VISCAL: CALIBRATING VISUAL INTERMEDIATE-STATE CAPACITY BY DOWNSTREAM TASK UTILITY

Anonymous authors
Paper under double-blind review

ABSTRACT

Interleaved visual reasoning models generate intermediate visual states. Yet the number of visual tokens or latents assigned to each state is usually fixed or calibrated by reconstruction fidelity. Holding reasoning trajectories fixed while varying capacity, we find that fidelity does not reliably order downstream answer utility, useful capacity depends on the problem and reasoning state, and capacity effects interact across steps. Visual capacity is therefore a budgeted sequential decision rather than a representation-quality parameter. We introduce **VisCal** at two levels: **VisCal-Ext** keeps the backbone frozen and distills hindsight trajectory allocations into a lightweight causal controller, isolating the deployment value of utility calibration and joint allocation. **VisCal-OPD** places a budget-conditioned capacity action before visual generation and constructs an outcome-calibrated teacher from the correct answer and complete outcomes of structured cross-step branches. Distilling this teacher jointly into capacity, visual, and text policies lets the model learn what each visual state contains and how later reasoning uses it. Across four models and eight transfer benchmarks disjoint from the post-training pool, VisCal-OPD improves matched-budget accuracy by **5.14 percentage points** over the released models and matches their accuracy with an estimated **65.1% fewer visual tokens or latents** from one budget-conditioned checkpoint per model.

1 INTRODUCTION

Interleaved visual reasoning models generate intermediate images, visual tokens, or latent states, then consume them to continue reasoning (Li et al., 2025; Gu et al., 2026; Wang et al., 2026b;a). These states form a visual workspace, but every use of that workspace requires a capacity decision: how many visual tokens or latents should this state receive?

Current systems treat capacity primarily as a representation setting. Bagel-Zebra-CoT and ThinkMorph use fixed published configurations across problems and steps (Li et al., 2025; Gu et al., 2026); Monet fixes a global latent length, while DeltaV deploys a router calibrated to reconstruction fidelity (Wang et al., 2026a). Reference-oriented supervision also reaches feature space: Monet aligns latent states to hidden features induced by ground-truth images (Wang et al., 2026b). These approaches assume that a single capacity setting is appropriate for all reasoning states, or that reconstruction fidelity is a reliable proxy for downstream utility.

An intermediate state is not a final image waiting to be faithfully reconstructed. Its purpose is to advance the answer. Does fidelity rank the capacities that best serve that purpose? Holding the reasoning trace fixed while changing capacity, we find that it does not reliably rank answer utility in either Bagel-Zebra-CoT or DeltaV (Figure 2). The best capacity also varies with the current problem and reasoning step, and effects across steps are not additive. A capacity policy therefore needs downstream-outcome supervision, state-dependent decisions, and joint allocation across the trajectory.

Input-side visual-token reduction retains evidence from observed images to save compute (Huang et al., 2025; Ding et al., 2026); variable-length visual thoughts adapt the length of generated states (Tong et al., 2026), while utility-aware latent training improves answer relevance under a fixed latent budget (Kang et al., 2026). The decision here is how to allocate a shared budget across model-generated visual states according to their joint effect on the final answer.

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.

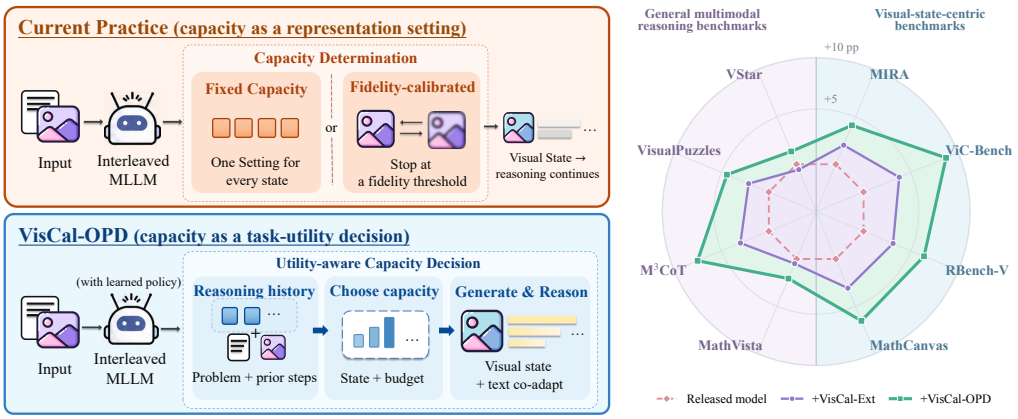


Figure 1: **Visual capacity as a task-utility decision.** Right: matched-budget accuracy gains over the released models (pp), averaged across four backbones.

VisCal addresses these requirements at two levels. **VisCal-Ext** freezes the reasoning model and trains a causal controller to imitate hindsight trajectory allocations. It tests utility calibration and joint allocation both as a new decision layer and as a replacement for a deployed fidelity router. **VisCal-OPD** moves the capacity action before visual generation. Its hindsight teacher uses the correct answer and full downstream outcomes of cross-step candidate branches to train capacity, visual content, and later text together. The resulting policy can change not only how much visual workspace a state receives, but also what it contains and how later reasoning uses it (Figure 1, left).

Across four models and eight transfer benchmarks disjoint from training, VisCal-OPD improves the released models by **5.14 percentage points** at matched visual capacity (**6.43 points** on visual-state-centric benchmarks; Figure 1, right). At the released models' accuracy, the same checkpoint uses an estimated **65.1% fewer visual tokens or latents** on average.

Our contributions are:

- **Diagnosing capacity as a task-utility decision.** Across fixed-capacity and fidelity-routed systems, controlled interventions reveal fidelity–utility mismatch, state-dependent optima, and cross-step interactions.
- **Establishing utility calibration as a deployable principle.** With the backbone frozen and controller architecture, inputs, data, and training procedure shared, replacing fidelity labels with utility labels improves deployment; trajectory-level labels add a further gain.
- **Internalizing capacity with VisCal-OPD.** Beyond standard outcome-weighted distillation, VisCal-OPD adds budget-conditioned capacity actions and cross-step branch interventions, using one hindsight teacher to train capacity, visual generation, and text together.

2 Problem Setup

Given an input x (the prompt and question image) with correct answer $y^\star$, a visual reasoning model produces intermediate states $V_1, \ldots, V_T$ along an interleaved trajectory. For model σ, state V_t has capacity $k_t \in \mathcal{K}_\sigma$, measured in visual tokens or latents; $\mathbf{k} = (k_1, \ldots, k_T)$ denotes the trajectory's capacity allocation. The question image follows the model's native processing and is excluded from this intermediate-state capacity space and its cost.

Frozen capacity interventions. For a reference trajectory τ_x^{ref}, an intervention $\tau_x^{\mathrm{ref}}[\mathbf{k}]$ keeps the input, source visual content, and text trace fixed while re-representing each visual state at capacity k_t. Its joint downstream utility is the correct-answer log-likelihood,

$$\mathcal{U}_x(\mathbf{k}) = \log p\big(y^\star \mid \tau_x^{\mathrm{ref}}[\mathbf{k}]\big)\,. \tag{1}$$

For reference allocation $\mathbf{k}^{\mathrm{ref}}$, the controlled single-state slice

$$U_{x,t}(k) = \mathcal{U}_x\big(k_1^{\mathrm{ref}}, \ldots, k_{t-1}^{\mathrm{ref}}, k, k_{t+1}^{\mathrm{ref}}, \ldots, k_T^{\mathrm{ref}}\big) \tag{2}$$

changes only the capacity of V_t while all other states retain their reference capacities; $\mathcal{U}_x(\mathbf{k})$ can vary the full allocation.

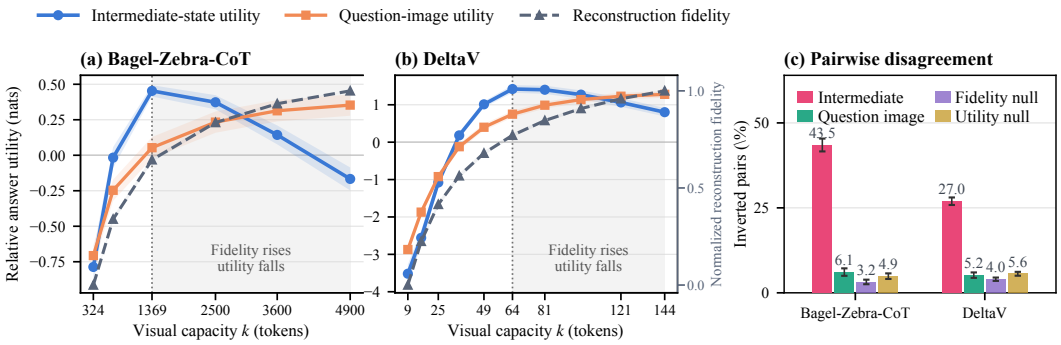


Figure 2: **Fidelity does not reliably rank intermediate-state utility.** Answer utility in (a–b) is centered within each model–role pair; shading and error bars show 95% confidence intervals.

Fidelity and total capacity. Let $\phi_{x,t}^{(j)}(k)$ measure agreement between V_t represented at capacity k and the original ground-truth intermediate state. The family includes pixel, perceptual, semantic-feature, and system-native reconstruction or router metrics, with larger values indicating higher fidelity. Local token uncertainty is a utility-side signal, not a fidelity metric. For multi-step interventions, we compare allocations within the same model at equal raw total capacity $C_{\text{raw}}(\mathbf{k}) = \sum_{t=1}^{T} k_t$.

These definitions let us test fidelity ranking, state dependence, and cross-step additivity.

3 Diagnosing Visual Capacity

Diagnostic scope. We study two models with different capacity mechanisms and deployment practices. Bagel-Zebra-CoT has no capacity-decision layer: its released configuration fixes a single visual capacity globally, across all problems and reasoning steps (Li et al., 2025). Its capacity is the number of ViT tokens used to re-consume an intermediate image, which we vary by resizing that image. DeltaV instead uses a learned variable-length visual tokenizer and deploys a TSIM router calibrated to reconstruction fidelity (Wang et al., 2026a). We analyze trajectories in which intermediate visual states measurably affect answer likelihood.

Does fidelity select the useful capacity? For every pair of capacities, we ask whether reconstruction fidelity and correct-answer likelihood prefer the same representation. We count a robust disagreement only when both differences exceed the variation induced by nuisance perturbations, and call it an ε-inversion. Two controls give this rate a noise scale: the fidelity–fidelity null compares reconstruction metrics, while the utility–utility null compares nuisance-equivalent answer scorers. We apply the same capacity ladder and encoding path to the question image while holding the reasoning trace fixed, testing whether the intervention itself disrupts visual inputs.

In both models, intermediate-state utility reaches an interior optimum and then declines even as fidelity continues to improve (Figure 2). The resulting fidelity–utility inversion rate is 27–44%, whereas the question-image control and both nuisance nulls remain at or below 6.1%. The failure is therefore specific to using reconstruction fidelity to rank model-generated reasoning states, rather than a generic consequence of changing visual-input capacity.

Can one rule serve every state? The fidelity mismatch leaves open whether one fixed capacity suffices for most reasoning states. We inspect each state's utility-maximizing capacity: no single tier is optimal for a majority (Figure 3a–b).

We compare four ways to choose capacity. *Best constant* fixes one capacity across problems and steps; *Step schedule* varies it only by visual-step index; and *Fidelity saturation* stops when reconstruction quality saturates. The *per-state ceiling* instead selects the best capacity from each state's full utility curve; it is a hindsight reference, not a deployable policy. Each deployable rule uses its best validation setting.

Even the strongest deployable rule, Step schedule, leaves 0.28 and 0.31 nats of headroom below the per-state ceiling in Bagel-Zebra-CoT and DeltaV, respectively (Figure 3c). Useful capacity must

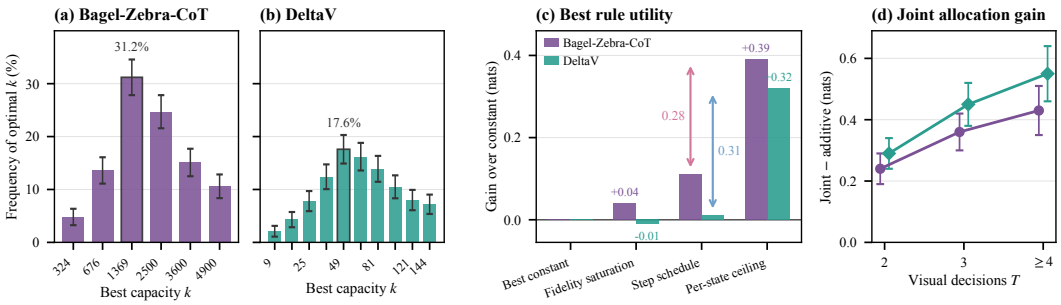


Figure 3: **Useful capacity varies by state and interacts across steps.** Joint and additive allocations in (d) have equal total capacity; error bars show 95% confidence intervals.

therefore be chosen from the current problem and reasoning state, not from a global constant, step index, or fidelity threshold alone.

Can visual steps be optimized independently? State dependence does not by itself imply interaction across steps. A natural strategy is to measure each step while all other steps remain at their reference capacities, then add those single-step effects to score a full allocation:

$$\mathcal{U}_{x,\mathrm{add}}(\mathbf{k}) = u_{x,0} + \sum_{t=1}^{T} u_{x,t}(k_t). \tag{3}$$

We test this model on capacity combinations withheld during fitting. If the steps were independent, its prediction error would approach the utility noise scale. Instead, held-out error spans $2.44\times$–$5.18\times$ that scale and grows with trajectory length. A plausible alternative is that the additive model misses a global saturation or curvature as total capacity increases, rather than an interaction that depends on which steps receive that capacity. We test it with a budget-only quadratic correction,

$$\mathcal{U}_{x,\mathrm{bud}}(\mathbf{k}) = \mathcal{U}_{x,\mathrm{add}}(\mathbf{k}) + a_1 \widetilde{C}_{\mathrm{raw}}(\mathbf{k}) + a_2 \widetilde{C}_{\mathrm{raw}}(\mathbf{k})^2, \tag{4}$$

where $\widetilde{C}_{\mathrm{raw}}$ is standardized total capacity. Even after this correction, 85–89% of the additive residual variance remains, showing that the missing structure concerns *where* capacity is placed, not merely how much is spent.

