TIRESIAS: TOWARDS LANGUAGE-CONDITIONED WORLD MODELING

Anonymous authors
Paper under double-blind review

ABSTRACT

World models enable planning through internal simulation, while natural language provides intuitive task specification. Yet, language-conditioned world modeling remains fragmented across domains with disparate action spaces, datasets, and evaluation protocols. We introduce LCWM, a shared task that maps an initial agent-view observation and a language instruction to an action sequence that fulfills the instruction. Its accompanying benchmark, LCWM-Bench, integrates 15 datasets across navigation, robotics, 3D gaming, and driving under a shared action interface and evaluation protocol, pairing 142,987 episodes with 428,961 instructions in three styles. We further propose *Tiresias*, an agent that plans via selective imagination: a language-conditioned diffusion policy generates candidate trajectories, a trajectory ranker shortlists them, and a video-pretrained world model imagines their visual consequences. A visual ranker evaluates these rollouts, complementing trajectory scores to select the final plan. *Tiresias* achieves competitive planning and imagination performance on LCWM-Bench. Ablations demonstrate the complementary benefits of trajectory ranking and visual verification, while experiments with alternative proposal methods show that the trained rankers remain effective beyond our diffusion policy. Together, LCWM-Bench and *Tiresias* provide a unified evaluation setting and an effective starting point for studying language-guided planning through world models.

1 INTRODUCTION

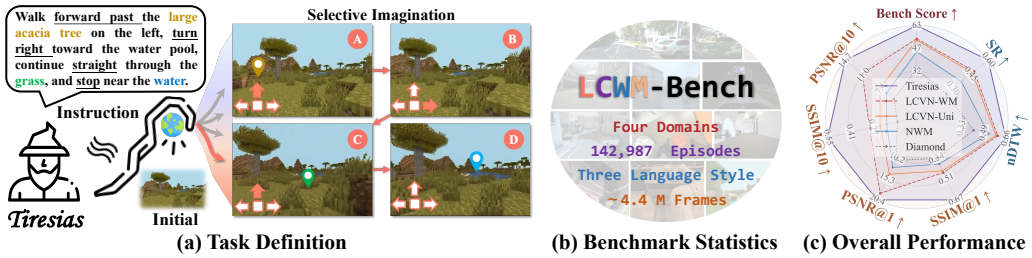
"The blind seer, whose mind abides steadfast."
— Homer, *The Odyssey*, Book X, trans. A. T. Murray

Generalist agents operating in complex environments (Chu et al., 2026), such as a robot in a kitchen, a vehicle in traffic, or an avatar in a game, require both world modeling and language grounding. World models predict how observations evolve under actions, enabling agents to consider possible outcomes before execution (Brooks et al., 2024; Zhu et al., 2024), while language grounding connects natural-language instructions to intended behavior (Wu et al., 2025). Bringing these capabilities together requires more than generating visually plausible futures: agents need to identify actions that follow the instruction and assess their consequences in the current scene.

Existing approaches have largely been evaluated within individual domains (Dong et al., 2026), with differences in control interfaces, datasets, and metrics across navigation, robotics, gaming, and driving making cross-domain comparisons difficult (Liang et al., 2025). This fragmentation also makes it difficult to assess whether the same modeling and planning approach can accommodate different embodiments and instruction styles. A shared evaluation setting is therefore needed to study language-conditioned world modeling across domains while preserving control requirements.

We address this need with LCWM, a shared task, and its accompanying benchmark, LCWM-Bench. As illustrated in Fig. 1 (a–c), LCWM requires an agent to map an initial agent-view observation and a natural-language instruction to an action sequence that fulfills the instruction. LCWM-Bench integrates 15 datasets across navigation, robotics, 3D gaming, and driving through a shared action parameterization with domain-specific control channels. It pairs 142,987 episodes with 428,961 instructions in three styles, allowing language grounding to be evaluated at different levels of descriptive detail. A common metric suite measures goal completion and trajectory quality, while separate seen- and unseen-source splits assess generalization beyond the training sources.

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.



(a) Task Definition (b) Benchmark Statistics (c) Overall Performance

Figure 1: **Overview of LCWM, LCWM-Bench, and *Tiresias*. (a)** From an initial observation and instruction, *Tiresias* plans through imagination (A↦D); colored markers identify landmarks and highlighted arrows indicate actions. **(b)** LCWM-Bench spans 15 datasets, four domains, and 142,987 episodes in three instruction styles. **(c)** Planning and imagination performance on LCWM-Bench.

We further propose *Tiresias*, a reference agent for LCWM-Bench named after the blind seer of Greek mythology. The agent plans via selective imagination: its language-conditioned diffusion policy proposes candidate trajectories without observing the scene. A trajectory ranker shortlists these proposals using language and motion alone. A video-pretrained world model then uses the initial observation and learned visual dynamics to imagine the shortlisted candidates' consequences, which a visual ranker evaluates. The trajectory and visual scores jointly select the final plan. This separates instruction-guided proposal generation from scene-dependent evaluation, allowing a broad candidate pool while limiting costly imagination to a small subset. Experiments demonstrate the effectiveness of this design, with ablations supporting complementary ranking signals and diminishing returns from larger imagination budgets. The trained rankers also benefit alternative candidate generators, making the selection pipeline reusable beyond our diffusion policy.

In summary, our main contributions are:

- We formulate LCWM and introduce LCWM-Bench, bringing together 15 datasets across four domains under a shared action interface, three instruction styles, and a standardized evaluation protocol for evaluating cross-domain language grounding.
- We propose *Tiresias*, which couples a language-conditioned diffusion policy with a video-pretrained world model through complementary trajectory and visual rankers, enabling planning via selective imagination.
- We evaluate planning and imagination across four domains, demonstrate the complementary benefits of trajectory ranking and visual verification, and analyze candidate coverage, imagination budgets, and the reuse of trained rankers with alternative proposal methods.

2 Related Work

World Modeling. World models predict environment dynamics under agent actions, enabling outcomes to be simulated before execution (Ha & Schmidhuber, 2018). The field has progressed from recurrent latent dynamics (Ha & Schmidhuber, 2018; Hafner et al., 2023) and predictive Transformers (Assran et al., 2023; Baldassarre et al., 2025) to generative backbones acting as interactive simulators (Bruce et al., 2024; Bar et al., 2025; Agarwal et al., 2025; Wan et al., 2025; Xiong et al., 2026a). Language is increasingly integrated here: actions dictate step-level dynamics, while language specifies goals and unifies heterogeneous agents. These advances now cover our four studied domains, including text-guided video plans for manipulation (Du et al., 2023; Zhou et al., 2024), prompt-enriched driving generation (Wang et al., 2024a; Zhao et al., 2025; Xiong et al., 2026b), free-text game simulation (Xiang et al., 2024; Yu et al., 2025), and instruction-grounded navigation futures (Dong et al., 2026). Yet, these efforts remain confined to isolated domains with distinct action spaces, leaving it untested whether a shared formulation can support language-conditioned simulation across domains. LCWM formalizes this setting, and *Tiresias* addresses it by utilizing the world model not just as a generator, but as an evaluator of language-conditioned plans.

