WORLDMIND: TEST-TIME SCALING FOR VISUAL SPATIAL REASONING THROUGH LATENT WORLD SIMULATION

ABSTRACT

Vision-Language Models (VLMs) have continuous deficiency in spatial reasoning tasks that require *mental spatial transformation*, such as reasoning under hypothetical changes in viewpoint or position, known as perspective-taking tasks. Existing approaches either scale linguistic reasoning or augment VLMs with static geometric representations, but neither directly enables active transformation of the represented scene. World models offer this capability, yet existing systems typically communicate through synthesized RGB images, introducing a latent-to-pixel-to-latent bottleneck. We introduce *WorldMind*, a framework for test-time scaling of spatial reasoning through latent world simulation. Given a query and observation, the VLM selects useful spatial transformations, while a world model executes them and exposes aligned intermediate latent features directly to the VLM. The resulting latent evidence is incorporated into subsequent reasoning steps, enabling iterative spatial simulation without repeatedly decoding imagined states into pixels. To optimize this interaction, WorldMind further employs process-verifiable reinforcement learning that rewards reasoning progress only when the associated action and latent transition are valid. WorldMind thus scales spatial reasoning by allocating additional inference-time computation to active world simulation rather than generating more language or pixels. Across VSI-Bench, MMSI-Bench, and MindCube-Tiny, WorldMind achieves an average improvement of 16.0% over the base VLM and 8.3% over the strongest RGB-based world simulation.

1 INTRODUCTION

Vision-Language Models (VLMs) have achieved remarkable success in multimodal perception, and demonstrated strong capabilities in semantic recognition and extraction of basic spatial cues from individual frames [5; 24; 8]. However, their performance degrades substantially when spatial reasoning requires going beyond what is directly observed, as shown in Figure 1, such as determining how objects would appear from another viewpoint, mentally rotating a scene, or planning a sequence of viewpoint changes [25; 44; 3; 27]. From the perspective of cognitive science, the limitation of VLMs in these perspective-taking tasks is due to their intrinsic deficiency in reproducing the human's spatial intelligence, which builds on *mental spatial transformation*, the ability to construct an internal representation of the environment and transform that representation under hypothetical changes in viewpoint, orientation, or position [36; 13; 15]. Current VLMs, since being trained for semantic understanding, lack a mechanism for similarly performing such internal spatial transformation, and hence struggle on complex downstream tasks that require reasoning over unobserved or transformed states, including multi-step navigation, dynamic perspective taking, and reasoning about occluded geometry [45; 52; 6].

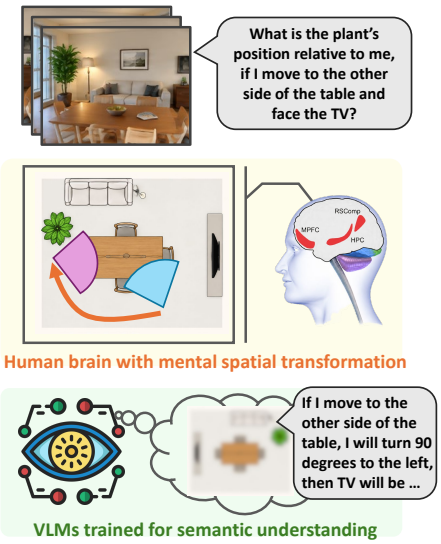


Figure 1: Spatial transformation in perspective-taking reasoning tasks

Existing efforts on addressing this gap primarily follow two approaches but both have fundamental limitations. The first approach is to scale linguistic test-time computation for mental spatial transformation, as shown in Figure 2(a). This strategy is highly effective in symbolic and mathematical reasoning [42; 37], but simply generating more texts at test time often fails to improve or even degrade spatial reasoning, because the missing spatial information about the real-world scene cannot

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

be made available by extra linguistic texts, which may instead introduce extra bias or confusions [45; 17; 16]. The second approach is to inject encoded 3D representations (e.g. depth maps or point clouds) into the VLMs' inputs [2; 23; 7]. While these representations provide useful geometric cues for the currently observed scene and benefit localized metric tasks such as object measurement, they remain limited to the existing observations and cannot support tasks that require active viewpoint transformation or exploration, such as reasoning from an opposing observer's perspective [50; 18].

Recently emerged models for novel view synthesis, such as world action models [57; 60; 34], are naturally a better alternative to enable spatial transformations and provide spatial information beyond current observations, because they are trained to predict how visual observations change under camera motions or explicitly specified actions. As a result, their learning objectives inherently require modeling how the environment changes under hypothetical transformations of viewpoint and state. More importantly, recent analyses suggest that their intermediate latent representations encode rich information about scene geometry, camera motion, object correspondence, and spatial structures, even without explicit 3D supervision [14; 54; 28]. This means that these models serve as **spatial simulators**, whose internal representations provide structured spatial information that can be directly exploited for spatial reasoning.

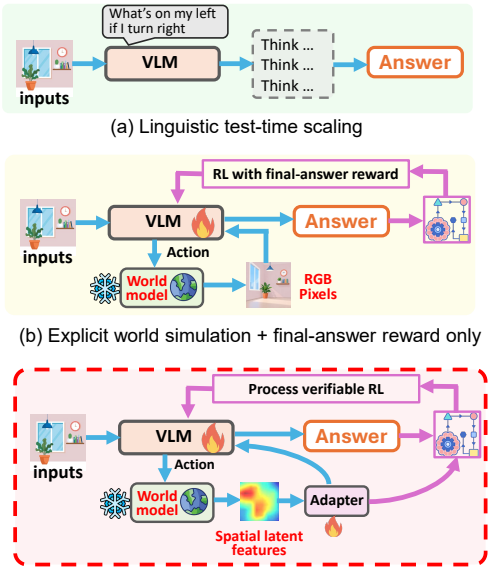


Figure 2: Comparisons of test-time scaling paradigms for spatial reasoning

Some existing approaches have coupled VLMs with world models for active visual imagination [59; 62; 58]. Given a spatial query, the VLM selects a target viewpoint, and the world model generates the novel view being needed as additional frames for reasoning. These methods establish an important paradigm of test-time scaling for spatial reasoning, through actively imagined observations in new viewpoints. However, they use the world model primarily as an ***explicit*** *image renderer* [4; 49]: the world model first converts its internal spatial representation into explicit pixels, from which the VLM recovers the spatial structures through its visual encoder, as shown in Figure 2(b). Since the visual encoder used in VLMs (e.g., ViT [9]) are mostly optimized for visual-semantic understanding, such recovery in VLMs would certainly cause loss or ineffective representation of important spatial information, such as geometry, object correspondence and relationship among objects.

In this paper, we present *WorldMind*, a new framework for test-time scaling of spatial reasoning through **latent world simulation**. As shown in Figure 2(c), given a spatial query and the current observation, the VLM iteratively decides *what to simulate* by selecting a useful viewpoint transformation, and the world model performs the scene transformation and returns its *latent representation* to VLM as spatial features. In this way, rather than relying on the synthesized pixels, WorldMind directly extracts the spatially informative latent representations from the world model, aligns them with the VLM, and incorporates them into VLM's reasoning. This interaction continues until sufficient evidence has been gathered to answer the query.

Realizing this latent world simulation poses two primary technical challenges. The first challenge is to determine which representations of the world model should be exposed to the VLM. The internal representations of diffusion world models are distributed across multiple network blocks and denoising steps, and features at different locations encode different types of information, ranging from low-level appearance and texture to object semantics, geometry and global scene structure [41]. WorldMind identifies spatially informative latent representations by evaluating features across diffusion layers and denoising steps, retaining those most useful for spatial question answering, and aligns them with the VLM through a lightweight adapter trained with answer supervision (Section 3.1).

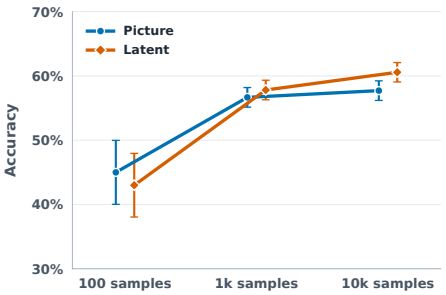
The second challenge is to effectively learn multi-step latent simulation. Final-answer supervision alone provides little guidance on whether each intermediate simulation step is useful or correctly executed [39]. To address this challenge, WorldMind trains the simulation policy in two stages:

supervised fine-tuning on synthetic reasoning trajectories to bootstrap viewpoint selection (Section 3.2), followed by reinforcement learning with step-level rewards that encourage task progress, correct execution of spatial transformations, and reliable simulation (Section 3.3).

To the best of our knowledge, WorldMind is the first that scales spatial reasoning at test time by directly exposing the world model's latent spatial representations to a VLM. It integrates latent feature selection, alignment, and multi-step world simulation into a unified reasoning framework, so that the VLM can actively acquire, interpret, and directly reason over transformed spatial states without repeatedly decoding them into pixels. We evaluated WorldMind across multiple spatial reasoning benchmarks, and experiment results show that WorldMind improves the average success rate of spatial reasoning tasks by 16.0% over the base VLMs and 8.3% over explicit world simulation, with particularly strong gains on viewpoint transformation and multi-step spatial reasoning tasks.

