, and none is ablated under the present recipe. The feature and token loss weights are $1$ and $0.1$. Training uses batch size $16$, AdamW learning rate $5 \times 10^{-5}$, betas $(0.9, 0.95)$, zero weight decay, and gradient clipping $0.5$. After $2{,}000$ warmup steps, the learning rate follows a cosine schedule to $10\%$ of its peak. The self-fed input probability increases linearly to $0.75$ at step $30{,}000$ and stays there. No KL term is used in that initial objective; Equation 5 describes a later training stage.

**Factorization.** We use a damped variant of the data-whitened factorization of SVD-LLM (Wang et al., 2025b). For each projection, form $\Sigma = X^\top X/|X| + \epsilon I$, with $\epsilon = 10^{-4}\,\mathrm{mean}(\mathrm{diag}(X^\top X/|X|))$. Let $LL^\top = \Sigma$ and decompose $WL$. With retained singular factors $\widetilde{U}_d, S_d, \widetilde{V}_d^\top$,

$$U = \widetilde{U}_d S_d^{1/2}, \qquad D = S_d^{1/2} \widetilde{V}_d^\top L^{-1}.$$

Calibration uses a fixed set of verifier-derived observations from the original training data. Factorization is in double precision before BF16 conversion. Damping is frozen, not validation-tuned.

**Recovery.** Thirty trajectories provide $11{,}423$ underlying observations. Each original ANN/SNN supplies its own six-position histories and production-shape target logits. Only hidden down/up and context down/up factors are updated: 20 epochs, $14{,}260$ updates, batch size $16$, AdamW learning rate $10^{-5}$, betas $(0.9, 0.95)$, zero weight decay, and clipping $0.5$. Forward/backward execution is BF16; FP32 master weights and optimizer state are rounded back after each update. KL temperature and both loss weights are one. There is no new sparsity loss or scheduled sampling during recovery. All frozen core, token-projection, and readout tensors remain unchanged.

**Adaptive development.** The archive contains $64$ recovery training attempts through the selected 20-epoch round, including eight early attempts invalidated by a data-shuffle defect. Completed alternatives include rank-$256/512$ five-epoch recovery, hard-CE recovery, joint recovery with preservation losses, projection-only recovery, and a 20-epoch projection-only precursor. The rank-$256$ and rank-$512$ five-epoch candidates each completed a closed-loop pilot of $360$ comparisons on the pilot initial states and failed its $-5$ pp gates (seed-43 ANN $-6.67$ pp against its cached control at rank $256$; seed-42 SNN minus ANN $-6.67$ pp and seed-43 ANN $-13.33$ pp at rank $512$); the auxiliary action filters below were introduced after these failures, and every primary mask excludes the pilot states. The last precursor passed its action/cost filters but retained $94.45\%$ of seed-42 SNN prefix, below its $95\%$ threshold. A fixed midpoint interpolation was tested without counting it as training. Failures remain archived as failures.

An auxiliary exact-action/bin-error filter rejected the eventual own-history candidate: seed-42 SNN action agreement was $70.25\%$ versus control $71.50\%$, and bin MAE was $0.7446$ versus $0.7368$. Before the direct-SR pilot these auxiliary filters were changed to diagnostics, with prefix-retention and cost gates retained. The pilot SR criteria did not change. This prospective protocol change does not reclassify earlier failures. Checkpoints were frozen before the pilot and the $4{,}000$-episode study of seeds $42$ and $43$. The pilot used $180$ new candidate episodes and $180$ audited reused controls; the full study reran all configurations.

After viewing that study, a fixed 40-epoch candidate was explored. Its completed pilot failed: seed-43 SNN minus ANN SR was $-8.89$ pp. A further 40-epoch candidate that exposed recovery to the compressed drafter's own histories with probability $0.5$ also completed its pilot and failed the same unchanged criterion, again at $-8.89$ pp for seed 43, despite improving on its own cached control for both seeds. Neither candidate received a full rollout study, and neither replaces the 20-epoch results here. Future candidates selected using the present confirmation outcomes require newly isolated evaluation, rather than relabeling those outcomes as fresh holdout evidence.

Seed 44 was then added as a fixed additional-seed replication: both original drafters were independently initialized and trained with the archived recipe to the same $60{,}000$-update endpoint, recovered with the same 20-epoch projection-only procedure on their own histories, and evaluated in all four configurations for $2{,}000$ new episodes, with no seed replacement, checkpoint choice, or gate applied to its outcome. The seed-44 data loader used an explicitly isolated random stream to match ANN/SNN data order, whereas the archived earlier seeds used the global loader stream; this difference is disclosed rather than assumed away. Because the benchmark and recipe were informed by the first two seeds, the extension is an additional-seed replication on a reused benchmark, not an untouched test set.

## B Closed-Loop Analysis and Sensitivity

The full study is four configurations, three training seeds, ten tasks, and 50 initial states per task. Actual initial-state hashes and shared numerical code are checked across paired runs. Each GPU runs one worker, with bounded CPU thread counts and matched rendering-device affinity. Episodes allow at most 520 control steps. The first three steps are excluded from timing/cost means, not SR. Episode means are averaged within task, then across tasks; across-seed means and sample SDs use the three task-macro values.

Primary 413 excludes every decision-exposed initial-state ID. The 455 sensitivity excludes pilot IDs only; full-500 includes development-exposed states and is not independently held out. Masks are fixed lists, not selected by observed success. Table 3 reports every seed and configuration on the primary set, and Table 4 reports all three masks for every seed.

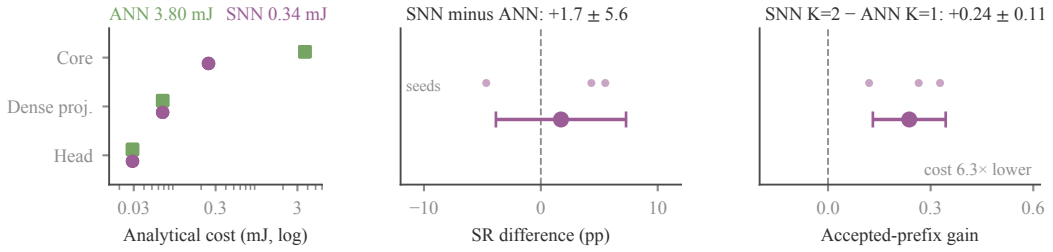
Table 3: **Per-seed closed-loop results.** Per-seed $K=1$ results, cached and compressed; the compressed rows underlie Table 1. Primary decision-excluded 413 episodes per configuration and training seed, equal task weights. Seed-level differences change sign; the aggregate is not evidence that any individual seed improves. Arrows give the preferred direction; bold marks the better value within each matched ANN/SNN pair of a seed.

| Seed | Drafter | SR (%) ↑ | AL ↑ | Cost (mJ) ↓ | Policy (ms) ↓ |
|---|---|---|---|---|---|
| 42 | ANN, cached | 46.39 | **2.654** | 4.274 | **138.05** |
| 42 | SNN, cached | **48.51** | 2.583 | **0.790** | 144.58 |
| 42 | ANN, compressed | 47.36 | **2.684** | 3.802 | **138.91** |
| 42 | SNN, compressed | **51.68** | 2.536 | **0.336** | 143.33 |
| 43 | ANN, cached | **52.37** | **2.619** | 4.274 | **139.08** |
| 43 | SNN, cached | 47.01 | 2.543 | **0.781** | 142.50 |
| 43 | ANN, compressed | **51.17** | **2.599** | 3.802 | **140.38** |
| 43 | SNN, compressed | 46.50 | 2.597 | **0.333** | 142.56 |
| 44 | ANN, cached | **50.49** | **2.645** | 4.274 | **139.70** |
| 44 | SNN, cached | 45.77 | 2.551 | **0.783** | 142.70 |
| 44 | ANN, compressed | 46.10 | **2.643** | 3.802 | **139.25** |
| 44 | SNN, compressed | **51.62** | 2.546 | **0.336** | 143.79 |