This misspecification changes the allocation itself: at equal total capacity, choosing by measured joint utility outperforms choosing by summed single-step slices, with a widening gap as trajectories contain more visual decisions (Figure 3d). Multi-step capacity must therefore be allocated jointly over the trajectory.

Do frozen rankings survive free decoding? The preceding measurements hold the future trace fixed. To test whether their ordering persists in actual generation, we force each candidate capacity from the same causal prefix and then let the model freely regenerate the remainder of the trajectory. Frozen-trajectory utility and free-rollout correct-answer log-likelihood rank capacities concordantly in both models, with Kendall's τ of 0.62 and 0.58 for Bagel-Zebra-CoT and DeltaV, respectively. The measured utility rankings therefore remain predictive when the model is allowed to alter its subsequent trajectory. Appendix C gives the intervention and control protocols, full capacity-rule curves, and rollout checks.

What the diagnosis requires of a capacity policy. **(i)** Its learning signal must reflect downstream outcomes rather than reconstruction fidelity. **(ii)** Its action must depend on the current problem and reasoning state. **(iii)** Its allocation must account for interactions across visual steps.

These requirements are revealed by hindsight targets that use $y^\star$, which is available during calibration but not deployment. We next turn the outcome signal into causal capacity policies while keeping the underlying reasoning model frozen.

4 From Diagnosis to Causal Deployment

A controlled causal deployment test. The diagnosis evaluates capacity choices using the correct answer, which is unavailable at deployment. We test whether its findings can guide causal capacity

decisions by freezing the reasoning model and varying only the supervision of a lightweight external controller.

At deployment, the controller should maximize trajectory outcome under an average visual-capacity budget. Let $\mathcal{O}(\tau)$ denote the task outcome, $c_\sigma(k) = k/k_{\max,\sigma}$ the normalized step cost, and $C(\tau) = \sum_{t=1}^{T(\tau)} c_\sigma(k_t)$ the trajectory cost. The constrained problem and its Lagrangian are

$$\begin{aligned} &\max_{\pi} \quad \mathbb{E}_{x,\tau\sim\pi(\cdot|x)}[\mathcal{O}(\tau)] && \text{s.t.} \quad \mathbb{E}_{x,\tau\sim\pi(\cdot|x)}[C(\tau)] \le B, \\ &\mathcal{J}_\lambda(\pi) = \mathbb{E}_{x,\tau\sim\pi(\cdot|x)}[\mathcal{O}(\tau) - \lambda C(\tau)], && \lambda \ge 0. \end{aligned} \tag{5}$$

An optimizer π_λ is optimal at its induced expected budget $B_\lambda = \mathbb{E}[C(\tau)]$, so the capacity price λ indexes a family of operating points.

One controller, three supervision targets. Let h_t be the prefix before the t-th capacity decision. After the frozen model produces V_t, an external controller selects the capacity used to re-represent and consume it from

$$o_t = g(h_t, \lambda) = \big[t, \delta_t, \nu_t, z_{t-1}, \bar{c}_{<t}, \lambda\big],$$

where δ_t measures visual change, ν_t is normalized text uncertainty, z_{t-1} is the previous log-capacity position, and $\bar{c}_{<t}$ is mean past expenditure. This observation contains neither the correct answer nor future states or actions. A lightweight GRU and model-specific adapter map $o_{\le t}$ to the nearest legal capacity tier.

We train three controllers with the same architecture, causal inputs, state pools, and training procedure. At each sampled price, only their hindsight labels differ:

$$k^{\text{fid}}_{x,t}(\lambda) = \arg\max_{k\in\mathcal{K}_\sigma} \big[\widetilde{\phi}_{x,t}(k) - \lambda c_\sigma(k)\big], \qquad k^{\text{step}}_{x,t}(\lambda) = \arg\max_{k\in\mathcal{K}_\sigma} \big[\bar{U}_{x,t}(k) - \lambda c_\sigma(k)\big],$$

$$\mathbf{k}^{\text{traj}}_x(\lambda) \approx \arg\max_{\mathbf{k}\in\mathcal{K}_\sigma^T} \left[\bar{\mathcal{U}}_x(\mathbf{k}) - \lambda \sum_t c_\sigma(k_t)\right].$$

Bars denote answer-length-normalized correct-answer log-likelihood. Fidelity uses the best validation-selected metric from a fixed reconstruction family. Per-Step Utility replaces fidelity with downstream utility, but chooses each step while holding the others at their reference capacities. Trajectory Utility instead searches the joint allocation. Search and label-construction details are in Appendix D.

To train VisCal-Ext, a hindsight teacher uses the correct answer and complete trajectory utilities to produce capacity-allocation labels. A GRU learns these labels from $o_{\le t}$ alone. At deployment, the teacher is discarded; each visual state requires only online feature extraction, one GRU prediction, and the capacity adapter—not the correct answer, hindsight utility, joint search, or future trajectory.

Table 1: **Utility improves external control.**

Method	Bagel-Zebra-CoT	DeltaV
(a) Accuracy at matched visual capacity		
Best non-state rule (Step schedule)	67.2	57.5
Identical controllers, three targets		
Fidelity	68.1	58.0
Per-Step Utility	70.7(+2.6)	61.4(+3.4)
Trajectory Utility	**73.5**(+5.4)	**65.4**(+7.4)
Trajectory − Per-Step, 1 visual state	+0.3	+0.4
Trajectory − Per-Step, 2+ visual states	+3.7	+5.2
(b) Released model vs. VisCal-Ext		
Released accuracy (%)	64.6	57.2
VisCal-Ext accuracy (%)	**73.5**(+8.9)	**65.4**(+8.2)
Released tokens / problem	11,642	176.8
VisCal-Ext tokens / problem	2,773(−76%)	128.5(−27%)

Utility supervision improves causal deployment. Table 1(a) reports complete free rollouts at matched capacity. Per-Step Utility exceeds Fidelity in both models, showing that downstream utility is a better calibration target than reconstruction fidelity.

Trajectory Utility improves further over Per-Step Utility, showing the benefit of joint trajectory-level labels. The gain matches the cross-step pattern in Section 3: negligible with one visual state, substantial when multiple visual decisions are present (Table 1(a)).

We carry the Trajectory Utility controller forward as **VisCal-Ext**. It is the strongest controller in both models, filling a missing capacity-decision layer in Bagel-Zebra-CoT and replacing DeltaV's deployed fidelity router. Against each released system's operating policy, VisCal-Ext gains $8.9/8.2$ accuracy points while using 76%/27% fewer intermediate visual tokens (Table 1(b)).

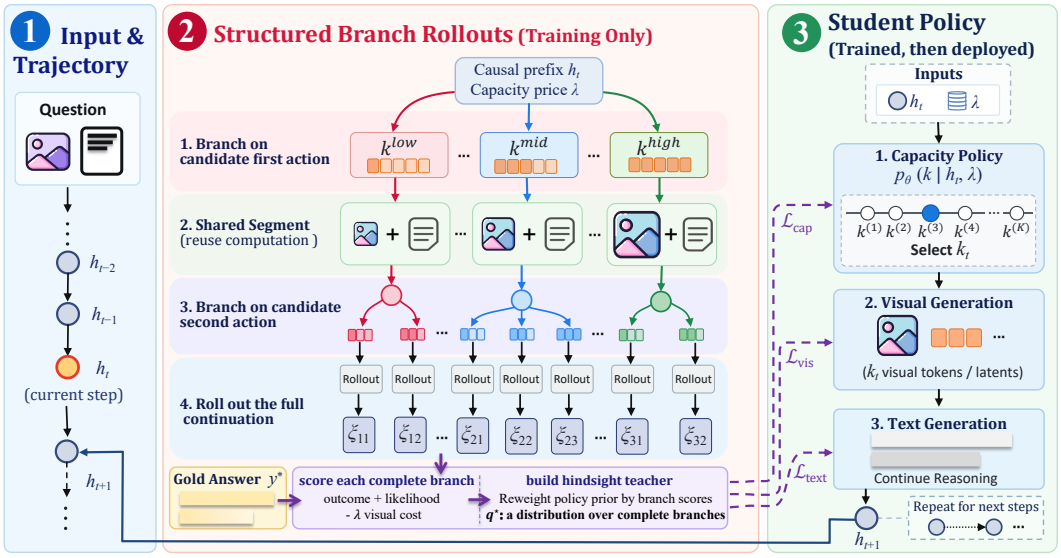


Figure 4: **Overview of VisCal-OPD.**

5 VisCal-OPD

VisCal-Ext selects capacity after a visual state is produced, but the frozen model cannot learn to adapt the visual content or subsequent text to that choice. VisCal-OPD internalizes the capacity decision by placing it before visual generation, allowing capacity, visual content, and subsequent reasoning to adapt together. Complete downstream outcomes then jointly supervise capacity, visual, and text generation (Figure 4).

Budget-conditioned capacity before generation. At each visual decision, the model selects capacity before generating the visual state:

$$h_t, \lambda \longrightarrow k_t \sim \pi_\theta^{\text{cap}}(\cdot \mid h_t, \lambda) \longrightarrow V_t \sim \pi_\theta^{\text{vis}}(\cdot \mid h_t, k_t, \lambda) \longrightarrow \text{future visual and text states.} \quad (6)$$

A capacity head selects $k_t \in \mathcal{K}_\sigma$ and inserts its token into the context before visual generation; a model-specific actuator realizes the choice. During training, we sample $\lambda \sim p_\Lambda$ per problem and condition the policy on it, training one checkpoint across the operating points of Equation 5. At deployment, λ selects an operating point without retraining. Model-specific interfaces are in Appendix E.

Structured capacity interventions. At a prefix h_t visited by the current student π_{old}, we consider two consecutive capacity actions as a block $\kappa = (k_t, k_{t+1})$, reverting to $\kappa = (k_t)$ when no further visual decision is reached. The finite candidate set $\mathcal{A}_t$ includes the actual block, adjacent capacity changes, cross-step reallocations, and policy-sampled exploration. The reallocations preserve total cost as closely as the discrete tiers permit, directly comparing where capacity is useful at comparable total expenditure.

Candidates expand as a tree: blocks with the same first action k share one generated visual state and intervening text up to $h_{t+1}^{(k)}$, then branch at the second action and independently continue to the final answer. Second actions are thus compared at the same reasoning prefix. Each complete branch ξ contains its capacity block and subsequent visual and text states; we denote the resulting finite branch set by Ξ_t.

Outcome-calibrated hindsight teacher. We score each branch using the task outcome and its remaining visual cost, instantiating Equation 5 as

$$G_\lambda(\xi) = r_{\text{task}}(\xi) + w_{\text{ll}} \bar{\ell}_x(\xi) - \lambda C_{t:\text{end}}(\xi). \quad (7)$$

Here r_{task} is the task reward, and the length-normalized correct-answer log-likelihood $\bar{\ell}_x$ supplies a dense signal for comparing branches, with fixed weight w_{ll}. Cost includes all visual states from the intervention onward, including extra steps caused by an initially smaller capacity. The hindsight teacher increases the weight of high-return branches:

$$q_t^\star(\xi \mid h_t, \lambda) = \frac{1}{Z_t} p_{\text{old}}(\xi \mid h_t, \lambda) \exp(G_\lambda(\xi)/\eta)\,. \quad (8)$$

The prior p_{old} restricts the old policy to $\mathcal{A}_t$ and divides each block's mass uniformly among its continuations. Z_t normalizes the teacher over Ξ_t, and $\eta > 0$ controls the strength of this KL-regularized reweighting. The teacher uses the correct answer $y^\star$ and the complete downstream outcome of each branch; the deployed student observes only the causal prefix and capacity price. The derivation is in Appendix E.

Joint on-policy distillation. Each branch's teacher weight supervises its capacity choice, visual content, and subsequent text. The capacity loss $\mathcal{L}_{\text{cap}}$ fits the teacher's first-action marginal at h_t and its second-action conditional at the shared prefix $h_{t+1}^{(k)}$, preserving cross-step dependencies. The resulting joint objective is

$$\mathcal{L}_m = \sum_{\xi \in \Xi_t} q_t^\star(\xi \mid h_t, \lambda) \mathcal{L}_m^{\text{native}}(\xi), \qquad m \in \{\text{text}, \text{vis}\}, \tag{9}$$

$$\mathcal{L}_{\text{OPD}} = \mathcal{L}_{\text{cap}} + w_{\text{text}}\mathcal{L}_{\text{text}} + w_{\text{vis}}\mathcal{L}_{\text{vis}} + w_{\text{ret}}\mathcal{L}_{\text{ret}}. \tag{10}$$

Text uses token-normalized negative log-likelihood; vision uses each model's native token-likelihood, flow-matching, or continuous-latent objective. The retention term $\mathcal{L}_{\text{ret}}$ uses general replay to preserve the released model's capabilities. Detailed losses are in Appendix E.

Each round uses the current student to produce decision prefixes, construct fresh branches and teachers, and apply Equation 10; the updated student repeats this process in the next round. The resulting budget-conditioned generative policy internalizes the teacher's outcome-informed choices and deploys without branch search or access to the correct answer.

6 Experiments

6.1 Experimental setup

We evaluate Bagel-Zebra-CoT, DeltaV, Monet, and ThinkMorph, spanning resized image tokens, a learned variable-length tokenizer, and continuous visual latents. Each trainable method uses the same 9,600-problem post-training pool from Zebra-CoT (Li et al., 2025) for its model and is frozen before evaluation. The test suite contains four visual-state-centric benchmarks (MIRA (Zhou et al., 2025), ViC-Bench (Wu et al., 2025), RBench-V (Guo et al., 2025), and MathCanvas (Shi et al., 2026)) and four general multimodal reasoning benchmarks (MathVista (Lu et al., 2024), M^{3}CoT (Chen et al., 2024), VisualPuzzles (Song et al., 2025), and VStar (Wu & Xie, 2024)), with training overlaps removed. We reserve a problem-disjoint Zebra-CoT split for the mechanism study.