2 Motivation and Related Work

Recent studies show that intermediate representations of video diffusion and world models encode substantial 3D and dynamic information [14; 54]. These observations motivate our design of WorldMind to directly expose the selected latent representations of world model to the VLM, avoiding repeated decoding and re-encoding of explicit pixels as the existing approaches do [49; 4; 62; 58], while retaining active, query-driven world simulation at test time.



(a) Performance under different fine-tuning data scales.

Input	obj.cnt.	rel.dir	route	app.order
RGB	68.4	65.2	35.8	69.7
WM Latent	**69.3**	**67.2**	**38.1**	**69.4**

(b) Task-wise comparison on VSI-bench.

Figure 3: **Preliminary comparison between RGB and world-model latent representations.** (a) Comparison under different fine-tuning data scales. (b) Comparison across VSI-bench tasks.

To examine whether the world model's latent states are essentially better spatial representations compared to explicit pixels in VLM's spatial reasoning, we conduct a preliminary study using Qwen3-VL-8B-Instruct [1] on the MindCube benchmark [50], by comparing the model fine-tuned with RGB inputs against the model fine-tuned with world-model latent features under different training-data scales. For this preliminary study, we uniformly sample latent feautures over layers and denoise steps. As shown in Figure 3, RGB inputs perform better with small data scale, because the pretrained VLM is already well aligned with the image modality. As the amount of fine-tuning data increases, the latent-based model catches up with and eventually surpasses the RGB-based counterpart, suggesting that world-model latents can provide more effective spatial representations once sufficient alignment data are available. We further break down the comparison across different spatial reasoning tasks on VSI-bench 45. The advantage of latent representations is most pronounced on tasks requiring explicit transformation of the observed spatial state, such as relative direction (rel.dir) and route plan (route).

3 Method

As shown in Figure 4, given a sequence of current RGB observations $\{I_1, \ldots, I_N\}$ and a question Q, WorldMind augments the standard VLM visual representation with latent features extracted from a pre-trained world model. For each frame, the VLM produces H_i^{rgb} that encodes the observed RGB frame I_i in the VLM's native visual representation space, and the corresponding world-model latent Z_i is mapped into the VLM feature space through a learnable adapter as $H_i^w = P_w(Z_i)$. The initial context is constructed as $X_0 = [H_1^{\text{rgb}}, H_1^w, \ldots, H_N^{\text{rgb}}, H_N^w, Q]$.

During reasoning, the VLM iteratively generates a textual reasoning token R_t and a camera action A_t. This action is executed by the world model, which returns a new latent state $\widetilde{H}_t^w$ that is appended to the VLM's reasoning context. The complete inference sequence is $X_0 \to [R_1, A_1, \widetilde{H}_1^w] \to \cdots \to [R_T, A_T, \widetilde{H}_T^w] \to [J, Y]$, where J and Y denote the final justification and answer. In this way, additional test-time computation is used to acquire new latent spatial evidence rather than merely generating more texts.

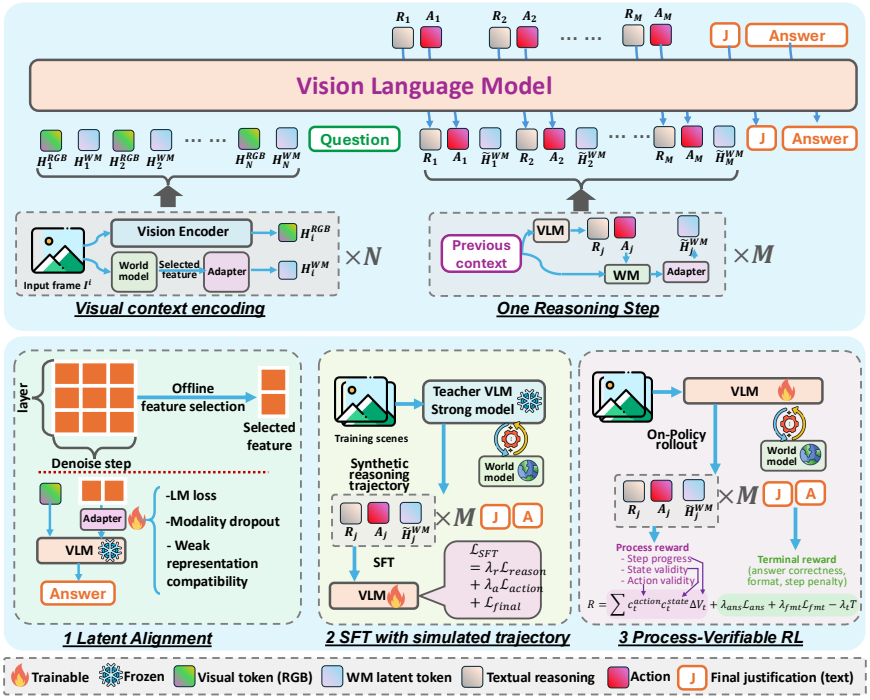


Figure 4: Overview of WorldMind design. Given the currently observed RGB frames and a spatial query, WorldMind augments standard VLM visual tokens with the selected latent features from a pre-trained world model, and performs iterative reasoning through camera actions and latent world simulation. To ensure proper latent world simulation, WorldMind first trains an adapter that selects latent features from the world model and aligns them with a frozen VLM. Then, it fine-tunes the VLM in two stages: (1) supervised fine-tuning with simulated reasoning trajectories; and (2) process-verifiable reinforcement learning, which further optimizes action selection and reasoning using intermediate and final rewards.

The key to ensuring proper latent world simulation described above is to select spatially informative world-model representations, align them with the VLM's feature space, and train the VLM to actively query and utilize these latent states during multi-step reasoning. To achieve this, we first train an adapter for **latent feature selection and alignment**, to identify a compact set of spatially informative world model representations and align them with a frozen VLM's representation space. Afterwards, it fine-tunes the VLM being used for spatial reasoning in two stages: (1) **simulation-trajectory supervised fine-tuning** trains the adapter and VLM's LoRA modules on synthetic trajectories, to interleave reasoning, camera-action selection, world-model simulation, and latent-state conditioning; and (2) **process-verifiable reinforcement learning** further optimizes the policy on its own sampled trajectories, rewarding the reasoning progress only when the intended spatial transformation is successfully realized and the resulting latent state is reliable, while retaining terminal answer supervision and penalizing unnecessary world-model calls. Together, these stages progressively teach WorldMind to interpret, reason with, and actively acquire latent spatial evidence.

3.1 Latent Selection and Alignment

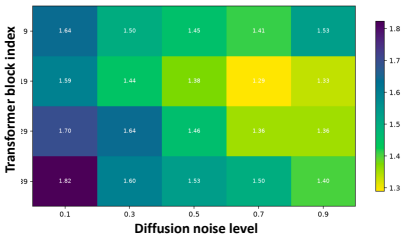


Figure 5: Spatial information across world-model latents.

World-model representations vary substantially across network layers and denoising timesteps, and not all latent features are equally informative for spatial reasoning. To show this empirically, we conduct a simple probing experiment in which a lightweight MLP is trained to reconstruct the scene's 3D point cloud from latent features extracted at different transformer blocks and diffusion noise levels. As shown in Fig. 5, reconstruction error varies considerably across layer–timestep combinations, with some intermediate representations preserving substantially more geometric information than others. This observation motivates us to avoid arbitrarily selecting a single

latent feature. Instead, we first identify a compact subset of spatially informative representations and then align the selected features with the pretrained VLM for downstream reasoning.

Learned latent selection. We construct a candidate feature bank $\mathcal{F} = \{h^{(\ell,\tau)}\}$ from multiple diffusion layers ℓ and denoising timesteps τ, and train a temporary fusion adapter over the full set. We then estimate the contribution of each layer–timestep pair using attribution or feature ablation and retain the top-K features, forming a fixed subset $\mathcal{S}^* = \{(\ell_1, \tau_1), \ldots, (\ell_K, \tau_K)\}$. This selection is performed offline and kept fixed during subsequent training and inference.

Using the selected features, we learn a lightweight adapter $\mathcal{P}_w$ that maps world-model representations into a form usable by the pretrained VLM. For frame I_i, the resulting latent representation is $H_i^w = \mathcal{P}_w(\{h_i^{(\ell,\tau)} : (\ell, \tau) \in \mathcal{S}^*\})$. During alignment, the world model and VLM backbone remain frozen, while only $\mathcal{P}_w$ is optimized. Rather than enforcing token-level matching with RGB features, we perform *functional alignment*, allowing the projected latents to preserve complementary geometric information such as depth, viewpoint, and 3D structure. Given the target response $Y^{\text{align}} = [J, Y]$, we optimize the standard autoregressive objective $\mathcal{L}_{\text{ans}} = -\sum_k \log p_\theta(y_k \mid X_{\text{align}}, y_{<k})$.

