UniPath: Adaptive Coordination of Understanding and Generation for Unified Multimodal Reasoning

Hayes Bai[1] **Yinyi Luo**[2] **Wenwen Wang**[2] **Qingsong Wen**[3] **Jindong Wang**[1*]
[1]William & Mary [2]Carnegie Mellon University [3]Squirrel Ai Learning

Abstract

Unified multimodal models (UMMs) aim to integrate understanding and generation within a single architecture. However, it remains underexplored how to effectively coordinate these two capabilities for more effective and efficient reasoning. Existing coordination approaches either perform coupling during training, without explicit inference-time coordination, or impose a fixed coordination pattern for all inputs. In this work, we show that multimodal tasks exhibit substantial coordination-path diversity: different inputs favor different coordination paths. This suggests that exploiting such diversity is key to improving performance. We propose **UniPath**, a framework for adaptively modeling and exploiting coordination-path diversity. Instead of enforcing a single coordination pattern, we represent task solving as the selection and execution of a path, ranging from direct answering to textual inference, visual-thought construction, and hypothesis-based exploration. We construct role-aligned trajectories to train a path-conditioned executor and introduce a lightweight planner mechanism to enable input-dependent path selection. Experiments show that leveraging coordination-path diversity improves performance over fixed coordination strategies while providing interpretable intermediate behaviors. The code is available at: `https://github.com/AIFrontierLab/TorchUMM`.

1 Introduction

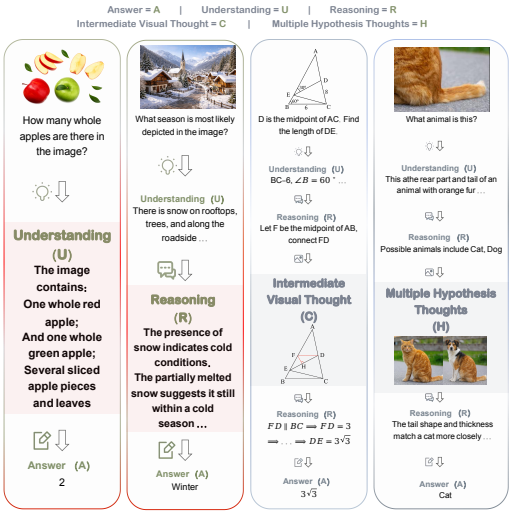
Unified multimodal models (UMMs) are a new family of models that can perform both understanding and generation tasks within a single architecture. Recent models have shown strong results on visual question answering and image generation [42, 39, 1, 44, 6, 31], suggesting that one single model can possess both capabilities. A natural next step is to move from capability coexistence toward effective *coordination:* understanding should provide useful evidence for generation, and generation-side visual signals should in turn support subsequent reasoning.

Coordination affects more than accuracy. If the model chooses a suitable reasoning path, it can use deeper multimodal steps only when useful, reduce unnecessary output tokens, and provide a readable explanation of why a particular solving strategy was used. Poor coordination has the opposite effect: simple paths may overlook problems that need intermediate reasoning, while forcing every input through a long coordination pattern wastes computation and is more prone to errors.

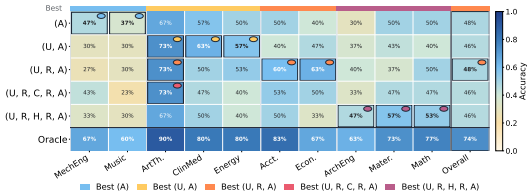
Existing work has explored coordination from different angles. Some methods promote coordination by coupling understanding and generation during training, such as self-play or reconstruction alignment [38, 50], improving consistency between perception and synthesis. However, they usually do not explicitly specify when and how coordination should occur at inference time, which limits how much learned cooperation can be exploited. Other methods introduce intermediate textual or visual representations [35], or use explicit coordination patterns such as analyzing-drafting loops and interleaved reasoning-generation traces [46, 18]. They make coordination more visible, but

*Corresponding author: `jdw@wm.edu`.

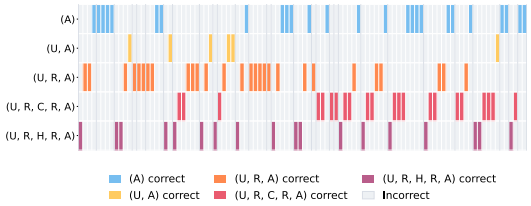
40th Conference on Neural Information Processing Systems (NeurIPS 2026).



(a) Different inputs favor different coordination paths.



(b) Subject-level path affinity on MMMU.



(c) Instance-level path correctness on MMMU.

Figure 1: **Coordination-path diversity in unified multimodal models.** Different coordination paths exhibit complementary strengths across inputs. The large oracle gap over fixed strategies suggests that exploiting coordination-path diversity can significantly improve UMM performance.

the protocol is usually fixed during training and inference that do not sufficiently account for the properties of different tasks and questions, making coordination less flexible than needed.

Do different inputs actually benefit from different coordination strategies? To answer it, we evaluate BAGEL [10] under several paths, including direct answering, explicit understanding, textual reasoning, visual-thought construction, and hypothesis exploration (formalized in §3.1). Figure 1 illustrates these paths with representative examples: simple perception questions may be answered after understanding alone, while others benefit from textual reasoning, visual-thought construction, or hypothesis exploration. We then examine whether such path differences translate into measurable performance variation on MMMU [57], a multidisciplinary benchmark spanning expert-level questions across diverse subjects. At the subject level, Figure 1b shows that no single path consistently dominates: different subjects favor different paths, and the best path varies across subjects. At the instance level, Figure 1c further shows that correctness varies sharply across paths, with many inputs being solved by only a subset of paths. The complete results for the two heatmaps are in Appendix A. The oracle results shown in Figure 1b provide direct evidence for the value of this diversity. By selecting the best path per input, the oracle substantially outperforms any fixed path, showing that coordination-path diversity is not redundant but can translate into large performance gains.

While it is promising to exploit the coordination-path diversity, turning this observation into a practical system raises three challenges. First, coordination categorization is needed: what kinds of contributions understanding and generation can make, and which forms are appropriate for different inputs? Without categorization, coordination tends to collapse into a single fixed pattern, ignoring task-specific needs and making costly cooperation less likely to yield matching gains. Second, even after categorization, we need training data and objectives that enable a single UMM to reliably execute different paths rather than merely follow their surface format. For visual roles, the intermediate state may be an abstract construction or a set of hypotheses, so supervision should not require every such step to become a complete image. Third, learning a path planner is a generalization problem under scarce supervision. Labels are expensive to obtain, domain biases vary significantly across datasets, and even with dataset-level knowledge, accurate instance-level path selection remains difficult.

In this paper, we propose **UniPath**, a planner-executor framework for adaptive coordination. We first abstract recurring operations in prior multimodal reasoning systems [15, 26, 35, 7, 8, 59, 46, 18, 11] into five functional roles: understanding, reasoning, construction, hypothesis, and answer. To keep the search space trainable, we define five representative coordination paths, each centered on one core role: answering directly, adding explicit understanding, adding textual reasoning, constructing a visual thought, or exploring hypotheses. We then train a path-conditioned executor on role-aligned trajectories so the same UMM can follow different reasoning paths. For visual roles, we use aligned

visual thought: the trace remains readable text, while the hidden states of visual-thought spans are supervised by visual summaries. Finally, a planner selects an input-dependent path, and a lightweight query-form calibration step combines learned path scores with simple structural priors.

Our contributions are threefold. (1) We formulate UMM reasoning as coordination-path selection and empirically show strong path diversity across subjects and instances. (2) We introduce a compact role/path space and train a path-conditioned executor with aligned visual thought, enabling one UMM to realize multiple coordination behaviors. (3) We build a planner-executor system that selects paths per input, improving accuracy with lower token cost while producing interpretable reasoning traces.

2 Related Work

Unified Multimodal Models. UMMs aim to integrate understanding and generation within a single architecture [56, 60]. Recent advances span a diverse set of designs, ranging from models that treat multimodal inputs as unified token sequences for next-token prediction [42, 39, 1], to approaches that incorporate diffusion or flow-based components for improved visual synthesis [51, 52, 44, 6, 31], as well as systems that explore different design choices to balance efficiency, scalability, and generation quality [54, 40, 47]. Despite their differences, these models share a common goal of capability unification, i.e., equipping a single model with multiple multimodal functionalities. However, multimodal reasoning is largely handled implicitly within the model, without explicit mechanisms to coordinate understanding and generation during inference. This often leads to inconsistencies between the two capabilities [30], revealing a gap between unified capabilities and structured reasoning. Recent work argues that structural unification alone does not ensure functional synergy [28]. FlexUMM [27] addresses this issue through adaptive allocation of shared and specialized computation. Our work studies a complementary level of adaptation: selecting the functional roles and reasoning path used to solve each input.

Coordinating Understanding and Generation. Role-based coordination and self-play have received increasing attention in multimodal learning [38, 16]. Some work couples the two processes during training such as self-play frameworks [38] and reconstruction alignment [50]. They improve global consistency between perception and synthesis, but do not specify how the two capabilities should be coordinated at inference time. LatentUMM [29] aligns cross-modal representations and transformations used for generation and re-encoding to improve consistency. These training approaches improve representations or learning signals, while our framework selects among coordination paths at inference time. Another direction extends chain-of-thought reasoning to multimodal settings, where intermediate visual representations may influence reasoning [35]. However, the coordination structure is still largely predetermined by the prompting or training format, making it difficult to adapt the amount and type of coordination to each input. More recent methods introduce explicit coordination mechanisms, such as iterative analyzing-drafting loops [46] and interleaving reasoning and generation for iterative refinement [18]. While they integrate generation into the reasoning process during inference, they rely on fixed coordination patterns and do not explicitly distinguish which functional roles are needed for different inputs. In contrast, we model understanding-generation coordination as path-based coordination: the system first selects a coordination path, then executes the corresponding role sequence. This shifts the focus from designing a single universal coordination protocol to adaptive exploitation.

3 Methodology

3.1 Problem Formulation

We denote the input of a UMM as $x = (q, \mathcal{I})$, where q is the textual question or instruction and $\mathcal{I}$ is the input image. Given x, the model can perform perceptual understanding and generative operations. While both capabilities are available, different inputs may benefit from different ways to organize them, raising a key challenge: how to represent multiple coordination patterns and select an appropriate one for each input? We address this by formulating understanding-generation coordination as path-based coordination. Instead of directly mapping x to an output y, we introduce a coordination path p that specifies a structured coordination strategy. Executing p produces intermediate states that lead to the final output y. This formulation avoids assuming a fixed coordination pattern and instead provides a unified interface for representing different strategies within a single model. Formally, for a

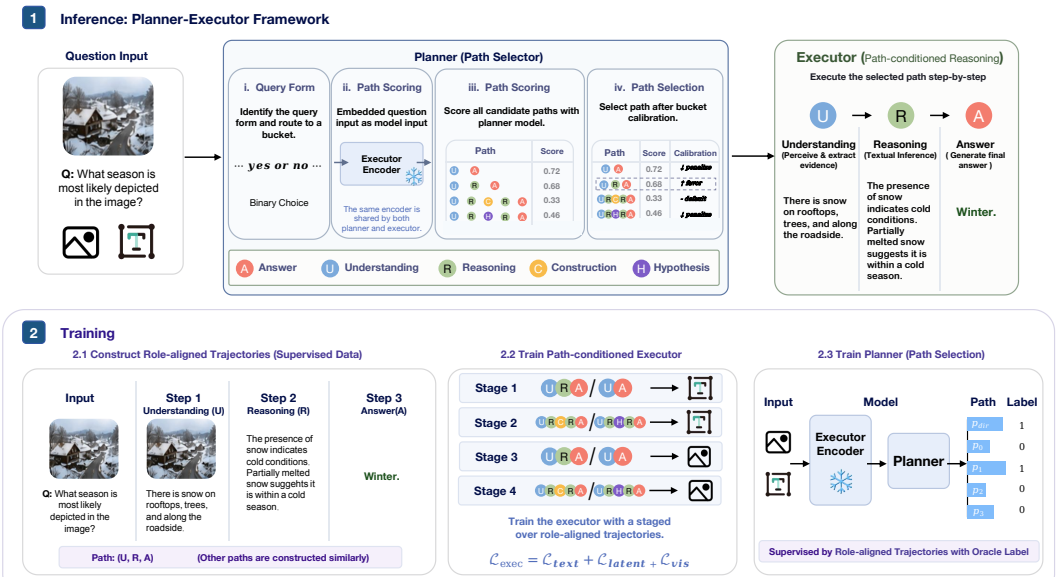


Figure 2: Overview of the training and inference process of UniPath.

path space $\mathcal{P}$, the planner is a path selector $\mathcal{G}_\psi$ that returns a single path before execution:

$$\hat{p} = \mathcal{G}_\psi(x) \in \mathcal{P}, \qquad y = \mathcal{E}_\theta(x, \hat{p}), \tag{1}$$

where $\hat{p}$ denotes the selected path and $\mathcal{E}_\theta$ is the executor that follows this path. The executor is the UMM itself after path-conditioned training: it receives both the original input and the selected path, then generates the corresponding trace and final output. The planner is a lightweight routing module that selects the path before the UMM executes it.