Benchmarks for World Models. Early evaluation followed video-generation protocols that score visual fidelity and prompt alignment (Huang et al., 2023; Duan et al., 2025). As world models became interactive simulators, benchmarks turned to spatiotemporal consistency and physical plausibility (Kwon et al., 2025; Lian et al., 2025), action following (Wang et al., 2024b; Yue et al., 2025),

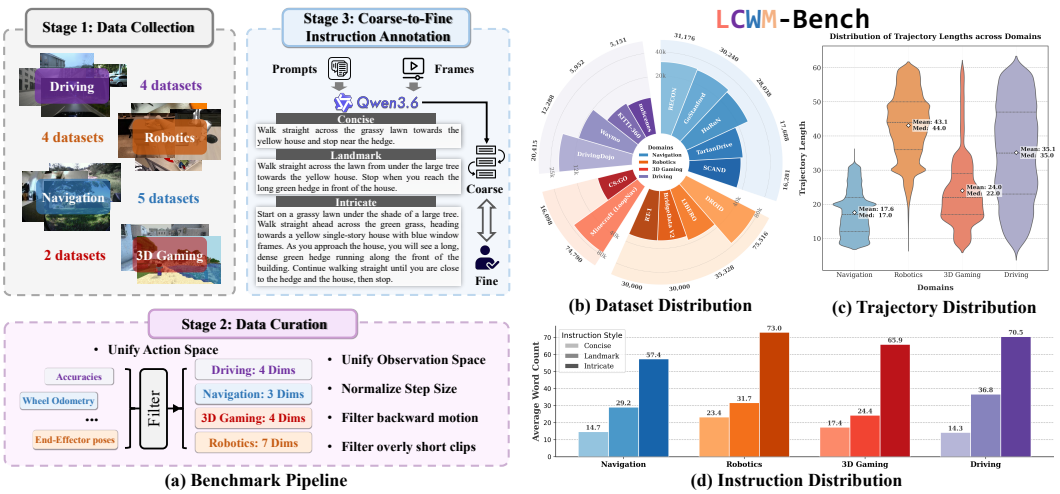


Figure 2: **Construction and statistics of LCWM-Bench. (a)** Dataset collection, action and observation standardization, and instruction annotation (Sec. 3.2). **(b–d)** Dataset sample counts, trajectory-length distributions, and average instruction lengths by style (Sec. 3.3).

and functional utility such as closed-loop policy evaluation and procedural planning (Zhang et al., 2025; Fan et al., 2026). Most target a single domain such as driving, manipulation, or gaming; the few multi-domain suites (Wu et al., 2026) still condition rollouts on actions or camera controls, with language serving only as a prompt. Language-conditioned evaluation instead centers primarily on policies, such as CALVIN for manipulation (Mees et al., 2022), or is restricted to single domains, like LCVN (Dong et al., 2026) evaluating world models only for navigation. Whether a world model can turn free-form instructions into reliable simulations and executable plans across domains remains untested. LCWM-Bench fills this gap by integrating 15 datasets under a shared action interface and evaluation suite.

3 LCWM-Bench

3.1 Task Formulation

We formulate language-conditioned world modeling (LCWM) as follows. Given an initial agent-view RGB observation o_s and a natural-language instruction I, an agent must predict an action sequence $\hat{\mathcal{A}}_T$ that fulfills the instruction. We denote the instruction and action sequence by:

$$I = \langle w_1, \ldots, w_n \rangle, \qquad \hat{\mathcal{A}}_T = \langle \hat{a}_1, \ldots, \hat{a}_T \rangle. \tag{1}$$

Each action $\hat{a}_t$ is a control command with a base parameterization (u_t, ϕ_t) shared across all domains, where $u_t \in \mathbb{R}^2$ is a planar displacement covering forward, backward, and lateral motions, and $\phi_t \in \mathbb{R}$ is a yaw rotation. Actions are executed sequentially, with a null action signaling episode termination. The LCWM task spans four domains, $\mathcal{D} = \{\textit{Navigation}, \textit{Robotics}, \textit{3D Gaming}, \textit{Driving}\}$, which share the base parameterization but incorporate additional domain-specific control channels: 3D gaming adds a vertical displacement $z_t \in \mathbb{R}$; driving introduces a scalar velocity $v_t \in \mathbb{R}$; and robotics, controlling an end-effector, appends the vertical displacement z_t alongside roll, pitch, and a gripper command $(\rho_t, \vartheta_t, g_t)$. The raw controls from all sources are normalized into this shared parameterization (Sec. 3.2).

3.2 Benchmark Construction

Instantiating LCWM requires episodes sharing a shared action interface across diverse domains. We build LCWM-Bench in three stages (Fig. 2a).

Dataset Collection. LCWM-Bench integrates 15 datasets across four domains, mixing real and simulated sources for scene diversity and metric actions (Fig. 2b). *Navigation*: GoStanford (Hirose et al., 2018), RECON (Shah et al., 2021), SCAND (Karnan et al., 2022), HuRoN (Hirose et al., 2023), and TartanDrive (Triest et al., 2022) cover indoor, outdoor, social, and off-road ground-robot traversal. *Robotics*: BridgeData V2 (Walke et al., 2023), DROID (Khazatsky et al., 2024), RT-1 (Brohan et al., 2022), and LIBERO (Liu et al., 2023a) provide 6-DoF gripper control across

household and office scenes. *3D Gaming*: Minecraft (Lian et al., 2025) contributes open worlds, while CS:GO (Pearce & Zhu, 2022) adds fast-paced human matches. *Driving*: nuScenes (Caesar et al., 2020), KITTI-360 (Liao et al., 2022), Waymo (Sun et al., 2020), and DrivingDojo (Wang et al., 2024b) span urban to suburban traffic, representing the benchmark's highest speeds.

Shared Action and Observation Spaces. We standardize heterogeneous raw controls (e.g., wheel odometry, end-effector poses) via a shared pipeline. Episodes are converted into per-step motion increments in the local frame (x forward, y left, yaw counter-clockwise, z up), derived from consecutive pose differences when explicit actions are absent. Following Dong et al. (2026), displacements are normalized by each source's average step size to unify magnitudes. Agent-view frames are resized to 256×256. We then filter out backward, near-static, or overly short clips, segmenting the remainder into semantically coherent episodes following Bar et al. (2025); Dong et al. (2026).

Instruction Styles. We augment each episode with language instructions in three styles, probing grounding at varying depths: **Concise** (essential directional cues); **Intricate** (rich descriptions of objects, layouts, and motion); and **Landmark-based** (anchoring maneuvers to salient landmarks, e.g., "pass the parked van, then stop at the white door"). Instructions are generated via a coarse-to-fine pipeline. In the coarse stage, Qwen3.6-27B (Qwen Team, 2026) generates all three styles from sampled frames, supplemented by trajectory metadata and (for manipulation) native annotations. In the fine stage, these machine-generated instructions are manually verified against the videos for correctness and naturalness. Thus, each episode yields three evaluation samples, one per style.

Evaluation Protocol and Metrics. Each domain is partitioned at the source level into train, val seen (held-out episodes from training sources), val unseen (sources completely excluded from training), and test (a mixture of both for future leaderboards). For instance, in navigation, TartanDrive is reserved for val unseen to evaluate generalization against significant visual domain shifts. To assess goal completion and trajectory quality, we report four metrics: *SR* (Success Rate) requires the terminal position error to be within one average step (Dong et al., 2026), plus a matching terminal gripper state for robotics; *nDTW* (Ilharco et al., 2019) measures path fidelity via order-preserving soft alignment; *ATE* and *RPE* (Sturm et al., 2012) compute the RMSE of absolute and relative position errors to capture global drift and local accuracy, respectively. The overall LCWM-Bench score is the macro-average of SR and nDTW across the four domains.

3.3 Benchmark Statistics

Fig. 2(b–d) summarizes LCWM-Bench. The benchmark comprises 142,987 episodes (≈4.4M frames) from 15 datasets (12 real, 3 simulated), which the three-style annotation expands into 428,961 evaluation samples, partitioned into training (300,276), val seen (38,607), val unseen (34,317), and test (55,761). Robotics contributes the largest share of samples (40%), followed by navigation (29%), gaming (21%), and driving (10%), and the split proportions are preserved within every domain. The domains complement one another in temporal structure (Fig. 2c): robotics episodes are the longest (43.1 steps on average) while navigation ones are the shortest (17.6), and the three instruction styles occupy separated verbosity bands in every domain (Fig. 2d), from 14–23 words for Concise to 57–73 for Intricate.

4 *Tiresias*: Planning with a Language-Conditioned World Model

We propose *Tiresias* to solve LCWM through selective imagination, combining language-guided trajectory proposals with scene-conditioned evaluation of their imagined consequences (Fig. 3). A language-conditioned diffusion policy (**LDP**) proposes trajectories, and a trajectory ranker (**Tr-Ranker**) shortlists them using language and motion alone. A language-conditioned world model imagines the retained candidates, whose rollouts are scored by a visual ranker (**V-Ranker**). The two ranker scores are fused to select the final plan.