We compare each released model under its native capacity policy with VisCal-Ext and VisCal-OPD at matched actual visual capacity. VisCal-Ext uses a frozen generator; VisCal-OPD jointly adapts capacity, visual content, and text. Standard OPD follows the same post-training recipe without capacity actions or structured cross-step branches. Complete protocols and model-specific interfaces are in Appendices B and E.

6.2 Performance across models and benchmarks

At matched visual capacity, VisCal-OPD outperforms both the released model and VisCal-Ext in all 32 model–benchmark pairs, improving the released models by 5.14 percentage points on average (Table 2). Gains are larger on the visual-state-centric suite than on the general multimodal suite (6.43 vs. 3.85 points), consistent with capacity calibration having its greatest effect when reasoning depends on generated visual states. With 2B–7B open-weight backbones, VisCal-OPD substantially narrows the gap to reported proprietary MLLM scores (Table 2).

Table 3 separates the two gains. With the generator frozen, VisCal-Ext improves the released models by 2.13 points (2.99 on visual-state-centric benchmarks); internalizing the decision adds another 3.01 points (3.44 on visual-state-centric benchmarks). Under matched post-training, Standard OPD gains only 0.80 points over the released models, while VisCal-OPD adds 4.44 points beyond it. VisCal-OPD outperforms both VisCal-Ext and Standard OPD in all 32 model–benchmark cells. These comparisons show that utility calibration works as an external plug-in and that internalization adds capacity-specific learning beyond generic outcome-weighted post-training.

Table 2: Accuracy (%) across eight benchmarks.

Model	# Params	Visual-State-Centric Benchmarks				General Multimodal Reasoning Benchmarks			
		MIRA	ViC-Bench	RBench-V	MathCanvas	MathVista	M^3CoT	VisPuzzles	VStar
Closed-source MLLMs									
GPT-4o (OpenAI, 2024)	—	11.2	40.6	14.1	19.4	63.8	64.3	41.3	61.8
Gemini-2.5-Pro (Gemini Team, 2025)	—	16.9	60.3	20.2	58.2	84.6	85.9	49.5	79.2
o3 (OpenAI, 2025)	—	16.4	56.1	25.8	—	86.8	83.8	54.0	95.7
o4-mini (OpenAI, 2025)	—	15.6	58.9	20.9	—	84.3	80.9	57.0	94.6
Open-weight Understanding MLLMs									
Qwen2.5-VL (Bai et al., 2025)	7B	—	—	8.3	18.7	68.2	62.4	33.7	76.4
Qwen2.5-VL (Bai et al., 2025)	72B	13.1	47.9	10.6	32.8	74.8	70.2	42.3	85.9
InternVL3.5 (Wang et al., 2025)	8B	—	—	—	26.4	74.1	59.9	34.8	68.6
LLaVA-OneVision (Li et al., 2024)	72B	—	34.1	9.0	—	67.5	61.5	30.8	73.8
Open-weight Unified MLLMs									
BAGEL (Deng et al., 2025)	7B	7.5	—	—	18.5	72.5	31.4	35.0	55.5
Janus-Pro (Chen et al., 2025)	7B	4.9	—	—	—	42.7	49.1	33.5	38.2
Interleaved Visual Reasoning Models									
Bagel-Zebra-CoT	7B	9.9	37.8	9.7	17.1	72.1	62.6	31.1	64.9
+VisCal-Ext	7B	11.9	41.8	12.7	20.0	71.6	66.9	33.5	64.6
+VisCal-OPD	7B	14.7 (+4.8)	45.8 (+8.0)	15.8 (+6.1)	23.5 (+6.4)	74.6 (+2.5)	69.2 (+6.6)	34.2 (+3.1)	66.5 (+1.6)
DeltaV	2B	10.4	24.3	8.8	16.4	69.3	54.5	29.7	75.9
+VisCal-Ext	2B	12.4	28.2	13.0	20.1	69.7	56.4	32.0	76.1
+VisCal-OPD	2B	14.0 (+3.6)	33.3 (+9.0)	15.6 (+6.8)	24.0 (+7.6)	70.8 (+1.5)	61.5 (+7.0)	35.6 (+5.9)	76.3 (+0.4)
Monet	7B	9.5	36.6	11.8	21.8	62.5	44.2	35.0	79.1
+VisCal-Ext	7B	12.0	40.9	14.6	24.6	63.8	47.2	37.4	78.4
+VisCal-OPD	7B	14.2 (+4.7)	47.5 (+10.9)	19.0 (+7.2)	28.5 (+6.7)	64.7 (+2.2)	53.3 (+9.1)	39.2 (+4.2)	80.3 (+1.2)
ThinkMorph	7B	8.6	40.9	11.2	20.6	67.8	48.8	39.0	67.0
+VisCal-Ext	7B	10.2	43.8	13.8	23.5	68.6	51.6	40.6	65.6
+VisCal-OPD	7B	12.0 (+3.4)	47.8 (+6.9)	16.6 (+5.4)	26.0 (+5.4)	69.9 (+2.1)	56.1 (+7.3)	43.5 (+4.5)	69.3 (+2.3)

Table 3: **Controlled accuracy differences (pp), averaged over four models.**

Accuracy difference (pp)	*Visual-State-Centric*				*General Multimodal Reasoning*				*Mean*	
	MIRA	ViC-Bench	RBench-V	MathCanvas	MathVista	M^3CoT	VisPuzzles	VStar	All 8	Visual 4
External calibration and internalization										
VisCal-Ext − released model	+2.01	+3.77	+3.11	+3.08	+0.49	+3.00	+2.13	−0.57	+2.13	+2.99
VisCal-OPD − VisCal-Ext	+2.09	+4.94	+3.27	+3.45	+1.57	+4.52	+2.29	+1.92	+3.01	+3.44
Controlled post-training comparison										
Standard OPD − released model	+0.63	+1.91	−0.02	+1.10	+0.29	+2.24	+0.35	−0.13	+0.80	+0.90
VisCal-OPD − Standard OPD	+3.45	+7.18	+6.52	+5.23	+1.71	+5.43	+4.22	+1.75	+4.44	+5.60
Full method										
VisCal-OPD − released model	+4.11	+8.71	+6.38	+6.53	+2.07	+7.52	+4.42	+1.35	+5.14	+6.43

6.3 Accuracy–Capacity Frontier

Sweeping VisCal-OPD's inference-time capacity price lets one checkpoint trace the accuracy–capacity frontier without retraining.

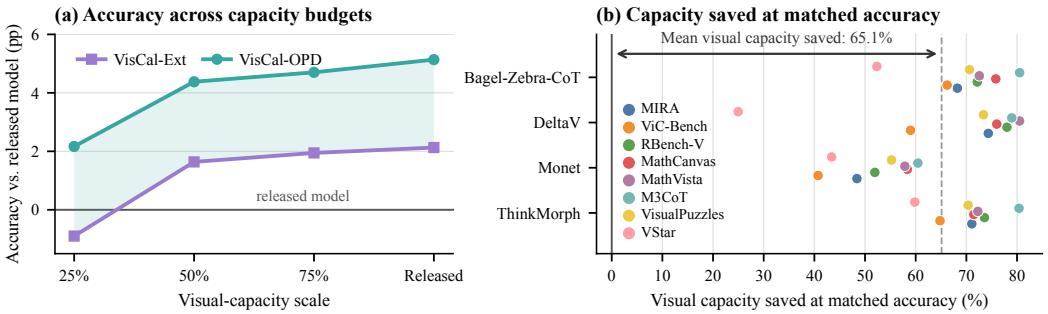


Figure 5: **VisCal-OPD improves the accuracy–capacity tradeoff.** Left: mean over 32 model–benchmark pairs. Right: one dot per pair.

The 32-cell average for VisCal-OPD stays above VisCal-Ext at all four displayed budgets (Figure 5, left). At the released models' accuracy, the same checkpoints use an estimated 65.1% fewer visual tokens or latents on average (right). Task-aligned capacity therefore improves both sides of the tradeoff: more accurate reasoning at a fixed budget and less visual capacity at fixed accuracy.

6.4 Ablations and mechanisms of capacity internalization

Beyond external allocation, VisCal-OPD learns cross-step capacity choices within the generative policy and adapts the visual and textual content they produce. We test these components on a problem-disjoint, in-distribution Zebra-CoT split for Bagel-Zebra-CoT and DeltaV, where the full method gains 14.08 points over the released models.

Joint allocation matters when visual steps interact. Suppose a two-step capacity block is $\kappa = (k_t, k_{t+1})$. The full teacher compares single-action changes with cross-step reallocations at comparable total capacity; the single-step ablation evaluates only one action at a time. Joint branches provide no measurable gain on trajectories with one visual decision, but improve both models when multiple steps compete for capacity (Table 4(a)).

Internalization improves what the capacity carries. The capacity-only ablation keeps the same capacity actions, cross-step branches, and outcome teacher but freezes visual and text generation. It can allocate capacity without adapting what it carries or how later reasoning uses it. Updating the visual and text generators adds more than five accuracy points at the released budget, with still larger gains at the lowest tested budget (Table 4(b)).

At a fixed reasoning prefix and capacity, we measure each trained model's visual-state distance from the released model relative to ordinary run-to-run variation (the $1\times$ generation-noise floor). Capacity-only states remain near this floor, whereas full VisCal-OPD moves beyond $2\times$; most of these changes also increase downstream answer utility.

To test whether the gain instead comes from moving visual information into text, we compare correct-answer likelihood with and without the intermediate visual states. After full VisCal-OPD, retaining these states provides a substantially larger likelihood advantage, showing that later reasoning relies more, not less, on the learned visual states.

Table 4: **Joint allocation and visual–text co-adaptation matter.**

Comparison / measurement	Bagel-Zebra-CoT	DeltaV
(a) Cross-step allocation: VisCal-OPD − single-step (pp)		
One visual decision in trajectory	−0.49	0.00
Multiple visual decisions in trajectory	+4.70	+7.49
(b) Visual and text co-adaptation		
VisCal-OPD − capacity-only (pp), released-model budget	+5.10	+6.37
VisCal-OPD − capacity-only (pp), lowest tested budget	+5.70	+9.53
State distance / noise floor: capacity-only → VisCal-OPD	$1.08\times \rightarrow 2.18\times$	$1.01\times \rightarrow 2.60\times$
Visual states with higher utility than released (%)	68.3	74.9
Visual reliance gain (nats): capacity-only → VisCal-OPD	$+0.27 \rightarrow +0.84$	$+0.18 \rightarrow +1.14$

Together, these tests show that visual capacity must be allocated jointly across steps and learned with its content; inference FLOPs remain essentially unchanged. Further mechanism breakdowns, intervals, and inference-cost measurements are in Appendix F.

7 Conclusion

We show that visual intermediate-state capacity should be calibrated by its downstream effect on the answer, rather than by fidelity to a reference state. The diagnosis exposes this mismatch; controlled external calibration shows that utility supervision and joint allocation each improve decisions with a frozen backbone; VisCal-OPD makes capacity a budget-conditioned action that co-adapts with visual content and later reasoning. This recasts capacity as visual workspace invested in the current reasoning state, whether carried by re-encoded image tokens or continuous latents. It should be priced by task outcomes, allocated jointly across the trajectory, and learned with the content it carries. Less visual capacity and stronger reasoning are compatible.

AI use statement

In this work, we used generative AI tools solely for retrieval and discovery, such as identifying potentially relevant related work and assisting with literature search. Any references or papers sug-

gested by generative AI tools were independently checked by the authors against the original sources before being considered or cited.

We did not use generative AI tools to aid or polish the writing, for research ideation or execution, to draft sections of the paper, to generate synthetic datasets, or to prove mathematical claims. We did not use generative AI tools for any other purposes requiring disclosure.

We reviewed all AI-assisted retrieval and discovery results and take responsibility for the final content of this work, including all text, claims, experiments, analyses, and citations.

Reproducibility statement

The problem setup, controller, VisCal-OPD method, and evaluations are described in Sections 2, 4, 5, and 6. Appendix B documents data construction and partitions, capacity and budget accounting, rollout settings, and statistical procedures. Appendices C, D, and E provide the diagnostic interventions, controller calibration, and training details; Appendix F reports the extended results and ablations.

References

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes, 2023. URL https://arxiv.org/abs/2306.13649.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025. URL https://arxiv.org/abs/2502.13923.

Qiguang Chen, Libo Qin, Jin Zhang, Zhi Chen, Xiao Xu, and Wanxiang Che. M^3CoT: A novel benchmark for multi-domain multi-step multi-modal chain-of-thought. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8199–8221. Association for Computational Linguistics, August 2024. doi: 10.18653/v1/2024.acl-long.446. URL https://arxiv.org/abs/2405.16473.

Xiaokang Chen, Zhiyu Wu, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, and Chong Ruan. Janus-Pro: Unified multimodal understanding and generation with data and model scaling. *arXiv preprint arXiv:2501.17811*, 2025. URL https://arxiv.org/abs/2501.17811.

Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, Guang Shi, and Haoqi Fan. Emerging properties in unified multimodal pretraining. *arXiv preprint arXiv:2505.14683*, 2025. URL https://arxiv.org/abs/2505.14683.

Zizhong Ding, Junxian Li, Kai Liu, Shaoqiu Zhang, Xiao Xiao, Linghe Kong, and Yulun Zhang. ET-Prune: Evidence-aware dynamic budgeting for visual token pruning in text-rich MLLMs, 2026. URL https://arxiv.org/abs/2608.01979.

Gemini Team. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025. URL https://arxiv.org/abs/2507.06261.

Jiawei Gu, Yunzhuo Hao, Huichen Will Wang, Linjie Li, Michael Qizhe Shieh, Yejin Choi, Ranjay Krishna, and Yu Cheng. ThinkMorph: Emergent properties in multimodal interleaved chain-of-thought reasoning, 2026. URL https://arxiv.org/abs/2510.27492.