3.2 Coordination Categorization

We model understanding-generation coordination through a set of structured coordination paths. The key idea is to make coordination explicit at the level of *what role each step plays*, rather than treating a trajectory as an arbitrary sequence of tokens. This lets us compare, train, and select different ways of using understanding and generation during inference.

Functional roles. Different inputs require different uses of the capabilities. For example, one input mainly needs visual evidence understanding, while another may need comparison among possible visual hypotheses. We therefore categorize coordination by the functional role. The role design is motivated by recurring patterns in existing multimodal reasoning systems. Visual question answering emphasizes explicit understanding [15]. Multimodal chain-of-thought separates textual reasoning [26, 35]. Interleaved understanding-generation methods suggest that generation can serve roles such as intermediate construction or hypothesis exploration [46, 18, 11]. Abstracting these observations, we use five functional roles: understanding (U), reasoning (R), construction (C), hypothesis (H), and answer (A). U extracts observations from the input, R performs textual reasoning, and A produces the final answer. C and H are visual-thought roles: construction creates a visual thought for the next step, while hypothesis maintains candidate visual thoughts for comparison. This role set is not intended to be exhaustive. Instead, it provides a compact interface that captures common useful functions while remaining simple enough to train and support path selection.

Coordination path space. Given these roles, coordination can be viewed as selecting among different coordination paths. Enumerating every role sequence would create a large search space with weak supervision and many redundant variants. We instead define a compact set of representative paths. Each path is centered on one core role, with only the surrounding steps needed to make the path executable. This keeps the space small enough to train and evaluate while still covering qualitatively different coordination patterns: $\mathcal{P} = \{p_{\text{A}}, p_{\text{U}}, p_{\text{R}}, p_{\text{C}}, p_{\text{H}}\}$, where

$$\underbrace{p_{\text{A}} = (\mathbf{A})}_{\text{Direct Answering}}, \quad \underbrace{p_{\text{U}} = (\mathbf{U}, A)}_{\text{Explicit Understanding}}, \quad \underbrace{p_{\text{R}} = (U, \mathbf{R}, A)}_{\text{Textual Reasoning}}, \quad \underbrace{p_{\text{C}} = (U, R, \mathbf{C}, R, A)}_{\text{Visual-thought Construction}}, \quad \underbrace{p_{\text{H}} = (U, R, \mathbf{H}, R, A)}_{\text{Hypothesis exploration}}.$$

3.3 Planner-Executor Framework

We instantiate path-based coordination with a planner-executor framework. The planner implements $\mathcal{G}_\psi$, selecting a coordination path $\hat{p} \in \mathcal{P}$ conditioned on the input x. The executor $\mathcal{E}_\theta$ is the UMM that follows the selected path and returns the intermediate states and final output y.

Role-aligned trajectories. Training the executor aims to follow a selected path and make intermediate states useful for the next step. We therefore convert heterogeneous examples into **role-aligned trajectories**. Each trajectory contains the input x, a path label p, and segments arranged in the role order specified by p. We use tagged text to mark each role in the trace (e.g., `Understanding` for U). For paths with visual-thought roles, the tagged `Visual`/`Hypothesis` span remains readable text, while its hidden states are aligned to a visual summary. This provides a lightweight coordination channel that passes visual information to subsequent reasoning, while avoiding the high cost and inaccuracy of explicit image generation and the granularity mismatch introduced by raw visual latent insertion. Additionally, further analysis of aligned visual thoughts is provided in Appendix D, the prompt-level wrappers used at evaluation time are list in Appendix H, and representative trajectories are provided in Appendix J.

Executor training. Given a selected path, the executor must follow the role-tagged interface and make each intermediate state meaningful. A single mixed objective can make this difficult because path following, answer prediction, final image generation, and visual-thought alignment impose different signals. We therefore train the executor with a staged curriculum over role-aligned trajectories. The final run follows a four-stage LoRA chain: textual understanding, visual-thought understanding, plain image answering, and image answering with visual-thought supervision. Implementation details are in Appendix G. Specifically, each trajectory provides an input x, a path label p, and target text tokens $z = (z_1, \ldots, z_T)$ for the textual roles. We optimize a role-weighted language modeling loss:

$$\mathcal{L}_{\text{text}} = -\frac{1}{\sum_{t=1}^{T} w_t} \sum_{t=1}^{T} w_t \log \pi_\theta(z_t \mid x, p, z_{<t}). \tag{2}$$

Here, π_θ is the executor's token distribution and w_t are role-dependent token weights, allowing the same sequence interface to supervise understanding, reasoning, and answer tokens without requiring a separate objective for each role. For paths with construction or hypothesis roles, each `Visual`/`Hypothesis` segment is trained as an aligned visual thought. Let $\bar{h}_j$ denote pooled hidden representation over the j-th visual-thought span, and let v_j denote the visual summary embedded from the corresponding reference image. A lightweight projection head g_ϕ aligns the executor state to this target:

$$\mathcal{L}_{\text{vis}} = \frac{1}{J} \sum_{j=1}^{J} \left\| g_\phi(\bar{h}_j) - v_j \right\|_2^2. \tag{3}$$

For trajectories whose final answer is an image, we also keep BAGEL's final image-latent reconstruction loss $\mathcal{L}_{\text{latent}}$. The objective for the executor is

$$\mathcal{L}_{\text{exec}} = \lambda_{\text{text}} \mathcal{L}_{\text{text}} + \lambda_{\text{mse}} \mathcal{L}_{\text{latent}} + \lambda_{\text{vis}} \mathcal{L}_{\text{vis}}. \tag{4}$$

Terms that are not present in a trajectory, such as visual-thought supervision for $p_{\text{U}}/p_{\text{R}}$ or final image reconstruction for answer-only examples, are omitted. The λ coefficients balance text, final-image, and visual-thought supervision.

Planner supervision. The planner is trained to predict which paths lead to correct outcomes (detail in Sec. 4.1). For each input, this yields binary outcomes $r_p \in \{0, 1\}$ for paths $p \in \mathcal{P}$. The learned planner produces a path-wise score

$$a_\psi(x, p) = \sigma(f_\psi(x)_p), \tag{5}$$

which estimates the probability that path p will succeed on input x, with σ denoting the sigmoid function. Since multiple paths can solve the same input, we train the planner as a multi-label predictor rather than imposing a single best-path target. For a minibatch $\mathcal{B}$, the objective is a weighted binary cross-entropy with regularization:

$$\mathcal{L}_{\text{plan}} = \frac{1}{\sum_{i \in \mathcal{B}} \omega_i} \sum_{i \in \mathcal{B}} \omega_i \frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} \beta_{i,p} \, \text{BCEWithLogits}(f_\psi(x_i)_p, r_{i,p}) + \lambda_{\text{reg}} \mathcal{R}(\psi). \tag{6}$$

Here, ω_i is a sample weight and $\beta_{i,p}$ is a path-level label weight. In practice, samples with fewer successful paths receive larger weight because they provide sharper routing supervision, and positive labels outside p_{A} are mildly upweighted to reduce collapse to p_{A}. $\mathcal{R}(\psi)$ denotes standard planner regularization, implemented as weight decay on the planner parameters. This objective preserves the multi-path nature of the supervision. The final single path is chosen only after query-form calibration.

Query-form calibrated path selection. At inference time, directly selecting the path with the highest predicted score can be unstable, as the planner must generalize across dataset-specific domain biases under limited supervision. We therefore add a query-form prior based on surface structure. This prior does not replace the planner. Instead, it calibrates planner scores using cues that often correlate with the required coordination. For example, simple counting or binary-choice questions tend to favor simple paths, while geometry or chart reasoning may benefit from more structured coordination. Concretely, we introduce a lightweight calibration mechanism that adjusts path selection based on query form. Inputs are grouped into coarse query-form buckets using simple surface patterns rather than dataset identity. For each bucket, we apply temperature scaling and path-specific biases to the planner scores, and select a path only when its advantage over a default path exceeds a margin.

4 Experiments

4.1 Experimental Setup

Backbone and Training. We instantiate the executor with BAGEL [10] and train lightweight LoRA adapters [17, 32] for path execution. Evaluation is conducted with TorchUMM [30] for fair comparison. For executor training, we train BAGEL on path-aligned trajectories with supervision across the coordination paths in Sec. 3.1. Notably, our executor uses a comparatively *smaller* training set that shows our empirical gains come from exploiting the right form of understanding-generation coordination, not simply from using a larger post-training corpus. Executor training is organized into four staged splits that activate different links of the path. We report answer accuracy, format accuracy, CE, visual-thought alignment loss, and image-latent MSE where applicable, with staged training diagnostics provided in Appendix G.2. For planner training, the supervision is built after executor training by running all five candidate paths on roughly 8k calibration examples and recording which paths solve each query. More training details and results are in Appendix B, G, E.1, and F.

Planner calibration. We treat bucket construction as a calibration step rather than a fully hand-written procedure. Buckets and routing rules are derived from an auxiliary calibration pool, including the planner-construction split, the MMBench validation split, a subset of MathVerse [58]. GPT-5.5 is used for summarization, identifying recurring query-form patterns, which we consolidate into shared query-form buckets and path-based rules. At evaluation time, the planner relies only on the input query form and learned path scores.

Benchmarks. We evaluate understanding on MMMU [57], MMBench-EN/CN [24], MathVista [25], and MMStar [5], covering expert knowledge, cross-lingual visual QA, mathematical reasoning, and fine-grained visual reasoning. We evaluate generation on GenEval [14] and WISE [33], and evaluate understanding-generation consistency on UnifiedBench [53]. For generation and consistency benchmarks, the trained executor is run under the original query input, so these results show the effects of executor training and aligned visual-thought supervision rather than input-dependent routing. Detailed benchmark descriptions and scoring protocols are provided in Appendix C.

Baselines. We compare against unified multimodal generation/understanding models when reported under the same benchmark protocol, including BAGEL [10], BLIP3-o [3], Janus/Janus-Pro [44, 6], JanusFlow [31], Show-o/Show-o2 [51, 52], Emu3 [42], Emu3.5 [9], OmniGen2 [45], TokenFlow [13], DeepGen [40], MMaDA [54], and Ovis-U1 [41]. We also compare with BAGEL-based or unified-reasoning variants including RecA [50], UniGame [38], IRG [18], UniCoT [35] and AD-Loop [46].

4.2 Main Results

Table 1 summarizes the main comparison across understanding and generation. On understanding benchmarks, our method consistently improves over the BAGEL backbone across all datasets. For example, it achieves gains of +4.3% on MMMU and +4.4% on MMBench-EN, and shows a particularly strong improvement of +7.7% on MMStar. The improvement on MathVista is comparatively smaller (+0.8%), which we attribute to the relatively homogeneous problem types in this benchmark. In such

Table 1: Main results on understanding and generation benchmarks. Missing entries indicate unavailable results. Relative-gain rows report improvements over the corresponding BAGEL baseline, with AD-Loop computed from the values reported in its paper.