Compressed SNN minus ANN SR is $+4.32$ pp (95% CI $[-1.07, 9.91]$) for seed 42, $-4.67$ pp ($[-10.47, 1.07]$) for seed 43, and $+5.51$ pp ($[0.11, 11.20]$) for seed 44 (Figure 4b). Relative to its own cached control, compression with recovery raised SNN SR in two of three seeds and lowered ANN SR in two of three; the only interval excluding zero, seed 44, coincides with the largest ANN

Table 4: **SR sensitivity to the analysis mask.** Percent success (↑) with equal task weights, all training seeds. Controls and candidates are newly executed in the same study. Bold marks the better value within each matched ANN/SNN pair.

| Set | Seed | Cached ANN | Cached SNN | Comp. ANN | Comp. SNN |
|---|---|---|---|---|---|
| Primary 413 | 42 | 46.39 | **48.51** | 47.36 | **51.68** |
| Primary 413 | 43 | **52.37** | 47.01 | **51.17** | 46.50 |
| Primary 413 | 44 | **50.49** | 45.77 | 46.10 | **51.62** |
| Pilot-excluded 455 | 42 | 47.37 | **50.14** | 48.66 | **51.80** |
| Pilot-excluded 455 | 43 | **52.14** | 47.34 | **52.71** | 47.40 |
| Pilot-excluded 455 | 44 | **50.71** | 47.40 | 45.49 | **51.17** |
| Full 500 | 42 | 47.00 | **49.20** | 48.60 | **51.80** |
| Full 500 | 43 | **51.80** | 47.00 | **52.40** | 47.80 |
| Full 500 | 44 | **50.40** | 45.40 | 45.40 | **51.80** |




(a) Where the cost goes (b) Paired SR differences (c) Budget buys longer prefixes

Figure 4: **Three views of the tradeoff.** (a) Fixed-input analytical cost per component of the compressed drafters (log axis; the three seeds overlap; squares ANN, circles SNN): the dense projections and action head are shared, so only the core differs. (b) Compressed SNN minus ANN SR and (c) fixed-input prefix gain for SNN $K = 2$ versus ANN $K = 1$, each as the across-seed mean with its sample standard deviation; small markers are individual seeds. All panels cover the three training seeds; (a,c) use development inputs, not independent SR evidence.

loss ($-4.38$ pp), and against the cached ANN the seed-44 SNN gain is $+1.13$ pp. Figure 4 also shows the fixed-input cost per component (Table 7) and the per-seed tree gains (Table 9).

Across seeds, the primary ANN/SNN cost ratios are $11.32$ ($[11.251, 11.399]$), $11.42$ ($[11.348, 11.490]$), and $11.31$ ($[11.238, 11.389]$). Their narrowness concerns episode variation in the arithmetic model, not hardware uncertainty or training-seed generalization. The headline $11.35\times$ is the mean of seed-specific ratios (sample SD $0.06$), not arbitrary per-step ratios; the ratio of across-seed mean costs is also $11.35\times$. SNN-minus-own-control intervals are $[-2.23, 8.64]$, $[-6.31, 5.30]$, and $[0.39, 11.32]$ pp; ANN-minus-own-control intervals are $[-4.89, 6.73]$, $[-6.90, 4.55]$, and $[-10.35, 1.60]$ pp. The inherited $-5$ pp point-estimate engineering gates pass on the primary set, but the no-decline point criterion (compressed SNN SR at least its own cached control for every seed) fails for seed $43$ ($-0.51$ pp). The $455$ sensitivity has seed-43 SNN-minus-ANN ($-5.31$ pp) and seed-44 ANN-minus-own-control ($-5.23$ pp) below $-5$ pp; on full-500 the latter is $-5.00$ pp. Pilot-excluded-455 means are $50.12\%$ (SNN) and $48.95\%$ (ANN); full-500 means are $50.47\%$ and $48.80\%$. Neither gate is formal equivalence or non-inferiority. Against the cached ANN baseline, which involves no recovery recipe, the compressed SPEC-SPIKE ratio would be $12.76\times$. With three seeds, a $t$-interval for the mean compressed SNN$-$ANN difference is $[-12.1, 15.6]$ pp, and at the cached point the per-seed differences are $+2.12$, $-5.36$, and $-4.72$ pp; we claim no across-seed equivalence.

**Against verifier-only decoding.** The same-study autoregressive run ($500$ new episodes on the paired initial states) reaches $53.56\%$ on the $413$ primary states, the value of the earlier study's autoregressive arm (Table 10), as greedy decoding is deterministic. Paired per seed ($42/43/44$),

SR minus autoregressive is $-1.88$ ($[-7.29, 3.52]$), $-7.06$ ($[-12.93, -1.25]$), and $-1.95$ pp ($[-7.36, 3.69]$) for one-branch SPEC-SPIKE; $-2.97$ ($[-8.55, 2.62]$), $-2.28$ ($[-8.25, 3.40]$), and $-4.79$ pp ($[-10.65, 0.79]$) with two branches; $-6.21$ ($[-11.84, -0.56]$), $-2.40$ ($[-7.80, 3.24]$), and $-7.46$ pp ($[-13.27, -1.58]$) for the Spec-VLA-style baseline with one branch and $-3.14$, $-2.91$, and $-3.99$ pp with two, every two-branch interval containing zero; and $-0.12$ pp ($[-6.14, 5.92]$) for the released Spec-VLA system.

**Relaxed acceptance and executed actions.** On the $400$ development inputs (seeds $42/43$), the action executed after speculation matches the verifier's own single-token greedy action in all bins on $72.75/73.25\%$ of control steps for the compressed ANN and $70.25/71.75\%$ for the compressed SNN, and on $70.25$–$73.25\%$ for the cached drafters, so the deviation precedes compression. For the compressed drafters, $92.3$–$93.0\%$ of bins match, the mean absolute bin difference is $0.62$–$0.74$, and $0.24$–$0.26$ tokens per step are accepted within the tolerance without being the verifier's argmax. These inputs overlap the training rollouts, and the comparison is a diagnostic, not a prediction of task success.

## C ADDITIONAL SUITES: LIBERO-SPATIAL AND LIBERO-GOAL

LIBERO-Spatial repeats the LIBERO-Long recipe, first for training seed $44$ (final checkpoint) with no tuning and no selection; seeds $42$ and $43$ follow in the last paragraph. The verifier is the suite's own released OPENVLA checkpoint; the two released checkpoints share the token-embedding table and differ in the action head, so the drafters use this suite's head. Both drafters are trained from scratch on a new dump of verifier rollouts on all $500$ initial states ($65{,}311$ control steps). The only departures from LIBERO-Long are the suite constants (verifier checkpoint; at most $220$ control steps), about $14.7$ rather than $4.8$ passes over the smaller dump at the same $60{,}000$ base updates, and a recovery update count from the same 20-epoch rule; acceptance keeps $r = 5$, whereas Spec-VLA uses $r = 9$ on this suite (Wang et al., 2025a). Seed $44$ was fixed before any LIBERO-Spatial run because its code and audits were the most recent; its LIBERO-Long result was known then. The primary $440$ states exclude initial states 0–5 of every task, which supplied factorization calibration, recovery histories, and the fixed inputs; they are not training-unseen, because the rollout dump covers all $500$ states. Intervals are paired by task and initial state and are conditional on the trained models.