Meng-Hao Guo, Xuanyu Chu, Qianrui Yang, Zhe-Han Mo, Yiqing Shen, Pei-lin Li, Xinjie Lin, Jinnian Zhang, Xin-Sheng Chen, Yi Zhang, Kiyohiro Nakayama, Zhengyang Geng, Houwen Peng, Han Hu, and Shi-Min Hu. RBench-V: A primary assessment for visual reasoning models with multi-modal outputs, 2025. URL https://arxiv.org/abs/2505.16770.

Zican Hu, Xuyang Hu, Yiming Liu, Zuwei Long, Wei Liu, Yunzhuo Hao, Jiawei Gu, Linjie Li, Yu Cheng, Zhenhong Sun, Weibo Gu, Xing Sun, and Zhi Wang. Bridging interleaved multimodal reasoning as a unified decision process, 2026. URL https://arxiv.org/abs/2607.03748.

Wenxuan Huang, Zijie Zhai, Yunhang Shen, Shaosheng Cao, Fei Zhao, Xiangfeng Xu, Zheyu Ye, and Shaohui Lin. Dynamic-LLaVA: Efficient multimodal large language models via dynamic vision-language context sparsification. In *International Conference on Learning Representations*, volume 2025, pp. 69927–69955, 2025.

Jiaxuan Kang, Siyu Chen, Mingda Li, Mingjie Liu, Tianyue Wang, Zhaoyang Wei, Yongheng Zhang, Yanchao Hao, and Zheng Wei. LUT: Latent utility training for visual reasoning, 2026. URL https://arxiv.org/abs/2608.00743.

Ang Li, Charles Wang, Deqing Fu, Kaiyu Yue, Zikui Cai, Wang Bill Zhu, Ollie Liu, Peng Guo, Willie Neiswanger, Furong Huang, Tom Goldstein, and Micah Goldblum. Zebra-CoT: A dataset for interleaved vision language reasoning, 2025. URL https://arxiv.org/abs/2507.16746.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. LLaVA-OneVision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024. URL https://arxiv.org/abs/2408.03326.

Pengyu Li, Zhitao Gao, Lingling Zhang, Muye Huang, Yuanming Li, Zesheng Yang, Fangzhi Xu, and Jun Liu. Visual-OPSD: Cross-modal on-policy self-distillation for efficient unified multimodal reasoning, 2026. URL https://arxiv.org/abs/2606.18974.

Kun Liang, Clive Bai, Xin Xu, Chenming Tang, Sanwoo Lee, Weijie Liu, Saiyong Yang, and Yunfang Wu. ORBIT: On-policy exploration-exploitation for controllable multi-budget reasoning, 2026. URL https://arxiv.org/abs/2601.08310.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. MathVista: Evaluating mathematical reasoning of foundation models in visual contexts. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/abs/2310.02255.

OpenAI. GPT-4o system card. *arXiv preprint arXiv:2410.21276*, 2024. URL https://arxiv.org/abs/2410.21276.

OpenAI. OpenAI o3 and o4-mini System Card, 2025. URL https://openai.com/index/o3-o4-mini-system-card/.

Zikun Qu, Min Zhang, Mingze Kong, Zhiwei Shang, Zhengyu Chen, Yikun Ban, Shuang Qiu, and Zhongxiang Dai. SPOT: Sparse probing and outcome calibration for on-policy distillation, 2026. URL https://arxiv.org/abs/2608.04419.

Weikang Shi, Aldrich Yu, Rongyao Fang, Houxing Ren, Ke Wang, Aojun Zhou, Changyao Tian, Xinyu Fu, Yuxuan Hu, Zimu Lu, Linjiang Huang, Si Liu, Rui Liu, and Hongsheng Li. MathCanvas: Intrinsic visual chain-of-thought for multimodal mathematical reasoning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 27933–27954, San Diego, California, United States, July 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.acl-long.1287. URL https://arxiv.org/abs/2510.14958.

Yueqi Song, Tianyue Ou, Yibo Kong, Zecheng Li, Graham Neubig, and Xiang Yue. VisualPuzzles: Decoupling multimodal reasoning evaluation from domain knowledge, 2025. URL https://arxiv.org/abs/2504.10342.

Jintao Tong, Shilin Yan, Hongwei Xue, Xiaojun Tang, Kunyu Shi, Guannan Zhang, Ruixuan Li, and Yixiong Zou. SwimBird: Eliciting switchable reasoning mode in hybrid autoregressive MLLMs, 2026. URL https://arxiv.org/abs/2602.06040.

Pengjie Wang, Linger Deng, Zujia Zhang, Shaojie Zhang, Zhenbo Luo, Pei Fu, Jian Luan, Xiang Bai, and Yuliang Liu. DeltaV: Thinking with visual state updates in unified large multimodal models, 2026a. URL https://arxiv.org/abs/2607.08434.

Qixun Wang, Yang Shi, Yifei Wang, Yuanxing Zhang, Pengfei Wan, Kun Gai, Xianghua Ying, and Yisen Wang. Monet: Reasoning in latent visual space beyond image and language. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12030–12040, June 2026b.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, Zhaokai Wang, Zhe Chen, Hongjie Zhang, Ganlin Yang, Haomin Wang, Qi Wei, Jinhui Yin, Wenhao Li, Erfei Cui, Guanzhou Chen, Zichen Ding, Changyao Tian, Zhenyu Wu, Jingjing Xie, Zehao Li, Bowen Yang, Yuchen Duan, Xuehui Wang, Zhi Hou, Haoran Hao, Tianyi Zhang, Songze Li, Xiangyu Zhao, Haodong Duan, Nianchen Deng, Bin Fu, Yinan He, Yi Wang, Conghui He, Botian Shi, Junjun He, Yingtong Xiong, Han Lv, Lijun Wu, Wenqi Shao, Kaipeng Zhang, Huipeng Deng, Biqing Qi, Jiaye Ge, Qipeng Guo, Wenwei Zhang, Songyang Zhang, Maosong Cao, Junyao Lin, Kexian Tang, Jianfei Gao, Haian Huang, Yuzhe Gu, Chengqi Lyu, Huanze Tang, Rui Wang, Haijun Lv, Wanli Ouyang, Limin Wang, Min Dou, Xizhou Zhu, Tong Lu, Dahua Lin, Jifeng Dai, Weijie Su, Bowen Zhou, Kai Chen, Yu Qiao, Wenhai Wang, and Gen Luo. InternVL3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency, 2025. URL https://arxiv.org/abs/2508.18265.

Penghao Wu and Saining Xie. V*: Guided visual search as a core mechanism in multimodal LLMs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13084–13094, June 2024. doi: 10.1109/CVPR52733.2024.01243. URL https://arxiv.org/abs/2312.14135.

Xuecheng Wu, Jiaxing Liu, Danlei Huang, Yifan Wang, Yunyun Shi, Kedi Chen, Junxiao Xue, Yang Liu, Chunlin Chen, Hairong Dong, and Dingkang Yang. ViC-Bench: Benchmarking visual-interleaved chain-of-thought capability in MLLMs with free-style intermediate state representations, 2025. URL https://arxiv.org/abs/2505.14404.

Tianyi Zhang, Mahtab Bigverdi, and Ranjay Krishna. Ablate-to-Validate: Are vision-language models really using continuous thought tokens?, 2026. URL https://arxiv.org/abs/2605.21642.

Yiyang Zhou, Haoqin Tu, Zijun Wang, Zeyu Wang, Niklas Muennighoff, Fan Nie, Yejin Choi, James Zou, Chaorui Deng, Shen Yan, Haoqi Fan, Cihang Xie, Huaxiu Yao, and Qinghao Ye. When visualizing is the first step to reasoning: MIRA, a benchmark for visual chain-of-thought, 2025. URL https://arxiv.org/abs/2511.02779.

A Related Work

A.1 Generated visual states and capacity mechanisms

Zebra-CoT and ThinkMorph generate visual states within interleaved reasoning trajectories and consume them in subsequent steps (Li et al., 2025; Gu et al., 2026). Monet instead uses continuous visual latents as an intermediate workspace (Wang et al., 2026b). Their released configurations assign a fixed capacity to each visual state. DeltaV uses a variable-length visual tokenizer with a TSIM router calibrated by marginal reconstruction gain, while SwimBird learns when to end a visual-thought span (Wang et al., 2026a; Tong et al., 2026). They show how visual-state capacity is set through fixed configurations, reconstruction-calibrated routing, or learned termination.

Visual-OPSD examines the contribution of generated visual thoughts, while Ablate-to-Validate uses interventions to test whether later reasoning relies on visual content (Li et al., 2026; Zhang et al., 2026). The same question motivates our visual-dependence checks; our diagnosis also asks how the capacity of a relied-upon state changes downstream answer utility.

A.2 Visual-token reduction from observed inputs

Dynamic-LLaVA learns to sparsify visual and language context, and ET-Prune retains question-relevant tokens from an observed image (Huang et al., 2025; Ding et al., 2026). In interleaved visual reasoning, capacity is assigned repeatedly to states generated by the model itself. Each choice changes the visual material available to later steps, so its value depends on other capacity choices along the same trajectory.

A.3 Outcome-guided learning and budget control

On-policy distillation trains a student on its own trajectories using teacher supervision (Agarwal et al., 2023). SPOT probes next-token candidates and uses their downstream outcomes to form soft targets; ORBIT distills policies with different text-reasoning effort levels into one controllable checkpoint (Qu et al., 2026; Liang et al., 2026). For visual reasoning, LUT trains answer-useful latent trajectories under a predefined latent-step budget, while BRAID applies shared outcome feedback to interleaved text and image generation (Kang et al., 2026; Hu et al., 2026).

VisCal makes the capacity of each generated visual state a decision within a shared reasoning trajectory. Its diagnosis exposes the mismatch between reconstruction fidelity and answer utility, together with the dependence between capacity choices across steps. A frozen-generator controller isolates the value of utility-calibrated allocation. VisCal-OPD then commits a budget-conditioned capacity action before visual generation and learns that action, visual content, and subsequent reasoning from complete branch outcomes.

B Experimental Protocol

Data partitions. We construct one question pool from the approximately 180,000 Zebra-CoT questions. At its published default visual capacity, Bagel-Zebra-CoT answers each question five times with a fixed prompt and decoding configuration and five independent seeds. We retain question IDs answered correctly two to four times, placing the pool near the screening model's empirical competence boundary. This screening is performed once, before the studies: models and methods in each comparison use the same question IDs from this pool, rather than being screened separately. Partitions are made by original question ID, so different rollouts of one question cannot cross a study's train, validation, and test partitions. Evaluation uses seeds independent of the five screening seeds.

The frozen-trajectory diagnosis draws 600 Zebra-CoT trajectories per model from this pool for Bagel-Zebra-CoT and DeltaV; only this diagnosis applies the additional visual-dependence and leakage filters detailed in Appendix C.1. The external-controller study uses 600 training, 200 validation, and 800 test questions per model from the screened pool. After this initial screen, its test partition is not filtered again by answer correctness or whether a visual state is generated; questions with no generated visual state remain in both accuracy and cost.

For post-training, all four models use the same 9,600 Zebra-CoT training questions. The 1,000-question validation and 1,000-question held-out mechanism-test partitions also come from the screened pool and are disjoint from training and each other by question ID. Post-training additionally uses 1,200 retention problems. Cross-benchmark deduplication removed 143 Zebra-CoT examples. The eight transfer benchmarks are not selected by this Zebra-CoT screening rule. The training questions supply input images and final answers; their reference reasoning traces are not used as training targets.

Capacity and cost. Capacity k_t counts the visual tokens or latents actually used for an intermediate state, not the question-image tokens or a nominal configuration value. Table 5 gives each model's legal tiers. The question image is processed natively and is excluded from the capacity action space. A trajectory with $T = 0$ visual states has zero intermediate-state capacity cost.

Table 5: Legal intermediate-state capacity tiers. The detailed actuator implementation is given with each method.

Model	Capacity unit and legal k
Bagel-Zebra-CoT	Re-consumed ViT tokens: $\{324, 676, 1369, 2500, 3600, 4900\}$
ThinkMorph	Re-consumed ViT tokens: $\{324, 676, 1369, 2500, 3600, 4900\}$
DeltaV	Variable-length visual tokens: $k = m^2$, $m \in \{3, \ldots, 12\}$
Monet	Continuous visual latents: $\{2, 4, 6, 8, 10, 12\}$

For model σ, we record raw expenditure $C_{\mathrm{raw}}(\tau) = \sum_{t=1}^{T(\tau)} k_t$ and optimize normalized expenditure $C(\tau) = \sum_{t=1}^{T(\tau)} k_t / k_{\max,\sigma}$. The diagnostic rule comparison matches mean raw capacity per analyzed visual state; the deployment comparisons match mean normalized cost per original problem over complete free rollouts. Both deployment studies select capacity prices using realized visual expenditure on their respective Zebra-CoT validation splits and freeze them before test evaluation. Test-set expenditure supplies the reported budget coordinates; it is not used to select prices. If the formal evaluation exceeds the 2% budget-matching tolerance, we read the metric by linear interpolation between adjacent evaluated prices; we never extrapolate.

Rollouts, scoring, and uncertainty. Within each model–benchmark cell, methods use the same questions, prompts, decoding settings, stopping rules, answer extraction, and benchmark scoring. The post-training transfer evaluation uses sampling with temperature 0.7 and top-$p = 0.95$, with paired rollout seeds $0, 1, 2$; learned capacity actions use argmax at test time. Invalid-format answers, empty outputs, timeouts, and out-of-memory runs count as incorrect rather than being removed. The frozen-trajectory diagnosis instead scores the teacher-forced log-likelihood of the correct answer and reports utility in nats; its selection criteria and free-rollout check are specified with the diagnostic results.

Diagnostic intervals resample source trajectories; deployment intervals resample original problems. Repeated rollout seeds are averaged within each problem before computing paired method differences. The external-controller comparison uses three controller-training seeds and three rollout seeds at its primary budget; its budget curve uses three pre-paired seed combinations. Post-training uses one training seed per model–method pair and three paired rollout seeds. Its confidence intervals quantify variation across test problems after averaging these fixed rollout seeds, not variation across independent retrainings. The deployment studies use 10,000 problem-cluster bootstrap draws and percentile 95% intervals. The 32-cell transfer summary weights model–benchmark cells equally, irrespective of their problem counts.