Modality dropout. Since I_i and Z_i describe the same scene, the frozen VLM may rely on RGB alone and ignore the world-model tokens. To mitigate this modality dominance [31], we randomly train with either the joint input $X_{\text{joint}} = [H^{\text{rgb}}, H^w, Q]$ or a latent-only input $X_{\text{latent}} = [H^w, Q]$, sampled with probabilities $1 - p_{\text{drop}}$ and p_{drop}, respectively. The latent-only samples force the latent interface to expose task-relevant information from the world-model representation.

Weak representation compatibility. We further introduce an optional global compatibility objective that encourages corresponding RGB and world-model representations to occupy a similar semantic space without enforcing token-level matching. We compute pooled projections $c_i^w = g_w(\text{Pool}(H_i^w))$ and $c_i^{\text{rgb}} = \text{sg}(g_v(\text{Pool}(H_i^{\text{rgb}})))$, normalize them, and apply a contrastive loss $\mathcal{L}_{\text{compat}}$ over matched and mismatched pairs within the batch [11; 43]. The overall alignment objective is

$$\mathcal{L}_{\text{align}} = \mathcal{L}_{\text{ans}} + \lambda_{\text{compat}} \mathcal{L}_{\text{compat}}, \tag{1}$$

where λ_{compat} is kept small so that functional answer supervision remains dominant. The compatibility branch is used only during training and discarded at inference.

3.2 Simulation-Trajectory Supervised Fine-Tuning

After latent alignment, the model can interpret world-model features but has not yet learned how to use them within an iterative reasoning process. This stage teaches the VLM to interact with the world model by identifying missing spatial information, selecting useful camera transformations, and incorporating the resulting latent states into its reasoning.

Such training requires step-level supervision over both textual reasoning and camera actions. Given an initial context, we assume a target trajectory of the form $[R_1^*, A_1^*, \widetilde{H}_1^w, \ldots, R_T^*, A_T^*, \widetilde{H}_T^w, J^*, Y^*]$, where R_t^* and A_t^* denote the target reasoning step and camera action, and $\widetilde{H}_t^w$ is the world-model latent returned after executing A_t^*. The latent states are treated as observations rather than autoregressive prediction targets; supervision is applied only to the reasoning tokens, camera actions, and final response.

The overall SFT objective is

$$\mathcal{L}_{\text{SFT}} = \lambda_r \mathcal{L}_{\text{reason}} + \lambda_a \mathcal{L}_{\text{action}} + \lambda_f \mathcal{L}_{\text{final}}, \tag{2}$$

where

$$\mathcal{L}_{\text{reason}} = -\sum_{t=1}^{T} \sum_{k \in R_t^*} \log p_\theta \left(r_{t,k}^* \mid C_t, r_{t,<k}^* \right), \tag{3}$$

and

$$\mathcal{L}_{\text{action}} = -\sum_{t=1}^{T} \log p_\theta \left(A_t^* \mid C_t, R_t^* \right). \tag{4}$$

Here, C_t contains the initial multimodal context together with all preceding reasoning steps, actions, and returned world-model latents. The final-response loss $\mathcal{L}_{\text{final}}$ is the standard autoregressive loss over the final justification and answer. We use separate terms because reasoning, camera actions, and final responses constitute distinct supervised outputs, whereas the returned world-model latents are conditioning observations rather than prediction targets [61; 46].

In practice, such interleaved reasoning–action–latent trajectories are not directly available in existing spatial-reasoning datasets. We therefore construct synthetic reasoning trajectories using a teacher model. For more details, please check Appendix B.

During this stage, we jointly optimize the latent adapter $\mathcal{P}_w$ and lightweight LoRA modules inserted into the VLM, while keeping the pretrained VLM backbone and world model frozen. Unlike latent alignment, which teaches the model to interpret world-model features, trajectory SFT teaches it how to use those features within an iterative spatial reasoning process.

3.3 Process Verifiable RL

While supervised fine-tuning provides a strong initialization for spatial reasoning and action generation, it primarily teaches the model to imitate teacher-provided trajectories rather than directly optimize task performance. We therefore further train WorldMind with reinforcement learning, allowing the model to explore its own reasoning and camera-action trajectories and optimize them according to task-level rewards. This stage encourages the model to discover more effective spatial transformations, improve final-answer accuracy, and avoid unnecessary simulation steps.

At step t, the policy samples $(R_t, A_t) \sim \pi_\theta(\cdot \mid C_t)$. The action A_t is executed by the world model to produce $\widetilde{Z}_t = \mathcal{W}(S_{t-1}, A_t)$, which is projected as $\widetilde{H}_t^w = \mathcal{P}_w(\widetilde{Z}_t)$ and appended to the context for the next reasoning step. This yields an interactive rollout

$$S_0 \xrightarrow{\pi_\theta} A_1 \xrightarrow{\mathcal{W}} S_1 \xrightarrow{\pi_\theta} \cdots \xrightarrow{\mathcal{W}} S_T \to Y. \tag{5}$$

A central challenge is assigning credit not only to the final answer, but also to the intermediate reasoning and simulation steps that produce it. Conventional outcome rewards, such as answer correctness and format compliance, provide little guidance on whether a camera transformation is useful or whether the resulting world-model state contributes to subsequent reasoning [22; 12]. We therefore augment it with a step-wise reward.

Conceptually, we impose three requirements on each reasoning step. First, the newly acquired evidence should improve the model's ability to answer the question compared with reasoning from the previous context alone. We therefore define the **progress reward** [35] as

$$r_t^{\text{prog}} = \Delta V_t = V(C_{t+1}) - V(C_t), \tag{6}$$

where $V(C_t)$ denotes the estimated probability of reaching the correct answer from context C_t. Second, the selected camera action should correctly realize the intended spatial instruction. We capture this with the **action validity** c_t^{action}, which measures if the generated action A^j is consistent with the textual instruction R^j in this reasoning step. Third, because the synthesis capability of a world model is inherently bounded (e.g. it may become unreliable when extrapolating far beyond the region covered by the observed views) we introduce the **state validity** c_t^{state}, which measures whether the resulting latent state remains within the reliable operating range of the world model.

Rather than adding these three terms independently, we use action and state validity as multiplicative gates [29; 30] on progress:

$$r_t^{\text{step}} = c_t^{\text{action}} c_t^{\text{state}} \Delta V_t. \tag{7}$$

This design prevents the policy from earning process reward merely by selecting easy-to-execute actions or producing plausible-looking states. High reward is obtained only when a step simultaneously makes useful progress, correctly executes the intended transformation, and produces a reliable world-model state. It therefore discourages degenerate behaviors such as repeatedly taking safe but uninformative actions or exploiting hallucinated latent states that spuriously increase answer confidence.

The overall trajectory reward is then

$$R(\tau) = \lambda_{\text{ans}} r_{\text{ans}} + \lambda_{\text{fmt}} r_{\text{fmt}} + \sum_{t=1}^{T} r_t^{\text{step}} - \lambda_c T, \tag{8}$$

where r_{ans} evaluates final-answer correctness, r_{fmt} enforces the required output format, and $\lambda_c T$ penalizes unnecessary world-model calls.

In practice, we compute c_t^{latent} by measuring the similarity between the world-model latent generated after executing A_t and the latent representation of the corresponding ground-truth target view.

For synthetic 3D scenes, we render the ground-truth observation O_t^{GT} at the same target camera pose and extract its latent representation H_t^{GT} using the same world-model layer and diffusion timestep as the generated latent H_t^w. Since the two representations lie in the same latent space and share the same spatial organization, we measure their consistency using the average cosine similarity between spatially corresponding latent tokens. Specifically, let $\widetilde{H}_t^w = \{h_{t,i}^w\}_{i=1}^N$ and $\widetilde{H}_t^{\mathrm{GT}} = \{h_{t,i}^{\mathrm{GT}}\}_{i=1}^N$. We define

$$c_t^{\mathrm{state}} = \frac{1}{N}\sum_{i=1}^{N} \frac{1+\cos\left(h_{t,i}^w, h_{t,i}^{\mathrm{GT}}\right)}{2}, \tag{9}$$

where the cosine similarity is rescaled from $[-1,1]$ to $[0,1]$. A high c_t^{latent} indicates that the generated latent remains consistent with the ground-truth scene state at the requested viewpoint, whereas a low value suggests that the transformation exceeds the reliable operating range of the world model and may introduce hallucinated or spatially inconsistent content.

4 EXPERIMENTS

Evaluation Benchmarks. We evaluate WorldMind on multiple benchmarks that include complementary aspects of visual spatial reasoning, such as metric understanding, relative spatial relations, perspective taking, multi-view reasoning, and navigation-oriented reasoning. These aspects widely cover video-based spatial understanding, viewpoint transformation and compositional reasoning, allowing us to assess where latent world simulation provides the largest benefit.