Table 5 lists every configuration. For seed $44$ on the primary set, compressed SNN minus ANN SR is $+3.18$ pp ($[-0.91, 7.27]$) at an $11.24\times$ lower analytical cost, and cached SNN minus ANN is $+3.64$ pp ($[0.00, 7.27]$, lower bound at zero) at $5.24\times$. Relative to verifier-only decoding the compressed SNN and ANN differ by $+1.82$ pp ($[-2.05, 5.91]$) and $-1.36$ pp ($[-5.45, 2.50]$), and relative to their cached controls compression with recovery changes SR by $+0.45$ and $+0.91$ pp, every interval containing zero. On all $500$ states compressed SNN minus ANN is $+2.00$ pp ($[-2.00, 5.80]$). On $400$ fixed inputs the SNN reaches AL $2.538$ per control step against $2.485$ for the ANN at $K = 1$ (difference $+0.053$, $[-0.076, 0.164]$) and $2.79$ against $2.86$ at $K = 2$ ($-0.070$, $[-0.193, 0.067]$); SNN $K = 2$ exceeds ANN $K = 1$ by $0.305$ tokens ($[0.163, 0.442]$) at a $6.24\times$ lower analytical cost, the pattern of Section 5.3. Closed-loop accepted prefixes are shorter than on LIBERO-Long for both drafters, and policy time is descriptive only.

**LIBERO-Goal.** A third suite was added under a protocol frozen before its runs; to fit the time available, the cached controls and the fixed-input comparison were omitted in advance, so only the compressed drafters and verifier-only decoding run in closed loop. All else follows LIBERO-Spatial: the suite's own released OPENVLA verifier (at most $300$ control steps), whose action head the drafters use; a new rollout dump on all $500$ initial states ($80{,}356$ control steps, about $11.9$ passes at the same $60{,}000$ base updates); recovery updates from the same 20-epoch rule; $r = 5$; and the same primary $440$ states, which are not training-unseen. Seed $44$ was fixed with its LIBERO-Long and LIBERO-Spatial results known. For seed $44$ on the primary set, compressed SNN minus ANN SR is $-4.55$ pp ($[-8.86, -0.23]$) at a $10.74\times$ lower analytical cost ($[10.68, 10.81]$), and relative to verifier-only decoding the SNN and ANN differ by $-5.91$ pp ($[-10.45, -1.59]$) and $-1.36$ pp ($[-5.91, 3.18]$); on all $500$ states these are $-5.00$ ($[-9.20, -0.80]$), $-5.80$ ($[-10.00, -1.60]$), and $-0.80$ pp ($[-5.00, 3.40]$). Per control step the seed-$44$ SNN and ANN reach AL $2.30$ and $2.39$ and cut verifier passes from $7.00$ to $5.81$ and $5.73$. Looser acceptance does not account for the seed-$44$

Table 5: **LIBERO-Spatial and LIBERO-Goal closed loop** (three training seeds; seed 44 ran first and seeds 42/43 were added afterwards with the compressed drafters only; each suite with its own verifier checkpoint, 50 initial states for each of ten tasks; LIBERO-Goal has no cached controls). SR is task-macro on the primary 440 states (initial states 6–49 of every task) and on all 500; the other columns use the 440. Cost is the warm-table analytical synaptic cost of the full drafter. Training seed in parentheses; bold marks SNN entries at least as good as their paired ANN.

| Drafter | $SR_{440}$ (%) | $SR_{500}$ (%) | AL | Calls | Cost (mJ) | Policy (ms) |
|---|---|---|---|---|---|---|
| LIBERO-*Spatial* | | | | | | |
| Verifier only | 79.32 | 78.80 | 1.000 | 7.00 | 0 | 156.64 |
| ANN, cached (44) | 77.05 | 77.00 | 2.123 | 5.94 | 4.274 | 147.53 |
| SNN, cached (44) | **80.68** | **80.40** | 2.005 | 6.05 | **0.815** | 151.72 |
| ANN, compressed (44) | 77.95 | 77.60 | 2.034 | 6.02 | 3.802 | 149.64 |
| SNN, compressed (44) | **81.14** | **79.60** | 1.910 | 6.14 | **0.338** | 154.28 |
| ANN, compressed (42) | 80.00 | 79.40 | 1.999 | 6.06 | 3.802 | 146.40 |
| SNN, compressed (42) | 79.32 | 78.60 | 1.883 | 6.16 | **0.351** | 148.66 |
| ANN, compressed (43) | 77.95 | 77.00 | 2.027 | 6.03 | 3.802 | 144.66 |
| SNN, compressed (43) | **80.91** | **81.00** | 1.916 | 6.13 | **0.348** | 149.46 |
| LIBERO-*Goal* | | | | | | |
| Verifier only | 77.95 | 78.00 | 1.000 | 7.00 | 0 | 153.88 |
| ANN, compressed (44) | 76.59 | 77.20 | 2.394 | 5.73 | 3.802 | 141.04 |
| SNN, compressed (44) | 72.05 | 72.20 | 2.302 | 5.81 | **0.354** | 144.29 |
| ANN, compressed (42) | 72.95 | 74.20 | 2.398 | 5.72 | 3.802 | 139.23 |
| SNN, compressed (42) | **73.86** | **74.40** | 2.311 | 5.81 | **0.360** | 141.87 |
| ANN, compressed (43) | 75.23 | 75.60 | 2.417 | 5.71 | 3.802 | 139.40 |
| SNN, compressed (43) | **76.14** | **76.40** | 2.324 | 5.79 | **0.351** | 141.14 |

lower SR: per control step the SNN accepts 0.37 tokens that lie within the tolerance but are not the verifier's argmax, against 0.43 for the ANN ($-0.065$, $[-0.093, -0.036]$), and 74.3% of its executed actions equal the verifier's greedy action in all seven bins, against 70.5% ($+3.8$ pp, $[2.4, 5.1]$); this diagnostic reads the recorded traces and does not identify a cause. The pre-registered rule reports an interval below zero as lower SR, which Sections 5.1 and 6 state, and the seed-44 runs were not retrained, reselected, or rerun.

**Added training seeds.** After the seed-44 results of both suites were known, we trained seeds 42 and 43 for both suites under a protocol fixed before their runs: the same code, dumps, masks, and verifier-only runs, the compressed drafters only, the same wording rules, and seed 44 reported unchanged. A seed counts only when both drafters complete all ten tasks and pass the per-file audits. On LIBERO-Goal, seed 42 gives compressed SNN minus ANN SR $+0.91$ pp ($[-3.64, 5.45]$) at $10.56\times$ lower analytical cost; relative to verifier-only decoding the SNN differs by $-4.09$ pp ($[-8.41, 0.23]$) and the ANN by $-5.00$ pp ($[-9.55, -0.68]$), and on all 500 states SNN minus ANN is $+0.20$ pp ($[-4.00, 4.40]$). Seed 43 gives $+0.91$ pp ($[-3.64, 5.68]$) at $10.82\times$, with the SNN and ANN at $-1.82$ ($[-6.59, 2.95]$) and $-2.73$ pp ($[-7.05, 1.82]$) from verifier-only decoding. Across the three LIBERO-Goal seeds, SR is $74.92 \pm 1.84\%$ (ANN) and $74.02 \pm 2.05\%$ (SNN), SNN minus ANN is $-0.91 \pm 3.15$ pp, and the cost ratio is $10.71 \pm 0.13\times$. On LIBERO-Spatial, seed 42 gives $-0.68$ pp ($[-4.77, 3.41]$) at $10.82\times$, with the SNN and ANN at 0.00 ($[-3.86, 3.86]$) and $+0.68$ pp ($[-3.41, 4.77]$) from verifier-only decoding, and seed 43 gives $+2.95$ pp ($[-0.91, 7.05]$) at $10.92\times$; on all 500 states seed 43 has higher SR in this training seed ($+4.00$ pp, $[0.40, 7.60]$), on that sensitivity set only. Across the three LIBERO-Spatial seeds, SR is $78.64 \pm 1.18\%$ (ANN) and $80.45 \pm 0.99\%$ (SNN), SNN minus ANN is $+1.82 \pm 2.17$ pp with no difference detected in any seed on the primary states, and the cost ratio is $10.99 \pm 0.22\times$.