		Understanding						Generation	
Method	Params	MMMU	MMB-EN	MMB-CN	MathVista	MMStar	Avg.	GenEval	WISE
BAGEL	7B+7B	51.90	82.65	80.94	71.60	63.20	70.06	78.81	0.3989
Janus-Pro	7B	40.70	67.64	64.94	42.80	42.00	51.62	78.92	0.3811
Janus	1.3B	27.30	27.69	37.19	26.60	26.67	29.09	40.04	0.2222
JanusFlow	1.3B	29.00	64.16	60.31	34.80	39.13	45.48	49.99	0.2954
Show-o2	7B	47.90	77.25	76.68	51.50	54.53	61.57	59.87	0.3595
Show-o2	1.5B	37.10	65.09	61.04	37.90	43.87	49.00	55.49	0.3349
Show-o	1.3B	26.10	43.48	11.01	29.00	27.87	27.49	65.06	0.3037
Emu3	8B	31.40	59.53	45.66	44.90	43.60	45.02	45.76	0.3373
OmniGen2	3B+4B	46.00	76.36	75.43	63.50	52.47	62.75	78.53	0.4029
MMaDA	7B	28.90	28.16	20.57	24.90	31.73	26.85	46.12	**0.6560**
Ovis-U1	3B	43.70	78.55	76.99	68.50	58.27	65.20	**90.05**	0.3755
Emu3.5	34B	29.20	17.56	18.13	41.67	30.27	27.37	81.83	0.6331
BLIP3-o	4B	–	–	–	–	–	–	81.36	0.4138
TokenFlow	7B+14B	–	–	–	–	–	–	52.21	0.3056
DeepGen	3B+2B	–	–	–	–	–	–	86.59	0.5470
RecA	–	52.30	82.65	80.94	51.60	68.47	67.19	83.05	0.4225
UniGame	–	52.40	82.75	80.94	72.20	68.73	71.40	86.17	0.4032
IRG	–	48.00	60.47	57.35	68.00	61.53	59.07	72.06	0.3842
UniCoT	–	53.10	83.12	80.99	**73.00**	**70.00**	72.04	77.91	0.4037
Ours	–	**54.11**	**86.31**	**83.57**	72.20	68.07	**72.85**	80.00	0.4100
AD-Loop rel. gain		+3.6	+3.1	–	**+4.9**	+5.5	–	–	–
Ours rel. gain		**+4.3**	**+4.4**	**+3.2**	+0.8	**+7.7**	–	–	–

cases, path diversity provides limited additional benefit, and the similarity of inputs also makes it more challenging for the planner to reliably distinguish between suitable coordination paths. In contrast, datasets such as MMMU and MMStar contain more diverse inputs, where different coordination strategies can be more effectively exploited. Compared with BAGEL-based and unified-reasoning variants, our method achieves the best results on MMMU, MMBench-EN, and MMBench-CN, while remaining competitive on MathVista and MMStar. This supports our central claim that exploiting coordination-path diversity is more effective than enforcing a single fixed coordination pattern.

The two rightmost columns report generation benchmarks. Since generation uses the trained executor without planner routing, these results reflect executor training rather than adaptive path selection. Ours improves the BAGEL backbone from 78.81 to 80.00 on GenEval (+1.5% relative gain) and from 0.3989 to 0.4100 on WISE (+2.8% relative gain). Compared with other post-training methods, our approach achieves competitive or better performance on these benchmarks. The gains indicate that aligned visual-thought supervision can improve or preserve generation ability while supporting the understanding-side coordination paths. Complete category-level generation results of each benchmark are provided in Appendix F.4.

4.3 Understanding-Generation Consistency

To evaluate understanding-generation consistency, we use UnifiedBench [53]. This diagnostic complements the standalone understanding and generation benchmarks by testing whether executor training preserves information across a reconstruction loop. We compare against the BAGEL backbone and report both absolute scores and relative gains. Table 2 shows that our executor improves the BAGEL backbone from 0.8346 to 0.8380 overall. This positive improvement supports the intended effect of aligned visual-thought supervision, where image-derived supervision encourages textual visual thoughts to carry visual information that remains useful when the model later regenerates an image. Appendix D further analyzes this design by comparing it with explicit visual latent feedback and complete image feedback.

Table 2: Understanding-generation consistency on UnifiedBench. Higher similarities are better.

Method	CLIP	DINOv2	DINOv3	LongCLIP	Overall
BAGEL	0.8947	0.7877	0.7240	0.9321	0.8346
Ours	0.8958	0.7865	0.7338	0.9358	0.8380
Ours rel. gain	+0.1%	-0.2%	+1.4%	+0.4%	+0.4%

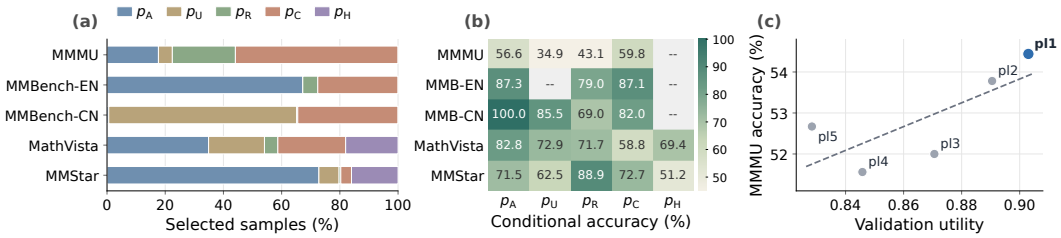


Figure 3: Planner behavior across benchmarks and validation transfer. (a) Selected path distribution. (b) Conditional accuracy on samples selected for each path. Missing entries indicate paths selected for zero samples. (c) Planner training validation utility versus routed MMMU accuracy for five planner checkpoints or configurations.

4.4 Planner Analysis

Planner behavior across benchmarks. Figure 3 reports three views of the planner. Panel (a) shows the fraction of questions assigned to each path across five understanding benchmarks. The selected path distribution varies across datasets, suggesting that the planner is not simply applying a fixed preference for deeper reasoning. The planner assigns most MMMU examples to p_{C}, consistent with its expert-level questions where intermediate visual-thought construction and structured reasoning are often useful. In contrast, MMBench-EN and MMStar are dominated by p_{A}, reflecting their larger fraction of recognition, commonsense, and option-matching questions where direct answering is usually sufficient. MathVista shows a more balanced path distribution. This may reflect the relatively homogeneous problem types in the benchmark, where coordination patterns are less distinct and harder to separate. Panel (b) reports conditional accuracy within the examples selected for each path, showing that non-dominant paths can still achieve reasonable accuracy on the subsets. This indicates that the planner does not collapse to a single dominant strategy, but instead distributes inputs across multiple paths. At the same time, the comparable accuracy across paths suggests that the path design is broadly well-aligned with different types of inputs. Panel (c) compares planner training validation utility against routed MMMU accuracy for five planner versions. Higher validation utility generally corresponds to better downstream routing, and the final planner achieves both the highest validation utility and the best MMMU accuracy.

Planner ablations. Table 3 ablates fixed paths, random selection, Model, Bucket, BAGEL path choice, our planner, and two reference settings. Model uses only the learned planner scores, while Bucket uses only query-form bucket rules to select a fixed path without learned scores. These subset results are intended to diagnose planner behavior rather than replace the full benchmark results in Table 1. The ablation highlights three points. First, model scores and query-form buckets are both useful, but neither is sufficient alone: raw scores can collapse toward a narrow preference, while bucket-only routing misses fine-grained distinctions. Although Model or Bucket can be competitive on individual datasets, our planner is better or tied on four of five benchmarks against each variant and obtains the best average among deployable planners. Second, dataset-adapted calibration can be stronger when the target benchmark distribution is known, but this sacrifices generalizability. Our planner gives up some target-specific headroom in exchange for a shared query-form calibration policy. Third, BAGEL's own path choice remains below our planner, indicating that path selection is not automatically solved by a capable UMM and requires explicit planner design.

Additionally, we analyze the planner from several perspectives and provide more discussion in the appendix. Appendix E.2 shows the path distributions of planner checkpoints, and Appendix E.3 visualizes the planner feature space, showing why domain shift and overlapping path labels make planner generalization difficult.

4.5 Token-Accuracy Tradeoff

Adaptive coordination improves accuracy without simply increasing the amount of generated reasoning. Figure 4 compares average output-token cost and accuracy against post-training reasoning methods. Across MMMU, MMBench-EN, MMBench-CN, MathVista, and MMStar, our method is consistently closer to the upper-left region. It uses substantially fewer output tokens while matching or improving accuracy on most benchmarks. This supports that the gain comes from invoking the

Table 3: Planner analysis and ablations on understanding benchmarks. For this ablation, MMMU is evaluated on the full set, while MMBench-EN, MMBench-CN, MathVista, and MMStar use 200-example subsets to keep the cost of evaluating a wide range of planner variants manageable.

Dataset	p_{A}	p_{U}	p_{R}	p_{C}	p_{H}	Rand.	Model	Bucket	BAGEL	Ours	Adapted	Oracle
MMMU	51.89	50.11	52.22	52.33	51.00	49.78	52.22	51.78	52.33	54.11	56.11	72.00
MMBench-EN	92.00	90.00	90.00	90.00	89.00	90.00	90.00	92.00	89.00	92.00	94.50	98.00
MMBench-CN	88.00	86.50	85.00	83.00	86.00	82.00	83.00	86.50	87.00	85.00	91.00	97.00
MathVista	67.50	65.50	67.50	72.00	67.50	66.50	69.00	66.50	67.00	68.50	77.50	88.00
MMStar	65.50	60.00	62.50	54.50	58.00	62.00	58.00	69.50	62.00	70.00	74.50	84.00
Average	72.98	70.42	71.44	70.37	70.30	70.06	70.44	73.26	71.47	73.99	78.72	87.80

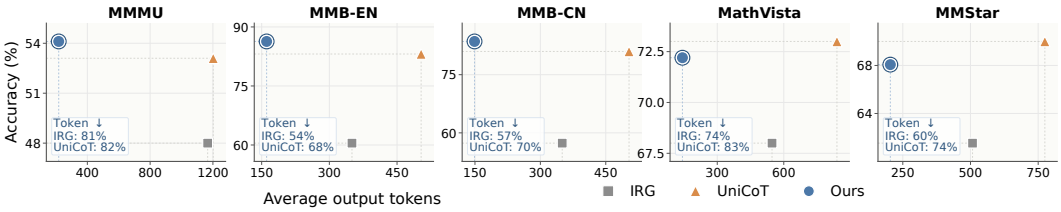


Figure 4: Accuracy versus average output-token cost on understanding benchmarks.

appropriate path and roles when useful, rather than forcing every query through a long reasoning trace. The blue annotations report the token reduction of our method relative to IRG and UniCoT. We omit AD-Loop from this cost comparison because the per-example outputs required for token accounting are unavailable.

4.6 Qualitative Analysis

We show example from the full five-path evaluation. Only the highlighted path gives the correct answer, making the role of path-specific execution visible at the question level. Additional examples for the other paths are provided in Appendix I.

p_{C}: visual-thought construction

QUESTION. In a bridge circuit with resistors 1Ω, 3Ω, 2Ω, and R, the ammeter reads zero. What is R? Options: A. 1.5Ω, B. 4Ω, C. 5Ω, D. 6Ω.

PREDICTIONS.

p_{A}	p_{U}	p_{R}	p_{C}	p_{H}	GT
A	C	C	**D**	B	**D**

TRACE.
Understanding: A circuit diagram shows a battery connected to four resistors in a diamond shape, with an ammeter in the middle.
Reasoning: The circuit is a Wheatstone bridge. For the ammeter to read zero, the resistance ratios must balance, so $1/3 = 2/R$.
Construction: A constructed bridge-state visual thought isolates the diamond bridge, places the ammeter at the center, and labels the four resistor arms.
Reasoning: Solving the balanced-arm relation gives $R = 6\Omega$.
Answer: D

WHY THIS PATH. The useful intermediate state is a structured visual relation over circuit arms. This is exactly where visual-thought construction helps: it turns a cluttered diagram into the relation needed for subsequent reasoning.