- **VSI-Bench** [62] evaluates video-based spatial reasoning in indoor environments across diverse task categories, including object counting, absolute and relative distance, object and room size, relative direction, route planning, and appearance order. Following the official evaluation protocol, we report Accuracy for multiple-choice questions and Mean Relative Accuracy (MRA) for numerical questions.
- **MindCube** [50] evaluates spatial mental modeling from limited multi-view observations. It contains three settings (Rotation, Among, and Around)that test whether a model can construct and manipulate an internal representation of a scene from only a few views. We report accuracy for each setting together with the overall score.
- **MMSI-Bench** [47] provides a broader evaluation of multimodal spatial intelligence across 11 task categories, covering camera–camera, camera–object, camera–region, object–object, object–region, and region–region relations, as well as measurement, appearance, camera motion, object motion, and multi-step reasoning. We report per-category accuracy together with the overall benchmark score.

Evaluation Baselines. We compare WorldMind with three groups of competitive baselines:

- **General-purpose VLMs.** We use Qwen3-VL-8B-Instruct [1], LLaVA-OneVision-7B [21], and LLaVA-Video-7B [56] as general-purpose multimodal baselines. They directly reason over the observed visual inputs without explicit 3D representations or world model simulation.
- **Spatial-Aware VLMs.** We compare WorldMind against VLM-3R-7B [10] and Cambrian-S-7B [48], which enhance general-purpose VLMs with explicit geometric or spatial representations. These baselines test whether directly injecting 3D-aware features is sufficient for spatial reasoning, without actively transforming the represented scene.
- **VLMs with world-model simulation.** We further compare with SpatialDreamer 4, ASTRA 62 and AVIC 51, which couple VLMs with world models to acquire additional observations through active visual imagination. Unlike WorldMind, these methods primarily return synthesized RGB pixel views to the VLM, providing the closest comparison against WorldMind's reasoning directly from latent world simulation.

Training Data. WorldMind is trained on a mixed collection of data from multiple sources. Besides using multiple real-world datasets such as VSI-590K 48, SPAR-234K 53, SpatialMosaic 20, and SAT-Real 33, we additionally incorporate synthetic datasets, including SAT-Synthesized 33 in AI2-THOR 19 and Infinibench 32; 40 scenes, for trajectory construction and reinforcement learning, to provide ground-truth novel views and scene states at the target camera poses in world simulation. For latent alignment, we sample a total of 40,000 examples across the different real-world data sources. We then construct 10,000 synthetic reasoning trajectories for supervised fine-tuning using GPT-5.6 Sol as the teacher model. For reinforcement learning, we sample 5,000 questions and allow the SFT-initialized policy to generate its own on-policy reasoning and simulation trajectories. No samples from the evaluation splits of the benchmarks reported in the main results are used in training. Full details of data construction, trajectory filtering, and leakage prevention, as well as examples of data samples, are provided in Appendix A.

Method	Obj. Count	Abs. Dist.	Rel. Dist.	Obj. Size	Room Size	Rel. Dir.	Route Plan.	App. Order	Avg.
LLaVA-Video-7B	48.5	14.0	43.5	47.8	24.2	42.4	34.0	30.6	35.6
LLaVA-OneVision-7B	47.7	20.2	42.5	47.4	12.3	35.2	29.4	24.4	32.4
Qwen3-VL-8B-Instruct	66.7	38.9	59.0	74.1	53.5	65.4	32.0	69.4	57.4
VLM-3R-7B	70.2	49.4	65.4	69.2	67.1	80.5	45.4	40.1	60.9
Cambrian-S-7B	73.2	50.5	71.1	74.9	**72.2**	76.2	41.8	**80.1**	67.5
SpatialDreamer	71.1	48.1	67.8	67.0	65.9	86.4	47.5	37.2	62.2
AVIC	72.7	49.8	65.3	74.8	57.9	81.2	37.9	63.8	66.6
WorldMind (SFT)	**73.2**	**59.5**	**72.9**	74.9	65.3	**86.7**	**50.6**	76.2	**67.9**
WorldMind (SFT+RL)	71.4	**62.3**	**73.2**	**75.7**	66.8	**87.1**	**55.4**	77.1	**71.1**

Table 1: Comparison of task accuracy with the existing methods on the VSI-Bench benchmark. "Obj." indicates Objects, "Abs. Dist." indicates Absolute Distance, "Rel. Dist." indicates Relative Distance, "Rel. Dir." indicates Relative Direction, "Route Plan." indicates Route Planning, and "App. Order" indicates Appearance Order.

Model	MMSI-Bench					MindCube-Tiny			
	PR.	Attr.	Mot.	MSR	All	Rot.	Ard.	Amg.	All
LLaVA-NeXT-Video-7B	23.5	30.0	21.3	27.8	25.7	35.7	30.1	43.6	36.5
LLaVA-OneVision-7B	28.0	27.7	26.7	11.6	23.5	36.5	44.1	48.4	43.0
Qwen3-VL-8B-Instruct	33.7	26.2	28.0	29.8	29.4	53.6	38.0	31.1	40.9
VLM-3R-7B	33.0	31.5	32.1	30.3	31.7	36.7	24.5	44.2	35.1
Cambrian-S-7B	29.5	28.4	21.4	24.2	25.9	33.0	39.2	39.0	37.1
Astra (Qwen3-VL-8B-Instruct)	42.3	41.0	32.1	33.6	37.3	60.1	43.5	36.8	46.8
AVIC	31.1	**45.4**	32.0	28.2	34.2	36.8	40.5	38.4	38.6
WorldMind (SFT)	**44.5**	44.9	**35.6**	**37.7**	40.7	**65.2**	**46.1**	48.1	53.1
WorldMind (SFT+RL)	**47.7**	42.8	**40.9**	**42.2**	**43.4**	**71.6**	**50.3**	**55.8**	**59.2**

Table 2: Comparison of task success rate with the existing methods on the MMSI-Bench and MindCube benchmarks. For MMSI-Bench, "PR." indicates Positional Relationship, "Attr." indicates Attribute, "Mot." indicates Motion, and "MSR" indicates Multi-Step Reasoning. For MindCube, "Rot." indicates Rotation, "Ard." indicates Around, and "Amg." indicates Among. "All" denotes the unweighted mean across the reported task categories. Higher is better (↑).

Implementation. We use SVC [60] as the frozen world model. The latent features of world model are projected into the VLM embedding space using a lightweight Transformer adapter, while LoRA modules are added to the VLM during SFT and reinforcement learning. We extract candidate features from four world-model layers and four denoising timesteps, and select the top-8 layer-timestep pairs offline using feature ablation. The adapter is first aligned for 1 epochs, followed by 3 epochs of simulated-trajectory SFT. We then perform GRPO initialized from the SFT checkpoint with 8 rollouts per question. All experiments use BF16 precision. Full architectural and optimization details are provided in Appendix B.

4.1 Main Results

Tables 1 and 2 summarize the main experiment results on the three spatial benchmarks. Overall, WorldMind consistently improves over the underlying VLM backbone and remains competitive with or outperforms specialized spatial-reasoning baselines. In particular, WorldMind (SFT+RL) achieves an average improvement of 16.0% over the base VLM and 8.3% over the strongest RGB-based world-model baseline across the evaluated benchmarks.

The gains are especially pronounced on tasks that require reasoning under transformed viewpoints, multi-step spatial relations, or camera/object motion. This trend is consistent with the design of WorldMind, which explicitly acquires additional spatial evidence through latent world simulation rather than relying only on the original observations. By contrast, the improvements are relatively smaller on tasks that can be solved directly from the visible scene, such as Object Counting and Object Size, suggesting that the main benefit of WorldMind comes from reasoning about spatial states that are not explicitly observed.

Comparison with the two WorldMind variants further shows that simulation-trajectory SFT already provides substantial gains by teaching the model how to interact with the world model and interpret

latent states, while process-verifiable reinforcement learning provides an additional improvement of 4.0%. This indicates that optimizing the policy on its own trajectories helps it select more useful camera transformations and avoid unreliable or unnecessary simulations.

4.2 Ablation Studies

Effect of latent world simulation. We first isolate the contribution of latent world simulation by comparing the base VLM using only observed RGB frames, additional linguistic test-time reasoning, RGB-based world simulation, and WorldMind with latent world simulation. As shown in Tab. 3, linguistic scaling provides only limited and inconsistent benefits. In contrast, RGB-based world simulation substantially improves performance across all three benchmarks. Replacing RGB outputs with directly exposed world-model latents yields further gains, improving the average performance from 48.5 to 50.0 and supporting the benefit of avoiding the latent-to-pixel-to-latent reasoning bottleneck. These results are from a small-scale experiment and therefore differ from the full-scale results in the main comparison.

Method	VSI-Bench	MMSI-Bench	Mindcube
Base VLM	57.4	29.4	40.9
+ Linguistic Scaling	55.9	29.7	40.7
+ RGB Simulation	63.4	33.6	48.4
+ Latent Simulation	**65.2**	**36.2**	**48.5**

Table 3: Ablation of the reasoning interface.