Table 6: **Component versus system per control step** ($K$=1, 400 fixed inputs). Time: synchronized, RTX 5090, BF16, seeds 42/43; the first three rows come from one session, in which Verify for AR is its six single-token passes (an earlier session measured totals of 135.1 and 137.2 ms for the two drafters). Arithmetic: Equation 6, seed 44, excluding the vision encoder, memory traffic and elementwise work; AR is a static count. Spec-VLA and KERV: official code with the released Spec-VLA drafter; their times include instrumentation, and their verifier arithmetic counts every packed tree node verified after the prompt prefill; their totals also include other per-step work, so the components do not sum. Bold: SPEC-SPIKE entries lower than every other proposer.

| | Measured time (ms) | | | | GPU MACs | Verifier arithmetic | | | Drafter |
|---|---|---|---|---|---|---|---|---|---|
| Proposer | Prefill | Draft | Verify | Total | drafter (B) | Calls | Prefill (J) | Rest (J) | (mJ) |
| None (AR) | 42.6 | 0 | 112.0 | 154.7 | 0 | 7.00 | 8.816 | 0.184 | 0 |
| Spec-VLA-style | 42.6 | 6.98 | 78.2 | 127.8 | 0.83 | 4.79 | 8.816 | 0.315 | 3.802 |
| SPEC-SPIKE | 42.6 | 8.53 | 78.7 | 129.9 | 3.24 | 5.01 | 8.816 | 0.321 | **0.343** |
| Spec-VLA, official | 47.4 | 19.54 | 66.4 | 142.0 | 111.3 | 4.47 | 8.816 | 5.348 | 511.8 |
| KERV, official | 60.0 | 12.94 | 72.6 | 177.5 | 98.9 | 3.46 | 8.816 | 21.886 | 454.8 |

Table 7: **Cost audit of the full drafter** per control step (six draft calls, 400 fixed inputs, seeds 42/43); each row adds one interface change. Both drafters execute the shared MACs; the ANN core adds 805.5 M MACs per row, the SNN core only amplitude-weighted ACs. Ops count MACs and ACs alike; cost prices them at 4.6 and 0.9 pJ. The last two rows charge the 19.757 mJ table construction every $M$ distinct observations.

| | Operations (M) | | Analytical cost (mJ) | | ANN/SNN | |
|---|---|---|---|---|---|---|
| Interface | Shared MAC | SNN AC | ANN | SNN | Ops | Cost |
| Plain (Eq. 4) | 308.3 | 242.4 | 5.123 | 1.636 | 2.02 | 3.13 |
| + context cache | 224.4 | 242.4 | 4.737 | 1.250 | 2.21 | 3.79 |
| + token lookup (*cached*) | 123.7 | 242.3 | 4.274 | 0.787 | 2.54 | 5.43 |
| + rank 256, recovered (*compressed*) | 21.0 | 272.8 | **3.802** | **0.342** | **2.81** | **11.12** |
| table rebuilt every 150 observations | 49.6 | 272.8 | 3.933 | 0.474 | 2.65 | 8.30 |
| table rebuilt every observation | 4315.9 | 272.8 | 23.558 | 20.099 | 1.12 | 1.17 |

## D NUMERICAL AND COST-ACCOUNTING BOUNDARIES

Table 6 lists the per-step time and arithmetic of each proposer that Section 5.4 summarizes.

Table 7 audits the fixed-input cost of the full drafter one interface change at a time (Section 5.2).

**Mapping and correction.** Raw token conditioning is distinct from action-bin equality: 256 head rows map to 255 physical bins at the boundary. This edge case is documented in OpenVLA's reference tokenizer.[1] Draft row 255 is clamped to bin 254 before the next embedding, matching the declared production mapping. Verifier raw tokens are not silently merged because actions coincide. The decoder checks prefixes under legal conditioning, uses an ancestor-only packed-tree mask, and reconstructs the selected contiguous path for suffix generation. Selected-prefix replay is charged. Finite tests do not imply universal BF16 losslessness, even at $r = 0$.

**Shape-sensitive arithmetic.** A target-logit diagnostic initially batched six positions into one head matrix multiplication. Relative to production-shape single-position replay, that batching changed 587 of 274,152 argmax tokens; the production-shape replay matched all of them, and the failed diagnostic remains archived. A repeated pilot replay found zero differences across 360 pairs in success, length, input sequence, raw tokens, and action bins. These are finite reproducibility checks, not universal theorems. Likewise, the observation cache and the action-token lookup table (Section 3.3) are algebraic rearrangements of the plain projections; they are not claimed to be bitwise identical under every floating-point execution shape.

[1] OpenVLA action tokenizer, fixed source revision, lines 49–68.

**Synaptic scope.** For integer input events, accumulations are $n_{\text{out}} \sum_i s_i$, not the number of nonzero inputs times fan-out. Dense interfaces, head products, all draft calls, and ANN attention products are counted. Normalization, activations, neuron-state updates, sorting, copies, and memory are outside the energy estimate, and verifier arithmetic is reported separately (Table 6); none is assumed free on a device. Reference FP32 coefficients are not calibrated to BF16 GPU execution. On the GPU, the SNN's four internal steps are unrolled into dense matrix products; its amplitude-weighted accumulations are only $8.4$–$8.5\%$ of the $3.22$ billion synaptic operations of that dense core work, which is why its analytical cost is an order of magnitude below the ANN's while its GPU time is not. Lookup construction is 2 MiB and $256H^2$ MACs, or $19.757$ mJ. If the table is rebuilt every $M$ distinct observations, the amortized cost is

$$E(M) = E_{\text{warm}} + E_{\text{build}}/M, \tag{7}$$

where $E_{\text{warm}}$ is the warm-table cost and $E_{\text{build}} = 19.757$ mJ; the ratio is about $1.17\times$ at $M = 1$, $8.28$–$8.32\times$ at $M = 150$, and $9.81$–$9.87\times$ at $M = 400$ across the three seeds. Amortization counts distinct observations between invalidations, not repeated timing invocations on one observation. Context resets each observation; the token table may persist.