4.1 Language-Conditioned Diffusion Policy

The LDP (Fig. 3a) is a DiT-based (Peebles & Xie, 2023) diffusion policy (Chi et al., 2025) that denoises action sequences under language guidance. Aside from diffusion timestep, conditioning is derived entirely from language, utilizing frozen umT5 (Chung et al., 2023) instruction embeddings processed via multi-head cross-attention within every DiT block. Trained to denoise expert

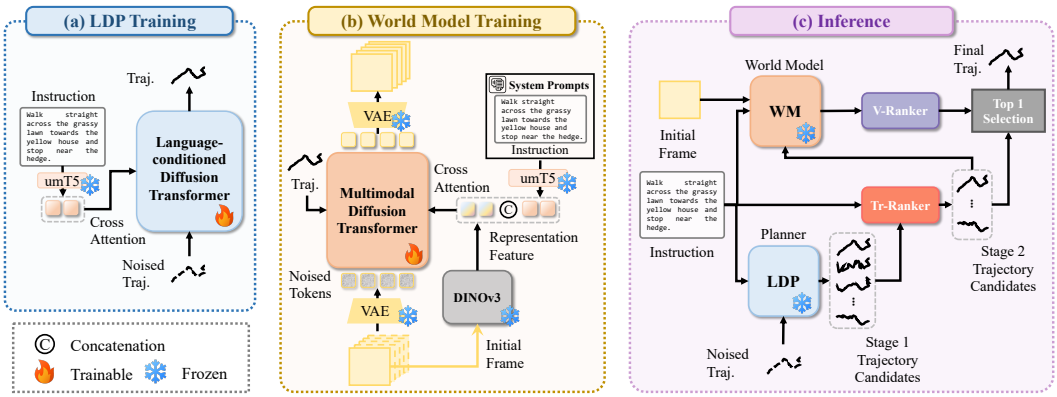


Figure 3: **Overview of *Tiresias*. (a)** The LDP learns to denoise action sequences conditioned on instructions. **(b)** The Wan2.2-based world model predicts future observations conditioned on the initial frame, trajectory, and instruction. **(c)** At inference, the Tr-Ranker shortlists LDP proposals for imagination; fused trajectory and visual scores select the final plan.

trajectories, the LDP learns the distribution of motions corresponding to an instruction, allowing independent noise seeds at inference to yield diverse candidates for subsequent trajectory ranking and visual verification. By design, the LDP is scene-agnostic: its samples span the possibilities permitted by language, delegating the scene-dependent selection to the world model.

4.2 Language-Conditioned World Model

The language-conditioned world model of *Tiresias* (Fig. 3b) is a multimodal diffusion Transformer that adapts the pretrained Wan2.2 video backbone (Wan et al., 2025) to predict futures under trajectory, scene, and language conditions while preserving its native interfaces. Video frames follow the backbone's original visual pathway, encoded by the frozen VAE (Wan et al., 2025), noised, and denoised as video tokens. Two conditioning routes steer this denoising process. First, the candidate trajectory, embedded via a lightweight MLP, directly conditions the denoiser to dictate embodiment motion. Second, cross-attention concatenates the initial frame's DINOv3 (Siméoni et al., 2025) features with the umT5 instruction embedding and a per-domain prompt, supplying critical context: scene content, intended behavior, and embodiment specifics. Formally, the generative process is:

$$\hat{X} = (\hat{x}_1, \ldots, \hat{x}_H) \sim p_\theta\big(\cdot \mid x_s,\, [f_s;\, e_I],\, \hat{\mathcal{A}}\big), \tag{2}$$

where $\hat{X}$ represents the imagined video latents over horizon H; x_s and f_s are the VAE latent and DINOv3 features of the initial observation o_s; e_I is the prompted umT5 instruction embedding; and $\hat{\mathcal{A}}$ is the conditioning trajectory.

The initial frame thus enters twice: its VAE latent anchors visual appearance, while its DINOv3 features supply high-level semantics often lacking in generative latents, complementing the backbone's motion and scene priors. Training optimizes the flow-matching objective:

$$\mathcal{L}(\theta) = \mathbb{E}_{X,\, c,\, \epsilon,\, \tau} \left\| u_\theta(X_\tau, \tau, c) - (\epsilon - X) \right\|_2^2, \tag{3}$$

where X denotes the VAE latents of the ground-truth future frames, $X_\tau = (1-\tau)X + \tau\epsilon$ represents their interpolation toward Gaussian noise ϵ at flow time τ, u_θ is the denoiser predicting velocity $\epsilon - X$, and c encapsulates the conditions in Eq. (2).

4.3 Trajectory and Visual Rankers

The Tr-Ranker scores each candidate $\hat{\mathcal{A}}^{(m)}$ using language and motion alone: action tokens query instruction tokens via cross-attention, a temporal Transformer summarizes the trajectory, and four heads estimate SR, nDTW, ATE, and RPE. These predictions are combined into a score $S_{\text{tr}}^{(m)}$. The V-Ranker evaluates the candidate's imagined visual consequences. Conditioning Eq. (2) on $\hat{\mathcal{A}} = \hat{\mathcal{A}}^{(m)}$ yields video latents $\hat{X}^{(m)}$, which are pooled over horizon H and aligned with instruction embedding e_I to produce $S_{\text{v}}^{(m)}$. The two scores provide motion and visual evidence for selecting instruction-consistent trajectories. Due to space limits, kindly refer to **Appendix** for additional details.



Figure 4: **Qualitative results on Navigation, Robotics, Driving, and 3D gaming.** For each case, we show the initial observation and instruction, the imagined rollout of the selected plan against the ground-truth future, and the predicted trajectory against the reference.

4.4 Inference with Selective Imagination

At inference (Fig. 3c), the independently trained modules are composed into a two-stage selection pipeline without joint fine-tuning. The LDP first samples a diverse pool of stage-one candidates $\{\hat{\mathcal{A}}^{(m)}\}_{m=1}^{N}$ using independent noise seeds. The Tr-Ranker then filters this pool into a stage-two shortlist $\mathcal{S}$ of size k prior to any simulation. Only these retained candidates are subsequently rolled out by the world model and evaluated by V-Ranker. *Tiresias* ultimately executes:

$$\hat{\mathcal{A}}^{*} = \hat{\mathcal{A}}^{(m^{*})}, \quad m^{*} = \arg\max_{m \in \mathcal{S}} \; \lambda_{\mathrm{tr}}\, S_{\mathrm{tr}}^{(m)} + \lambda_{\mathrm{v}}\, S_{\mathrm{v}}^{(m)}, \tag{4}$$

where λ_{tr} and λ_{v} weight the two scores, which are normalized within $\mathcal{S}$ prior to fusion.

5 Experiments

5.1 Experimental Setup

Baselines. We first include two policy-only baselines, GNM (Shah et al., 2022) (lang) and NoMaD (Sridhar et al., 2024) (lang), which replace the image-goal encoder with a frozen CLIP text encoder projecting instructions into the goal-token space. The remaining baselines all leverage world models: Diamond (Alonso et al., 2024) (UNet-based diffusion), NWM (Bar et al., 2025) (CDiT-based diffusion), NWM (lang) which integrates CLIP-encoded instructions via the CDiT AdaLN block, and LCVN-WM (Dong et al., 2026). Each of these world models is paired with the latent actor-critic LCVN-AC to produce actions. We additionally include LCVN-Uni (Dong et al., 2026), an autoregressive unified model that predicts observations and actions jointly. All baselines are trained on LCWM-Bench and evaluated under the shared protocol. Finally, *Tiresias* (MPC) replaces the LDP with sampling-based MPC for candidate generation while retaining two-stage ranking, thereby isolating contribution of learned language-conditioned proposal distribution.

Table 1: **Planning comparison** on val seen and val unseen splits of LCWM-Bench. The score is the mean of SR and nDTW macro-averaged over four domains; higher is better for Score, SR, and nDTW, lower for ATE and RPE. Best results are in bold.