5 Conclusion and Limitations

We study how a unified multimodal model should decide whether, when, and how to coordinate understanding and generation. We introduce a compact path space, train an executor to follow these paths through a unified interface, and use a query-form calibrated planner to select a path for each input. Our experiments show that different examples favor different paths, and oracle path selection remains far above any fixed path, indicating strong complementarity among coordination patterns. At the same time, raw planner scores, query-form rules alone, and BAGEL's own path choice are insufficient, suggesting that path selection is itself a central modeling problem. Overall, the results support treating coordination policy as a first-class component of unified multimodal reasoning.

The main limitation is path selection: the deployable planner still leaves a large gap to oracle routing, and learning a planner that generalizes robustly across domains remains challenging. We provide a fuller discussion in Appendix J.

Acknowledgments and Disclosure of Funding

We thank William & Mary Research Computing for providing computational resources and/or technical support that have contributed to the results reported within this paper. URL: https://www.wm.edu/it/rc.

References

[1] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

[2] Jan Brejcha and Martin Čadík. Geopose3k: Mountain landscape dataset for camera pose estimation in outdoor environments. *Image and Vision Computing*, 66:1–14, 2017.

[3] Jiuhai Chen, Zhiyang Xu, Xichen Pan, Yushi Hu, Can Qin, Tom Goldstein, Lifu Huang, Tianyi Zhou, Saining Xie, Silvio Savarese, et al. Blip3-o: A family of fully open unified multimodal models-architecture, training and dataset. *arXiv preprint arXiv:2505.09568*, 2025.

[4] Junying Chen, Zhenyang Cai, Pengcheng Chen, Shunian Chen, Ke Ji, Xidong Wang, Yunjin Yang, and Benyou Wang. Sharegpt-4o-image: Aligning multimodal models with gpt-4o-level image generation. *arXiv preprint arXiv:2506.18095*, 2025.

[5] Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, et al. Mmstar: Are we on the right way for evaluating large visionlanguage models. *arXiv preprint arXiv:2403.20330*, 5, 2024.

[6] 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.

[7] Zihui Cheng, Qiguang Chen, Xiao Xu, Jiaqi Wang, Weiyun Wang, Hao Fei, Yidong Wang, Alex Jinpeng Wang, Zhi Chen, Wanxiang Che, et al. Visual thoughts: A unified perspective of understanding multimodal chain-of-thought. *arXiv preprint arXiv:2505.15510*, 2025.

[8] Zihui Cheng, Qiguang Chen, Jin Zhang, Hao Fei, Xiaocheng Feng, Wanxiang Che, Min Li, and Libo Qin. Comt: A novel benchmark for chain of multi-modal thought on large vision-language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 23678–23686, 2025.

[9] Yufeng Cui, Honghao Chen, Haoge Deng, Xu Huang, Xinghang Li, Jirong Liu, Yang Liu, Zhuoyan Luo, Jinsheng Wang, Wenxuan Wang, et al. Emu3. 5: Native multimodal models are world learners. *arXiv preprint arXiv:2510.26583*, 2025.

[10] Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, et al. Emerging properties in unified multimodal pretraining. *arXiv preprint arXiv:2505.14683*, 2025.

[11] Rongyao Fang, Chengqi Duan, Kun Wang, Linjiang Huang, Hao Li, Shilin Yan, Hao Tian, Xingyu Zeng, Rui Zhao, Jifeng Dai, et al. Got: Unleashing reasoning capability of multimodal large language model for visual generation and editing. *arXiv preprint arXiv:2503.10639*, 2025.

[12] Rongyao Fang, Aldrich Yu, Chengqi Duan, Linjiang Huang, Shuai Bai, Yuxuan Cai, Kun Wang, Si Liu, Xihui Liu, and Hongsheng Li. Flux-reason-6m & prism-bench: A million-scale text-to-image reasoning dataset and comprehensive benchmark. *arXiv preprint arXiv:2509.09680*, 2025.

[13] Michal Geyer, Omer Bar-Tal, Shai Bagon, and Tali Dekel. Tokenflow: Consistent diffusion features for consistent video editing. *arXiv preprint arXiv:2307.10373*, 2023.

[14] Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. *Advances in Neural Information Processing Systems*, 36:52132–52152, 2023.

[15] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6904–6913, 2017.

[16] Yicheng He, Chengsong Huang, Zongxia Li, Jiaxin Huang, and Yonghui Yang. VisPlay: Self-evolving vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26274–26284, 2026.

[17] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3, 2022.

[18] Wenxuan Huang, Shuang Chen, Zheyong Xie, Shaosheng Cao, Shixiang Tang, Yufan Shen, Qingyu Yin, Wenbo Hu, Xiaoman Wang, Yuntian Tang, et al. Interleaving reasoning for better text-to-image generation. *arXiv preprint arXiv:2509.06945*, 2025.

[19] Dongzhi Jiang, Ziyu Guo, Renrui Zhang, Zhuofan Zong, Hao Li, Le Zhuo, Shilin Yan, Pheng-Ann Heng, and Hongsheng Li. T2i-r1: Reinforcing image generation with collaborative semantic-level and token-level cot. *arXiv preprint arXiv:2505.00703*, 2025.

[20] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024.

[21] Chengzu Li, Wenshan Wu, Huanyu Zhang, Yan Xia, Shaoguang Mao, Li Dong, Ivan Vulić, and Furu Wei. Imagine while reasoning in space: Multimodal visualization-of-thought. *arXiv preprint arXiv:2501.07542*, 2025.

[22] Bin Lin, Zongjian Li, Xinhua Cheng, Yuwei Niu, Yang Ye, Xianyi He, Shenghai Yuan, Wangbo Yu, Shaodong Wang, Yunyang Ge, et al. Uniworld-v1: High-resolution semantic encoders for unified visual understanding and generation. *arXiv preprint arXiv:2506.03147*, 2025.

[23] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

[24] Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? In *European conference on computer vision*, pages 216–233. Springer, 2024.

[25] 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. *arXiv preprint arXiv:2310.02255*, 2023.

[26] Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in neural information processing systems*, 35:2507–2521, 2022.

[27] Yinyi Luo, Haoyue Bai, Yi Ge, Nilton Cesar Rojas Vales, Marios Savvides, and Jindong Wang. FlexUMM: Flexible representation learning for unified multimodal models. *arXiv preprint arXiv:2610.22682*, 2026.

[28] Yinyi Luo, Hayes Bai, Yiqiao Jin, Wenyue Hua, Srijan Kumar, Marios Savvides, Qingsong Wen, and Jindong Wang. Beyond structural unification: A perspective and roadmap on unified multimodal models. *arXiv preprint arXiv:2610.35906*, 2026.

[29] Yinyi Luo, Wenwen Wang, Hayes Bai, Marios Savvides, and Jindong Wang. LatentUMM: Dual latent alignment for unified multimodal models. *arXiv preprint arXiv:2605.17766*, 2026.

[30] Yinyi Luo, Wenwen Wang, Hayes Bai, Hongyu Zhu, Hao Chen, Pan He, Marios Savvides, Sharon Li, and Jindong Wang. Torchumm: A unified multimodal model codebase for evaluation, analysis, and post-training. *arXiv preprint arXiv:2604.10784*, 2026.

[31] Yiyang Ma, Xingchao Liu, Xiaokang Chen, Wen Liu, Chengyue Wu, Zhiyu Wu, Zizheng Pan, Zhenda Xie, Haowei Zhang, Xingkai Yu, et al. Janusflow: Harmonizing autoregression and rectified flow for unified multimodal understanding and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7739–7751, 2025.

[32] Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, Benjamin Bossan, and Marian Tietz. PEFT: State-of-the-art parameter-efficient fine-tuning methods. `https://github.com/huggingface/peft`, 2022.

[33] Yuwei Niu, Munan Ning, Mengren Zheng, Weiyang Jin, Bin Lin, Peng Jin, Jiaqi Liao, Chaoran Feng, Kunpeng Ning, Bin Zhu, et al. Wise: A world knowledge-informed semantic evaluation for text-to-image generation. *arXiv preprint arXiv:2503.07265*, 2025.

[34] Brandon Ong, Tej Deep Pala, Vernon Toh, William Chandra Tjhi, and Soujanya Poria. Training vision-language process reward models for test-time scaling in multimodal reasoning: Key insights and lessons learned. *arXiv preprint arXiv:2509.23250*, 2025.

[35] Luozheng Qin, Jia Gong, Yuqing Sun, Tianjiao Li, Mengping Yang, Xiaomeng Yang, Chao Qu, Zhiyu Tan, and Hao Li. Uni-cot: Towards unified chain-of-thought reasoning across text and vision. *arXiv preprint arXiv:2508.05606*, 2025.

[36] Hao Shao, Shengju Qian, Han Xiao, Guanglu Song, Zhuofan Zong, Letian Wang, Yu Liu, and Hongsheng Li. Visual cot: Advancing multi-modal language models with a comprehensive dataset and benchmark for chain-of-thought reasoning. *Advances in Neural Information Processing Systems*, 37:8612–8642, 2024.

[37] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565, 2018.

[38] Zhaolong Su, Wang Lu, Hao Chen, Sharon Li, and Jindong Wang. Unigame: Turning a unified multimodal model into its own adversary. *arXiv preprint arXiv:2511.19413*, 2025.

[39] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[40] Dianyi Wang, Ruihang Li, Feng Han, Chaofan Ma, Wei Song, Siyuan Wang, Yibin Wang, Yi Xin, Hongjian Liu, Zhixiong Zhang, et al. Deepgen 1.0: A lightweight unified multimodal model for advancing image generation and editing. *arXiv preprint arXiv:2602.12205*, 2026.

[41] Guo-Hua Wang, Shanshan Zhao, Xinjie Zhang, Liangfu Cao, Pengxin Zhan, Lunhao Duan, Shiyin Lu, Minghao Fu, Xiaohao Chen, Jianshan Zhao, et al. Ovis-u1 technical report. *arXiv preprint arXiv:2506.23044*, 2025.

[42] Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Zhen Li, Qiying Yu, et al. Emu3: Next-token prediction is all you need. *arXiv preprint arXiv:2409.18869*, 2024.

[43] Navve Wasserman, Noam Rotstein, Roy Ganz, and Ron Kimmel. Paint by inpaint: Learning to add image objects by removing them first. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 18313–18324, 2025.

[44] Chengyue Wu, Xiaokang Chen, Zhiyu Wu, Yiyang Ma, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, Chong Ruan, et al. Janus: Decoupling visual encoding for unified multimodal understanding and generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 12966–12977, 2025.

[45] Chenyuan Wu, Pengfei Zheng, Ruiran Yan, Shitao Xiao, Xin Luo, Yueze Wang, Wanli Li, Xiyan Jiang, Yexin Liu, Junjie Zhou, et al. Omnigen2: Exploration to advanced multimodal generation. *arXiv preprint arXiv:2506.18871*, 2025.

[46] Shengqiong Wu, Bobo Li, Xinkai Wang, Xiangtai Li, Lei Cui, Furu Wei, Shuicheng Yan, Hao Fei, and Tat-seng Chua. Synergizing understanding and generation with interleaved analyzing-drafting thinking. *arXiv preprint arXiv:2602.21435*, 2026.

[47] Size Wu, Zhonghua Wu, Zerui Gong, Qingyi Tao, Sheng Jin, Qinyue Li, Wei Li, and Chen Change Loy. Openuni: A simple baseline for unified multimodal understanding and generation. *arXiv preprint arXiv:2505.23661*, 2025.

[48] Size Wu, Wenwei Zhang, Lumin Xu, Sheng Jin, Zhonghua Wu, Qingyi Tao, Wentao Liu, Wei Li, and Chen Change Loy. Harmonizing visual representations for unified multimodal understanding and generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17739–17750, 2025.

[49] Shitao Xiao, Yueze Wang, Junjie Zhou, Huaying Yuan, Xingrun Xing, Ruiran Yan, Chaofan Li, Shuting Wang, Tiejun Huang, and Zheng Liu. Omnigen: Unified image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13294–13304, 2025.

[50] Ji Xie, Trevor Darrell, Luke Zettlemoyer, and XuDong Wang. Reconstruction alignment improves unified multimodal models. *arXiv preprint arXiv:2509.07295*, 2025.

[51] Jinheng Xie, Weijia Mao, Zechen Bai, David Junhao Zhang, Weihao Wang, Kevin Qinghong Lin, Yuchao Gu, Zhijie Chen, Zhenheng Yang, and Mike Zheng Shou. Show-o: One single transformer to unify multimodal understanding and generation. *arXiv preprint arXiv:2408.12528*, 2024.