C Diagnostic Details

C.1 Intervention validity and analysis population

For single-state slices, the diagnostic intervention replays a fixed Zebra-CoT reasoning trace and re-represents one generated visual state at a chosen capacity, leaving its source content and all other states unchanged. The joint-surface test varies several states while likewise holding their source content and the text trace fixed. For Bagel-Zebra-CoT we hold the VAE resize fixed and change only the ViT resize used when the state is returned to context. For DeltaV we do not truncate its autoregressive visual output: we re-encode that same state at the requested token count and rebuild

the interleaved visual placeholders and projected embeddings used by its routing path. We assert the actual token count, rather than trusting the requested configuration. All 8,712 Bagel-Zebra-CoT and 14,520 DeltaV forced-capacity writes passed this check.

Utility is the answer-span log-likelihood under teacher forcing, in nats. The scorer assigns the correct answer substantially higher likelihood than a length-matched random answer and has negligible repeated-forward variation (Table 6). We retain a visual-state cell only if removing that state decreases utility by more than ε_U at $k_{\max,\sigma}$, and remove an entire trajectory if its text-only trace already predicts the answer within ε_U of the full trace:

$$U_{x,t}(k_{\max,\sigma}) - U_{x,t}^{\text{none}} > \varepsilon_U, \qquad U_x^{\text{full}} - U_x^{\text{text}} > \varepsilon_U. \tag{11}$$

These filters define the visual-dependence population for the frozen-trajectory diagnosis; the deployment evaluations do not apply these additional filters to their test partitions. The retained trajectories have mean visual-state count 2.42, with 77.6% containing at least two states. Diagnostic intervals resample trajectories, not their constituent visual states.

Table 6: Capacity, scorer, and analysis-population checks. The dynamic range is the within-state range of answer utility across legal capacities.

Check	Bagel-Zebra-CoT	DeltaV
Forced-capacity writes passed	8,712 / 8,712	14,520 / 14,520
Correct vs. random answer gap (median nats)	$+11.7$	$+9.4$
Repeated-forward error / ε_U	0.28%	0.26%
ε_U (nats)	0.11	0.18
Dynamic range / ε_U (median)	$13.5\times$	$21.6\times$
Visual-dependence cell pass rate	82.4%	85.1%
Answer-leakage trajectory exclusion	16.8%	17.6%
Retained trajectories / visual cells	467 / 931	468 / 963
Full minus text-only utility (median nats)	3.41	4.08

C.2 Fidelity ranking and role controls

The fidelity family includes PSNR and SSIM, perceptual LPIPS, DINOv2 feature similarity, and each model's own visual representation metric. DeltaV also provides the TSIM value used to calibrate its deployed router. Pixel, perceptual, and semantic metrics compare the capacity-limited representation with the original ground-truth intermediate state; the model-native feature comparison uses that state's reference encoding. We orient every metric so larger values mean higher fidelity. Local token uncertainty is instead a utility-side signal, not a reconstruction measure. Fidelity itself rises with capacity across this family (within-state $\tau_{\mathrm{K}}(\phi, k) \geq 0.95$), while the utility ranking can reverse. The rank association with utility is $0.130/0.461$ for PSNR and $0.194/0.503$ for the model-native feature or deployed TSIM metric in Bagel-Zebra-CoT/DeltaV, respectively.

For a capacity pair (k, k'), let $\Delta U = U_{x,t}(k) - U_{x,t}(k')$ and $\Delta\phi^{(m)} = \phi_{x,t}^{(m)}(k) - \phi_{x,t}^{(m)}(k')$. We count an inversion when $\Delta U \Delta\phi^{(m)} < 0$, $|\Delta U| > \varepsilon_U$, and $|\Delta\phi^{(m)}| > \varepsilon_\phi^{(m)}$. The denominator is *all* capacity pairs, not only those exceeding both tolerances. We set ε_U to the 95th percentile of absolute utility changes under semantically equivalent answers or multiple-choice option permutations. Each $\varepsilon_\phi^{(m)}$ is the corresponding nuisance percentile after changing Bagel-Zebra-CoT's resampling filter or shifting DeltaV's center crop by one pixel before re-encoding. The fidelity–fidelity null compares reconstruction metrics, and the utility–utility null compares nuisance-equivalent answer scorers.

Figure 6 varies the utility tolerance while keeping the fidelity tolerance fixed. The mismatch remains well above both nulls in both models. The same encoding intervention applied to the question image yields only $6.1\%/5.2\%$ inversions; its utility has a clear capacity response, with a median range of $9.8/23.7$ times ε_U. Thus the low question-image inversion is not a floor effect. For Bagel-Zebra-CoT, replacing the default extrapolated position treatment with interpolation reduces intermediate-state inversion from 43.5% to 39.2%, but does not remove it. DeltaV's deployed TSIM criterion itself inverts 24.8% of capacity pairs under the same tolerance definition.

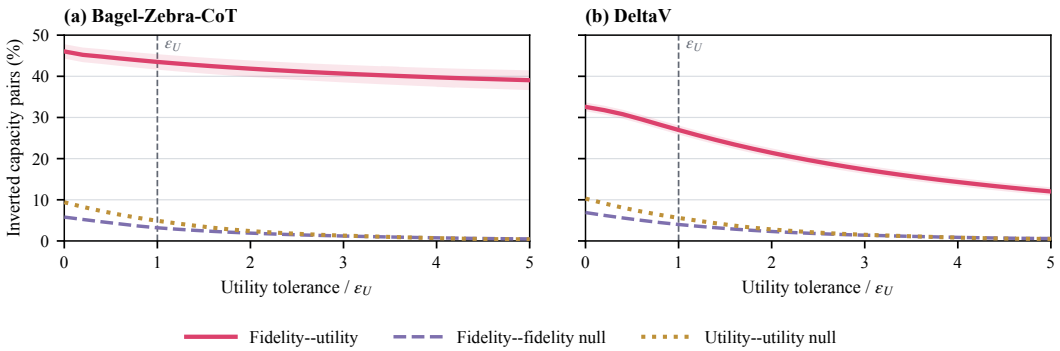


Figure 6: **The fidelity–utility mismatch survives stricter utility tolerances.** Curves use all capacity pairs as the denominator; shaded regions are 95% intervals for fidelity–utility inversion. The dashed vertical line marks the prespecified ε_U.

C.3 State dependence and cross-step structure

The utility-maximizing tier varies across states: the most common tier accounts for 31.2% of states in Bagel-Zebra-CoT and 17.6% in DeltaV. Visual-step index explains $14.2\%/8.1\%$ of the variance in $\log_2 k^\star_{x,t}$. The fixed-capacity, step-schedule, and fidelity-saturation rules are fitted on one partition and evaluated on held-out trajectories. The per-state ceiling uses a separate nuisance utility version to select each tier and another to evaluate it, avoiding an upward-biased maximum. Its $0.28/0.31$-nat headroom compares each rule at its own best validation setting; it is not a same-budget difference. Figure 7(a–b) gives the full matched-budget curves. The fidelity-saturation action has Kendall correlation only $0.19/0.15$ with the utility-optimal tier, so a fidelity threshold does not recover state-specific allocation.

To test cross-step separability, we fit the additive model in Eq. 3 on half the capacity vectors and evaluate on the rest. We enumerate the joint surface for two- and three-state trajectories; for longer trajectories we sample 200 vectors. The study uses 100, 60, and 60 trajectories in these three length groups per model. Adding the linear and quadratic total-capacity terms in Eq. 4 removes only 11–15% of the additive residual variance (Figure 7c). Four-point interactions also expose where this residual comes from: the median absolute interaction between adjacent visual steps is roughly twice that of non-adjacent steps in both models.

Finally, we compare the allocation selected from the measured joint surface with that selected by the additive fit at *identical raw total capacity*. Capacity choices use one nuisance version and are evaluated with another. For three-state trajectories the gap is largest at intermediate budgets and vanishes at the endpoints (Figure 7d). The aggregate gain also grows with trajectory length; for at least four states its reported $0.43/0.55$-nat values are sampling lower bounds rather than exact joint optima.

C.4 Free-rollout validation

To test whether a frozen ranking predicts actual decoding, we choose one intervention position in each of 60 held-out trajectories per model, covering early, middle, and last visual states. From the same causal prefix, the prespecified capacity alternatives are forced, and the model freely generates the remaining text and visual states with three paired rollout seeds. Frozen utility ranks both free-rollout correct-answer log-likelihood and answer accuracy positively (Table 7). Early-position correlations are also positive ($0.54/0.49$). Across rollout seeds, capacity rankings have pairwise Kendall correlation $0.82/0.78$.

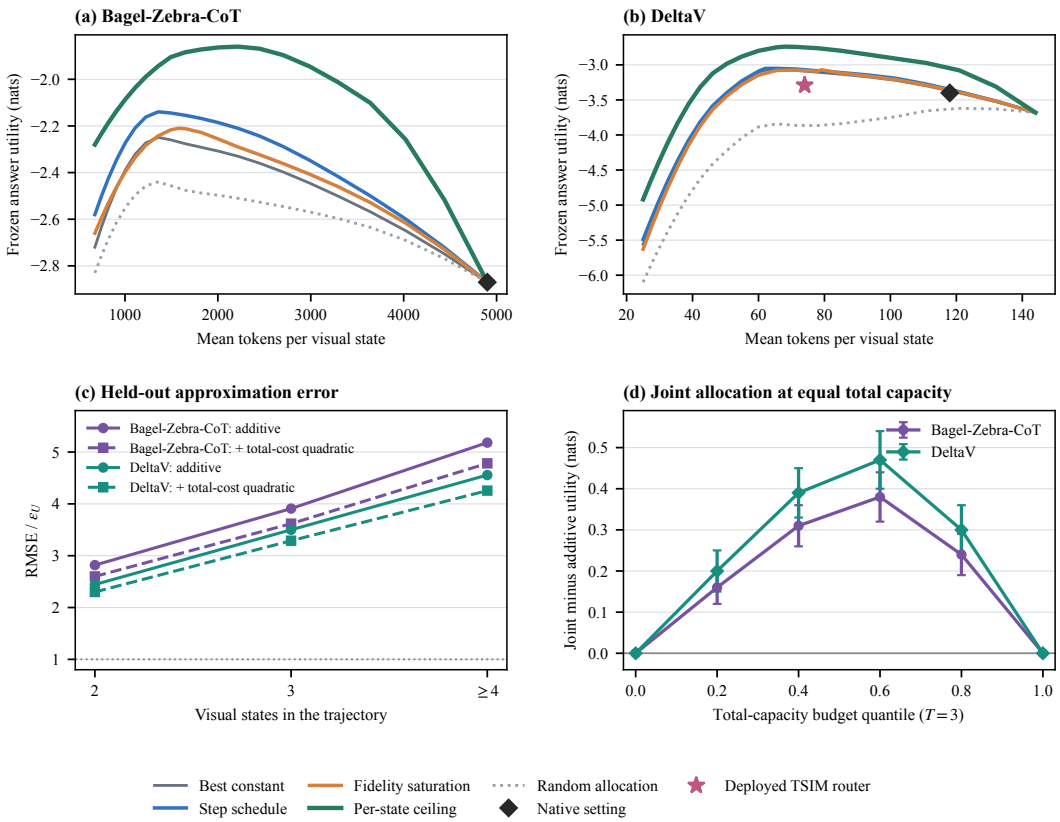


Figure 7: **Full capacity-rule curves and joint-allocation checks.** (a–b) Frozen utility versus mean visual-state capacity; diamonds mark the Bagel-Zebra-CoT published default and DeltaV's unrouted native setting, while the star marks DeltaV's deployed TSIM router. (c) Held-out error of additive and total-cost-corrected fits. (d) Joint-minus-additive utility at equal total capacity for three-state trajectories.

Table 7: Frozen rankings under free decoding and a separate matched-budget noncausal reference. Positive cells have positive within-cell capacity rank correlation. Accuracy uses target mean raw capacities 1,369 and 64 for Bagel-Zebra-CoT and DeltaV, respectively; the per-state ceiling uses correct-answer information and is not deployable. Brackets give 95% confidence intervals.

(a) Frozen ranking versus free decoding

Model	τ_K(rollout log-prob)	$\tau_{K,b}$(accuracy)	Positive cells
Bagel-Zebra-CoT	0.622 [0.510, 0.720]	0.458 [0.329, 0.571]	79%
DeltaV	0.578 [0.462, 0.681]	0.421 [0.289, 0.543]	74%

(b) Matched-budget free-rollout accuracy (%)

Model	Best constant	Fidelity saturation	Per-state ceiling
Bagel-Zebra-CoT	66.9 [55.0, 77.2]	67.8 [55.8, 77.7]	71.1 [59.4, 80.1]
DeltaV	57.2 [44.6, 68.7]	59.0 [43.9, 68.1]	62.3 [48.9, 72.1]

At matched visual capacity, the noncausal per-state ceiling improves free-rollout accuracy by 4.2/5.1 points over the best constant in Bagel-Zebra-CoT/DeltaV, validating the frozen ranking within the diagnosed visual-dependence population.

D External Calibration Details

D.1 Controlled supervision and hindsight-label checks

The three supervised controllers differ only in their capacity labels. Each uses the six causal scalars in Section 4, a one-layer GRU with 64 hidden units, and the same 32-unit prediction head. A

sigmoid output gives a normalized log-capacity position, which the adapter maps to the nearest legal tier; ties go to the lower tier. Each controller has 15,937 trainable parameters and is fitted with the same masked Smooth-L1 loss, AdamW settings, early-stopping rule, and three training seeds. The reasoning backbone and its visual and text generators remain frozen.

Training begins with 600 released-policy trajectories per model. After a first controller fit, we collect 600 trajectories from each of the three controllers and pool them with the released-policy trajectories. All three final controllers receive this same 2,400-trajectory pool, containing 4,872 valid visual decisions in Bagel-Zebra-CoT and 5,016 in DeltaV. We regenerate all three label sets on the common pool and train the final controllers under identical optimization settings, isolating the effect of capacity labels.