		LCWM-Bench	Navigation				Robotics				3D Gaming				Driving			
	Method	Score (%)	SR	nDTW	ATE	RPE	SR	nDTW	ATE	RPE	SR	nDTW	ATE	RPE	SR	nDTW	ATE	RPE
Val Seen	GNM (lang)	36	0.24	0.44	1.10	0.34	0.25	0.42	0.74	0.33	0.25	0.39	3.18	0.48	0.36	0.52	1.29	0.27
	NoMaD (lang)	40	0.28	0.48	0.95	0.29	0.29	0.45	0.68	0.31	0.27	0.41	3.05	0.45	0.42	0.57	1.14	0.24
	Diamond	33	0.19	0.40	1.31	0.40	0.20	0.37	0.82	0.36	0.31	0.45	2.88	0.42	0.29	0.46	1.48	0.30
	NWM	45	0.31	0.50	0.72	0.23	0.33	0.49	0.64	0.29	0.40	0.53	2.30	0.35	0.46	0.60	1.04	0.22
	NWM (lang)	48	0.33	0.52	0.69	0.25	0.35	0.51	0.59	0.27	0.43	0.55	2.14	0.33	0.49	0.63	0.95	0.20
	LCVN-WM	54	0.44	0.59	0.37	0.12	0.44	0.57	0.47	0.22	0.51	0.58	1.88	0.29	0.54	0.65	0.83	0.17
	LCVN-Uni	53	0.43	0.58	0.39	0.11	0.43	0.56	0.50	0.21	0.49	0.57	1.96	0.28	0.52	0.63	0.89	0.16
	Tiresias (MPC)	59	0.56	0.64	0.36	0.10	0.51	0.59	0.34	0.16	0.58	0.60	1.74	0.26	0.58	0.66	0.73	0.14
	***Tiresias* (ours)**	**63**	**0.60**	**0.67**	**0.35**	**0.09**	**0.55**	**0.62**	**0.24**	**0.10**	**0.63**	**0.64**	**1.49**	**0.19**	**0.62**	**0.70**	**0.66**	**0.11**
Val Unseen	GNM (lang)	28	0.12	0.33	2.45	0.81	0.18	0.34	0.93	0.42	0.19	0.33	3.58	0.55	0.27	0.45	1.69	0.35
	NoMaD (lang)	31	0.14	0.36	2.28	0.74	0.21	0.38	0.86	0.39	0.20	0.35	3.46	0.52	0.31	0.50	1.52	0.32
	Diamond	24	0.10	0.31	2.76	0.91	0.12	0.28	1.01	0.45	0.22	0.37	3.35	0.50	0.16	0.33	1.95	0.41
	NWM	33	0.15	0.37	2.10	0.70	0.22	0.39	0.82	0.38	0.30	0.45	2.62	0.41	0.29	0.46	1.47	0.31
	NWM (lang)	35	0.16	0.39	2.13	0.68	0.24	0.41	0.76	0.35	0.33	0.48	2.44	0.38	0.32	0.50	1.35	0.29
	LCVN-WM	38	0.20	0.43	1.47	0.54	0.28	0.44	0.67	0.31	0.38	0.49	2.18	0.35	0.33	0.50	1.34	0.28
	LCVN-Uni	42	0.28	0.49	1.18	0.45	0.31	0.47	0.63	0.29	0.42	0.52	2.08	0.33	0.36	0.53	1.25	0.26
	Tiresias (MPC)	46	0.38	0.56	0.94	0.18	0.35	0.49	0.43	0.20	0.48	0.55	1.84	0.28	0.37	0.53	1.15	0.24
	***Tiresias* (ours)**	**51**	**0.41**	**0.61**	**0.89**	**0.17**	**0.39**	**0.55**	**0.31**	**0.14**	**0.54**	**0.61**	**1.61**	**0.25**	**0.40**	**0.56**	**1.07**	**0.22**

Table 2: **Imagination comparison** on the val seen and val unseen splits of LCWM-Bench, averaged over the four domains. @n metrics score n-step open-loop rollouts under ground-truth actions against the real frames. Best results are in bold.

			One-step (@1)				Rollout (@10)			
	Method	FVD@32 ↓	SSIM↑	PSNR↑	LPIPS↓	DreamSim↓	SSIM↑	PSNR↑	LPIPS↓	DreamSim↓
Val Seen	Diamond	648.7	0.336	10.642	0.401	0.128	0.118	4.731	0.668	0.301
	NWM	382.1	0.382	12.364	0.326	0.111	0.184	6.218	0.621	0.242
	NWM (lang)	365.3	0.396	12.578	0.319	0.108	0.178	6.176	0.598	0.236
	LCVN-WM	321.6	0.452	18.724	0.268	0.101	0.289	11.204	0.471	0.219
	LCVN-Uni	344.8	0.438	14.018	0.287	0.096	0.224	7.562	0.492	0.214
	***Tiresias* (ours)**	**274.0**	**0.674**	**20.351**	**0.242**	**0.089**	**0.546**	**14.732**	**0.454**	**0.197**
Val Unseen	Diamond	2453.8	0.272	9.341	0.468	0.173	0.083	3.914	0.721	0.344
	NWM	1869.3	0.318	10.982	0.381	0.142	0.137	5.421	0.681	0.286
	NWM (lang)	1748.6	0.331	11.164	0.374	0.137	0.132	5.368	0.652	0.276
	LCVN-WM	1328.6	0.386	16.452	0.319	0.119	0.238	9.843	0.515	0.246
	LCVN-Uni	1563.7	0.366	12.384	0.342	0.112	0.174	6.614	0.548	0.238
	***Tiresias* (ours)**	**744.9**	**0.550**	**19.695**	**0.303**	**0.088**	**0.448**	**14.571**	**0.504**	**0.199**

Evaluation Metrics. Planning is evaluated using the LCWM-Bench metrics detailed in Sec. 3: SR, nDTW, ATE, RPE, and the overall score. Additionally, we assess the imagination quality of the world models: @1 metrics compare the predicted next frame against the ground truth, while @n metrics evaluate n-step open-loop generation conditioned on ground-truth actions. Specifically, we adopt SSIM, PSNR, LPIPS (Zhang et al., 2018), and DreamSim (Fu et al., 2023) for @1 and @10 evaluations, alongside FVD (Unterthiner et al., 2019) at @32.

Implementation Details. The world model is initialized from Wan2.2 and the LDP is trained from scratch; umT5, DINOv3, and the VAE remain frozen throughout training. Visual encoders process all observations at a uniform 256×256 resolution, and scene features are extracted using a frozen DINOv3 ViT-L/16 distilled encoder (300M parameters). The world model is trained for 10 epochs on LCWM-Bench, utilizing 8×NVIDIA A100 GPUs (80GB). At inference, the LDP samples $N{=}128$ candidates from a fixed set of independent noise seeds, the Tr-Ranker shortlists $k{=}8$ candidates, and the two evaluation scores are fused with a ratio of $\lambda_{\text{tr}}{:}\lambda_{\text{v}} = 2{:}1$.

Due to space limits, kindly refer to **Appendix** for more implementation and experimental details.

5.2 Performance Comparison on LCWM-Bench

Planning. Table 1 compares the evaluated agents across four domains. Under the reported configurations, *Tiresias* obtains the highest overall score on both val seen and val unseen, reaching 63 and 51, respectively. Its MPC variant obtains scores of 59 and 46; replacing sampling-based proposals with learned diffusion proposals improves performance while retaining the same two-stage ranking pipeline. All evaluated methods show lower overall scores on val unseen, indicating that generaliza-

Table 3: Ablation on **cross-attention conditioning** of world model on val seen, with per-domain system prompt kept in all variants. Best results are in bold.

	Planning		Imagination	
WM Conditioning	SR↑	nDTW↑	SSIM@1 ↑	FVD@32 ↓
Instruction only	0.56	0.65	0.61	328.7
DINOv3 only	0.54	0.64	0.58	367.4
Instruction + DINOv3	**0.60**	**0.66**	**0.67**	**274.0**

Table 4: Ablation on **LDP conditioning**: random candidate execution versus two-stage selection (128 → 8). Best results are in bold.

	Unranked exec.		Full pipeline	
LDP Conditioning	SR↑	nDTW↑	SR↑	nDTW↑
Observation only	0.06	0.26	0.41	0.54
Language only (default)	**0.13**	**0.34**	**0.60**	**0.66**
Language + Observation	0.11	0.31	0.56	0.63

Figure 5: Planning performance across three different **instruction styles** on the val seen split.

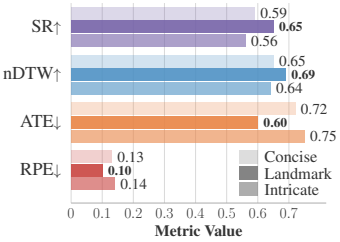


Table 5: **Selector comparison** on val seen using the same eight randomly sampled LDP candidates. Imag. denotes methods requiring world-model rollouts. Best results are in bold.