**Event-driven CPU timing.** To measure how much of the analytical ratio a substrate that skips silent neurons realizes, we timed drafting on a commodity 96-core x86 server CPU (threads pinned to one NUMA node, one process at a time, no concurrent GPU work) for the three compressed seeds on the 400 fixed inputs. The SNN kernel does synaptic work only for emitted spikes, reading only the weight rows those spikes address; the ANN uses a dense library matrix product at the same weight precision. Before timing, the event kernel had to return candidates identical to the production CPU implementation on every input for $K = 1$ and $K = 2$ (2,400 pairs). On CPU, BF16 execution of the SNN changes 5–6% of $K = 1$ candidates relative to the GPU records, identically for dense and event execution; the ANN changes 0–1%. Per step the event kernel reads $0.22$ GB of logical weight bytes against $1.65$ GB for the BF16 ANN ($7.4$–$7.5\times$ fewer; logical bytes, not measured DRAM traffic). Its time advantage depends on the thread count (Table 8): $2.52\times$ at 16 threads, where the ANN reaches $80\%$ of the measured read-only memory bandwidth, $1.38\times$ at 4, and $0.82\times$ on one thread, where the dense kernel is faster. With skipping disabled the same kernel takes $2.5\times$ as long as the ANN at 16 threads, so the gain comes from skipping, not from a faster kernel. The FP32 ANN did not reach the pre-registered bandwidth criterion at 16 threads and did not speed up with more threads, so the FP32 ratios overstate the saving. These are drafter-only latency measurements on commodity hardware, not device energy, neuromorphic deployment, or an end-to-end policy speedup.

Table 8: **Event-driven CPU drafting time.** Per-step drafting time of the dense parameter-matched ANN over that of the compressed SNN whose kernel performs synaptic work only for emitted spikes, on the 400 fixed inputs ($K = 1$, quiet blocks); ratios $> 1$ favor the SNN. Mean of seed ratios $\pm$ across-seed sample SD, seeds 42–44; ms are across-seed means. *Skipping* divides the event kernel time with skipping disabled by that with skipping enabled.

| | BF16 weights | | | FP32 weights | |
|---|---|---|---|---|---|
| Threads | ANN (ms) | SNN (ms) | ANN/SNN | ANN/SNN | Skipping |
| 1 | 146.7 | 178.7 | $0.82 \pm 0.01$ | $1.25 \pm 0.01$ | $8.33 \pm 0.13$ |
| 4 | 75.7 | 54.7 | $1.38 \pm 0.01$ | $3.79 \pm 0.01$ | $7.12 \pm 0.04$ |
| 16 | 45.5 | 18.0 | $2.52 \pm 0.02$ | $7.86 \pm 0.09$ | $6.35 \pm 0.03$ |

## E CANDIDATE-TREE EVIDENCE

**Closed-loop two-branch runs.** The compressed SNN drafter with $K = 2$ ran on all ten tasks and 50 initial states per task for each seed (1,500 new episodes each for the SNN and the matched-width ANN), with the same verifier, $r = 5$, and policy implementation as Table 1; its single-branch controls are the $K = 1$ runs of that table, reused after a 15-episode parity run reproduced them exactly. On the 413 mask, SNN $K = 2$ minus ANN $K = 1$ SR is $+3.24$ ($[-2.73, 9.22]$), $+0.12$ ($[-5.57, 5.65]$), and $+2.67$ pp ($[-3.21, 8.32]$), and minus SNN $K = 1$ it is $-1.08$, $+4.79$, and $-2.84$ pp, all intervals containing zero. The accepted-prefix differences to ANN $K = 1$ are $+0.028$ ($[-0.100, 0.158]$), $+0.211$ ($[0.085, 0.337]$), and $+0.147$ ($[0.016, 0.279]$),

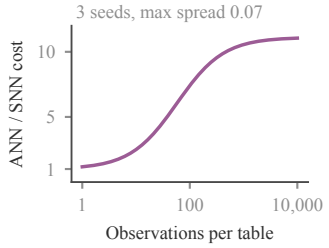



Figure 5: **Cold-start sensitivity.** Fixed-input ANN/SNN cost ratio with lookup construction amortized over $M$ distinct observations (Equation 7); the line is the mean of the three seeds, which differ by less than $0.08$ at every $M$.

Table 9: **Candidate expansion under a shared bounded tree** (400 fixed inputs, compressed drafters, mean $\pm$ sample SD over three training seeds). Gain is the accepted-prefix difference to ANN $K{=}1$ on the same inputs; cost charges every executed draft call, including pruned parents. Per-seed intervals are in Appendix E.

| Drafter | Draft calls | Gain vs. ANN $K{=}1$ $\uparrow$ | Cost (mJ) $\downarrow$ | Cost ratio to ANN $K{=}1$ |
|---|---|---|---|---|
| ANN, $K{=}1$ | 6 | ref. | 3.802 | $1.00\times$ |
| ANN, $K{=}2$ | 11 | $0.349 \pm 0.053$ | $6.962 \pm 0.000$ | $0.55\times$ |
| SNN, $K{=}1$ | 6 | $-0.100 \pm 0.154$ | $0.342 \pm 0.001$ | $11.11\times$ |
| SNN, $K{=}2$ | 11 | $0.238 \pm 0.106$ | $0.606 \pm 0.002$ | $6.28\times$ |

and to SNN $K = 1$ they are $+0.177$, $+0.213$, and $+0.244$, all excluding zero. The matched-width ANN $K = 2$ arm (the 1,500 ANN episodes above) gives SNN minus ANN SR of $+0.17$ ($[-5.47, 5.97]$), $+0.63$ ($[-5.47, 6.60]$), and $-0.80$ pp ($[-6.32, 4.83]$) and prefix differences of $-0.077$ ($[-0.211, 0.051]$), $+0.085$ ($[-0.039, 0.206]$), and $-0.006$ ($[-0.129, 0.117]$), at ANN/SNN cost ratios of $11.63/11.69/11.59$; ANN $K = 2$ minus ANN $K = 1$ SR is $+3.07$, $-0.51$, and $+3.47$ pp, all intervals containing zero.

**Fixed-input tree records.** The compressed study crosses 400 shared inputs with ANN/SNN and $K = 1/2$: 3,200 records for seeds $42/43$ and 1,600 for seed 44, not independent episodes. At matched $K = 1$, SNN minus ANN prefix is $+0.060$ ($[-0.096, 0.244]$), $-0.112$ ($[-0.322, 0.065]$), and $-0.247$ ($[-0.464, -0.011]$) for seeds 42, 43, and 44; at matched $K = 2$ it is $-0.080$ ($[-0.290, 0.124]$), $-0.070$ ($[-0.259, 0.106]$), and $-0.185$ ($[-0.454, 0.076]$). SNN $K = 2$ minus ANN $K = 1$ is $+0.328$ ($[0.139, 0.539]$), $+0.265$ ($[0.080, 0.445]$), and $+0.120$ ($[-0.133, 0.380]$), at ANN/SNN cost ratios of $6.27/6.30/6.26$; at matched $K = 2$ the ratios are $11.49/11.54/11.46$. Seed-44 records give ANN/SNN cost ratios of 11.09 at $K = 1$, 11.46 at $K = 2$, and 6.26 for ANN $K = 1$ versus SNN $K = 2$, with accepted-prefix means 2.500 and 2.805 for ANN at $K = 1/2$ and 2.253 and 2.620 for SNN. These fixed-input prefixes exceed the closed-loop ones (Table 1 AL minus one), so only within-set differences, not absolute values, are interpreted. These results do not identify an SNN-exclusive branching mechanism; the residual evaluation of Section 5.3 relies on integer activations, not on spiking as such.

Table 10 places the drafters against non-neural references measured in an earlier six-arm study on the same tasks and initial states. In that study, speculation with either learned drafter cut verifier calls but did not resolvably cut wall-clock, and a zero-arithmetic proposer that copies the previous accepted bins reaches $53.73\%$ against $53.56\%$ for no speculation at 4.87 calls, above both learned drafters on that benchmark. Consecutive control-step actions often repeat their bins, as the copy proposer's 2.90 accepted tokens show; the combined arms accept more than either source alone, so the two proposal sources are complementary rather than redundant. These are pre-compression checkpoints under a different decoder and a single evaluation seed, so they bound expectations rather than measuring the models in Table 1.