Fidelity target. Its candidate family is fixed before evaluation: LPIPS, DINOv2 or other semantic-feature similarity, and system-native reconstruction features. For DeltaV it also includes the TSIM quantity used by the deployed router. Each metric is scaled by its training-split robust range; we select and calibrate the family member using frozen-trajectory utility on validation data. Thus the Fidelity controller uses the most favorable validated reconstruction signal, while the deployed TSIM router remains a separate policy baseline. Neither test answers nor utility-side token uncertainty enter this selection.

Trajectory target. Per-Step Utility chooses each tier on a reference-capacity slice of the answer-length-normalized utility surface. Trajectory Utility instead searches the full allocation objective in Section 4. It starts from the per-step vector, the best constant vector, and a fixed-seed feasible vector. From each start, at most two full scans test adjacent-tier changes to one step and transfers or exchanges between pairs of steps, accepting only strict objective improvements. The best endpoint is retained; near-ties within the normalized utility tolerance choose lower cost, then a fixed lexicographic order. Measured joint utilities are reused across capacity prices.

Table 8 checks both search approximation and label stability. For two- and three-state trajectories the complete joint surface supplies an exact comparator. For longer trajectories, the reported three-start spread checks initialization sensitivity without treating the searched endpoint as an exact optimum. A nuisance-equivalent utility version selects each label and an independent version evaluates its regret. Causal suffix-invariance, legal-tier, actual-token-write, and common-training-manifest checks all passed.

Table 8: Hindsight Trajectory Utility label checks. The independent-utility rows are evaluated on a nuisance-equivalent scorer not used to select the labels; exact matching is measured per visual decision.

Check	Bagel-Zebra-CoT	DeltaV
$T = 2, 3$: ε_U-optimal search recovery	99.8%	99.4%
$T \geq 4$: median three-start spread / ε_U	0.15	0.20
Selection/evaluation exact label agreement	67.9%	58.7%
Independent-utility ε_U-optimal labels	88.7%	84.2%
Independent-utility gain retained	88%	84%

D.2 Complete accuracy–capacity comparison

We scan a frozen nonnegative price grid on validation data. Within the mean cost range reachable by every adjustable method, the four targets are $B(q) = B_{\text{low}} + q(B_{\text{high}} - B_{\text{low}})$ for $q \in \{0.2, 0.4, 0.6, 0.8\}$. The endpoints are 1,000/8,000 raw visual tokens per problem in Bagel-Zebra-CoT and 30/260 in DeltaV. Each method uses the validation-selected price or rule setting closest to each target, fixed before test rollout.

Figure 8 gives the full test-budget sweep. The same ordering holds in both models: Trajectory Utility leads Per-Step Utility, which leads Fidelity, while the frozen Training-Free Proxy also exceeds the rule-based baselines. The curves peak before the largest tested budget; their highest-budget points have lower test accuracy. At every plotted point, each method's realized mean capacity lies within 2% of its target.

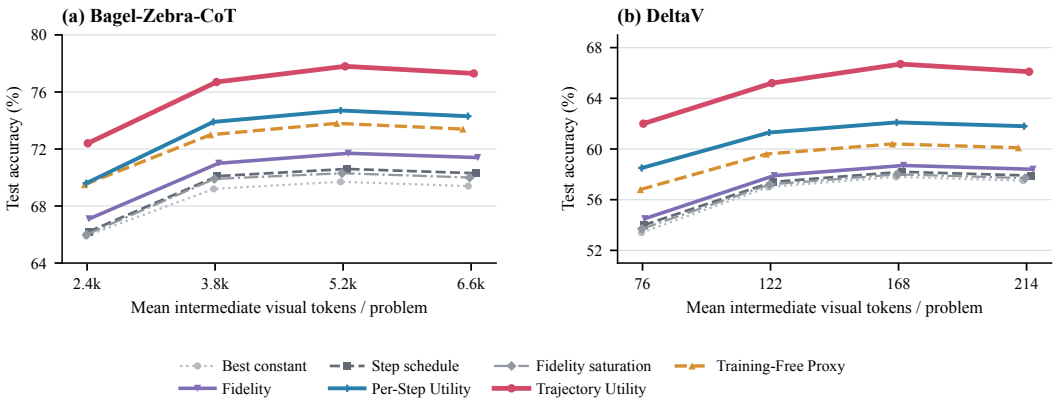


Figure 8: **Utility-calibrated external control improves the full accuracy–capacity curve.** Each point is accuracy on 800 held-out questions from the screened Zebra-CoT pool at the method's realized mean intermediate visual-token cost; lines connect the four evaluated budget levels.

The primary comparison uses B_{const}, the validation-frozen cost of the best constant rule, with target mean raw costs of 2,760 and 128 visual tokens per problem for Bagel-Zebra-CoT and DeltaV. The actual costs of all seven methods range from 2,728 to 2,788 and from 126.5 to 129.3, respectively. Table 9 gives absolute accuracies, paired differences, and the joint-supervision gain by pre-intervention trajectory length. Its intervals use the problem-cluster procedure in Appendix B. The three trained controllers use all nine training-seed $\times$ rollout-seed combinations at this budget; the rule-based baselines and frozen proxy use three rollout seeds.

Table 9: Full free-rollout test comparison at B_{const}. Brackets are 95% problem-cluster intervals; method differences are paired within problem. The successive Per-Step–Fidelity and Trajectory–Per-Step tests use Holm correction. Accuracy is in percent and gains in percentage points.

Method or comparison	Bagel-Zebra-CoT	DeltaV
(a) Test accuracy (%)		
Best constant	66.8 [63.9, 69.6]	57.1 [53.7, 60.4]
Step schedule	67.2 [64.3, 70.0]	57.5 [54.1, 60.8]
Fidelity saturation	67.0 [64.1, 69.8]	57.3 [53.9, 60.6]
Training-Free Proxy	70.4 [67.6, 73.1]	59.7 [56.3, 63.0]
Fidelity	68.1 [65.2, 70.9]	58.0 [54.6, 61.3]
Per-Step Utility	70.7 [67.9, 73.4]	61.4 [58.0, 64.7]
Trajectory Utility (VisCal-Ext)	**73.5** [70.8, 76.1]	**65.4** [62.0, 68.6]
(b) Paired accuracy gain (pp)		
Training-Free Proxy − Step schedule	+3.2 [2.2, 4.2]	+2.2 [1.1, 3.3]
Per-Step Utility − Fidelity	+2.6 [1.5, 3.7]	+3.4 [2.1, 4.7]
Trajectory Utility − Per-Step Utility	+2.8 [1.6, 4.0]	+4.0 [2.5, 5.5]
Trajectory Utility − Fidelity	+5.4 [4.1, 6.7]	+7.4 [5.8, 9.0]
Trajectory Utility − Step schedule	+6.3 [4.9, 7.7]	+7.9 [6.2, 9.6]
(c) Trajectory Utility − Per-Step Utility by reference length (pp)		
$T_{\text{ref}}^{(2)} = 1$	+0.3 [−0.8, 1.4]	+0.4 [−0.9, 1.7]
$T_{\text{ref}}^{(2)} \geq 2$	+3.7 [2.5, 4.9]	+5.2 [3.6, 6.8]
Multi-step minus single-step gain	+3.4 [1.8, 5.0]	+4.8 [2.8, 6.8]

Here $T_{\text{ref}}^{(2)}$ is fixed before intervention from one released-model rollout at its published capacity, not from the trajectory a controller later produces. The same validation-selected capacity price is used in both length groups. The full test set also retains 40/32 questions with no visual decision; these contribute to overall accuracy and cost but cannot contribute to a capacity-dependent gain. Across the four budget points, Trajectory Utility exceeds Per-Step Utility in normalized partial AUC by $+2.93\,[1.80, 4.10]$ and $+4.13\,[2.70, 5.60]$ points in the two models. Per-Step Utility beats Fidelity and Trajectory Utility beats Per-Step Utility in all nine matched training-seed $\times$ rollout-seed combinations in each model. Across methods, mean visual-state counts differ by at most 0.04; there are no out-of-memory runs, and other failures (0.17–0.33%) count as incorrect answers.

The released Bagel-Zebra-CoT policy obtains $64.6\,[61.5, 67.6]\%$ accuracy while spending 11,642.5 intermediate visual tokens per problem. DeltaV's deployed TSIM router obtains $57.2\,[53.7, 60.6]\%$ at 176.8 tokens. Against these natural operating policies, VisCal-Ext at B_{const} gains $8.9/8.2$ accuracy points while spending $76.2/27.3\%$ less visual capacity.

D.3 The frozen three-signal proxy

The Training-Free Proxy uses only the current step index, visual change, and text uncertainty. Its shallow multinomial classifier, feature basis, and capacity-class coefficients were learned during diagnostic development and then frozen. No classifier or controller parameter is refitted here; only the capacity price is chosen on validation data. If $s_\sigma(t, \delta_t, \nu_t; k)$ is the frozen class logit, its deployment action is

$$k_t^{\text{proxy}}(\lambda) = \arg\max_{k \in \mathcal{K}_\sigma} \big[s_\sigma(t, \delta_t, \nu_t; k) - \lambda c_\sigma(k) \big]. \tag{12}$$

Thus the proxy has neither the correct answer nor future states at inference.

To measure how much of the diagnostic opportunity these online signals recover, we use a separate set of 220 audit trajectories per model and an independent utility evaluation. Writing $\text{Util}_{\text{audit}}$ for mean utility under that scorer, we compute

$$\text{HR}_{\text{proxy}} = \frac{\text{Util}_{\text{audit}}(\pi_{\text{proxy}}) - \text{Util}_{\text{audit}}(\pi_{\text{base}})}{\text{Util}_{\text{audit}}(\pi_{\text{hindsight}}) - \text{Util}_{\text{audit}}(\pi_{\text{base}})}. \tag{13}$$

The hindsight policy in the denominator selects from measured joint utility with the correct answer; the denominator is positive at every measured budget. The non-utility reference π_{base} is selected on validation: Step schedule for Bagel-Zebra-CoT and Best constant for DeltaV. The resulting ratio measures frozen-utility headroom on audit trajectories; the test-accuracy comparisons use Step schedule as the strongest non-state-conditioned rule in both models.

Table 10: Fraction of independently evaluated hindsight utility headroom recovered by the frozen three-signal proxy. Brackets give 95% trajectory-bootstrap intervals.

Capacity budget	Bagel-Zebra-CoT	DeltaV
20% of sweep	$61\%\,[55, 67]$	$48\%\,[41, 55]$
40% of sweep	$68\%\,[62, 74]$	$55\%\,[48, 62]$
60% of sweep	$70\%\,[64, 76]$	$58\%\,[51, 65]$
80% of sweep	$60\%\,[53, 67]$	$49\%\,[42, 56]$
B_{const}	$67\%\,[61, 73]$	$56\%\,[49, 63]$

The frozen classifier contains 486/810 coefficients in Bagel-Zebra-CoT and DeltaV; its own decision costs 1,680/2,800 FLOPs. The three supervised controllers each have 15,937 trainable parameters and use 32,704 FLOPs per decision. Feature extraction is accounted for separately: the visual-change signal costs 4.624 billion FLOPs per decision in Bagel-Zebra-CoT and 262,144 in DeltaV, with 0.310/0.016 GiB incremental peak memory. None of these policies requires an additional reasoning-backbone forward pass. At B_{const}, VisCal-Ext's measured total FLOPs are $1.011\times$ /$1.008\times$ those of Step schedule at matched visual capacity.

E VisCal-OPD Details

E.1 Capacity actions and model interfaces

The capacity head reads the pre-generation hidden state and a price embedding, selects $k_t \in \mathcal{K}_\sigma$, and inserts a capacity token into the generative context before the first unit of V_t. The token conditions visual generation and subsequent text; the actuator enforces the selected native capacity. Table 11 specifies this interface for each model. The legal tiers are in Table 5; none includes a zero-capacity action.

Table 11: The shared capacity action is applied before visual generation, using each model's native capacity mechanism.

Model	Action position and actuator
Bagel-Zebra-CoT, ThinkMorph	Insert the capacity token before `gen_image`; map k_t to the corresponding image-generation canvas and set the `vit_transform` target to k_t re-consumed ViT tokens. Train the image generator at the selected canvas with its native flow-matching objective.
DeltaV	Commit k_t before the image-token block, end generation at that token count, and re-encode for consumption with `num_tokens`$= k_t$.
Monet	Insert the capacity token before the continuous-latent block and set the per-block latent limit to $k_t + 1$: its end marker occupies one position, leaving exactly k_t generated latents.

The action is learned jointly with the generative parameters; at inference, the trained capacity head uses the model-specific actuator as a fixed mapping to native execution. Capacity prices are sampled per training question from a frozen finite distribution with support Λ_{train}. At test time the chosen price conditions the same checkpoint, while the capacity head takes its argmax action.

E.2 Constructing branches and the hindsight teacher

At a prefix h_t visited by the current policy π_{old}, the candidate set $\mathcal{A}_t$ contains the realized one- or two-action block, adjacent-tier changes, approximately cost-preserving transfers or exchanges between two steps, and policy-sampled exploration. When discrete tiers admit no exactly equal-cost transfer, we use the feasible reallocation with closest total normalized cost. In the measured configuration, these four sources contribute one, three, two, and two candidate blocks per prefix, respectively.

For each distinct first action k, we force k and generate its visual state and intervening text only once, up to the next visual-decision prefix $h_{t+1}^{(k)}$. Blocks (k, k') then branch from this same prefix at their second action and independently continue to the final answer. If no second visual decision is reached, they collapse to the completed single-action branch (k). This shared tree lets second actions be compared under the same intervening content, while distinct first actions may change both that content and the number of later visual steps. Every later visual cost is included in $C_{t:\text{end}}$ in Equation 7; cost incurred before h_t is common to all branches and cancels from their comparison.