Selector	Imag.	SR↑	nDTW↑	ATE↓	RPE↓
Random	–	0.13	0.34	2.35	0.29
Consensus (mean traj.)	–	0.24	0.47	1.43	0.20
Cluster-vote (medoid)	–	0.18	0.40	1.91	0.25
Qwen3.6 verifier	✓	0.26	0.49	1.28	0.19
Tr-Ranker	–	0.30	0.53	1.14	0.17
V-Ranker	✓	0.28	0.51	1.20	0.19
Tr + V (fused)	✓	**0.39**	**0.59**	**0.98**	**0.14**
Oracle-of-8	–	0.52	0.66	0.85	0.12

tion to held-out sources and environments remains challenging. These results establish *Tiresias* as an effective starting point for further development on LCWM-Bench.

Imagination. Table 2 compares imagination quality on val seen and val unseen under ground-truth actions. *Tiresias* obtains competitive results across the reported metrics among the evaluated configurations on both splits. These results characterize the predictive quality of the world model, while the ranker ablations below assess the contribution of visual verification to candidate selection.

5.3 Ablation Studies

a1. World-Model Conditioning. Table 3 ablates the instruction and DINOv3 features supplied through cross-attention, with the per-domain system prompt and initial-frame VAE conditioning retained in all variants. Combining both inputs yields better planning and imagination results than either alone, supporting value of instruction conditioning and additional scene-semantic features.

a2. LDP Conditioning. Table 4 compares conditioning the LDP on the observation, the instruction, or both. Language-only conditioning yields the best results in both unranked execution and the full pipeline. This supports its use for proposal generation in *Tiresias*, with scene information incorporated through subsequent imagination and selection.

a3. Instruction Style. Fig. 5 reports planning performance per style. Landmark-based instructions perform best and concise ones edge out intricate ones, suggesting that grounding in observable references matters more than descriptive detail. Landmarks may anchor key maneuvers to recognizable objects and locations, while extra description can dilute action-relevant cues.

b1. Ranker Comparison. Table 5 compares selection and aggregation strategies using the same 8 randomly sampled candidates. We also evaluate a Qwen3.6-based verifier that applies MVoT-inspired reasoning (Li et al., 2025) to the same imagined rollouts as the V-Ranker to assess instruction fulfillment. Both learned rankers outperform random selection, trajectory averaging, and cluster voting. Fusing the two scores achieves the best non-oracle performance, surpassing the Qwen verifier, while the remaining gap to Oracle-of-8 indicates further room to improve candidate selection.

b2. Test-Time Scaling. With k=8 fixed, final SR largely plateaus around N=128, motivating this default to balance performance and proposal costs (Fig. 6a). Unranked execution remains unchanged, while Oracle-of-N continues to improve as the candidate pool grows, indicating that additional successful candidates are not fully exploited by selection.

b3. Candidate Funnel. Fig. 6(b–d) examines shortlist size, score fusion, and shortlisting strategy. Increasing the shortlist size initially improves planning, but gains diminish as more candidates require imagination. A 2:1 fusion of Tr-Ranker and V-Ranker scores achieves the highest SR, outper-

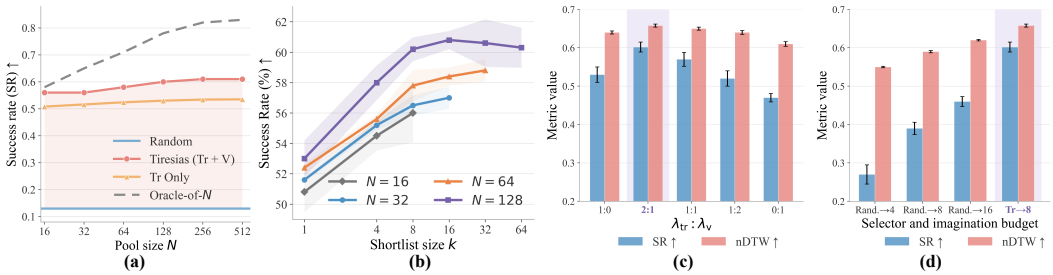


Figure 6: **Ablations on candidate sampling and selection** on val seen. **(a)** Test-time scaling with pool size N at fixed k=8; Oracle-of-N provides the SR upper bound over each candidate pool. **(b)** Effect of shortlist size k for different pool sizes. **(c)** Ranker fusion weights $\lambda_{\text{tr}} : \lambda_{\text{v}}$ under the default $128 \rightarrow 8$ funnel. **(d)** Random versus Tr-Ranker shortlisting with different imagination budgets at N=128. Shaded bands in (b) and error bars in (c–d) indicate $\pm$ one standard deviation.

Table 6: **Stage-one shortlist quality** on val seen (N=128). Retention is ratio of shortlist oracle SR to full-pool oracle SR. Hit@k is percentage of episodes whose oracle-best candidate is retained.

Metric	k=1	k=4	k=8	k=16
Retention↑	0.68	0.80	**0.85**	0.86
Final SR_{Tr}↑	0.53	0.58	**0.60**	0.61
Final $\text{SR}_{\text{Random}}$↑	0.13	0.27	**0.39**	0.46
Hit@k (%)↑	5.39	14.77	**22.34**	34.22

Table 7: **Plug-and-play trajectory ranking** on val seen. The trained rankers are applied to NoMaD (lang) and MPC without further fine-tuning.

Policy	Selector	SR↑	nDTW↑	ATE↓	RPE↓
MPC	Unranked	0.11	0.32	2.62	0.31
	Tr-Ranker	0.49	0.60	0.92	0.20
	Tr + V	0.56	0.62	0.79	0.17
NoMaD (lang)	Unranked	0.32	0.48	1.46	0.32
	Tr-Ranker	0.42	0.58	1.05	0.21
	Tr + V	0.47	0.60	0.92	0.18
LDP (ours)	Unranked	0.13	0.34	2.35	0.29
	Tr-Ranker	0.53	0.64	0.77	0.13
	Tr + V	**0.60**	**0.66**	**0.69**	**0.12**

forming either score alone. Tr-Ranker shortlisting also outperforms random shortlisting, even when the latter imagines twice as many candidates. We therefore adopt the $128 \rightarrow 8$ funnel with 2:1 score fusion to balance planning performance and imagination cost.

b4. Shortlist Quality. Table 6 measures retention of oracle SR, the fraction of episodes containing a successful candidate. Retention and Hit@k increase with shortlist size; keeping only 8 of 128 candidates preserves approximately 85% of full-pool oracle SR, showing that the coarse filter retains useful candidates while substantially reducing imagination.

b5. Plug-and-Play Trajectory Ranking. We replace the LDP with NoMaD (lang) or sampling-based MPC, reusing the trained rankers without further fine-tuning. Table 7 compares unranked execution, Tr-Ranker selection, and the full pipeline with N=128 and k=8. Both alternatives benefit from trajectory ranking and visual verification, demonstrating the applicability of our selection pipeline beyond LDP proposals. And the LDP-based pipeline still achieves the highest final SR.

c1. Qualitative Results and Limitations. Fig. 4 shows scene-consistent imagined rollouts and predicted trajectories close to references across four domains. Remaining errors include late-episode trajectory drift, where instructions leave motion magnitudes underspecified, and imagined futures that remain plausible but diverge from actual environment beyond what initial observation reveals.

6 Conclusion

We introduced LCWM, a shared task for language-conditioned world modeling, and LCWM-Bench, which brings together 15 datasets across four domains under a shared action interface and evaluation protocol, with three distinct instruction styles. Our agent, *Tiresias*, combines language-conditioned trajectory proposals with selective imagination and achieves competitive planning and imagination performance across all four evaluated domains. Ablations support the complementary roles of trajectory ranking and visual verification, characterize the trade-off between candidate coverage and imagination cost, and show that the trained rankers benefit alternative candidate generators. Together, LCWM-Bench and *Tiresias* provide a foundation for studying how agents can turn language into action, using imagined futures to find a way forward.