Latent feature selection. Tab. 4 examines which world-model representations are most useful for spatial reasoning. We conduct a small-scale experiment on VSI-Bench using partial latent alignment and SFT, without reinforcement learning. We compare a single layer–timestep feature, randomly selected features, the complete feature bank, and our offline selected subset. Our selected features achieve **60.4**, compared with **60.3** for the full feature bank, indicating that a compact subset of informative representations is sufficient while avoiding redundant or less task-relevant features. In contrast, randomly selecting four features degrades performance to 57.8, further highlighting the importance of selecting spatially informative layer–timestep representations rather than i

Selection	K	Task success rate
Single feature	1	59.8
Random	4	57.8
Full feature bank	All	60.3
Ours	8	**60.4**

Table 4: Ablation of latent feature selection on the VSI-Bench benchmark

Reward design. Tab. 5 evaluates the components of the process-verifiable reward. Starting from terminal answer supervision alone, we progressively introduce rollout-derived progress ΔV_t, action validity c_t^{action}, and latent validity c_t^{latent}. Progress supervision provides a modest improvement from **68.1** to **68.3**, while additionally incorporating action validity further improves performance to **68.9**. Latent validity alone does not improve over progress supervision, yielding 68.0; however, combining both validity terms with progress substantially increases performance to **70.2**. The full formulation therefore performs best, indicating that action and latent validity provide complementary signals and that useful intermediate progress should be credited only when the intended transformation is successfully realized and the resulting world-model latent remains reliable.

Reward	ΔV	c^{action}	c^{state}	Success rate
Final answer only	–	–	–	68.1
+ Progress	✓	–	–	68.3
+ Action validity	✓	✓	–	68.9
+ Latent validity	✓	–	✓	68.0
Full	✓	✓	✓	**70.2**

Table 5: Ablation of the process reward on VSI-Bench

5 Conclusion

We introduce **WorldMind**, a framework for test-time spatial reasoning via latent world simulation. By directly conditioning the VLM on world-model latents, it enables iterative spatial transformation without costly RGB conversion. Experiments show consistent gains, particularly on viewpoint-dependent and multi-step reasoning tasks.

AI USE STATEMENT

In this work, we used generative AI tools for Generate synthetic data sets, implement methods, clean and reformat dataset.

We have not used generative AI tools for help develop theoretical models or conceptual frameworks, design or provide feedback on research methodology or experiments, support qualitative and thematic data analysis, interpret results.

And formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypothese, assist with translation are not applicable to this work.

Additionally, we used generative AI tools for suggest experimental parameters, summarize or analyse existing literature, edit a research paper to improve readability.

We have reviewed all AI-assisted work. We checked all the AI editted text to make sure it's consistent with our implementation. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

[1] S. Bai, Y. Cai, R. Chen, K. Chen, X. Chen, Z. Cheng, L. Deng, W. Ding, C. Gao, C. Ge, W. Ge, Z. Guo, Q. Huang, J. Huang, F. Huang, B. Hui, S. Jiang, Z. Li, M. Li, M. Li, K. Li, Z. Lin, J. Lin, X. Liu, J. Liu, C. Liu, Y. Liu, D. Liu, S. Liu, D. Lu, R. Luo, C. Lv, R. Men, L. Meng, X. Ren, X. Ren, S. Song, Y. Sun, J. Tang, J. Tu, J. Wan, P. Wang, P. Wang, Q. Wang, Y. Wang, T. Xie, Y. Xu, H. Xu, J. Xu, Z. Yang, M. Yang, J. Yang, A. Yang, B. Yu, F. Zhang, H. Zhang, X. Zhang, B. Zheng, H. Zhong, J. Zhou, F. Zhou, J. Zhou, Y. Zhu, and K. Zhu. Qwen3-vl technical report, 2025.

[2] W. Cai, I. Ponomarenko, J. Yuan, X. Li, W. Yang, H. Dong, and B. Zhao. Spatialbot: Precise spatial understanding with vision language models. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9490–9498. IEEE, 2025.

[3] Z. Cai, R. Wang, C. Gu, F. Pu, J. Xu, Y. Wang, W. Yin, Z. Yang, C. Wei, T. Zhou, et al. Scaling spatial intelligence with multimodal foundation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7879–7890, 2026.

[4] M. Cao, X. Li, X. Liu, I. Reid, and X. Liang. Spatialdreamer: Incentivizing spatial reasoning via active mental imagery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7176–7187, 2026.

[5] B. Chen, Z. Xu, S. Kirmani, B. Ichter, D. Sadigh, L. Guibas, and F. Xia. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14455–14465, 2024.

[6] S. Chen, T. Zhu, R. Zhou, J. Zhang, S. Gao, J. C. Niebles, M. Geva, J. He, J. Wu, and M. Li. Why is spatial reasoning hard for vlms? an attention mechanism perspective on focus areas. *arXiv preprint arXiv:2503.01773*, 2025.

[7] A.-C. Cheng, Y. Fu, Y. Chen, Z. Liu, X. Li, S. Radhakrishnan, S. Han, Y. Lu, J. Kautz, P. Molchanov, et al. 3d aware region prompted vision language model. In *International Conference on Learning Representations*, volume 2026, pages 123286–123308, 2026.

[8] M. Deitke, C. Clark, S. Lee, R. Tripathi, Y. Yang, J. S. Park, M. Salehi, N. Muennighoff, K. Lo, L. Soldaini, et al. Molmo and pixmo: Open weights and open data for state-of-the-art vision-language models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 91–104. IEEE, 2025.

[9] A. Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[10] Z. Fan, J. Zhang, R. Li, J. Zhang, R. Chen, H. Hu, K. Wang, P. Wang, H. Qu, S. Zhou, et al. Vlm-3r: Vision-language models augmented with instruction-aligned 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 31054–31065, 2026.

[11] R. Girdhar, A. El-Nouby, Z. Liu, M. Singh, K. V. Alwala, A. Joulin, and I. Misra. Imagebind one embedding space to bind them all. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15180–15190. IEEE, 2023.

[12] J. Guo, Y. Chen, X. Wang, S. Li, X. Zhao, H. Liu, S. Dong, F. Zhai, and Y. Zhou. Beyond visual memory: Mechanistic diagnostics of latent visual reasoning. *arXiv preprint arXiv:2606.01287*, 2026.

[13] M. Hegarty and D. Waller. A dissociation between mental rotation and perspective-taking spatial abilities. *Intelligence*, 32(2):175–191, 2004.

[14] Z. Huang, X. Li, Z. Lv, and J. M. Rehg. How much 3d do video foundation models encode? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 384–394, 2026.

[15] J. Huttenlocher and C. C. Presson. The coding and transformation of spatial information. *Cognitive psychology*, 11(3):375–394, 1979.

[16] B. Ji, S. Agrawal, Q. Tang, and Y. Wu. Enhancing spatial reasoning in vision-language models via chain-of-thought prompting and reinforcement learning. *arXiv preprint arXiv:2507.13362*, 2025.

[17] S. S. Kancheti, A. S. Kanade, V. N. Balasubramanian, and T. Ganu. Chain-of-thought degrades visual spatial reasoning capabilities of multimodal llms. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 862–876, 2026.

[18] Y. Kim, J. Kim, Y. Jeong, and S.-B. Cho. Diagnosing spatial consistency across perspectives and viewpoints in large vision-language models. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 32803–32827, 2026.

[19] E. Kolve, R. Mottaghi, W. Han, E. VanderBilt, L. Weihs, A. Herrasti, M. Deitke, K. Ehsani, D. Gordon, Y. Zhu, et al. Ai2-thor: An interactive 3d environment for visual ai. *arXiv preprint arXiv:1712.05474*, 2017.

[20] K. Lee, J. Hong, S. Lee, I. Lee, M. Kwak, K. Ryu, and J. Park. Spatialmosaic: A multiview vlm dataset for partial visibility. *arXiv preprint arXiv:2512.23365*, 2025.

[21] B. Li, Y. Zhang, D. Guo, R. Zhang, F. Li, H. Zhang, K. Zhang, P. Zhang, Y. Li, Z. Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024.

[22] Y. Li, C. Chen, Y. Li, F. Zeng, K. Huang, J. Xu, and M. Sun. Imagination helps visual reasoning, but not yet in latent space. *arXiv preprint arXiv:2602.22766*, 2026.

[23] X. Liang, X. Guo, Z. Jin, W. Pan, P. Shang, D. Cai, B. Lin, and J. Ye. Enhancing spatial reasoning through visual and textual thinking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 23433–23441, 2026.

[24] H. Liu, C. Li, Y. Li, and Y. J. Lee. Improved baselines with visual instruction tuning. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 26286–26296. IEEE, 2024.

[25] W. Liu, Q. Xue, H. Wang, X. Yin, B. Yang, and W. Gao. Spatial reasoning in multimodal large language models: A survey of tasks, benchmarks and methods. *arXiv preprint arXiv:2511.15722*, 2025.

[26] G. Luo, L. Dunlap, D. H. Park, A. Holynski, and T. Darrell. Diffusion hyperfeatures: Searching through time and space for semantic correspondence. *Advances in Neural Information Processing Systems*, 36:47500–47510, 2023.

[27] X. Ma, S. Yong, Z. Zheng, Q. Li, Y. Liang, S.-C. Zhu, and S. Huang. Sqa3d: Situated question answering in 3d scenes. *arXiv preprint arXiv:2210.07474*, 2022.