[52] Jinheng Xie, Zhenheng Yang, and Mike Zheng Shou. Show-o2: Improved native unified multimodal models. *arXiv preprint arXiv:2506.15564*, 2025.

[53] Zhiyuan Yan, Kaiqing Lin, Zongjian Li, Junyan Ye, Hui Han, Zhendong Wang, Hao Liu, Bin Lin, Hao Li, Xue Xu, et al. Can understanding and generation truly benefit together–or just coexist? *arXiv e-prints*, pages arXiv–2509, 2025.

[54] Ling Yang, Ye Tian, Bowen Li, Xinchen Zhang, Ke Shen, Yunhai Tong, and Mengdi Wang. Mmada: Multimodal large diffusion language models. *arXiv preprint arXiv:2505.15809*, 2025.

[55] Junyan Ye, Dongzhi Jiang, Zihao Wang, Leqi Zhu, Zhenghao Hu, Zilong Huang, Jun He, Zhiyuan Yan, Jinghua Yu, Hongsheng Li, et al. Echo-4o: Harnessing the power of gpt-4o synthetic images for improved image generation. *arXiv preprint arXiv:2508.09987*, 2025.

[56] Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. A survey on multimodal large language models. *National Science Review*, 11(12):nwae403, 2024.

[57] Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9556–9567, 2024.

[58] Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu, Kai-Wei Chang, Yu Qiao, et al. Mathverse: Does your multi-modal llm truly see the diagrams in visual math problems? In *European Conference on Computer Vision*, pages 169–186. Springer, 2024.

[59] Xintong Zhang, Zhi Gao, Bofei Zhang, Pengxiang Li, Xiaowen Zhang, Yang Liu, Tao Yuan, Yuwei Wu, Yunde Jia, Song-Chun Zhu, et al. Adaptive chain-of-focus reasoning via dynamic visual search and zooming for efficient vlms. *arXiv preprint arXiv:2505.15436*, 2025.

[60] Shanshan Zhao, Xinjie Zhang, Jintao Guo, Jiakui Hu, Lunhao Duan, Minghao Fu, Yong Xien Chng, Guo-Hua Wang, Qing-Guo Chen, Zhao Xu, et al. Unified multimodal understanding and generation models: Advances, challenges, and opportunities. *arXiv preprint arXiv:2505.02567*, 2025.

[61] Kai Zou, Zhiwei Zhao, Bin Liu, and Nenghai Yu. Advancing aesthetic image generation via composition transfer. *International Journal of Computer Vision*, 134(5):252, 2026.

Appendix

Contents

A Additional Path-Diversity Visualizations **14**
B Training Data Construction **15**
C Benchmark and Metric Details **15**
D Aligned Visual Thought Analysis **16**
E More Details and Analysis on Planner **16**
E.1 Planner Training Details 16
E.2 Planner Training Transfer 17
E.3 Planner Feature-Space Analysis 17
F More Experimental Results **18**
F.1 Relative Gains of Post-training Methods 18
F.2 Path Format Compliance on MMMU 18
F.3 Different Backbone Experiments 19
F.4 Additional Generation Results 19
G Executor Training Details **20**
G.1 Executor Data Statistics 21
G.2 Executor Training Diagnostics 22
H Path Prompt Templates **23**
I Additional Qualitative Examples **25**
J Role-Aligned Trajectory Examples **25**
K Limitations **27**
L Broader Impact **27**

A Additional Path-Diversity Visualizations

Figure 5 and Figure 6 provide the full-resolution versions of the compact MMMU visualizations in Figure 1. The row labels use the role-sequence notation from Sec. 3.1: (A) corresponds to p_{A}, (U, A) to p_{U}, (U, R, A) to p_{R}, (U, R, C, R, A) to p_{C}, and (U, R, H, R, A) to p_{H}.

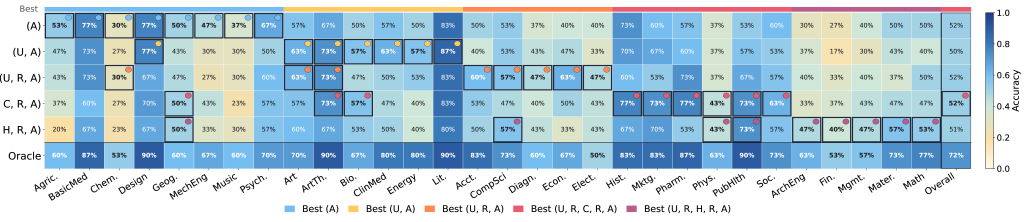


Figure 5: **Full subject-level path affinity on MMMU.** Each column corresponds to an MMMU subject, and each row reports the accuracy of one coordination path. The overall pattern shows that subject domains favor different paths, while the oracle row remains substantially higher than any fixed path, supporting the need for input-dependent path selection.

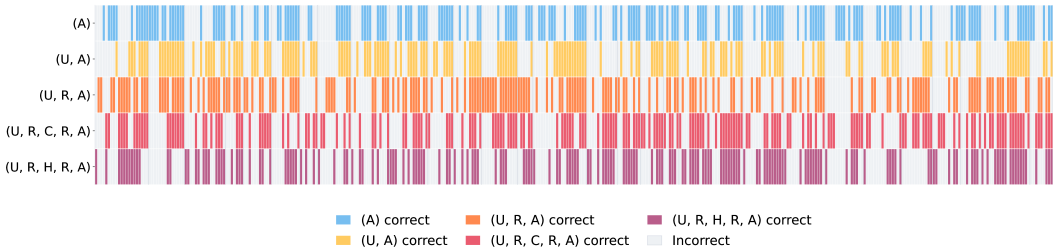


Figure 6: **Full instance-level path correctness on MMMU.** Each column is an MMMU example and each row is a coordination path. Colored cells indicate that the corresponding path answers the example correctly, while gray cells indicate failure. The sparse and non-identical correctness patterns show that path complementarity also appears at the individual-question level, not only after aggregating by subject.

B Training Data Construction

We construct path-aligned trajectories from both understanding and generation sources, following the path space in Sec. 3.1. For $p_{\mathrm{U}} = (U, A)$, we use VQAv2 [15] to form perception-only trajectories. For $p_{\mathrm{R}} = (U, R, A)$, we use ScienceQA [26] for understanding and LAION-Aesthetics-High-Resolution-GoT [11] for generation. For ScienceQA, we combine questions with answer choices, map the answer index to its option text, and use lecture and solution fields as textual reasoning. For GoT generation data, the prompt defines the generation target and the GoT annotation provides the reasoning trace.

For $p_{\mathrm{C}} = (U, R, C, R, A)$, we construct examples that require intermediate visual-thought construction. On the understanding side, we process CoMT [8] and CoF-SFT [59], using provided or zoomed visual references as construction targets. On the generation side, we use FLUX-Reason-6M [12], where the first reasoning step specifies partial visual elements, a reference image is synthesized, and the second reasoning step completes the remaining elements. For $p_{\mathrm{H}} = (U, R, H, R, A)$, we use GoT generation data and VL-PRM300K [34] to create examples with multiple candidate visual hypotheses followed by comparative reasoning. We sample approximately 5k examples for each nontrivial level and source group. When native intermediate annotations are unavailable, we use GPT-5.4 and Claude Code Opus 4.6 as text teachers to produce role-aligned textual segments, and use BAGEL to synthesize reference images needed for aligned visual-thought supervision. After executor training, planner supervision is constructed by running all five candidate paths on roughly 8k calibration examples and recording which paths solve each query. As summarized in Table 4, our executor uses a comparatively small training set. Our empirical gains therefore come from exploiting the right form of understanding-generation coordination, not simply from using a larger post-training corpus.

Table 4: Training data scale and sources.

Method	Training data scale	Sources
RecA	>500K	[23, 61]
UniGame	∼450K	[15, 37]
UniCoT	∼290K	[55, 4, 20, 2, 43]
AD-Loop	∼51K	[8, 59, 36, 21, 11, 49]
IRG	∼300K	[22, 3, 19]
Ours	∼38K	[15, 26, 11, 8, 59, 12, 34]

C Benchmark and Metric Details

In Table 1, understanding and GenEval scores are accuracies in percent, WISE reports WiScore, and Avg. is the mean over the five understanding benchmarks when all scores are available. Appendix F.1 reports a separate relative-gain comparison with AD-Loop to avoid mixing potentially different backbone and evaluation settings in the main table.

Understanding benchmarks. MMMU [57] evaluates college-level multimodal knowledge and reasoning. MMBench-EN and MMBench-CN [24] evaluate instruction-following visual question answering in English and Chinese, with the Chinese split testing behavior under a language distribution different from our mostly English training sources. MathVista [25] emphasizes mathematical and visual reasoning, while MMStar [5] focuses on fine-grained multimodal reasoning. We report the official accuracy for each benchmark, and analyze selected subsets and per-path behavior when path-level outputs are available.

Generation benchmarks. GenEval [14] measures text-to-image object binding and compositional accuracy across categories such as object count, color, position, and attribute binding. WISE [33] evaluates broader text-to-image alignment across domains including culture, time, space, biology, physics, and chemistry. Generation evaluation uses the trained executor under the same generation protocol across prompts, so these scores reflect executor training rather than planner routing.

Understanding-generation consistency. UnifiedBench [53] evaluates whether a UMM preserves image information across a reconstruction loop. The model first converts an input image into text and then regenerates an image from that text. The reconstructed image is compared with the original image using CLIP, DINOv2, DINOv3, and LongCLIP similarities. This benchmark probes the alignment effect of executor training and aligned visual-thought supervision, because success requires visual information encoded during understanding to remain useful for later generation.

D Aligned Visual Thought Analysis

We further test whether aligned visual thought is preferable to more explicit feedback mechanisms for paths that use construction or hypothesis roles. On MMMU, we compare our readable visual-thought trace against two variants: *latent feedback*, which replaces the $p_{\mathrm{C}}/p_{\mathrm{H}}$ visual-thought step with generated pure visual latents, and *image feedback*, which generates an intermediate image and feeds it back for subsequent reasoning. The comparison is reported both for fixed $p_{\mathrm{C}}/p_{\mathrm{H}}$ execution and for the routed MMMU setting where only the selected $p_{\mathrm{C}}/p_{\mathrm{H}}$ calls are replaced.

The motivation is that both explicit alternatives have practical drawbacks. Full image feedback is expensive, and some intermediate visual thoughts are difficult to synthesize as precise images, so errors in the generated image can hurt later reasoning. Pure visual-latent feedback avoids rendering an image, but it places nonlinguistic visual states inside a text reasoning context, which can break semantic continuity. Aligned visual thought keeps the intermediate step as readable text while using image-derived supervision to shape its hidden representation.

Table 5 shows that explicit feedback is not only slower, but also less accurate. Replacing aligned visual thoughts with newly generated latents drops routed MMMU accuracy by 4.44 points, while feeding back generated intermediate images drops it by 3.44 points. The path-specific results show the same trend on both p_{C} and p_{H}, where the distinction between visual-thought supervision and explicit visual feedback matters most. In runtime, aligned visual thought reduces per-sample cost by 27.3–30.3% relative to latent feedback and by 24.4–28.6% relative to image feedback. These results support the intended design: keeping the intermediate step as text preserves semantic continuity for later reasoning, while image-derived supervision injects visual information into the hidden representation without paying the cost or brittleness of generating and reinserting a concrete visual object.

Table 5: MMMU analysis of aligned visual thought versus explicit latent/image feedback. Accuracy is reported on 900 examples. Runtime is average seconds per sample for fixed p_{C} or p_{H} execution.