AI USE STATEMENT

In this work, we used generative AI tools to generate synthetic data: Qwen3.6-27B (Qwen Team, 2026) drafted the three-style language instructions of LCWM-Bench from sampled frames, trajectory metadata, and 150 human-written seed annotations (Sec. 3.2, Appendix A.2). Every generated instruction was verified and corrected by human annotators against the corresponding video, and a random subset was independently rated by human raters (Appendix A.3). Qwen3.6-27B is also evaluated as a compared verifier baseline in Table 5. We have not used generative AI tools to develop theoretical models or conceptual frameworks, formulate or prove mathematical claims, propose or refine hypotheses, design or provide feedback on methodology or experiments, implement methods, clean or reformat datasets, support data analysis, or interpret results; translation is not applicable to this work. Additionally, we used LLM-based writing assistants to check grammar, spelling, and phrasing and to improve the readability of text written by the authors. We have reviewed all AI-assisted work; all edits were approved by the authors and none changed the technical content or claims. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

LCWM-Bench integrates existing datasets spanning navigation, robotics, 3D gaming, and driving. Redistribution of the demonstration subset and the planned full benchmark is subject to the licenses and permissions of the original sources. Visual observations may contain bystanders or identifying scene information, making privacy review and appropriate handling of such content important considerations for release. The benchmark also inherits the coverage limitations and potential biases of its constituent datasets, which should be considered when interpreting cross-domain performance. Our evaluation measures offline trajectory prediction and visual imagination; these results do not establish the safety of deploying an agent in physical environments.

REPRODUCIBILITY STATEMENT

Section 3 describes the task formulation, dataset construction, instruction styles, and evaluation protocol. Sections 4 and 5 detail the model components, inference procedure, and experimental settings. The supplementary material includes a demonstration subset of LCWM-Bench and a README documenting the directory structure, instruction-style identifiers, frame indexing, and trajectory-data format. This subset illustrates the benchmark format and is not intended for training, evaluation, or reproducing the reported quantitative results. It is organized by source dataset rather than by evaluation split. The full benchmark, comprising 142,987 episodes and 428,961 instruction-annotated samples, will be released upon acceptance together with the official split files, subject to source-dataset redistribution permissions.

REFERENCES

Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, et al. Cosmos world foundation model platform for physical AI. *arXiv preprint arXiv:2501.03575*, 2025.

Eloi Alonso, Adam Jelley, Vincent Micheli, Anssi Kanervisto, Amos J Storkey, Tim Pearce, and François Fleuret. Diffusion for world modeling: Visual details matter in Atari. *Advances in Neural Information Processing Systems*, 37:58757–58791, 2024.

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15619–15629, 2023.

Federico Baldassarre, Marc Szafraniec, Basile Terver, Vasil Khalidov, Francisco Massa, Yann LeCun, Patrick Labatut, Maximilian Seitzer, and Piotr Bojanowski. Back to the features: DINO as a foundation for video world models. *arXiv preprint arXiv:2507.19468*, 2025.

Amir Bar, Gaoyue Zhou, Danny Tran, Trevor Darrell, and Yann LeCun. Navigation world models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15791–15801, 2025.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, et al. Video generation models as world simulators. *OpenAI Blog*, 2024.

Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *International Conference on Machine Learning*, 2024.

Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog.*, pp. 11621–11631, 2020.

Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. *Advances in Neural Information Processing Systems*, 37:24081–24125, 2024.

Ethan Chern, Jiadi Su, Yan Ma, and Pengfei Liu. Anole: An open, autoregressive, native large multimodal models for interleaved image-text generation. *arXiv preprint arXiv:2407.06135*, 2024.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025.

Haozhuang Chi, Jingsong Liang, Ziying Song, Lei Yang, Shihao Li, Haoruo Zhang, and Chen Lv. Pv-wm: A heterogeneous micro-macro world model for articulated pedestrian-vehicle co-rollout. *arXiv preprint arXiv:2609.07328*, 2026a.

Haozhuang Chi, Daosheng Qiu, Hao Su, Haochen Liu, Zirui Li, Haoruo Zhang, and Chen Lv. Driver-wm: A driver-centric traffic-conditioned latent world model for in-cabin dynamics rollout. *arXiv preprint arXiv:2605.05092*, 2026b.

Meng Chu, Xuan Billy Zhang, Kevin Qinghong Lin, Lingdong Kong, Jize Zhang, Teng Tu, Weijian Ma, Ziqi Huang, Senqiao Yang, Wei Huang, et al. Agentic world modeling: Foundations, capabilities, laws, and beyond. *arXiv preprint arXiv:2604.22748*, 2026.

Hyung Won Chung, Xavier Garcia, Adam Roberts, Yi Tay, Orhan Firat, Sharan Narang, and Noah Constant. Unimax: Fairer and more effective language sampling for large-scale multilingual pretraining. In *The Eleventh International Conference on Learning Representations*, 2023.

Yifei Dong, Fengyi Wu, Yilong Dai, Lingdong Kong, Guangyu Chen, Xu Zhu, Qiyu Hu, Tianyu Wang, Johnalbert Garnica, Feng Liu, et al. Language-conditioned world modeling for visual navigation. *arXiv preprint arXiv:2603.26741*, 2026.

Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Josh Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. *Adv. Neural Inf. Process. Syst.*, 36:9156–9172, 2023.

Haoyi Duan, Hong-Xing Yu, Sirui Chen, Li Fei-Fei, and Jiajun Wu. Worldscore: A unified evaluation benchmark for world generation. *arXiv preprint arXiv:2504.00983*, 2025.

Chun-Kai Fan, Xiaowei Chi, Xiaozhu Ju, Hao Li, Yong Bao, Yu-Kai Wang, Lizhang Chen, Zhiyuan Jiang, Kuangzhi Ge, Ying Li, et al. Wow, wo, val! a comprehensive embodied world model evaluation turing test. *arXiv preprint arXiv:2601.04137*, 2026.

Stephanie Fu, Netanel Tamir, Shobhita Sundaram, Lucy Chai, Richard Zhang, Tali Dekel, and Phillip Isola. DreamSim: Learning new dimensions of human visual similarity using synthetic data. *arXiv preprint arXiv:2306.09344*, 2023.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

Noriaki Hirose, Amir Sadeghian, Marynel Vázquez, Patrick Goebel, and Silvio Savarese. GONet: A semi-supervised deep learning approach for traversability estimation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3044–3051, 2018.

Noriaki Hirose, Dhruv Shah, Ajay Sridhar, and Sergey Levine. SACSoN: Scalable autonomous control for social navigation. *IEEE Robotics and Automation Letters*, 9(1):49–56, 2023.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. LoRA: Low-rank adaptation of large language models. *International Conference on Learning Representations*, 2022.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. Vbench: Comprehensive benchmark suite for video generative models, 2023. URL https://arxiv.org/abs/2311.17982.

Gabriel Ilharco, Vihan Jain, Alexander Ku, Eugene Ie, and Jason Baldridge. General evaluation for instruction conditioned navigation using dynamic time warping. *arXiv preprint arXiv:1907.05446*, 2019.

Haresh Karnan, Anirudh Nair, Xuesu Xiao, Garrett Warnell, Sören Pirk, Alexander Toshev, Justin Hart, Joydeep Biswas, and Peter Stone. Socially compliant navigation dataset (SCAND): A large-scale dataset of demonstrations for social navigation. *IEEE Robotics and Automation Letters*, 7 (4):11807–11814, 2022.

Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024.

Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Zahra Kolagar and Alessandra Zarcone. Aligning uncertainty: Leveraging llms to analyze uncertainty transfer in text summarization. In *Workshop on Uncertainty-Aware NLP*, pp. 41–61, 2024.

Xianghao Kong, Jinyu Chen, Wenguan Wang, Hang Su, Xiaolin Hu, Yi Yang, and Si Liu. Controllable navigation instruction generation with chain of thought prompting. In *European Conference on Computer Vision*, pp. 37–54. Springer, 2024.

Soonwoo Kwon, Jin-Young Kim, Hyojun Go, and Kyungjune Baek. Toward stable world models: Measuring and addressing world instability in generative environments, 2025. URL https://arxiv.org/abs/2503.08122.

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.

Kewei Lian, Shaofei Cai, Yilun Du, and Yitao Liang. Toward memory-aided world models: Benchmarking via spatial consistency. *arXiv preprint arXiv:2505.22976*, 2025.

Ao Liang, Lingdong Kong, Tianyi Yan, Hongsi Liu, Wesley Yang, Ziqi Huang, Wei Yin, Jialong Zuo, Yixuan Hu, Dekai Zhu, Dongyue Lu, Youquan Liu, Guangfeng Jiang, Linfeng Li, Xiangtai Li, Long Zhuo, Lai Xing Ng, Benoit R. Cottereau, Changxin Gao, Liang Pan, Wei Tsang Ooi, and Ziwei Liu. WorldLens: Full-spectrum evaluations of driving world models in real world. *arXiv preprint arXiv:2512.10958*, 2025.