[28] Y. Man, S. Zheng, Z. Bao, M. Hebert, L.-Y. Gui, and Y.-X. Wang. Lexicon3d: Probing visual foundation models for complex 3d scene understanding. *Advances in Neural Information Processing Systems*, 37:76819–76847, 2024.

[29] F. Pan, J. He, D. Tu, and Q. He. Trust the model when it is confident: Masked model-based actor-critic. *Advances in neural information processing systems*, 33:10537–10546, 2020.

[30] Y.-J. Park, K. Greenewald, K. Alimohammadi, H. Wang, and N. Azizan. Know what you don't know: Uncertainty calibration of process reward models. *Advances in neural information processing systems*, 38:38852–38895, 2026.

[31] J. Piekenbrinck, A. Hermans, N. Vaskevicius, T. Linder, and B. Leibe. Rgb-d cube r-cnn: 3d object detection with selective modality dropout. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1997–2006. IEEE, 2024.

[32] A. Raistrick, L. Mei, K. Kayan, D. Yan, Y. Zuo, B. Han, H. Wen, M. Parakh, S. Alexandropoulos, L. Lipson, et al. Infinigen indoors: Photorealistic indoor scenes using procedural generation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21783–21794. IEEE, 2024.

[33] A. Ray, J. Duan, E. Brown, R. Tan, D. Bashkirova, R. Hendrix, K. Ehsani, A. Kembhavi, B. A. Plummer, R. Krishna, et al. Sat: Dynamic spatial aptitude training for multimodal language models. *arXiv preprint arXiv:2412.07755*, 2024.

[34] X. Ren, T. Shen, J. Huang, H. Ling, Y. Lu, M. Nimier-David, T. Müller, A. Keller, S. Fidler, and J. Gao. Gen3c: 3d-informed world-consistent video generation with precise camera control. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6121–6132. IEEE, 2025.

[35] A. Setlur, C. Nagpal, A. Fisch, X. Geng, J. Eisenstein, R. Agarwal, A. Agarwal, J. Berant, and A. Kumar. Rewarding progress: Scaling automated process verifiers for llm reasoning. In *International Conference on Learning Representations*, volume 2025, pages 60808–60838, 2025.

[36] R. N. Shepard and J. Metzler. Mental rotation of three-dimensional objects. *Science*, 171 (3972):701–703, 1971.

[37] C. Snell, J. Lee, K. Xu, and A. Kumar. Scaling llm test-time compute optimally can be more effective than scaling parameters for reasoning. In *International Conference on Learning Representations*, volume 2025, pages 10131–10165, 2025.

[38] L. Tang, M. Jia, Q. Wang, C. P. Phoo, and B. Hariharan. Emergent correspondence from image diffusion. *Advances in neural information processing systems*, 36:1363–1389, 2023.

[39] J. Uesato, N. Kushman, R. Kumar, F. Song, N. Siegel, L. Wang, A. Creswell, G. Irving, and I. Higgins. Solving math word problems with process-and outcome-based feedback. *arXiv preprint arXiv:2211.14275*, 2022.

[40] H. Wang, Q. Xue, and W. Gao. Infinibench: Infinite benchmarking for visual spatial reasoning with customizable scene complexity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21594–21605, 2026.

[41] R. Wang, Z. Cai, F. Pu, J. Xu, W. Yin, M. Wang, R. Ji, C. Gu, B. Li, Z. Huang, et al. Demystifying video reasoning. *arXiv preprint arXiv:2603.16870*, 2026.

[42] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[43] L. Xue, M. Gao, C. Xing, R. Martín-Martín, J. Wu, C. Xiong, R. Xu, J. C. Niebles, and S. Savarese. Ulip: Learning a unified representation of language, images, and point clouds for 3d understanding. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1179–1189. IEEE, 2023.

[44] Q. Xue, H. Wang, W. Liu, S. Wang, Y. Wu, and W. Gao. Reasoning path and latent state analysis for multi-view visual spatial reasoning: A cognitive science perspective. In *European Conference on Computer Vision*, pages 512–531. Springer, 2026.

[45] J. Yang, S. Yang, A. W. Gupta, R. Han, L. Fei-Fei, and S. Xie. Thinking in space: How multimodal large language models see, remember, and recall spaces. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10632–10643. IEEE, 2025.

[46] J. Yang, W. Huang, J. Zhang, M. Hu, and H. Guo. In-context vla: Endowing vision-language-action models with language via in-context post-training and agentic tool use. *arXiv preprint arXiv:2608.05738*, 2026.

[47] S. Yang, R. Xu, Y. Xie, S. Yang, M. Li, J. Lin, C. Zhu, X. Chen, H. Duan, X. Yue, et al. Mmsi-bench: A benchmark for multi-image spatial intelligence. In *International Conference on Learning Representations*, volume 2026, pages 157051–157088, 2026.

[48] S. Yang, J. Yang, P. Huang, E. Brown, Z. Yang, Y. Yu, S. Tong, Z. Zheng, Y. Xu, M. Wang, et al. Cambrian-s: Towards spatial supersensing in video. In *International Conference on Learning Representations*, volume 2026, pages 78185–78225, 2026.

[49] Y. Yang, J. Liu, Z. Zhang, S. Zhou, R. Tan, J. Yang, Y. Du, and C. Gan. Mindjourney: Test-time scaling with world models for spatial reasoning. *Advances in Neural Information Processing Systems*, 38:109855–109885, 2026.

[50] B. Yin, Q. Wang, P. Zhang, J. Zhang, K. Wang, Z. Wang, J. Zhang, K. Chandrasegaran, H. Liu, R. Krishna, et al. Spatial mental modeling from limited views. In *Structural Priors for Vision Workshop at ICCV'25*, 2025.

[51] S. Yu, Y. Zhang, Z. Wang, J. Yoon, H. Yao, M. Ding, and M. Bansal. When and how much to imagine: Adaptive test-time scaling with world models for visual spatial reasoning. *arXiv preprint arXiv:2602.08236*, 2026.

[52] H. Zhang, C. Li, W. Wu, S. Mao, I. Vulić, Z. Zhang, L. Wang, T. Tan, F. Wei, et al. A call for new recipes to enhance spatial reasoning in mllms. *arXiv e-prints*, pages arXiv–2504, 2025.

[53] J. Zhang, Y. Chen, Y. Xu, Z. Huang, J. Mei, C. Chen, Y. Zhou, Y.-J. Yuan, X. Cai, G. Huang, et al. From flatland to space: Teaching vision-language models to perceive and reason in 3d. *Advances in Neural Information Processing Systems*, 38, 2026.

[54] K. Zhang, K. Ge, X. Chi, R. Zhang, S. Shi, Z. Dong, S. Han, and S. Zhang. Can world models benefit vlms for world dynamics? *arXiv preprint arXiv:2510.00855*, 2025.

[55] Y. Zhang, J. Xing, E. Lo, and J. Jia. Real-world image variation by aligning diffusion inversion chain. *Advances in Neural Information Processing Systems*, 36:30641–30661, 2023.

[56] Y. Zhang, J. Wu, W. Li, B. Li, Z. Ma, Z. Liu, and C. Li. Llava-video: Video instruction tuning with synthetic data. *arXiv preprint arXiv:2410.02713*, 2024.

[57] Y. Zhang, C. Cao, T. Wang, X. Zuo, J. Wu, J. Zhu, and C. Guo. Worldstereo: Bridging camera-guided video generation and scene reconstruction via 3d geometric memories. *arXiv preprint arXiv:2603.02049*, 2026.

[58] Y. Zhang, B. Wang, J. Liu, N. Jiang, J. Jiang, H. Sun, Y. Yang, S. Zheng, L. Song, H. Huang, et al. Thinking with novel views: A systematic analysis of generative-augmented spatial intelligence. *arXiv preprint arXiv:2605.10588*, 2026.

[59] Z. Zhang, Y. Wu, L. Jia, Y. Wang, Z. Zhang, Y. Li, B. Ran, F. Zhang, Z. Sun, Y. Peng, et al. Think3d: Thinking with space for spatial reasoning. *arXiv preprint arXiv:2601.13029*, 2026.

[60] J. Zhou, H. Gao, V. Voleti, A. Vasishta, C.-H. Yao, M. Boss, P. Torr, C. Rupprecht, and V. Jampani. Stable virtual camera: Generative view synthesis with diffusion models. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12405–12414. IEEE, 2025.

[61] Z. Zhou, T. Cai, S. Zhao, Y. Zhang, Z. Huang, B. Zhou, and J. Ma. Autovla: A vision-language-action model for end-to-end autonomous driving with adaptive reasoning and reinforcement fine-tuning. *Advances in Neural Information Processing Systems*, 38:27920–27956, 2026.

[62] C. Zhu, J. Lin, Y. Long, P. Cao, T. Wang, J. Pang, and X. Liu. Thinking with imagination: Agentic visual spatial reasoning with world simulators. *arXiv preprint arXiv:2606.06476*, 2026.