Variant	Routed MMMU	Fixed p_{C}	Fixed p_{H}	p_{C} time	p_{H} time
Aligned visual thought (ours)	54.44 (490/900)	52.33 (471/900)	51.00 (459/900)	29.05	25.37
Latent feedback	50.00 (450/900)	48.00 (432/900)	45.67 (411/900)	39.98	36.41
Image feedback	51.00 (459/900)	48.33 (435/900)	48.22 (434/900)	38.42	35.51

E More Details and Analysis on Planner

E.1 Planner Training Details

We use a compact planner for all routed understanding experiments. The planner takes a path-aware feature vector and produces one score for each path in $\mathcal{P} = \{p_{\mathrm{A}}, p_{\mathrm{U}}, p_{\mathrm{R}}, p_{\mathrm{C}}, p_{\mathrm{H}}\}$. The input feature

concatenates an image-summary feature with the last-token and mean text features extracted under each candidate path prompt. This yields a 39,424-dimensional feature vector. The planner itself is a two-hidden-layer MLP, where the two hidden layers have width 768 and the output layer has dimension 5. The five output logits are converted to path-wise scores with a sigmoid. The final route is selected after the query-form calibration step described in Sec. 4.1.

Planner supervision is multi-label, since more than one path can solve the same input. We train the planner with weighted binary cross-entropy over the five path labels. This paragraph maps the notation in Sec. 3.3 to the implementation. Samples where all five paths are incorrect are removed from planner training. For each remaining sample, let $n_i = \sum_{p \in \mathcal{P}} r_{i,p}$ be the number of positive paths. The sample weight is $\omega_i = 3.0$ when $n_i = 1$, $\omega_i = 2.0$ when $n_i = 2$, and $\omega_i = 1.0$ when $n_i \geq 3$. The path-level label weight is $\beta_{i,p} = 1.3$ only for positive labels outside p_{A}, namely $r_{i,p} = 1$ and $p \neq p_{\mathrm{A}}$, and $\beta_{i,p} = 1.0$ otherwise. The regularization coefficient λ_{reg} is implemented through AdamW weight decay, set to 5×10^{-5}. We use batch size 256 and learning rate 5×10^{-4}.

E.2 Planner Training Transfer

Figure 3(c) in the main text shows how held-out planner validation utility transfers to routed MMMU accuracy. Here we additionally report the selected-path distributions of the same five planner checkpoints or configurations. The path distributions reveal a clearer failure mode: lower-utility planners often collapse to a narrow path preference, such as routing 856/900 examples to p_{R} or 690/900 examples to p_{U}. This supports the role of planner training in learning nontrivial path selection, rather than relying only on hand-crafted query-form calibration.

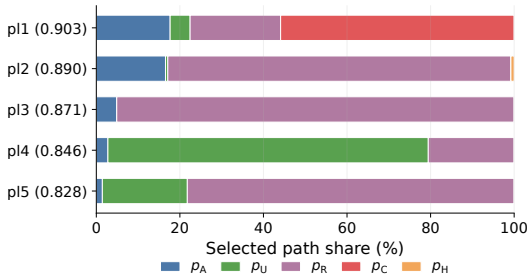


Figure 7: **Selected path distributions of planner checkpoints.** pl1 is the final planner used in our system. Lower-utility planner variants often collapse to a small subset of paths, while the final planner keeps a broader routing pattern.

E.3 Planner Feature-Space Analysis

To better understand why planner generalization remains difficult, we visualize planner input features from the five understanding benchmarks. Figure 8 shows that the global feature space is organized much more clearly by dataset/domain than by the path that happens to solve the example. Panel (b) colors each sample by an oracle-correct path label. For examples solved by multiple paths, we randomly sample one correct label for visualization. The resulting labels remain heavily mixed across the global embedding, indicating that limited planner supervision must contend with both domain shift and overlapping path labels. Panels (c)–(e) visualize local UMAP projections within three representative query-form buckets, denoted Bucket 1–3. Since all displayed samples are correctly solved by our routed model, these local panels color each point by the planner-selected successful path, which is also one of its correct paths. This avoids arbitrary coloring for multi-path examples and shows that bucket-conditioned views are more homogeneous and often expose clearer local path structure than the global path view. The result supports the role of query-form buckets as a lightweight inductive bias: they do not solve path selection by themselves, but they partition the input space before applying learned planner scores, making the routing problem less entangled.

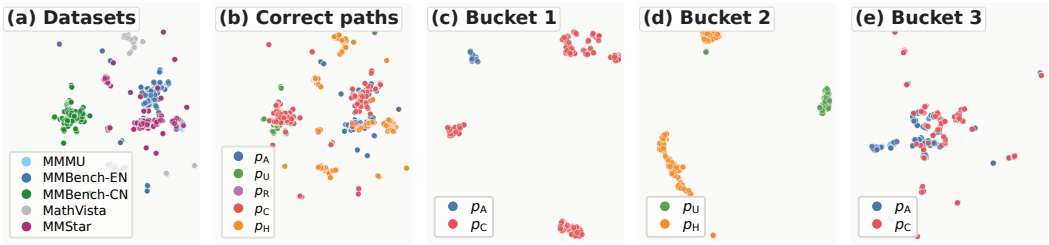


Figure 8: **Planner feature-space visualization on correctly solved sampled examples.** Panels (a)–(b) show the global PCA–UMAP embedding colored by dataset/domain and by one randomly sampled oracle-correct path label for multi-path examples. Panels (c)–(e) show local UMAP projections within three representative buckets, colored by the planner-selected successful path. The global space shows stronger dataset/domain clustering than path separation, while bucket-conditioned views reveal more localized path structure.

F More Experimental Results

F.1 Relative Gains of BAGEL-based Post-training Methods

Table 6 compares relative improvements over the corresponding BAGEL baseline for BAGEL-based post-training methods. For RecA, UniGame, UniCoT, and Ours, the gains are computed from Table 1. The AD-Loop row is computed from the values reported in its paper.

Table 6: Relative gains over the corresponding BAGEL baseline. RecA, UniGame, UniCoT, and Ours use the BAGEL baseline in Table 1. AD-Loop uses the BAGEL baseline reported in its paper. Missing entries indicate unavailable reports.

Method	MMMU	MMB-EN	MMB-CN	MathVista	MMStar
RecA	+0.8%	+0.0%	+0.0%	-27.9%	+8.3%
UniGame	+1.0%	+0.1%	+0.0%	+0.8%	+8.8%
UniCoT	+2.3%	+0.6%	+0.1%	+2.0%	**+10.8%**
AD-Loop	+3.6%	+3.1%	–	**+4.9%**	+5.5%
Ours	**+4.3%**	**+4.4%**	**+3.2%**	+0.8%	+7.7%

F.2 Path Format Compliance on MMMU

We audit whether the executor follows the requested path template during fixed-path MMMU evaluation. Section format checks whether the output contains the role sections required by the path, such as `Understanding`, `Reasoning`, `Visual`, `Hypothesis`, and `Answer`. Pred answer format checks whether the parsed `pred_answer` is a legal answer, namely a valid option letter for multiple-choice questions or a nonempty answer for open questions. Strict answer-text format additionally checks whether the raw answer span follows the expected answer surface form.

Table 7: Path-format compliance on fixed-path MMMU evaluation. Format checks are high across all paths, showing that the executor usually follows the requested coordination template.

Path	Section format	Pred. answer format	Strict answer-text format	Acc.	Avg. tokens
p_A	900 / 900 = 100.00%	900 / 900 = 100.00%	900 / 900 = 100.00%	51.89	1.2
p_U	899 / 900 = 99.89%	900 / 900 = 100.00%	899 / 900 = 99.89%	50.11	74.9
p_R	880 / 900 = 97.78%	900 / 900 = 100.00%	875 / 900 = 97.22%	52.22	231.4
p_C	886 / 900 = 98.44%	900 / 900 = 100.00%	886 / 900 = 98.44%	52.33	291.5
p_H	886 / 900 = 98.44%	900 / 900 = 100.00%	887 / 900 = 98.56%	51.00	295.7

The executor follows the requested section structure for nearly all examples and always produces a parsable final answer. Thus, downstream failures are unlikely to be dominated by simple template-following errors. Together with the planner analyses in the main text, this suggests that the larger remaining bottleneck is selecting the right path for each query rather than making the executor emit the requested path format.

F.3 Different Backbone Experiments

We further instantiate the same coordination path space with a smaller Harmon-1.5B [48] backbone. This experiment is intended to test whether path complementarity is tied only to the main BAGEL backbone. Table 8 shows that single-path performance remains close to raw Harmon performance, but the five-path oracle reaches 67.78% on MMMU. The gain over raw Harmon is 33.45 points, indicating that the path space still exposes substantial complementary behavior under a different and smaller executor.

Table 8: MMMU path complementarity with a Harmon-1.5B backbone. The oracle selects the correct answer whenever any of the five path-conditioned executions is correct.

Metric	Raw	p_{A}	p_{U}	p_{R}	p_{C}	p_{H}	Oracle
Correct / Total	309 / 900	302 / 900	296 / 900	272 / 900	289 / 900	252 / 900	610 / 900
Acc.	34.33	33.56	32.89	30.22	32.11	28.00	67.78

Table 9 reports deployable routed results on five understanding benchmarks. The routed system improves over raw Harmon on four of the five datasets, but the gains are much smaller than the oracle gap in Table 8. This suggests that the coordination paths remain useful across backbones, while converting path complementarity into reliable routed gains becomes harder with a weaker executor.

Table 9: Routed results with the Harmon-1.5B backbone. Scores are accuracies in percent.

Method	MMMU	MathVista	MMStar	MMB-EN	MMB-CN	Average
Raw	34.33	24.50	37.47	62.72	54.65	42.73
Ours	34.78	28.70	38.00	63.06	54.65	43.84

F.4 Additional Generation Results

Tables 10 and 11 expand the generation results summarized in the main text. We report the category-level GenEval and WISE scores for the same models as Table 1, so that the overall generation numbers can be traced back to their underlying subcategories.

Table 10: Complete GenEval results. Scores are accuracies in percent.

Method	Single object	Two object	Counting	Colors	Position	Color attr.	Overall
BAGEL	99.38	94.19	78.75	87.77	51.00	61.75	78.81
BLIP3-o	98.12	93.18	73.44	86.17	72.75	64.50	81.36
Show-o2	97.81	71.46	48.75	78.46	20.00	42.75	59.87
Show-o2 (1.5B)	96.88	64.39	46.88	76.06	16.75	32.00	55.49
Janus-Pro	97.81	86.62	57.50	89.36	76.00	66.25	78.92
Janus	85.62	37.63	18.75	53.46	17.50	27.25	40.04
JanusFlow	94.25	46.06	27.75	74.68	32.20	25.00	49.99
OmniGen2	99.69	93.94	68.75	88.03	53.25	67.50	78.53
TokenFlow	97.19	59.60	37.81	86.17	17.25	15.25	52.21
Emu3	94.69	55.81	30.00	76.06	8.50	9.50	45.76
DeepGen	98.75	98.99	81.25	92.55	75.00	73.00	86.59
MMaDA	89.06	49.75	31.25	73.67	12.50	20.50	46.12
Emu3.5	100.00	93.94	49.06	91.49	85.50	71.00	81.83
Ovis-U1	100.00	94.95	93.75	93.62	80.00	78.00	90.05
IRG	98.44	87.37	70.31	78.72	40.50	57.00	72.06
RecA	99.38	94.44	79.38	89.10	61.75	74.25	83.05
UniCoT	98.44	91.92	80.94	86.44	47.75	62.00	77.91
UniGame	98.44	95.96	81.25	93.62	72.00	75.75	86.17
Ours	98.00	95.00	77.00	86.00	54.00	69.00	80.00

Table 11: Complete WISE results. Scores are WiScore.