Yiyi Liao, Jun Xie, and Andreas Geiger. Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45 (3):3292–3310, 2022.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *arXiv preprint arXiv:2306.03310*, 2023a.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in Neural Information Processing Systems*, 36:34892–34916, 2023b.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks, 2022. URL https://arxiv.org/abs/2112.03227.

Tim Pearce and Jun Zhu. Counter-strike deathmatch with large-scale behavioural cloning. In *2022 IEEE Conference on Games (CoG)*, pp. 104–111. IEEE, 2022.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023.

Qwen Team. Qwen3.6-27B: Flagship-level coding in a 27B dense model, April 2026. URL https://qwen.ai/blog?id=qwen3.6-27b.

Dhruv Shah, Benjamin Eysenbach, Gregory Kahn, Nicholas Rhinehart, and Sergey Levine. Rapid exploration for open-world navigation with latent goal models. *arXiv preprint arXiv:2104.05859*, 2021.

Dhruv Shah, Ajay Sridhar, Arjun Bhorkar, Noriaki Hirose, and Sergey Levine. GNM: A general navigation model to drive any robot. *arXiv preprint arXiv:2210.03370*, 2022.

Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

Ajay Sridhar, Dhruv Shah, Catherine Glossop, and Sergey Levine. NoMaD: Goal masked diffusion policies for navigation and exploration. In *IEEE International Conference on Robotics and Automation*, pp. 63–70, 2024.

Jürgen Sturm, Wolfram Burgard, and Daniel Cremers. Evaluating egomotion and structure-from-motion approaches using the TUM RGB-D benchmark. In *IEEE/RSJ International Conference on Intelligent Robot Systems Workshops*, volume 13, pp. 6, 2012.

Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog.*, pp. 2446–2454, 2020.

Samuel Triest, Matthew Sivaprakasam, Sean J Wang, Wenshan Wang, Aaron M Johnson, and Sebastian Scherer. TartanDrive: A large-scale dataset for learning off-road dynamics models. In *IEEE International Conference on Robotics and Automation*, pp. 2546–2552, 2022.

Thomas Unterthiner, Sjoerd Van Steenkiste, Karol Kurach, Raphaël Marinier, Marcin Michalski, and Sylvain Gelly. Fvd: A new metric for video generation. 2019.

Homer Rich Walke, Kevin Black, Tony Z Zhao, Quan Vuong, Chongyi Zheng, Philippe Hansen-Estruch, Andre Wang He, Vivek Myers, Moo Jin Kim, Max Du, et al. Bridgedata v2: A dataset for robot learning at scale. In *Conference on Robot Learning*, pp. 1723–1736. PMLR, 2023.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models, 2025. URL https://arxiv.org/abs/2503.20314.

Xiaofeng Wang, Zheng Zhu, Guan Huang, Xinze Chen, Jiagang Zhu, and Jiwen Lu. Drivedreamer: Towards real-world-drive world models for autonomous driving. In *European conference on computer vision*, pp. 55–72. Springer, 2024a.

Yuqi Wang, Ke Cheng, Jiawei He, Qitai Wang, Hengchen Dai, Yuntao Chen, Fei Xia, and Zhaoxiang Zhang. Drivingdojo dataset: Advancing interactive and knowledge-enriched driving world model. *Advances in Neural Information Processing Systems*, 37:13020–13034, 2024b.

Fengyi Wu, Yifei Dong, Zhi-Qi Cheng, Yilong Dai, Guangyu Chen, Hang Wang, Qi Dai, and Alexander G Hauptmann. GoViG: Goal-conditioned visual navigation instruction generation. *arXiv preprint arXiv:2508.09547*, 2025.

Meiqi Wu, Zhixin Cai, Fufangchen Zhao, Xiaokun Feng, Rujing Dang, Bingze Song, Ruitian Tian, Jiashu Zhu, Jiachen Lei, Hao Dou, Jing Tang, Lei Sun, Jiahong Wu, Xiangxiang Chu, Zeming Liu, and Kaiqi Huang. Omni-worldbench: Towards a comprehensive interaction-centric evaluation for world models, 2026. URL https://arxiv.org/abs/2603.22212.

Jiannan Xiang, Guangyi Liu, Yi Gu, Qiyue Gao, Yuting Ning, Yuheng Zha, Zeyu Feng, Tianhua Tao, Shibo Hao, Yemin Shi, et al. Pandora: Towards general world model with natural language actions and video states. *arXiv preprint arXiv:2406.09455*, 2024.

Zhexiao Xiong, Yizhi Song, Hao Kang, Qing Yan, Liming Jiang, Jenson Yang, Zhoujie Fu, Stathi Fotiadis, Angtian Wang, Zichuan Liu, et al. Actworld: From explorable to interactive world model via action-aware memory. *arXiv preprint arXiv:2606.17730*, 2026a.

Zhexiao Xiong, Xin Ye, Burhan Yaman, Sheng Cheng, Yiren Lu, Jingru Luo, Nathan Jacobs, and Liu Ren. Unidrive-wm: Unified understanding, planning and generation world model for autonomous driving. *arXiv preprint arXiv:2601.04453*, 2026b.

Jiwen Yu, Yiran Qin, Xintao Wang, Pengfei Wan, Di Zhang, and Xihui Liu. GameFactory: Creating new games with generative interactive videos. *arXiv preprint arXiv:2501.08325*, 2025.

Hu Yue, Siyuan Huang, Yue Liao, Shengcong Chen, Pengfei Zhou, Liliang Chen, Maoqing Yao, and Guanghui Ren. Ewmbench: Evaluating scene, motion, and semantic quality in embodied world models. *arXiv preprint arXiv:2505.09694*, 2025.

Jiahan Zhang, Muqing Jiang, Nanru Dai, Taiming Lu, Arda Uzunoglu, Shunchi Zhang, Yana Wei, Jiahao Wang, Vishal M Patel, Paul Pu Liang, et al. World-in-world: World models in a closed-loop world. *arXiv preprint arXiv:2510.18135*, 2025.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 586–595, 2018.

Guosheng Zhao, Xiaofeng Wang, Zheng Zhu, Xinze Chen, Guan Huang, Xiaoyi Bao, and Xingang Wang. DriveDreamer-2: LLM-enhanced world models for diverse driving video generation. In *AAAI Conference on Artificial Intelligence*, volume 39, pp. 10412–10420, 2025.

Siyuan Zhou, Yilun Du, Jiaben Chen, Yandong Li, Dit-Yan Yeung, and Chuang Gan. Robodreamer: Learning compositional world models for robot imagination. In *Int. Conf. Mach. Learn.*, pp. 61885–61896. PMLR, 2024.

Zheng Zhu, Xiaofeng Wang, Wangbo Zhao, Chen Min, Nianchen Deng, Min Dou, Yuqi Wang, Botian Shi, Kai Wang, Chi Zhang, et al. Is Sora a world simulator? a comprehensive survey on general world models and beyond. *arXiv preprint arXiv:2405.03520*, 2024.

APPENDIX

This supplementary material provides comprehensive technical documentation, extended dataset analyses, and additional empirical results to support the main text. The appendix is organized into three primary sections: First, Appendix A details the construction of the LCWM-Bench benchmark, encompassing trajectory preprocessing, the automated annotation pipeline, human evaluation protocols, and extended dataset statistics. Second, Appendix B formalizes the *Tiresias* framework, providing exact architectural specifications for the constituent modules (LDP, World Model, Tr-Ranker, and V-Ranker), alongside domain-specific system prompts and inference pseudo-code. Finally, Appendix C outlines the extended experimental setups, including rigorous definitions of the evaluation metrics, training implementation details, and more experimental and qualitative results.

A LCWM-BENCH BENCHMARK DETAILS

A.1 TRAJECTORY SOURCES AND PREPROCESSING

The navigation domain aggregates five ground-robot datasets spanning indoor corridors, unstructured open-world scenes, human-populated social spaces, and off-road terrains. Across all datasets, the navigation action (u_t, ϕ_t) is derived from consecutive relative poses, with displacements normalized by the per-source average step size.