An earlier uncompressed study freshly evaluated 2,000 episodes: two seeds, ANN $K = 1$ versus SNN $K = 2$, ten tasks, 50 states per task. Primary 413 ANN/SNN SR was $48.27/49.86\%$

Table 10: **Historical reference arms.** An earlier six-arm study on the same ten tasks with 50 initial states per arm, recomputed on the primary 413 mask with equal task weights. It uses pre-compression checkpoints, one evaluation seed, and a one-round decoder rather than the packed tree, so these are reference points and not rows of Table 1. The copy proposer reuses the previous control step's accepted bins and performs no drafter arithmetic. Arrows give the preferred direction and bold the best arm per column; with one evaluation seed, paired SR intervals on this mask span about ±5 pp, and timing gaps smaller than the 16 ms spread of the no-speculation arm's task medians are not resolved. Accepted counts accepted draft tokens (AL−1), and policy time is the mean of per-task median step times, not the task-macro mean of Table 1.

| Arm | Proposal source | SR (%) ↑ | Verifier calls ↓ | Accepted ↑ | Policy (ms) ↓ |
|---|---|---|---|---|---|
| H0 | None (autoregressive) | 53.56 | 7.00 | 0.00 | 169.2 |
| H1 | Copy previous action | **53.73** | 4.87 | 2.90 | 141.4 |
| H2 | ANN drafter | 49.75 | 5.54 | 1.81 | 164.7 |
| H3 | SNN drafter | 47.36 | 5.61 | 1.74 | 167.6 |
| H4 | ANN drafter + copy | 52.36 | **4.55** | **3.33** | 136.0 |
| H5 | SNN drafter + copy | 49.47 | 4.59 | 3.28 | **128.3** |

for seed 42 and $49.01/46.95\%$ for seed 43. SNN-minus-ANN was $+1.59$ pp ($[-4.11, 7.24]$) and $-2.06$ pp ($[-7.87, 3.65]$). Analytical cost, with uncached and uncompressed interfaces, was ANN $5.123$ mJ and SNN $2.993/2.976$ mJ, only $1.7\times$ apart because the SNN drafted two branches (eleven calls) against one ANN branch (Table 7 gives $3.13\times$ at matched $K = 1$); policy time was ANN $150.28/152.84$ ms and SNN $161.90/159.75$ ms. Each packed verification checked $9.06/9.09$ nodes for SNN $K = 2$ versus $6.00$ for ANN $K = 1$, at $5.42/5.42$ versus $5.47/5.52$ verifier calls per control step and 97.0–98.0 versus 96.8–99.1 ms of verification and completion time. Full-500 SR was ANN $47.6/49.6\%$ and SNN $50.8/48.2\%$. This establishes feasibility for earlier checkpoints, not SR of the new compressed drafter or an SNN GPU speedup.

## F Published Results Beside Ours

Table 11 lists published LIBERO-Long results beside ours; the notes below define each paper's units.

Spec-VLA's length $L$ is "the number of tokens predicted in each forward pass" (Wang et al., 2025a, Table 1); it includes the verifier's correction token, since for exact matching on LIBERO-Long it equals one plus the mean accepted-draft count of their Table 2 ($1 + 0.665$). HeiSD does not define its AL, and KERV names AFEP (average first error position) without stating how it is indexed; HeiSD's retrieval-only AL falls below one on three of four suites (0.81 on LIBERO-Long; Zheng et al., 2026b, Table 3), so it cannot include a correction token, while its Spec-VLA row repeats the AFEP of KERV's $r = 15$ row (Zheng et al., 2026a, Table 2). WA-SpecDec reports accepted draft tokens per verification step but gives no per-suite value for LIBERO-Long. Our AL counts accepted draft tokens plus the verifier's first token $a_0$ in a single round, and the suffix after the first rejection is decoded one token per pass; per post-prefill verifier pass our runs reach 1.31–1.33 tokens and the official Spec-VLA run 1.69 by the same count (its own counter, which reports $L$, gives 2.13), so this count, not our AL, is the closer number to set beside $L$, and it is not the same unit. KERV and HeiSD report identical AR and exact-match SR and step counts; converted with KERV's $0.91\times$, KERV's $1.48\times$ is about $1.35\times$ AR. KERV and HeiSD execute some actions without verification and WA-SpecDec fine-tunes the verifier, so their speed-ups are not those of a drafter on a frozen verifier. On all 500 initial states our SR is 48.80 (ANN), 50.47 (SNN), and $54.80\%$ (AR).

**Official Spec-VLA on our benchmark.** We ran Spec-VLA's official code at a fixed commit with its released LIBERO-10 drafter on our verifier, initial states, and 400 fixed inputs, porting only imports and paths. Its dynamic tree (50 nodes, depth 4, top-8), multi-round re-drafting, prefix encoding, full-vocabulary head, and training on relabeled demonstrations are unchanged, and acceptance uses $r = 5$. Before any evaluation, a gate required exact-match decoding to reproduce the verifier's greedy tokens on 15 frozen inputs; 14 matched, and the remaining input diverged at a one-ulp BF16 tie (27.75 versus 27.625), which we accepted under a gate amendment fixed before any evaluation.

Table 11: **Published LIBERO-Long results beside ours; not a controlled comparison.** Published rows are copied from the cited tables (S: Wang et al. (2025a); H: Zheng et al. (2026b); K: Zheng et al. (2026a); W: Wen et al. (2026)), not re-run; settings and units differ across papers (Appendix F). $\Delta$SR is SR minus the autoregressive (AR) SR of the same paper; with 500 episodes a single-run difference below about 6 pp is within binomial noise. Tok./pass is the source's $L$ for published rows and the official counter for runs of official code; for our drafters it is $6/(\text{verifier calls}-1)$, a different count (Appendix F). KERV's speed-ups are relative to its exact-match speculation (SD, $0.91\times$ AR). For runs on our verifier, the speed-up is the measured closed-loop policy time of the same-study AR arm divided by that of the method on the same GPU (mean of three training seeds for our drafters), a descriptive measurement. n.r.: not reported.

| Method | Src. | SR (AR) | $\Delta$SR $\uparrow$ | Accept. (own unit) | Tok./pass | Speed-up |
|---|---|---|---|---|---|---|
| *Published, as reported* | | | | | | |
| Spec-VLA, exact match | S T1 | 50.8 (52.0) | $-1.2$ | $L = 1.67$ | 1.67 | $1.13\times$ AR |
| Spec-VLA, $r$=5 | S T1 | 55.0 (52.0) | $+3.0$ | $L = 2.10$ | 2.10 | $1.22\times$ AR |
| Neural SD, exact[a] | H T3 | 50.2 (54.4) | $-4.2$ | AL$= 1.59$ | n.r. | $0.91\times$ AR |
| KERV[b] | K T2 | 48.8 (54.4) | $-5.6$ | AFEP$= 4.64$ | 3.53[†] | $1.48\times$ SD |
| HeiSD[b] | H T3 | 47.0 (54.4) | $-7.4$ | AL$= 4.96$ | n.r. | $1.79\times$ AR |
| Spec-VLA, $\rho$=10[c] | W T1 | 49.4 (57.8) | $-8.4$ | $L = 2.22$[†] | 2.22[†] | $1.18\times$ AR |
| WA-SpecDec, $\rho$=10[d] | W T1 | 56.2 (57.8) | $-1.6$ | n.r. | n.r. | $1.21\times$ AR |
| *Run on our frozen* OpenVLA-*7B and initial states* | | | | | | |
| Spec-VLA, official code | Tab. 1 | 53.45 (53.56) | $-0.12$ | AL$= 2.06$ | 2.13 | $1.10\times$ AR[g] |
| ANN[e], compressed | Tab. 1 | 48.21 (53.56) | $-5.35$ | AL$= 2.64$ | 1.33 | $1.10\times$ AR |
| SNN, compressed | Tab. 1 | 49.93 (53.56) | $-3.63$ | AL$= 2.56$ | 1.31 | $1.08\times$ AR |
| Copy prev. bins (H1) | Tab. 10 | 53.73 (53.56) | $+0.17$ | AL$= 3.90$ | 1.55 | $1.20\times$ AR[f] |