A APPENDIX

A ADDITIONAL EXPERIMENTAL DETAILS

A.1 BASELINE SELECTION AND EVALUATION PROTOCOL

A large number of recent methods have been proposed for visual spatial reasoning, but not all of them can be evaluated consistently on the three benchmarks considered in this work. In particular, some methods report results on only a subset of VSI-Bench, MMSI-Bench, and MindCube, while others do not release trained checkpoints or executable evaluation code. We therefore include a baseline if it satisfies at least one of the following criteria:

1. the original paper reports sufficiently complete results on the benchmarks considered in our evaluation;
2. the authors release trained model weights that can be directly evaluated on the missing benchmarks; or
3. the authors release source code that allows us to reproduce performance comparable to the results reported in the original paper and evaluate the method under our benchmark protocol.

Among methods satisfying these requirements, we prioritize strong and recent baselines. For fair comparison with our Qwen3-VL-8B-Instruct backbone, we primarily consider models with approximately 7–8B parameters. We divide them into three groups: general-purpose VLMs, spatial-aware VLMs that incorporate explicit geometric representations, and VLMs augmented with world-model simulation. The last group is most closely related to WorldMind. Since considerably fewer methods currently belong to this category, we relax the model-size and result-availability criteria when necessary to provide representative comparisons.

When published results are available under the same benchmark protocol, we use the reported numbers. Otherwise, we evaluate the released checkpoint or implementation using the corresponding official benchmark protocol. We do not mix results obtained from different variants of a benchmark. For the aggregate columns reported in our tables, we use the same convention across methods: the VSI-Bench score is averaged across the reported task categories, the MMSI-Bench "All" score is the unweighted mean of PR., Attr., Mot., and MSR, and the MindCube "All" score is the unweighted mean of Rotation, Around, and Among.

A.2 NOTATION

Table 6 summarizes the notation used throughout the paper. To avoid overloading N, we use N_{obs} for the number of observed RGB views and M for the number of spatial latent tokens in this appendix.

A.3 CHOICE OF WORLD MODEL

WorldMind requires a world model that serves not only as a novel-view generator but also as a source of intermediate spatial representations. We therefore consider four properties when choosing the world-model backbone.

First, the model should support conditioning on the currently observed views and explicit camera transformations, since WorldMind uses camera motion as the action space for spatial simulation. Second, it should support sufficiently general target viewpoints rather than being limited to a small discrete action vocabulary. Third, intermediate features from multiple network layers and denoising timesteps should be directly accessible, allowing us to study and select spatially informative latent representations. Finally, we prefer a model that does not depend on an externally reconstructed persistent 3D representation, since otherwise it becomes difficult to distinguish the contribution of world-model latents from the contribution of explicitly injected 3D geometry.

Following these criteria, we use Stable Virtual Camera (SVC) [60] as the frozen world model. SVC provides a convenient interface between posed visual observations and target camera viewpoints, while its diffusion architecture exposes intermediate representations over both transformer layers and denoising timesteps. This directly matches our latent feature bank $\mathcal{F} = \{h^{(\ell,\tau)}\}$ and allows WorldMind to execute a camera transformation while obtaining the corresponding latent representation without first decoding the simulated state into RGB pixels.

Symbol	Meaning
I_i	The i-th observed RGB frame.
N_{obs}	Number of initially observed RGB frames.
Q	Spatial question given to the VLM.
H_i^{rgb}	Native VLM visual representation of observation I_i.
Z_i	World-model latent associated with observation I_i.
$h^{(\ell,\tau)}$	Intermediate world-model feature from layer ℓ and denoising timestep τ.
$\mathcal{F}$	Candidate bank of world-model layer–timestep features.
$\mathcal{S}^*$	Offline-selected subset of spatially informative layer–timestep features.
$\mathcal{P}_w$	Learnable adapter mapping world-model features into the VLM representation space.
H_i^w	Projected world-model representation of observation I_i.
X_0	Initial multimodal reasoning context.
C_t	Complete VLM context available before reasoning step t.
R_t	Textual reasoning generated at step t.
A_t	Camera transformation selected at step t.
$\mathcal{W}$	Frozen world model.
$\widetilde{Z}_t$	World-model latent produced after executing A_t.
$\widetilde{H}_t^w$	Projected latent observation returned to the VLM after step t.
S_t	World-model state after the t-th simulation step.
T	Number of world-model interactions in a reasoning trajectory.
J	Final textual justification.
Y	Final answer.
p_{drop}	Probability of using latent-only input during modality dropout.
$\mathcal{L}_{\text{ans}}$	Functional answer-supervision loss used for latent alignment.
$\mathcal{L}_{\text{compat}}$	Weak RGB–latent representation compatibility loss.
$\mathcal{L}_{\text{reason}}$	SFT loss over intermediate reasoning tokens.
$\mathcal{L}_{\text{action}}$	SFT loss over camera actions.
$\mathcal{L}_{\text{final}}$	SFT loss over the final justification and answer.
π_θ	WorldMind reasoning and action policy.
$V(C_t)$	Estimated answer value given reasoning context C_t.
ΔV_t	Step-wise reasoning progress $V(C_{t+1}) - V(C_t)$.
c_t^{action}	Validity of the executed camera transformation at step t.
c_t^{latent}	Reliability of the simulated world-model latent at step t.
r_t^{step}	Reliability-gated process reward at step t.
r_{ans}	Terminal answer-correctness reward.
r_{fmt}	Output-format reward.
λ_c	Cost assigned to each world-model call.
H_t^{GT}	World-model representation extracted from the ground-truth target view.
M	Number of spatial latent tokens used in the latent-validity computation.

Table 6: Summary of mathematical notation used in WorldMind.

Interactive video world models provide an alternative interface based on long autoregressive action sequences. However, their standard inference interfaces typically map an initial observation and an action trajectory to a video chunk rather than mapping a set of current observations and a single target camera transformation to one simulated spatial state. Adopting such models would therefore require modifying the simulation interface and temporal aggregation used by WorldMind. We leave this extension to future work.

A.4 Observed and Simulated World-Model Representations

WorldMind uses intermediate representations from the same world-model architecture for both observed and simulated views. For an observed RGB view, we encode the image into the world-model latent space, apply the noise level corresponding to timestep τ, and extract the hidden representation $h_{\text{obs}}^{(\ell,\tau)}$ from transformer layer ℓ. For a newly simulated view, we directly record $h_{\text{sim}}^{(\ell,\tau)}$ from the same layer and timestep during world-model denoising.

Using the same (ℓ, τ) ensures that the two representations have the same architecture, dimensionality, token organization, and representational role. However, we do *not* assume that observed and simulated representations are identically distributed. The former originates from a real observation injected into the diffusion process, whereas the latter lies on the model's own generative trajec-

tory. Distribution differences between representations associated with real and generated samples are expected in diffusion models [26; 38; 55].

Our alignment design is intended to tolerate this mismatch rather than explicitly forcing the two distributions to coincide. In particular, the adapter is trained through downstream answer supervision instead of token-level reconstruction, modality dropout forces the VLM to make functional use of the latent representation, and the weak compatibility objective operates only at the pooled semantic level. During simulation-trajectory SFT, the model is additionally exposed to world-model-generated latent states, further reducing the gap between alignment-time representations and those encountered during iterative simulation.

This distinction is also important for the latent-validity reward. Comparing a simulated latent with the latent extracted from a ground-truth target view does not assume that the two distributions are identical. Instead, their shared layer–timestep representation provides a common coordinate system in which spatial consistency can be measured.

B Training and Implementation Details

WorldMind is trained in three stages: latent feature selection and alignment, simulation-trajectory supervised fine-tuning, and process-verifiable reinforcement learning. In this appendix, we describe the data construction and optimization details of each stage together to avoid repeating the same training procedure across separate sections.

B.1 Data Sources and Separation

WorldMind uses a mixture of real-world and synthetic spatial-reasoning data. For real-world supervision, we use VSI-590K [48], SPAR-234K [53], SpatialMosaic [20], and SAT-Real [33]. These datasets provide complementary supervision covering metric relations, multi-view reasoning, relative direction, partial visibility, and viewpoint-dependent spatial reasoning.

We additionally use synthetic scenes from SAT-Synthesized [33] in AI2-THOR [19] and Infinibench [32; 40]. Unlike real-world datasets, these environments expose the underlying scene state and camera parameters, allowing us to render the ground-truth observation at arbitrary target camera poses. We primarily use these synthetic scenes for trajectory construction and reinforcement learning, where action execution and simulated-state reliability need to be verified.

No samples from the evaluation splits of VSI-Bench, MMSI-Bench, or MindCube are used during any training stage. When a training source and an evaluation benchmark originate from the same underlying scene collection, we separate them using scene identifiers whenever available. We additionally apply image-level and question-level duplicate filtering. Latent feature selection and hyperparameter tuning are performed only on training or held-out training samples.

B.2 Stage I: Latent Feature Selection and Alignment

Training data. For latent alignment, we sample 40,000 examples from the real-world training sources described above. Each example contains the observed RGB frames, a spatial question, and its target answer. We reserve 5% of these samples as a held-out split for latent feature selection and hyperparameter tuning.

World-model feature extraction. We use Qwen3-VL-8B-Instruct [1] as the VLM backbone and Stable Virtual Camera (SVC) [60] as the frozen world model. We extract intermediate SVC representations from four transformer layers and four denoising timesteps, producing a candidate bank of

$$16 \times 16 = 256$$

layer–timestep features.