Write $\xi_{\kappa,n}$ for continuation n of block κ, with $n = 1, \ldots, N_\kappa$, and let Ξ_t collect all completed branches. The old capacity policy assigns a two-action block the chain probability

$$\pi_{\text{old}}(\kappa \mid h_t, \lambda) = \pi_{\text{old}}^{\text{cap}}(k \mid h_t, \lambda)\pi_{\text{old}}^{\text{cap}}(k' \mid h_{t+1}^{(k)}, \lambda), \qquad \kappa = (k, k'), \tag{14}$$

with only the first factor for $\kappa = (k)$. Restricting these probabilities to $\mathcal{A}_t$ gives $\bar{\pi}_{\text{old}}(\kappa)$; the branch prior is

$$p_{\text{old}}(\xi_{\kappa,n} \mid h_t, \lambda) = \frac{\bar{\pi}_{\text{old}}(\kappa \mid h_t, \lambda)}{N_\kappa}. \tag{15}$$

The division by N_κ keeps a block's prior mass independent of how many continuations it receives. On the finite, explicitly constructed set Ξ_t, the teacher solves

$$\max_{q \in \Delta(\Xi_t)} \left\{ \sum_{\xi \in \Xi_t} q(\xi) G_\lambda(\xi) - \eta D_{\text{KL}}(q \| p_{\text{old}}) \right\}. \tag{16}$$

Its normalization-constrained stationary condition gives $\log(q(\xi)/p_{\text{old}}(\xi)) = G_\lambda(\xi)/\eta - \log Z_t$, which is Equation 8. Thus the old-policy factor is part of the teacher prior, not an importance weight to be removed: candidate blocks are constructed and explicitly summed, not sampled from that prior. Training uses one complete continuation per block, $N_\kappa = 1$.

Capacity and native generation losses. The teacher's block marginal is $q^\star_{\text{cap},t}(\kappa) = \sum_n q^\star_t(\xi_{\kappa,n})$. Let $q^\star_{1,t}(k)$ be its first-action marginal and $\text{Supp}_{2|1,t}(k) = \{k' : (k, k') \in \mathcal{A}_t\}$. For a first action with at least two supported second actions, define $q^\star_{2,t}(k' \mid k)$ by normalizing the masses of blocks

(k, k'). The capacity objective is

$$\mathcal{L}_{\mathrm{cap}} = D_{\mathrm{KL}}\big(q^{\star}_{1,t} \big\| \pi^{\mathrm{cap}}_{\theta}(\cdot \mid h_t, \lambda)\big) + \sum_{k:\,|\,\mathrm{Supp}_{2|1,t}(k)|\geq 2} q^{\star}_{1,t}(k) D_{\mathrm{KL}}\Big(q^{\star}_{2,t}(\cdot \mid k) \Big\| \pi^{\mathrm{cap}}_{\theta}(\cdot \mid h^{(k)}_{t+1}, \lambda)\Big). \tag{17}$$

When only one second action is supported, its apparent certainty is an artifact of candidate construction. We therefore omit only that conditional term; the branch still contributes to the first-action marginal and to both generation losses.

The text and visual terms in Equation 9 are explicit $q^{\star}_t$-weighted sums over Ξ_t, with each branch's generated content as its target. Text uses token-normalized autoregressive negative log-likelihood. DeltaV uses visual-token likelihood, Bagel-Zebra-CoT and ThinkMorph use image flow matching, and Monet uses its continuous-latent objective. These losses are normalized by their visual generation units; image branches admit a negative-aware update when preferred and dispreferred candidates are paired. The capacity term is normalized by the number of supervised decisions, and the fixed weights in Equation 10 combine it with the two generation terms and retention replay. The teacher's access to the correct answer and complete branch outcomes is confined to training; neither is an inference input.

E.3 Training schedule and controlled variants

1. Initialize from each released checkpoint. The round-zero capacity distribution favors its native policy while retaining support for every legal tier.
2. For each training question, sample λ and freely roll out the current student. Select up to two visited visual-decision prefixes per question.
3. At each prefix, construct $\mathcal{A}_t$, expand the shared tree, complete every branch, and score it with Equation 7.
4. Form $q^{\star}_t$ and update the capacity head, visual generator, and text generator with Equation 10, including retention replay. Roll out the updated student again in the next round.

We run four rounds of 2,400 questions, with at most two decision prefixes per question and one continuation per candidate block. The price support consists of zero and 52 logarithmically spaced positive values in $[0.002, 3.2]$, selected on Zebra-CoT validation and frozen for the full runs. Round-zero exploration probability is 0.06. The fixed weights are $w_{\mathrm{ll}} = 0.5$, $\eta = 0.25$, $(w_{\mathrm{text}}, w_{\mathrm{vis}}, w_{\mathrm{ret}}) = (1, 0.5, 0.1)$; the retention replay ratio is 15%. Each model uses rank-64 LoRA, AdamW with learning rate 10^{-4}, and 2,400 optimizer steps. Each model–method pair has one training seed. The shared data, evaluation seeds, and statistical protocol are in Appendix B.

Matched post-training control. Standard OPD follows the same on-policy rollout, outcome weighting, native text and visual updates, and retention procedure but has neither capacity actions nor structured capacity branches. In the main controlled comparison, it inherits VisCal-OPD's validation-selected training pool, numbers of rounds, questions and prefixes, complete continuations per prefix, optimizer schedule, retention setting, and trainable generation parameters, without a separate hyperparameter search. Its capacity follows the released model's native policy. Its teacher uses a single price λ_{std}: the price at which VisCal-OPD's Zebra-CoT validation expenditure is closest to the released model's. The student does not read this price. The single-step ablation keeps the training budget but replaces two-step blocks with one-action candidates; the capacity-only ablation keeps the full teacher and branches while freezing text and visual generation parameters.

Standard OPD tuning check. For Bagel-Zebra-CoT and DeltaV, we independently search Standard OPD's learning rate in $\{5 \times 10^{-5}, 10^{-4}, 2 \times 10^{-4}\}$ and teacher temperature η in $\{0.125, 0.25, 0.5\}$, selecting by Zebra-CoT validation accuracy. All other training settings follow the main control. The selected $(\text{learning rate}, \eta)$ pairs are $(10^{-4}, 0.125)$ for Bagel-Zebra-CoT and $(5 \times 10^{-5}, 0.25)$ for DeltaV.

E.4 Implementation checks

For all four models, we verified that capacity tokens precede visual output, forced tiers equal consumed visual units, candidates share the pre-action context, suffixes are regenerated after interven-

tion, remaining cost is recovered exactly from logs, and both $\bar{\pi}_{\mathrm{old}}$ and $q_t^\star$ normalize. The second-action loss is active only when its conditional support contains at least two candidates. Monet's initial actuator yielded $k-1$ latents when its limit was set to k; after accounting for the end marker with a limit of $k+1$, all 2,400 forced-capacity checks passed. A short training round updated capacity, text, and visual parameters under VisCal-OPD, only capacity parameters under the capacity-only ablation, and no capacity parameters under Standard OPD. Validation expenditure decreased monotonically with price in each model.

F Extended Results

F.1 Full transfer comparisons

Tables 12 and 13 complement the main comparison table with Standard OPD's test accuracy and paired differences for every model–benchmark cell. Write Δ_{base}, Δ_{ext}, and Δ_{std} for VisCal-OPD minus the released model, VisCal-Ext, and Standard OPD, respectively. The first two differences are read at the released model's mean visual capacity B_{base}; the third is read at Standard OPD's own mean capacity B_{std}. Across the 32 cells, $B_{\mathrm{std}}/B_{\mathrm{base}}$ ranges from 0.996 to 1.016; VisCal-OPD is read directly, without interpolation, at all 32 Standard OPD comparison points. Intervals are paired by question after averaging three rollout seeds, as specified in Appendix B.

Table 12: Visual-state-centric transfer benchmarks. Standard OPD accuracy is in percent; VisCal-OPD gains are in percentage points with 95% intervals. Δ_{std} reads VisCal-OPD at B_{std}, while the main table shows it at B_{base}; differences use unrounded accuracies.

Benchmark	Std. OPD	Δ_{base}	Δ_{ext}	Δ_{std}
Bagel-Zebra-CoT				
MIRA	10.6	$+4.76\,[2.81, 6.65]$	$+2.75\,[0.67, 4.88]$	$+4.09\,[2.20, 5.98]$
ViC-Bench	39.9	$+8.00\,[5.93, 10.03]$	$+4.03\,[2.03, 6.00]$	$+5.93\,[4.00, 7.83]$
RBench-V	10.0	$+6.10\,[4.32, 7.93]$	$+3.07\,[1.41, 4.77]$	$+5.77\,[4.07, 7.47]$
MathCanvas	18.1	$+6.41\,[5.43, 7.38]$	$+3.52\,[2.58, 4.44]$	$+5.41\,[4.45, 6.37]$
DeltaV				
MIRA	11.8	$+3.60\,[1.59, 5.62]$	$+1.65\,[-0.18, 3.48]$	$+2.14\,[0.12, 4.21]$
ViC-Bench	26.3	$+9.03\,[7.17, 10.90]$	$+5.10\,[3.30, 6.87]$	$+8.53\,[6.67, 10.40]$
RBench-V	8.4	$+6.81\,[5.19, 8.47]$	$+2.70\,[1.12, 4.28]$	$+7.68\,[6.10, 9.30]$
MathCanvas	17.7	$+7.58\,[6.60, 8.57]$	$+3.92\,[2.94, 4.89]$	$+5.50\,[4.54, 6.46]$
Monet				
MIRA	9.7	$+4.64\,[2.75, 6.59]$	$+2.14\,[0.18, 4.15]$	$+4.46\,[2.81, 6.17]$
ViC-Bench	38.7	$+10.87\,[8.73, 13.03]$	$+6.57\,[4.70, 8.43]$	$+8.77\,[6.83, 10.73]$
RBench-V	12.0	$+7.18\,[5.48, 8.84]$	$+4.44\,[2.61, 6.27]$	$+7.06\,[5.40, 8.76]$
MathCanvas	23.4	$+6.70\,[5.66, 7.73]$	$+3.88\,[2.87, 4.89]$	$+5.13\,[4.11, 6.13]$
ThinkMorph				
MIRA	8.9	$+3.42\,[1.65, 5.19]$	$+1.83\,[0.06, 3.54]$	$+3.11\,[1.59, 4.64]$
ViC-Bench	42.4	$+6.93\,[4.97, 8.97]$	$+4.07\,[2.13, 6.00]$	$+5.47\,[3.50, 7.40]$
RBench-V	11.1	$+5.44\,[3.74, 7.14]$	$+2.86\,[1.08, 4.65]$	$+5.56\,[3.90, 7.26]$
MathCanvas	21.1	$+5.42\,[4.43, 6.44]$	$+2.49\,[1.50, 3.45]$	$+4.89\,[3.88, 5.87]$

Table 13: General multimodal transfer benchmarks, using the same comparison points and units as Table 12.

Benchmark	Std. OPD	Δ_{base}	Δ_{ext}	Δ_{std}
Bagel-Zebra-CoT				
MathVista	73.1	$+2.47\,[0.57, 4.33]$	$+3.00\,[1.20, 4.80]$	$+1.43\,[-0.40, 3.20]$
M^3CoT	64.5	$+6.61\,[5.32, 7.91]$	$+2.32\,[1.08, 3.52]$	$+4.72\,[3.54, 5.92]$
VisualPuzzles	31.0	$+3.17\,[1.34, 4.99]$	$+0.77\,[-1.03, 2.54]$	$+3.28\,[1.46, 5.08]$
VStar	63.4	$+1.57\,[-2.97, 6.11]$	$+1.92\,[-1.92, 5.76]$	$+3.14\,[-1.05, 7.50]$
DeltaV				
MathVista	70.4	$+1.53\,[-0.33, 3.43]$	$+1.10\,[-0.57, 2.83]$	$+0.17\,[-1.63, 2.00]$
M^3CoT	56.2	$+7.02\,[5.65, 8.38]$	$+5.10\,[3.85, 6.38]$	$+5.90\,[4.62, 7.18]$
VisualPuzzles	29.6	$+5.88\,[4.08, 7.68]$	$+3.60\,[1.83, 5.39]$	$+6.62\,[4.88, 8.30]$
VStar	75.4	$+0.35\,[-3.66, 4.36]$	$+0.17\,[-3.49, 3.66]$	$+1.92\,[-1.92, 5.58]$
Monet				
MathVista	61.3	$+2.20\,[0.23, 4.13]$	$+0.93\,[-1.03, 2.93]$	$+3.43\,[1.57, 5.30]$
M^3CoT	47.1	$+9.10\,[7.75, 10.44]$	$+6.11\,[4.83, 7.39]$	$+6.26\,[4.98, 7.56]$
VisualPuzzles	35.7	$+4.20\,[2.37, 5.99]$	$+1.86\,[0.09, 3.57]$	$+3.51\,[1.57, 5.45]$
VStar	79.6	$+1.22\,[-2.97, 5.58]$	$+1.92\,[-1.92, 5.76]$	$+0.70\,[-3.14, 4.71]$
ThinkMorph				
MathVista	68.1	$+2.07\,[0.17, 3.97]$	$+1.27\,[-0.60, 3.20]$	$+1.80\,[0.03, 3.60]$
M^3CoT	51.3	$+7.35\,[5.98, 8.66]$	$+4.56\,[3.28, 5.82]$	$+4.86\,[3.54, 6.17]$
VisualPuzzles	40.0	$+4.45\,[2.63, 6.28]$	$+2.94\,[1.20, 4.68]$	$+3.48\,[1.71, 5.31]$
VStar	68.1	$+2.27\,[-2.62, 6.98]$	$+3.66\,[-0.52, 7.85]$	$+1.22\,[-3.49, 5.76]$

All three differences have positive point estimates in all 32 cells. Their 95% intervals exclude zero in 27 cells for Δ_{base}, 23 for Δ_{ext}, and 26 for Δ_{std}. The smallest gain over the released model is $+0.35\,[-3.66, 4.36]$ points on DeltaV–VStar, which has 191 test questions.

Standard OPD tuning robustness. We also evaluate the validation-tuned Standard OPD checkpoints for Bagel-Zebra-CoT and DeltaV (search details in Appendix E). Their realized visual capacities range from 0.97 to 1.02 times the released-model capacity B_{base} across the 16 model–benchmark cells. VisCal-OPD's accuracy at each tuned checkpoint's actual budget is read directly, without interpolation. Independent tuning improves Standard OPD by 0.6 points on average; VisCal-OPD remains 3.9 points ahead at the matched budgets.