Method	Culture	Time	Space	Biology	Physics	Chemistry	Overall
BAGEL	0.3883	0.4386	0.4714	0.3620	0.4205	0.2940	0.3989
BLIP3-o	0.4028	0.4186	0.5259	0.4025	0.4255	0.3000	0.4138
Show-o2	0.3641	0.3497	0.4519	0.3455	0.3690	0.2390	0.3595
Show-o2 (1.5B)	0.3111	0.3563	0.4357	0.3150	0.3840	0.2310	0.3349
Show-o	0.2865	0.3225	0.4132	0.2750	0.3300	0.1980	0.3037
Janus-Pro	0.3616	0.3853	0.4789	0.3605	0.4745	0.2485	0.3811
Janus	0.2080	0.2707	0.3508	0.1705	0.1910	0.1095	0.2222
JanusFlow	0.2731	0.3222	0.3947	0.3215	0.2860	0.1905	0.2954
OmniGen2	0.4180	0.4042	0.4887	0.3635	0.3875	0.2810	0.4029
TokenFlow	0.3253	0.3626	0.3357	0.2915	0.2605	0.1510	0.3056
Emu3	0.3463	0.3482	0.3711	0.3310	0.3685	0.2130	0.3373
DeepGen	0.5989	0.4955	0.6102	0.4765	0.5515	0.4080	0.5470
Emu3.5	0.7001	0.5683	0.6944	0.6435	0.6085	0.4060	0.6331
MMaDA	0.6502	0.6814	0.7492	0.6620	0.7420	0.4205	0.6560
Ovis-U1	0.3643	0.3991	0.4831	0.3405	0.4090	0.2390	0.3755
IRG	0.3674	0.4081	0.4650	0.3575	0.4495	0.2655	0.3842
RecA	0.4035	0.4147	0.5432	0.3985	0.4630	0.3340	0.4225
UniCoT	0.3998	0.4183	0.4797	0.3405	0.4550	0.3060	0.4037
UniGame	0.3956	0.4138	0.4876	0.3590	0.4355	0.3155	0.4032
Ours	0.3905	0.4287	0.4962	0.4360	0.4370	0.2890	0.4100

G Executor Training Details

This appendix summarizes the final staged LoRA executor used in the main experiments. We initialize from BAGEL-7B-MoT, freeze the base model, and train language-model LoRA adapters together with the lightweight projection head used for aligned visual-thought supervision. The LoRA adapter uses rank 16, alpha 32, and dropout 0.05. We report only the hyperparameters needed to interpret the method-level objectives in Sec. 3.

Executor training is organized into four staged splits that activate different links of the path: text-only understanding (S1), understanding with aligned visual thoughts (S2), final image generation without aligned visual-thought supervision (S3), and final image generation with construction or hypothesis supervision (S4). The four stages form a strict checkpoint chain: S1 best is used to initialize S2, S2 best initializes S3, and S3 best initializes S4. All stages use batch size 1, bf16 training, weight decay 0.01, evaluation every 200 steps, and GH200 GPUs. S1 and S2 are trained for one epoch with maximum sequence lengths 3072 and 6144, respectively. S3 and S4 are trained for 3000 steps. Best-checkpoint selection enforces no large degradation in format accuracy, with tolerance 0.01 for S1 and 0.03 for S2–S4. S1 uses early-stop patience 6 with a 1200-step warmup before stopping can trigger, while S2–S4 use patience 8 with a 1600-step warmup. For S3 and S4 generation-side validation, evaluation allows up to 2048 new tokens.

The loss weights in Table 12 correspond directly to the symbols in Sec. 3.3. In S1 and S2, w_t is implemented as a token-level `loss_weights` vector: prompt and context tokens have weight 0, intermediate role tokens use the "thought CE" value, and final `Answer` tokens use the "answer CE" value. The resulting normalized $\mathcal{L}_{\text{text}}$ is used with $\lambda_{\text{text}} = 1$. In S3 and S4, the native image-answer trainer uses unit token weights for supervised target text and controls the text term with the global CE coefficient, so the table reports this value as λ_{text}. The MSE and visual columns are λ_{mse} and λ_{vis}, with absent losses set to 0.

Table 12: Staged executor training hyperparameters. The final column gives the code-level settings for the token weights w_t and objective coefficients λ_{text}, λ_{mse}, and λ_{vis} in Sec. 3.3.

Algorithm 1 Staged executor training pipeline.

Require: Base UMM M_0, staged role-aligned data $\mathcal{D}_{1:4}$, latent cache $\mathcal{V}$
Ensure: Path-conditioned executor $\mathcal{E}_\theta$
Initialize the executor from M_0
Attach language-model LoRA adapters and projection head g_ϕ, then freeze the base UMM
Set $C_0 \leftarrow M_0$ with trainable LoRA parameters and g_ϕ
for $s = 1$ to 4 **do**
  Load stage data $\mathcal{D}_s$ and initialize from checkpoint C_{s-1}
  for minibatch $B \subset \mathcal{D}_s$ **do**
    Convert each example to tagged role segments according to its path p
    Compute the role-weighted text loss $\mathcal{L}_{\text{text}}$ on supervised text tokens
    if B contains `Visual` or `Hypothesis` spans **then**
      Retrieve the corresponding reference visual summaries $\{v_j\}$ from $\mathcal{V}$
      Pool hidden states $\{\bar{h}_j\}$ over the tagged visual-thought spans
      Compute $\mathcal{L}_{\text{vis}} = \frac{1}{J}\sum_j \|g_\phi(\bar{h}_j) - v_j\|_2^2$
    else
      Set $\mathcal{L}_{\text{vis}} \leftarrow 0$
    end if
    if B contains final image answers **then**
      Compute BAGEL's final image-latent reconstruction loss $\mathcal{L}_{\text{latent}}$
    else
      Set $\mathcal{L}_{\text{latent}} \leftarrow 0$
    end if
    Update trainable parameters with $\lambda_{\text{text}}\mathcal{L}_{\text{text}} + \lambda_{\text{mse}}\mathcal{L}_{\text{latent}} + \lambda_{\text{vis}}\mathcal{L}_{\text{vis}}$
  end for
  Select the best checkpoint C_s using the stage validation metric and format-preservation gate
end for
Set $\mathcal{E}_\theta \leftarrow C_4$
return $\mathcal{E}_\theta$

Stage	Role split	Initialization	Hardware	LR	w_t and λ settings
S1	Text understanding, $p_{\text{U}}/p_{\text{R}}$	BAGEL init	2 GH200 GPUs	3×10^{-6}	w_t=0.5 for thought tokens, w_t=4.0 for answer tokens, λ_{text}=1, λ_{mse}=0, λ_{vis}=0
S2	Visual-thought, $p_{\text{C}}/p_{\text{H}}$	S1 best	4 GH200 GPUs	4×10^{-6}	w_t=0.25 for thought tokens, w_t=6.0 for answer tokens, λ_{text}=1, λ_{mse}=0, λ_{vis}=0.05
S3	Plain image answer	S2 best	4 GH200 GPUs	3×10^{-6}	unit target-token weights, λ_{text}=2.0, λ_{mse}=0.3, λ_{vis}=0
S4	Image answer + visual	S3 best	4 GH200 GPUs	3×10^{-6}	unit target-token weights, λ_{text}=2.0, λ_{mse}=0.3, λ_{vis}=0.05

For staged executor diagnostics, answer accuracy reports final-answer correctness on understanding splits. Format accuracy measures whether the output follows the expected role structure, so it is a sanity check for path execution rather than evidence of correct reasoning. CE denotes the weighted language-modeling loss on supervised text tokens. Visual denotes the alignment loss between textual visual-thought hidden states and image-derived visual summaries. MSE denotes BAGEL's final image-latent reconstruction loss for generation-side stages.

G.1 Executor Data Statistics

Table 13 reports the split sizes used by the staged executor run after filtering and stage assignment. These counts correspond to the four validation splits used in Table 14. They describe the executor-stage data rather than the additional path-outcome runs used to build planner supervision.

Table 13: Staged executor split statistics.

Stage	Split name	Train	Val	Total
S1	`understanding_text`	12,733	164	12,897
S2	`understanding_visual`	5,232	68	5,300
S3	`image_answer_plain`	5,380	92	5,472
S4	`image_answer_visual`	6,282	115	6,397
Total	–	29,627	439	30,066

G.2 Executor Training Diagnostics

Before the planner is evaluated on downstream benchmarks, we compare several executor training strategies on the four staged validation splits. This diagnostic separates superficial format following from useful path execution. A valid trajectory should both obey the role structure and improve the task-relevant objective. Figure 9 shows the smoothed training losses of the final staged LoRA executor, and Table 14 reports the raw BAGEL executor and three finetuning variants on held-out validation splits. The *staged LoRA* setting trains one stage at a time and passes the previous adapter forward. This is the executor used in our final experiments. *Partial-SFT* unfreezes a subset of the language model instead of using PEFT adapters [32], while *multitask Partial-SFT* mixes all stages in a single training run.

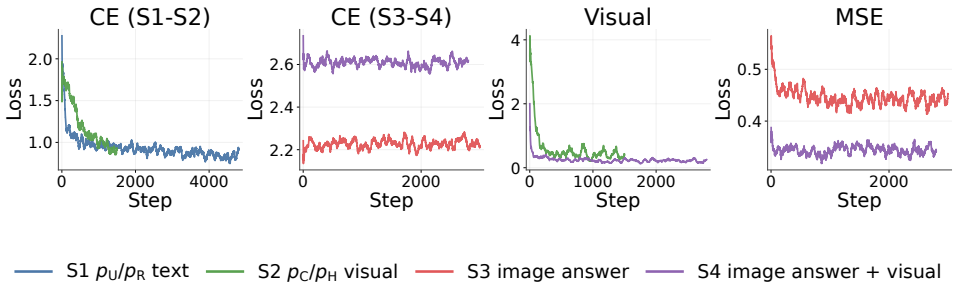


Figure 9: **Smoothed training losses for the staged LoRA executor.** Panels are arranged in one row and use the same stage names as Table 14. CE decreases clearly on S1/S2, while visual and MSE losses show that aligned visual-thought and image-latent objectives are actively optimized in the stages where they are enabled.

Table 14: Executor training diagnostics on the four staged validation splits. Acc. and Format are higher-is-better, while CE, Visual, and MSE are lower-is-better. Although raw BAGEL often preserves the requested format, its low answer accuracy and weak intermediate/final losses indicate that format compliance alone does not provide meaningful path execution. The final system uses staged LoRA.

Stage	Method	Acc.	Format	CE	Visual	MSE
S1 p_U/p_R text	BAGEL	0.2439	1.0000	1.7557	–	–
	Staged LoRA	0.8125	1.0000	0.9775	–	–
	Partial-SFT	0.3476	1.0000	1.4324	–	–
	Multitask Partial-SFT	0.4531	1.0000	1.9636	–	–
S2 p_C/p_H visual	BAGEL	0.2353	1.0000	2.8750	4.0301	–
	Staged LoRA	0.3253	0.7390	0.5384	0.3523	–
	Partial-SFT	0.3235	1.0000	1.8601	3.4149	–
	Multitask Partial-SFT	0.2500	0.9844	2.0606	6.4729	–
S3 image answer	BAGEL	–	1.0000	2.3748	–	0.5410
	Staged LoRA	–	1.0000	2.2334	–	0.4510
	Partial-SFT	–	1.0000	1.7335	–	0.5320
	Multitask Partial-SFT	–	1.0000	1.7537	–	0.5376
S4 image answer + visual	BAGEL	–	0.9478	2.7529	1.4585	0.4560
	Staged LoRA	–	0.9688	2.6233	0.2718	0.3446
	Partial-SFT	–	1.0000	2.6146	0.6878	0.4453
	Multitask Partial-SFT	–	1.0000	2.6545	1.0655	0.4535

Several observations guide the final experimental choice. First, the raw BAGEL backbone can often preserve the requested output structure, with near-perfect format accuracy in several splits. However, its S1/S2 answer accuracy remains low and its visual-summary loss is high, indicating that the pretrained model is mostly following the surface template rather than executing the intended intermediate reasoning or visual-grounding operations. Second, format accuracy alone is not sufficient: Partial-SFT and multitasking often maintain perfect formatting while improving task accuracy only weakly. The downstream format audit in Appendix F.2 further confirms that the final executor can follow path templates on MMMU. Finally, staged LoRA provides the best balance between stage-level

learning, format preservation, checkpoint size, and downstream path behavior, so we use it for the executor in the main experiments.

H Path Prompt Templates

This section lists the execution-time prompt wrappers used for path-conditioned evaluation. For p_{A}, the executor receives the benchmark query directly. For p_{U}, p_{R}, p_{C}, and p_{H}, the template below is prepended and the benchmark query is appended after `Query:`. These prompts are used after the planner has selected a path. They specify the executor's role order and output format rather than replacing the learned path selector.