[a]HeiSD's "Pure D-SD". [b]Executes some actions without verification. [c]Spec-VLA's rule as run by WA-SpecDec. [d]Verifier adapted with LoRA and a world-aware prefill bias. [e]Spec-VLA-style drafter, reimplemented (Section 3.2). [f]Earlier single-seed study (Table 10), not the same run. [g]Its time includes instrumentation, so this is a lower bound. [†]Not reported by the source; measured by us with the official KERV or Spec-VLA code and the released Spec-VLA drafter on our 400 fixed inputs.

On the fixed inputs its first-round AL is 2.37, below our compressed drafters at $K = 1$ by 0.86–1.13 (ANN) and 0.88–0.92 (SNN) tokens across seeds, every paired 95% interval excluding zero, while its multi-round tree needs 0.32–0.54 fewer verifier passes; these inputs overlap our drafters' training rollouts, which favors ours. Its analytical drafter cost is 135× our compressed ANN and 1,490–1,500× our SNN at $K = 1$ (74× and 840–850× at $K = 2$); its times include instrumentation and are descriptive. With the same code at exact matching and at $r = 10$, its first-round AL is 1.89 and 2.63, with 4.90 and 4.30 verifier passes, 195 and 165 packed nodes, and 535 and 502 mJ of drafter arithmetic per control step (Table 1); acceptance tolerance barely changes its drafter cost. In closed loop (500 new episodes, no errors), it reaches 53.45% SR on the 413 primary states and 54.00% on all 500, with first-round AL 2.06, 2.13 tokens per verification round, 4.55 verifier passes and 177 packed nodes per control step, and 515 mJ of drafter arithmetic. Paired with our compressed drafters at $K = 1$, its SR differs by +6.09 ($[0.28, 11.90]$), +2.28 ($[-3.65, 8.03]$), and +7.34 pp ($[1.47, 13.21]$) from the ANN and by +1.77 ($[-3.92, 7.51]$), +6.95 ($[0.86, 12.77]$), and +1.83 pp ($[-3.92, 7.45]$) from the SNN for seeds 42–44; against our $K = 2$ drafters every interval contains zero. The comparison does not separate its drafter from its decoder, and its exposure to the evaluation initial states is unknown.

**Proposers of the other methods.** HeiSD (Zheng et al., 2026b) switches per control step between a single-block neural drafter and retrieval: the query is a 4352-dimensional two-view image embedding, searched in an HNSW index over every training timestep, which for LIBERO-Long holds 101,469 vectors in 2.16 GB (6.5 GB over four suites; their Table 8), and an adaptive verify-skip executes retrieved actions without verification on some steps. One index query touches on the order of $10^3$ vectors, about $10^7$ MACs under default index parameters (our estimate), less than SPEC-SPIKE's drafter; the two image encoders add more unless their features are shared with the verifier. KERV (Zheng et al., 2026a) released its code but not its weights, and its drafter configuration is identical to Spec-VLA's, so we load the released Spec-VLA drafter into KERV's official code (one import line ported), run its default decoding path unchanged (FlagOS acceleration off), and measure it on the 400 fixed inputs after the same gates as Spec-VLA (14 of 15 exact-match inputs reproduced, the remaining one a one-ulp BF16 tie). It reaches first-round AL 3.53 with 3.46 verifier passes per control step and 98.9 billion drafter MACs, 455 mJ, which is about 750× two-branch SPEC-SPIKE and 1,330× one-branch SPEC-SPIKE; its AL differs from two-branch SPEC-SPIKE in no seed. The released code departs from the paper in three ways that we did not change: the acceptance tolerance follows a fixed schedule from 15 to 5 bins over the episode rather than a kinematic rule (so it is about 15 on these early-episode inputs), the Kalman completion is forced every fourth step rather than triggered by a rejection, and on forced steps the verified first token is dropped and the dimensions shift by one. Kalman values fill 13.1% of executed action dimensions without verification, and 57.8% of executed tokens differ from the verifier's argmax; the observations come from the verifier-only trajectories, so this is not a KERV closed loop, and no SR is reported. WA-SpecDec (Wen et al., 2026) fine-tunes the verifier, adds a video-encoder bias to its input, and re-distills its drafter to the modified verifier, so it changes the verifier itself rather than only the proposer.

## G PRE-REGISTERED INFERENCE-TIME VARIANTS

Four inference-only variants were fixed, with their endpoints, main-text bars, and wording, before any of their records existed, and all four are reported here. Each started from a design analogy; none is claimed as a biological mechanism. All use the archived compressed drafters of seeds 42–44, the 400 fixed inputs, $r = 5$, and paired intervals conditional on the trained models, with no training and no closed-loop runs.

**B1, exact residual evaluation.** Updating each core map only by the change in its integer inputs follows delta networks across time (O'Connor & Welling, 2017; Neil et al., 2017) and efficiently updatable evaluation across game-tree moves (Nasu, 2018); here the change is taken across branches. The per-seed 95% intervals of $D$ are $[0.356, 0.388]$, $[0.367, 0.400]$, and $[0.353, 0.382]$; the reference is the concurrent greedy call at the same depth, whose alternatives are siblings with a different token (34–35%) or cousins with the same (62–65%) or a different token (1.5–4%). Against an unrelated call the residual saves nothing (0.99), so the saving comes from the structure shared between branches. Charging the residual over time-summed inputs instead of per internal step would give $D = 0.17/0.18/0.17$ and ratios $8.48/8.50/8.49\times$; we report the per-step accounting. Cached refer-

Table 12: **Pre-registered inference-time variants.** Values are per seed $(42/43/44)$ on the 400 fixed inputs; only the MAIN outcome enters the main text.

| Variant | Main-text bar | Result | Outcome |
|---|---|---|---|
| B1: evaluate the $K{=}2$ alternative calls as exact residuals of the greedy call | charged/direct accumulations $D \leq 0.60$ in every seed, margin $\geq 0.10$ to an unrelated reference | $D = 0.37/0.38/0.36$; unrelated reference $0.99$; no spike change; ratio $6.27/6.30/6.26 \rightarrow 7.83/7.83/7.84\times$ | MAIN |
| B2: set-point gain on first-layer firing after rank-256 factorization | SNN retention gain $\geq 0.04$ with interval above zero in every seed | fitted gains restored only $0.57/0.55/0.55$ of SNN firing (ANN 1.00) | manipulation failed |
| B3: admit the second branch only if more than $2r = 10$ bins from the first | mean SNN $K{=}2$ prefix gain $\geq 0.05$, intervals above zero in two seeds | SNN $+0.085/ + 0.032/ + 0.018$; ANN $+0.105/ + 0.047/ + 0.197$ | appendix |
| B4: two instead of four internal steps at draft positions 4–6 | cost cut at retained prefix, beating length cap, mirrored and uniform schedules | cost $-17.4/ - 17.3/ - 17.4\%$ at retention $1.005/1.003/1.004$ | appendix |

ence values (81,920 per pair) are memory and excluded like all memory, every elementwise update (163,840 per alternative call) is charged at the accumulation price, and the closed-loop $6.35\times$ was not re-measured.