Table 14: Eight-benchmark mean transfer accuracy (%) under the two-model Standard OPD tuning check. The VisCal-OPD row uses tuned Standard OPD's realized visual budget.

Method	Bagel-Zebra-CoT	DeltaV	Two-model mean
Standard OPD (shared settings)	38.8	37.0	37.9
Standard OPD (tuned learning rate and η)	39.3	37.7	38.5
VisCal-OPD (at tuned Standard OPD budget)	43.0	41.8	42.4

F.2 Accuracy across the full capacity sweep

Table 15 reports all five evaluated budget coordinates. Each entry is the equal-weight mean accuracy difference from the released model across eight benchmarks, in percentage points. For model σ and benchmark e, the target budget is $B^{(3)}_{\sigma,e}(\gamma) = B^{\sigma,e}_{\text{low}} + \gamma(B^{\sigma,e}_{\text{base}} - B^{\sigma,e}_{\text{low}})$; $\gamma = 0$ denotes the minimum expenditure reached at the highest capacity price. At this endpoint both methods fall below the released models on average. Over $\gamma \in [0.25, 1]$, VisCal-OPD exceeds VisCal-Ext by $+2.84\,[2.64, 3.06]$ points in normalized area under the accuracy–capacity curve (nPAUC). Of the 320 method–cell–budget readings, 246 are direct and 74 are linearly interpolated between measured prices. Direct readings have a median absolute target-cost error of 0.73%.

Table 15: Accuracy gain over each released model (pp), averaged equally over eight benchmarks at each target budget coordinate.

Model	Method	$\gamma = 0$	0.25	0.50	0.75	1.00
Bagel-Zebra-CoT	VisCal-Ext	-5.61	-1.21	$+1.67$	$+2.11$	$+2.21$
	VisCal-OPD	-3.91	$+2.10$	$+4.29$	$+4.37$	$+4.89$
DeltaV	VisCal-Ext	-5.31	-0.77	$+1.74$	$+1.92$	$+2.31$
	VisCal-OPD	-3.10	$+2.50$	$+4.34$	$+5.29$	$+5.23$
Monet	VisCal-Ext	-5.36	-0.29	$+1.77$	$+2.16$	$+2.28$
	VisCal-OPD	-2.40	$+2.18$	$+4.90$	$+5.04$	$+5.76$
ThinkMorph	VisCal-Ext	-4.83	-1.32	$+1.38$	$+1.59$	$+1.71$
	VisCal-OPD	-3.79	$+1.89$	$+3.99$	$+4.11$	$+4.67$
All 32 cells	VisCal-Ext	-5.28	-0.90	$+1.64$	$+1.95$	$+2.13$
	VisCal-OPD	-3.30	$+2.17$	$+4.38$	$+4.70$	$+5.14$

F.3 Capacity at matched performance

The token or latent savings in Table 16 are estimated separately for every model–benchmark cell. Let adjacent points on VisCal-OPD's measured accuracy–*actual cost* curve be (C_i, A_i) and (C_{i+1}, A_{i+1}), with released-model accuracy A_{base} between them. We linearly interpolate the lowest-cost crossing and compare it with the released model's mean cost B_{base}:

$$\widehat{C}_{\times} = C_i + \frac{A_{\text{base}} - A_i}{A_{i+1} - A_i}(C_{i+1} - C_i), \qquad \widehat{s} = 1 - \frac{\widehat{C}_{\times}}{B_{\text{base}}}. \tag{18}$$

All 32 crossings lie between measured budget points: 26 between the $\gamma = 0$ and 0.25 readings, five between 0.25 and 0.50, and one between 0.50 and 0.75. The equal-cell mean estimated saving is 65.1% relative to the released models (range 24.9–80.5%). Using VisCal-Ext's accuracy at B_{base} as the target instead gives an estimated mean saving of 59.2% (range 23.6–78.8%).

Table 16: Estimated visual-token or latent savings (%) for VisCal-OPD at the released model's accuracy, using the per-cell interpolation in Eq. (18). Means weight the 32 cells equally.

	Visual-state-centric				General multimodal				
Model	MIRA	ViC-Bench	RBench-V	MathCanvas	MathVista	M^{3}CoT	VisualPuzzles	VStar	Mean
Bagel-Zebra-CoT	68.2	66.2	72.2	75.8	72.6	80.5	70.6	52.3	69.8
DeltaV	74.3	59.0	78.0	76.0	80.5	79.0	73.4	24.9	68.1
Monet	48.4	40.7	51.9	58.4	57.9	60.4	55.2	43.4	52.0
ThinkMorph	71.1	64.8	73.6	71.5	72.3	80.4	70.3	59.8	70.5
All cells									65.1

Local crossing audit. Because 26 of the 32 crossings lie between $\gamma = 0$ and 0.25, we measure three additional capacity prices near the predicted crossing in this interval, without retraining. We use the same two benchmarks for all four backbones: MathCanvas represents visual-state-centric reasoning and M^{3}CoT general multimodal reasoning. All eight original crossings lie in the interval being tested.

Table 17: Local interpolation audit. The direct point is the lowest-cost evaluated price whose accuracy point estimate reaches the original model; FLOPs and time ratios use complete rollouts.

		Crossing: capacity saved (%)		Direct point			
Model	Benchmark	Coarse	Dense	ΔAcc (pp)	Saved (%)	FLOPs ratio	Time ratio
Bagel-Zebra-CoT	MathCanvas	75.81	74.0	$+0.42$	70.9	$0.772\times$	$0.825\times$
	M^{3}CoT	80.51	81.7	$+0.29$	78.6	$0.746\times$	$0.802\times$
DeltaV	MathCanvas	76.00	78.1	$+0.36$	75.0	$0.785\times$	$0.842\times$
	M^{3}CoT	78.97	77.5	$+0.48$	74.8	$0.781\times$	$0.834\times$
Monet	MathCanvas	58.35	56.1	$+0.44$	52.8	$0.903\times$	$0.915\times$
	M^{3}CoT	60.43	62.0	$+0.31$	58.7	$0.876\times$	$0.899\times$
ThinkMorph	MathCanvas	71.47	73.4	$+0.37$	70.9	$0.765\times$	$0.817\times$
	M^{3}CoT	80.40	79.1	$+0.45$	76.5	$0.751\times$	$0.803\times$
Eight-cell mean	–	72.74	72.74	$+0.39$	69.78	$0.797\times$	$0.842\times$

Across these eight cells, the coarse and dense crossings both average 72.7% capacity savings; the mean absolute shift is 1.70 percentage points, with a maximum of 2.25. All eight directly evaluated points reach the original model's accuracy point estimate while using less visual capacity, total inference FLOPs, and wall-clock time.

F.4 Cross-step ablations and visual–text co-adaptation

The mechanism study uses 1,000 held-out Zebra-CoT questions per model, disjoint by question ID from post-training, with three paired rollout seeds. On this in-distribution test, VisCal-OPD gains $+14.08\,[12.60, 15.58]$ accuracy points over the released models. This sets the scale for the ablation differences below.

Table 18 gives the two controlled ablations at every evaluated budget. The single-step variant removes cross-step capacity reallocations from the teacher's candidate branches; the capacity-only variant keeps the full branches and teacher but freezes visual and text generation. Each comparison holds the target visual-capacity budget fixed. Both gaps remain positive over the four budget points in each model.

Table 18: VisCal-OPD accuracy gains over the indicated ablation (pp; paired 95% intervals). $\gamma = 1$ is the released model's mean capacity cost.

Model	$\gamma = 0.25$	0.50	0.75	1.00
Full minus single-step branching				
Bagel-Zebra-CoT	$+2.50\,[0.73, 4.33]$	$+3.33\,[1.60, 5.07]$	$+3.03\,[1.27, 4.80]$	$+3.27\,[1.50, 5.07]$
DeltaV	$+4.43\,[2.60, 6.30]$	$+5.87\,[4.03, 7.77]$	$+7.17\,[5.27, 9.03]$	$+5.57\,[3.67, 7.50]$
Two-model mean	$+3.47\,[2.17, 4.77]$	$+4.60\,[3.33, 5.88]$	$+5.10\,[3.80, 6.38]$	$+4.42\,[3.13, 5.70]$
Full minus capacity-only				
Bagel-Zebra-CoT	$+5.70\,[3.90, 7.57]$	$+5.43\,[3.63, 7.23]$	$+4.40\,[2.63, 6.20]$	$+5.10\,[3.33, 6.83]$
DeltaV	$+9.53\,[7.67, 11.43]$	$+8.90\,[7.03, 10.77]$	$+7.03\,[5.17, 8.93]$	$+6.37\,[4.57, 8.17]$
Two-model mean	$+7.62\,[6.32, 8.93]$	$+7.17\,[5.88, 8.45]$	$+5.72\,[4.40, 7.02]$	$+5.73\,[4.48, 6.95]$

For the cross-step test, T_{ref} is the number of visual states in the released model's pre-intervention rollout, shared across variants. The single-step gap is concentrated where at least two visual decisions occur (Table 19). The $T_{\mathrm{ref}} = 0$ questions—59 for Bagel-Zebra-CoT and 55 for DeltaV—remain in overall accuracy but have no visual decision to enter this stratified comparison.

Table 19: Full minus single-step branching by pre-intervention visual-state count (pp; paired 95% intervals). The one-state and multi-state groups contain 203/738 Bagel-Zebra-CoT and 206/739 DeltaV questions.

T_{ref} group	Bagel-Zebra-CoT	DeltaV	Two-model mean
$= 1$	$-0.49\,[-4.43, 3.44]$	$0.00\,[-4.05, 4.17]$	$-0.25\,[-3.07, 2.62]$
≥ 2	$+4.70\,[2.58, 6.82]$	$+7.49\,[5.27, 9.75]$	$+6.09\,[4.56, 7.63]$
Difference, ≥ 2 minus $= 1$	$+5.19\,[0.74, 9.62]$	$+7.49\,[2.89, 12.15]$	$+6.34\,[3.08, 9.53]$

The co-adaptation measurements use a fixed set of 500 questions per model with $T_{\mathrm{ref}} \geq 1$. At the same pre-action prefix and capacity, we compare a state from VisCal-OPD with one from the released generator; two independent released-model generations define the $1\times$ noise floor. Table 20 reports whole-set measurements. State changes under capacity-only training stay near that floor, while the larger changes under full VisCal-OPD usually increase downstream answer utility.

Table 20: Visual-state changes on the held-out mechanism set. Distance ratios divide median state distance by the median distance between independent released-model samples at the same prefix and capacity; brackets for ΔU_t are the 25th and 75th percentiles, not confidence intervals.

Model	Capacity-only	VisCal-OPD	Median ΔU_t (nats)	$\Delta U_t > 0$
Bagel-Zebra-CoT	$1.08\times$	$2.18\times$	$+0.328\,[-0.122, 0.808]$	68.3%
DeltaV	$1.01\times$	$2.60\times$	$+0.578\,[-0.008, 1.239]$	74.9%

We also measure the correct-answer log-likelihood advantage of retaining the intermediate visual states over removing them from the same model's rollout, $U_{\mathrm{full}} - U_{\mathrm{textonly}}$. Table 21 reports paired

changes in this quantity relative to the released model. The larger positive change under VisCal-OPD shows that later reasoning uses the learned visual states more, rather than bypassing them through text.

Table 21: Change in $U_{\text{full}} - U_{\text{textonly}}$ from the released model (nats; paired median and 95% interval) on 500 questions per model. Released-model medians are 2.51 and 3.10 nats, respectively.

Method	Bagel-Zebra-CoT	DeltaV
VisCal-Ext	$+0.15\,[0.08, 0.21]$	$-0.06\,[-0.11, 0.00]$
Capacity-only	$+0.27\,[0.19, 0.34]$	$+0.18\,[0.12, 0.25]$
VisCal-OPD	$+0.84\,[0.78, 0.93]$	$+1.14\,[1.08, 1.20]$

F.5 Inference Cost

At matched visual-capacity budgets, averaging across the eight transfer benchmarks, the ratio of VisCal-OPD to released-model total inference FLOPs is $1.006\times$ for Bagel-Zebra-CoT, $0.992\times$ for DeltaV, $1.130\times$ for Monet, and $1.018\times$ for ThinkMorph. The FLOPs accounting includes generation at the capacity-selected image canvas in Bagel-Zebra-CoT and ThinkMorph.

Table 22: Mean wall-clock time (s/problem) over eight transfer benchmarks. Original, VisCal-Ext, and VisCal-OPD use B_{ref}; Standard OPD uses B_{std}.

Model	Original	VisCal-Ext	Standard OPD	VisCal-OPD
Bagel-Zebra-CoT	24.6	25.2	24.5	24.9
DeltaV	6.2	6.4	6.1	6.2
Monet	9.6	10.1	9.4	9.8
ThinkMorph	14.8	15.0	14.9	14.7

At these budgets, VisCal-OPD improves accuracy with nearly unchanged inference FLOPs and wall-clock time.

At the per-cell accuracy crossings of Eq. (18), we apply the same interpolation weights to measured complete-rollout FLOPs and wall-clock time. Table 23 reports the resulting end-to-end cost at the released models' accuracy.

Table 23: Estimated end-to-end inference cost at released-model accuracy. Ratios compare VisCal-OPD with the original model; averages weight the 32 model–benchmark cells equally.

Model	Capacity saved (%)	FLOPs ratio	Time (s/problem)	Time ratio
Bagel-Zebra-CoT	69.8	$0.754\times$	$24.6 \to 19.9$	$0.809\times$
DeltaV	68.1	$0.787\times$	$6.2 \to 5.2$	$0.839\times$
Monet	52.0	$0.892\times$	$9.6 \to 8.7$	$0.906\times$
ThinkMorph	70.5	$0.762\times$	$14.8 \to 12.0$	$0.811\times$
All 32 cells	65.1	$0.799\times$	–	$0.841\times$

Across the 32 cells, this corresponds to 20.1% fewer total inference FLOPs and 15.9% shorter wall-clock time. The directly evaluated prices in Table 17 also reduce both quantities in all eight audited cells.