Prompt for p_{A}

```
<benchmark query>
```

Prompt for p_{U}

```
You are a multimodal reasoning model. For this task, you must follow this EXACT reasoning path:

Understanding -> Answer

Where:
- Understanding = Text Understanding: give exactly one sentence that directly describes the visible
      input and the task, without inference
- Answer = Final Answer: answer directly from perception only

Output format:

Understanding:
(Exactly one sentence directly describing the visible input and the task)

Answer:
(Direct answer based only on what is explicitly visible or stated)

Now solve the following query using ONLY this path (Understanding -> Answer):

Query:
<benchmark query>
```

Prompt for p_{R}

```
You are a multimodal reasoning model. For this task, you must follow this EXACT reasoning path:

Understanding -> Reasoning -> Answer

Where:
- Understanding = Text Understanding: give exactly one sentence that directly describes the visible
      input and the task, without inference
- Reasoning = Text Reasoning: give a short textual reasoning chain using only the provided information
- Answer = Final Answer: answer directly and concisely

Output format:

Understanding:
(Exactly one sentence directly describing the visible input and the task)

Reasoning:
(A short textual reasoning chain)

Answer:
(Final answer)

Now solve the following query using ONLY this path (Understanding -> Reasoning -> Answer):

Query:
<benchmark query>
```

Prompt for p_{C}

```
You are a multimodal reasoning model. For this task, you must follow this EXACT reasoning path:

Understanding -> Reasoning -> Visual -> Reasoning -> Answer

Where:
- Understanding = Text Understanding: give exactly one sentence that directly describes the visible
      input and the task, without inference
- Reasoning = Text Reasoning: give a short textual reasoning chain using only the provided information
- Visual = Visual Reasoning: describe one intermediate visual construct or visual step
- Answer = Final Answer: answer directly and concisely

Output format:

Understanding:
(Exactly one sentence directly describing the visible input and the task)

Reasoning:
(A short textual reasoning chain)

Visual:
(One intermediate visual construct or visual step)

Reasoning:
(A short textual reasoning chain that uses the visual step)

Answer:
(Final answer)

Now solve the following query using ONLY this path (Understanding -> Reasoning -> Visual -> Reasoning
      -> Answer):

Query:
<benchmark query>
```

Prompt for p_{H}

```
You are a multimodal reasoning model. For this task, you must follow this EXACT reasoning path:

Understanding -> Reasoning -> Hypothesis -> Reasoning -> Answer

Where:
- Understanding = Text Understanding: give exactly one sentence that directly describes the visible
      input and the task, without inference
- Reasoning = Text Reasoning: give a short textual reasoning chain using only the provided information
- Hypothesis = Visual Hypothesis: describe candidate visual states or hypotheses before answering
- Answer = Final Answer: answer directly and concisely

Output format:

Understanding:
(Exactly one sentence directly describing the visible input and the task)

Reasoning:
(A short textual reasoning chain)

Hypothesis:
(Candidate visual states or hypotheses)

Reasoning:
(A short textual reasoning chain that uses the visual hypotheses)

Answer:
(Final answer)

Now solve the following query using ONLY this path (Understanding -> Reasoning -> Hypothesis ->
      Reasoning -> Answer):

Query:
<benchmark query>
```

I Additional Qualitative Examples

We provide additional MMMU cases from the full five-path evaluation. In each case, only the highlighted path among $\{p_{\mathrm{A}}, p_{\mathrm{U}}, p_{\mathrm{R}}, p_{\mathrm{C}}, p_{\mathrm{H}}\}$ gives the correct answer.

p_{A}: direct answer

QUESTION. What are the yellow spheres connected to the roots of a potato? Options: A. Sclerotia, B. Cysts, C. Eggs, D. Galls.

PREDICTIONS.

p_{A}	p_{U}	p_{R}	p_{C}	p_{H}	GT
B	A	A	A	A	**B**

TRACE.

Direct output: B

WHY THIS PATH. The visual information needed by the answer is already described by the query itself: yellow spheres are connected to potato roots. Additional decomposition is unnecessary and can move the model toward plausible but wrong distractors.

p_{U}: visual understanding only

QUESTION. A velocity–time graph asks when the object again passes through its initial position. Options: A. 1 s, B. Between 1 and 2 s, C. 2 s, D. Between 2 and 3 s.

PREDICTIONS.

p_{A}	p_{U}	p_{R}	p_{C}	p_{H}	GT
C	**B**	C	C	C	**B**

TRACE.

Understanding: The velocity curve starts at $(0, 0)$, dips below zero before returning to 0 at $t = 1$, and then rises above zero toward 2 by $t = 2$. The return to the initial position is indicated between 1 and 2 s.

Answer: B

WHY THIS PATH. The decisive evidence is the numerical structure of the plotted curve. Explicit perception preserves the visual anchors needed to choose the interval, while additional roles lead the model toward incorrect alternatives.

p_{R}: textual reasoning

QUESTION. Given a chemical structure with a central Pt atom bonded to two ammine and two chlorine ligands, identify the untrue statement. Options: A. Chlorine groups are displaced before activation, B. The agent causes intrastrand DNA cross-linking, C. Adenine–thymine base pairing is disrupted, D. The compound acts as a metallating agent.

PREDICTIONS.

p_{A}	p_{U}	p_{R}	p_{C}	p_{H}	GT
A	A	**C**	A	A	**C**

TRACE.

Understanding: The image shows a platinum complex with a central Pt atom bonded to two ammonia groups and two chlorine atoms.

Reasoning: The structure is cisplatin. It acts as a metallating agent and leads to intrastrand DNA cross-linking. Base pairing between adenine and thymine is not disrupted by the agent.

Answer: C

WHY THIS PATH. The visual input must be interpreted, but the answer is determined by textual biochemical reasoning rather than by constructing an aligned visual thought.

p_{H}: visual hypotheses

QUESTION. Which graph shows how dissipated power P varies with current I for an Ohmic component? Options: A. upward-curving from the origin, B. decreasing linear, C. increasing linear, D. decreasing curved.

PREDICTIONS.

p_{A}	p_{U}	p_{R}	p_{C}	p_{H}	GT
C	C	C	C	**A**	**A**

TRACE.

Understanding: Four graphs labeled A, B, C, and D are shown, each plotting P against I.

Reasoning: For an Ohmic component, $V = IR$ and $P = I^2R$, so power is proportional to the square of current.

Hypothesis: Candidate visual hypotheses correspond to the four graph shapes: A is upward-curving from the origin, B is decreasing linear, C is increasing linear, and D is decreasing curved.

Reasoning: Graph A matches the expected upward-curving relation, while the linear and decreasing alternatives do not.

Answer: A

WHY THIS PATH. The difficulty is not only reading the figure, but choosing among plausible visual hypotheses. Explicit hypothesis comparison prevents the model from defaulting to the tempting linear option.

J Role-Aligned Trajectory Examples

We show one representative supervised trajectory for each coordination path beyond p_{A}. The examples are drawn from the staged executor data. Role contents are lightly shortened for readability while preserving the supervised role order. For p_{C} and p_{H}, the displayed visual references are used to supervise aligned visual thoughts during training. At inference time, the executor emits the tagged textual trace rather than inserting these images back into the context.

p_{U}: Understanding → Answer

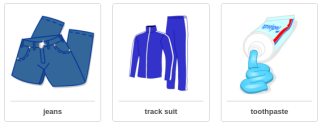


Input image

QUERY. What brand is the bike?

Understanding: A blue Kawasaki sport motorcycle with racing decals is parked on pavement, with people standing nearby and other vehicles in the background.
Answer: kawasaki

p_{R}: Understanding → Reasoning → Answer



Input image

QUERY. Which property do these three objects have in common? Choices: (A) fragile, (B) blue, (C) sticky.

Understanding: The image shows jeans, a track suit, and toothpaste, all containing visible blue regions.
Reasoning: Fragile and sticky are not shared by all three objects. Blue is the shared visual property.
Answer: blue

p_{C}: Understanding → Reasoning → Visual → Reasoning → Answer



Input image



Visual reference for supervision

QUERY. What is the status of the Samarinda flight?

Understanding: The image shows a Batik Air flight display with rows for Balikpapan, Makassar, and Samarinda.
Reasoning: The relevant evidence is the row whose destination is Samarinda, so the model should focus on that row before answering.
Visual: A visual thought isolates the Samarinda row, including flight ID 6672, time 14:30, destination Samarinda, and its status field.
Reasoning: The isolated row states "SESUAI JADWAL," which indicates that the flight is on schedule.
Answer: sesuai jadwal

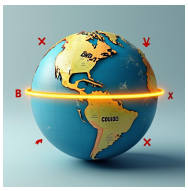
p_{H}: Understanding → Reasoning → Hypothesis → Reasoning → Answer



Input image



Candidate 1



Candidate 2

Hypothesis references

QUERY. Which of these continents does the equator intersect? Choices: (A) Europe, (B) Asia, (C) Antarctica.

Understanding: The image shows a world map with continents, oceans, and latitude lines, including the equatorial region.
Reasoning: The answer can be found by checking which candidate continent crosses the equator at 0 degrees latitude.
Hypothesis: Candidate references mark the equator on a world map or globe, making the possible land intersections easier to compare.
Reasoning: The equator does not pass through Europe or Antarctica. It crosses island regions of Asia, so the matching choice is Asia.
Answer: Asia

Limitations

The main limitation of our current system is path selection. Our results show a large remaining gap between the deployable planner and oracle path selection, which means that the proposed path space contains substantial unused potential. Closing this gap requires a planner that is both more accurate at the instance level and more robust across domains. This is challenging because planner supervision is expensive: to know which path works for an input, we must run multiple candidate paths and compare their outcomes. Such supervision is much more costly than ordinary single-trajectory training, especially for paths that involve visual-thought construction or hypothesis exploration.

The same issue also limits cross-dataset robustness. Different benchmarks have markedly different domain and query-form distributions, and our query-form buckets are calibrated on auxiliary data. As a result, the method implicitly assumes that the target test distribution is reasonably aligned with the calibration distribution. Dataset-adapted results suggest that performance can improve when the target distribution is known, but this weakens the goal of a single deployable planner. Future work should therefore focus on cheaper ways to collect path-outcome supervision and on planner objectives that generalize better under domain shift.

Broader Impact

This work studies how unified multimodal models coordinate understanding and generation. Potential positive impacts include more efficient multimodal reasoning, lower inference cost when a short path is sufficient, and more interpretable model behavior because the selected coordination path exposes when the model uses understanding, reasoning, visual thought, or hypothesis comparison. These properties may help developers analyze failures and build systems that spend computation more selectively.

The same capability can also create risks. Improvements in multimodal reasoning and generation can make synthetic visual content easier to produce or adapt, which may support disinformation or other deceptive uses if deployed without safeguards. Incorrect path selection can also make a system appear confident while using an inappropriate reasoning process, which is concerning in domains where visual answers affect user decisions. Because our planner uses query-form buckets calibrated on auxiliary data, performance may vary across domains, languages, or user groups whose inputs differ from the calibration distribution.

Mitigations include evaluating path selection across diverse domains and languages, reporting planner uncertainty or path traces to support auditing, and applying standard safeguards for generative systems such as content provenance, watermarking, misuse monitoring, and access controls when models are released. The system should not be used as an autonomous decision-maker in high-stakes settings without task-specific validation and human oversight.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims are stated in the abstract and introduction, supported in Sec. 4, and scoped by Appendix J.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The main limitations are discussed in Appendix J.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: The paper does not present theoretical theorems or proof claims.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Reproducibility details are provided in Sec. 4.1 and Appendices B, E.1, G, and C.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Code and processed trajectories are released at the URL in the abstract.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Settings are described in Sec. 4.1 and Appendices G, G.1, and E.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Error bars are omitted because repeating the full multi-path executor and planner pipeline is computationally expensive.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Compute resources are reported in Appendix G.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We confirm that the work follows the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Broader impacts are discussed in Appendix J.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The submission does not release a new high-risk pretrained model, image generator, or scraped image dataset.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Existing datasets, benchmarks, and models are credited.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The processed trajectories are documented in Appendix B and will be released.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The work does not involve crowdsourcing or human-subject experiments.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The work does not involve crowdsourcing or human-subject experiments, so IRB approval is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: The paper describes LLM use in Sec. 4.1 and Appendix B.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.