For an observed frame, we encode the frame into the SVC latent space and extract

$$h_{\text{obs}}^{(\ell,\tau)}$$

from layer ℓ at denoising timestep τ. For a simulated target view, we directly record

$$h_{\text{sim}}^{(\ell,\tau)}$$

at the same layer and timestep during SVC generation. Thus, simulated latent evidence can be directly passed to the VLM without decoding the new state into RGB and subsequently re-encoding it through the VLM visual encoder.

Offline feature selection. We first train a temporary fusion adapter using all 16 candidate representations. We then remove each layer–timestep representation in turn and measure the resulting performance degradation on the held-out alignment split. Candidates are ranked by this feature-ablation score, and the top-$K = 8$ representations are retained:

$$\mathcal{S}^* = \{(\ell_1, \tau_1), \ldots, (\ell_8, \tau_8)\}.$$

The selected subset remains fixed throughout subsequent SFT, RL, and inference; no feature selector is used at test time.

Latent adapter. The selected world-model features are mapped into the VLM representation space using a lightweight Transformer adapter $\mathcal{P}_w$. Each feature is first linearly projected to a hidden dimension of 1024, followed by two Transformer layers with 16 attention heads.

The adapter compresses the fused representation into 64 latent tokens for each world-model observation.

We additionally include embeddings indicating spatial position, world-model layer identity, and denoising timestep. The resulting representation is

$$H_i^w = \mathcal{P}_w\left(\{h_i^{(\ell,\tau)} : (\ell, \tau) \in \mathcal{S}^*\}\right).$$

Alignment optimization. During this stage, both the VLM backbone and SVC are frozen and only the latent adapter is optimized. We train for one epoch using AdamW with learning rate

$$1 \times 10^{-4},$$

weight decay 0.01, and an effective batch size of 64.

We use a modality-dropout probability of

$$p_{\text{drop}} = 0.5,$$

so that the model receives either

$$X_{\text{joint}} = [H^{\text{rgb}}, H^w, Q]$$

or

$$X_{\text{latent}} = [H^w, Q].$$

For weak representation compatibility, we set

$$\lambda_{\text{compat}} = 0.05,$$

yielding

$$\mathcal{L}_{\text{align}} = \mathcal{L}_{\text{ans}} + 0.05\mathcal{L}_{\text{compat}}.$$

The compatibility projection heads are used only during alignment and are discarded afterwards.

B.3 Stage II: Simulation-Trajectory Supervised Fine-Tuning

Trajectory construction. Existing spatial-reasoning datasets generally provide observations, questions, and final answers, but do not provide the intermediate reasoning–action–latent sequences required by WorldMind. We therefore construct 10,000 synthetic reasoning trajectories using GPT-5.6 Sol as the teacher model.

Given the initial observations and question, the teacher identifies spatial information that cannot be reliably obtained from the current views and predicts a sequence

$$\mathcal{T}^* = \{(R_1^*, A_1^*), \ldots, (R_T^*, A_T^*)\},$$

where R_t^* denotes the intermediate reasoning instruction and A_t^* specifies a camera transformation.

Importantly, the teacher predicts only R_t^* and A_t^*. It does not generate latent states. Instead, each predicted action is executed by the actual frozen world model:

$$\widetilde{Z}_t = \mathcal{W}(S_{t-1}, A_t^*), \qquad \widetilde{H}_t^w = \mathcal{P}_w(\widetilde{Z}_t).$$

The actual world-model latent is then inserted into the context for the next reasoning step. The resulting SFT sequence is

$$[R_1^*, A_1^*, \widetilde{H}_1^w, \ldots, R_T^*, A_T^*, \widetilde{H}_T^w, J^*, Y^*].$$

We allow at most $T = 10$ world-model interactions per trajectory.

Trajectory filtering. Before SFT, we remove trajectories containing malformed or unsupported camera actions, incorrect final answers, failed world-model executions, or transformations inconsistent with the corresponding reasoning instruction. We also discard trajectories containing repeated transformations that provide no additional spatial evidence.

For synthetic scenes, we additionally render the corresponding ground-truth target view. A trajectory is removed if the generated world-model state is severely inconsistent with the ground-truth state at the requested pose. The final SFT set contains 10,000 trajectories.

LoRA configuration. Starting from the aligned model, we insert LoRA modules into the attention and feed-forward projections of the VLM. We use rank

$$r = 16, \qquad \alpha = 32,$$

with LoRA dropout 0.05.

The original Qwen3-VL parameters remain frozen. Only the LoRA parameters and latent adapter are optimized.

SFT optimization. We jointly train the latent adapter and LoRA parameters for three epochs using AdamW. The learning rates are

$$1 \times 10^{-4}$$

for the latent adapter and

$$2 \times 10^{-5}$$

for LoRA, with weight decay 0.01 and effective batch size 64.

We use equal weights for the three supervised objectives:

$$\lambda_r = \lambda_a = \lambda_f = 1.$$

The returned world-model latent states are conditioning observations and therefore receive no autoregressive prediction loss.

B.4 Stage III: Process-Verifiable Reinforcement Learning

On-policy training data. For reinforcement learning, we sample 5,000 spatial questions from the training pool and initialize the policy from the SFT checkpoint. Unlike SFT, no teacher trajectory is supplied. The model generates its own reasoning steps and camera actions and interacts with the frozen world model on-policy.

We use GRPO with eight sampled trajectories per question. Each rollout contains at most three world-model interactions.

Progress estimation. For each intermediate reasoning context C_t, we estimate the model's ability to reach the correct answer using $V(C_t)$. For multiple-choice questions, a simple implementation is the normalized probability assigned to the ground-truth answer:

$$V(C_t) = \frac{\exp(s_{y^*})}{\sum_{y \in \mathcal{Y}} \exp(s_y)},$$

where s_y is the answer log-likelihood and $\mathcal{Y}$ denotes the candidate answers.

The progress associated with the new reasoning and simulated evidence is then

$$\Delta V_t = V(C_{t+1}) - V(C_t).$$

Action validity. The action-validity term evaluates whether the resulting world-model transition realizes the transformation intended by the reasoning step.

For synthetic environments, where both requested and executed camera poses are available, we verify the transformation programmatically. A transition is treated as fully valid when the rotational and translational errors satisfy

$$e_R < 5^\circ, \qquad e_T < 0.05 \text{ m}.$$

For examples without ground-truth camera states, we use a frozen VLM judge:

$$c_t^{\text{action}} = J_{\text{VLM}}(R_t, A_t, O_{t-1}, O_t),$$

which measures whether the transition from O_{t-1} to O_t is consistent with the transformation described by R_t and A_t. The score is normalized to $[0, 1]$.

Latent validity. For synthetic examples, we render the ground-truth observation O_t^{GT} at the requested target camera pose. We then extract its world-model representation using the same layer and denoising timestep used for the generated latent.

Let

$$H_t^w = \{h_{t,i}^w\}_{i=1}^M, \qquad H_t^{\text{GT}} = \{h_{t,i}^{\text{GT}}\}_{i=1}^M.$$

The latent-validity score is

$$c_t^{\text{latent}} = \frac{1}{M} \sum_{i=1}^{M} \frac{1 + \cos\left(h_{t,i}^w, h_{t,i}^{\text{GT}}\right)}{2}.$$

When multiple selected layer–timestep representations are used, we compute the score independently for each selected representation and average the resulting validity scores.

Step and trajectory rewards. Action and latent validity act as reliability gates rather than independent positive rewards:

$$r_t^{\text{step}} = c_t^{\text{action}} c_t^{\text{latent}} \Delta V_t.$$

This prevents a reasoning step from receiving high process credit merely because an unreliable generated state happens to increase confidence in the correct answer.

The complete trajectory reward is

$$R(\tau) = \lambda_{\text{ans}} r_{\text{ans}} + \lambda_{\text{fmt}} r_{\text{fmt}} + \sum_{t=1}^{T} r_t^{\text{step}} - \lambda_c T.$$

For the preview configuration, we use

$$\lambda_{\text{ans}} = 1.0, \qquad \lambda_{\text{fmt}} = 0.1, \qquad \lambda_c = 0.02.$$

For categorical questions, r_{ans} is based on answer correctness. For numerical tasks, the reward can use the corresponding benchmark-specific continuous correctness score.

GRPO optimization. We optimize the latent adapter and VLM LoRA parameters while keeping the original VLM parameters and SVC frozen. We use learning rates of

$$5 \times 10^{-6}$$

for the adapter and

$$1 \times 10^{-6}$$

for LoRA.

We use a GRPO clipping parameter of 0.2 and KL coefficient of 0.01.

B.5 Common Optimization and Hardware

All three stages use BF16 precision. We apply gradient clipping with a maximum norm of 1.0, use a warm-up ratio of 0.03, and follow the warm-up with cosine learning-rate decay.

Training is conducted on eight NVIDIA H100 GPUs.

The world model always operates in inference mode and does not retain gradients. Gradient accumulation is used when necessary to maintain the effective batch sizes reported above.