**B2–B4.** The fitted gains did not restore first-layer firing, so B2 does not test whether firing loss explains the SNN's factorization loss. Window-disjoint sibling admission lengthened the SNN $K = 2$ prefix in every seed but below the bar, so the main text keeps the standard rule. Running two internal steps at positions 4–6 cuts drafter cost by $17.4/17.3/17.4\%$ at retention $1.005/1.003/1.004$ and beats capping the draft length at equal cost, but at $K = 1$ it did not beat the mirrored schedule in seed 44, which the main-text rule required; we report it only as a runtime operating point, not as evidence for position-wise allocation.

## H Robustness of Drafter Acceptance

Because the verifier decides every executed action, a perturbation can change what speculation delivers only through the drafter's acceptance (and, under relaxed acceptance, the bins it gets accepted). With levels and decision rules fixed before any record, we measured how much of its accepted prefix each compressed drafter keeps when its core weights are quantized (per output channel, symmetric, round-to-nearest, no retraining) or when Gaussian noise scaled to each input vector's RMS is added to the verifier hidden state and context it reads; both drafters receive identical perturbations, and the shared interfaces, lookup table, embeddings and head stay in BF16. The rule credited either architecture only if every seed's interval of $G$ excluded zero in its favor at a primary level (INT4; $\sigma = 0.1$). Neither happened (Table 13), so no consistent difference in acceptance robustness was detected; both drafters keep at least $95\%$ of their accepted prefix under every tested perturbation. A pre-registered observation-corruption arm was skipped because the fixed inputs store processed pixel tensors rather than raw observations.

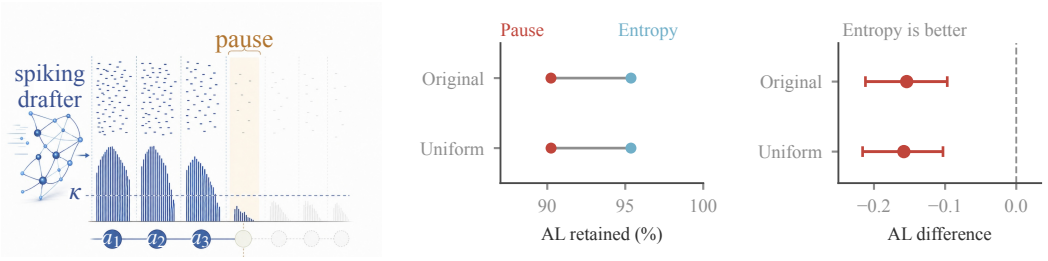
## I Pause as an Observational Finding

A pause here is low mean amplitude of the final integer-neuron output over internal spike steps, not a biological complex-spike measurement. Two historical seed-42 checkpoints are probed: one trained without the activity penalty and one with the uniform penalty (coefficient 1, as in Appendix A). Neither is the recovered rank-256 model in the main SR table; Figure 6 and Table 14 summarize the probe.

Each checkpoint has 2,000 chains: ten tasks, 20 trajectories per task, ten evenly spaced distinct steps per trajectory. Trajectories 0–9 calibrate and 10–19 test the gate, giving 1,000 chains in each

Table 13: **Robustness of drafter acceptance** on the 400 fixed inputs ($K = 1$, compressed drafters). Retained is the accepted prefix under the perturbation relative to the same drafter without it, mean over seeds 42–44 (every seed at least 0.95); $G$ is SNN minus ANN retention per seed with its paired 95% interval, every one of which contains zero. Both drafters receive identical perturbations; interfaces, lookup, embeddings and head stay in BF16. Starred rows are the pre-registered primary levels.

| | Retained | | $G$ with 95% interval | | |
|---|---|---|---|---|---|
| Perturbation | ANN | SNN | seed 42 | seed 43 | seed 44 |
| INT8 core weights | 1.002 | 1.006 | $+0.008$ $[-0.00, 0.02]$ | $+0.003$ $[-0.03, 0.03]$ | $+0.001$ $[-0.02, 0.03]$ |
| INT4 core weights* | 0.991 | 0.972 | $-0.049$ $[-0.11, 0.02]$ | $-0.011$ $[-0.07, 0.07]$ | $+0.005$ $[-0.05, 0.07]$ |
| feature noise $\sigma = 0.05$ | 0.999 | 1.010 | $-0.012$ $[-0.04, 0.01]$ | $+0.023$ $[-0.01, 0.07]$ | $+0.022$ $[-0.01, 0.05]$ |
| feature noise $\sigma = 0.1$* | 0.998 | 0.999 | $-0.043$ $[-0.09, 0.01]$ | $+0.006$ $[-0.05, 0.06]$ | $+0.039$ $[-0.01, 0.09]$ |
| feature noise $\sigma = 0.2$ | 0.992 | 0.984 | $-0.021$ $[-0.07, 0.03]$ | $-0.008$ $[-0.06, 0.04]$ | $+0.007$ $[-0.05, 0.07]$ |




(a) Stopping signal (schematic) (b) Acceptance retained (c) Pause minus entropy

Figure 6: **Activity is not the best stopping gate.** (a) Drafting stops after the first position whose mean final-neuron amplitude falls below a calibrated threshold $\kappa$; the paused position is still proposed. (b,c) The rule is probed on historical models only; entropy retains more AL at comparable call savings. Intervals in (c) are paired trajectory-bootstrap 95% intervals.

split. These trajectories were used for distillation: only gate calibration is held out. CPU BF16 evaluation initializes each six-position self-fed chain with verifier context/token and compares with stored verifier bins at $r = 5$. It is not production parallel decoding or closed-loop evaluation.

Stopping is *after* the current paid draft position. Calibration targets 42% fewer calls, mixing adjacent thresholds with a fixed random seed. The confidence comparator is selected on calibration only from maximum probability and entropy; entropy wins for both models. Test thresholds are not retuned to force equal realized budgets.

Pause minus entropy AL is $-0.154$ (95% CI $[-0.212, -0.097]$) and $-0.158$ ($[-0.216, -0.103]$). Intervals use 2,000 paired trajectory bootstrap replicates within tasks. Entropy saves 1.52 and 0.83 percentage points more calls, so this is comparable-cost, not exactly equal-cost.

A retrospective null permutes test stop lengths to match pause's realized budget. Pause exceeds it by 0.336 ($[0.240, 0.439]$) and 0.331 ($[0.239, 0.428]$) AL. This null uses test stop lengths and is not a deployable policy. Activity-conditioned stopping is informative relative to that null, but loses to a logit-confidence rule. The prespecified trigger for a new pause-gated closed-loop study was not met. We neither infer biological causality nor transfer these measurements to the compressed drafter.

Table 14: **Offline gate probe.** Each checkpoint has 1,000 test chains from 100 trajectories. Retention is relative to no-stop AL. Arrows give the preferred direction; bold marks the better of the two stopping gates.

| Checkpoint | Gate | Calls ↓ | Saved (%) ↑ | AL ↑ | Retained (%) ↑ |
|---|---|---|---|---|---|
| Original | No stop | 6.000 | 0.00 | 3.001 | 100.00 |
| | Pause | 3.481 | 41.98 | 2.708 | 90.24 |
| | Entropy | **3.390** | **43.50** | **2.862** | **95.37** |
| Uniform | No stop | 6.000 | 0.00 | 3.085 | 100.00 |
| | Pause | 3.449 | 42.52 | 2.784 | 90.24 |
| | Entropy | **3.399** | **43.35** | **2.942** | **95.36** |