Go Stanford (Hirose et al., 2018). This dataset comprises approximately 24 hours of indoor video (3 Hz) collected by a teleoperated TurtleBot2 equipped with a Ricoh THETA S fisheye camera across 25 buildings. The trajectories primarily feature corridors, doorways, and staircases, providing tight, structurally repetitive indoor layouts. The forward-facing view is rectified to a perspective crop, resized to 256×256, and combined with wheel odometry to compute per-step displacements.

RECON (Shah et al., 2021). This dataset contains over 5,000 autonomous trajectories collected by a Clearpath Jackal via time-correlated random walks. Recorded over 18 months across nine outdoor environments (e.g., parking lots, farmlands, residential areas), the data includes visual observations from a forward-facing 170° fisheye RGB camera.

SCAND (Karnan et al., 2022). SCAND provides 8.7 hours (40 km) of human-teleoperated, socially compliant demonstrations recorded indoors and outdoors. It features two morphologically distinct platforms: a wheeled Clearpath Jackal and a legged Boston Dynamics Spot. The trajectories capture naturally occurring social interactions, such as street crossing and passing pedestrians. Actions are derived from wheel odometry, paired with the front-facing RGB stream.

HuRoN (Hirose et al., 2023). HuRoN comprises 75 hours (58.7 km) of robot navigation across five office environments, capturing over 4,000 human-robot interactions. Unlike SCAND, HuRoN's autonomous data collection policy actively approaches pedestrians, resulting in episodes dense with dynamic obstacles and yielding behaviors.

TartanDrive (Triest et al., 2022). This dataset includes approximately 200,000 off-road interactions (5 hours at 10 Hz) recorded by a Yamaha Viking ATV traversing unstructured terrains (e.g., rocks, mud, steep slopes). Lacking indoor or urban structures and featuring visible robot parts in the camera view, it presents significant visual and dynamic disparities compared to other sources. Thus, TartanDrive is strictly excluded from training and reserved for Val Unseen and test splits to evaluate generalization under severe domain shifts.

The robotics domain involves 6-DoF gripper manipulation and constitutes the largest subset of LCWM-Bench (40%), encompassing household kitchens, offices, tabletop lab setups, and a procedurally generated simulator. Because all four datasets record absolute end-effector poses rather than displacement commands, each episode is converted into per-step increments within the local end-effector frame, defining a 7-dimensional action space $(u_t, z_t, \phi_t, \rho_t, \vartheta_t, g_t)$. Translations are normalized by the per-source average step size. The gripper state is maintained as a binary open/close command and explicitly verified at the terminal step.

DROID (Khazatsky et al., 2024). As the largest robotics dataset in our split, DROID contains 76k demonstration trajectories (350 hours) collected across 564 scenes, 52 buildings, and 86 tasks.

Listing A1: Annotation prompt used to generate navigation instructions.

Role. You are a visual navigation data annotation assistant.

Inputs. A sequence of evenly sampled first-person images, in temporal order; a 2D top-down trajectory map (color from dark to light indicates time; the red arrow indicates yaw heading); trajectory metadata and an angle summary (for reference).

Task. Combine the images and the 2D trajectory geometry to produce three English navigation instructions. The 2D trajectory map is for judging direction changes only. Every instruction must end with a sentence that includes `stop` (not a standalone "Stop"). Do not output your reasoning; output JSON only.

Output styles

> **`concise`.** Consists solely of a minimal sequence of action commands centered on the core movements, direction changes, and stopping location. A small number of concise positional cues may be included to help situate an action, but these should serve the path description rather than expand into descriptions of scenery, objects, or the environment along the way. A phrase that mainly reports what is visible during travel, rather than how to move or where to stop, should generally be omitted unless it is the sole positional information required for a turn or a stop. The overall style should be short and direct, as clean as a command list.

> **`intricate`.** The most detailed style, as detailed as possible: the sequence of actions plus extensive visible visual information and scene detail (objects, people, environmental features, atmosphere, etc.).

> **`landmark`.** Organizes the path description around concrete, recognizable landmarks, stating explicitly which action is performed at which landmark; landmarks should mark the key nodes of the path.

Shared constraints

> Do not invent details that are not visible.
> Each of the three styles must be no more than 100 words.

Reference example

One of several examples provided in context.

> **`concise`:** "Walk straight along the stone-paved walkway and then stop at the end."

> **`intricate`:** "Proceed along the open, stone-paved walkway, following the blue tactile paving strip that runs through its center. A black manhole cover lies on the left, with two pedestrians in dark clothing farther ahead on that side and another group approaching from the right. Dense trees line both sides of the walkway, and sunlight filtering through the canopy casts dappled shadows across the path. Continue forward and stop near the pedestrians ahead."

> **`landmark`:** "Follow the blue tactile paving strip along the stone-paved walkway. Pass the black manhole cover on the left, continue toward the pedestrians ahead, and stop before reaching them."

Demonstrations were gathered using identical Franka Panda hardware across diverse environments, ranging from controlled research labs to real households.

BridgeData V2 (Walke et al., 2023). This dataset comprises 60,096 trajectories collected using a WidowX arm across 24 environments. It features 50,365 human demonstrations spanning 13 skills (e.g., pick-and-place, pushing, folding) and over 100 distinct objects. Scenes are designed to support multiple feasible tasks simultaneously, providing diverse instruction conditioning from identical initial observations.

RT-1 (Brohan et al., 2022). The RT-1 corpus includes approximately 130k episodes collected by 13 mobile manipulators operating in office kitchen environments. The data covers over 700 manipulation tasks, including picking, placing, drawer articulation, and object rearrangement.

LIBERO (Liu et al., 2023a). As the sole simulated dataset in the robotics split, LIBERO provides 130 procedurally generated manipulation tasks categorized into four suites, varying by object types, spatial layouts, and task goals. Similar to TartanDrive, LIBERO is strictly excluded from training and reserved for the Val Unseen and test splits to evaluate generalization under visual and structural distribution shifts.

The 3D gaming domain constitutes 21% of LCWM-Bench and features two distinct embodiments: a first-person avatar in competitive maps and a player character in procedurally generated open

Listing A2: Annotation prompt used to generate robotics instructions.

Role. You are a robot-manipulation visual instruction generation assistant.

Inputs. A single storyboard image formed by concatenating the robot-arm images in temporal order (either a wrist-mounted first-person view or an external third-person view; the model must determine which), plus the per-frame 7-DoF end-effector action ($\Delta x, \Delta y, \Delta z, \Delta$roll, Δpitch, Δyaw, gripper). Combine the image sequence and the action sequence to produce three styles of instruction. Do not output your reasoning; output JSON only.

Analyze the manipulation. Determine whether the viewpoint is wrist (first-person wrist camera) or agentview (fixed third-person camera); identify the key objects visible in the images, their appearance, and their spatial relations; identify the main phases of the action sequence, but treat the action sequence as intent/attempt only, not proof of success; pay particular attention to keyframe roles such as `near_gripper_close_attempt` and `near_gripper_open_release`, as well as the event windows supplied by `event_metadata`; merge consecutive similar actions and ignore minor perturbations.

Organize the styles. Ensure that the three styles are clearly and visibly distinct.

Output styles

`concise`. The shortest and most abstract style, retaining only the core sequence of action verbs (e.g., move forward, grasp, lift, place, rotate, or close drawer), like a pure command list. If evidence of success is insufficient, write attempt, not confirmed, or failed rather than completing it into a successful sequence.

`intricate`. A vivid narrative style describing object appearance (color, material, and shape), the gripper's motion trajectory and state changes, the surrounding workspace, and the visual feedback during execution. It should be clearly richer and more vivid than landmark, like narrating a visualizable manipulation story. It must faithfully reflect whether the manipulation succeeded, explicitly stating uncertainty or failure when evidence is insufficient.

`landmark`. Structures the whole action sequence around the clearly visible, recognizable key objects in the image as spatial and manipulation anchors, with each major step explicitly tied to a specific object (e.g., "move to object X and grasp it" or "place it beside object X"). It should be more concise and structured than intricate, foregrounding objects as reference points. It must not describe a placement as successful when evidence is insufficient.

Shared constraints

Do not invent any details that do not appear in the images.
Each of the three styles must be no more than 100 words.
Every instruction must begin by explicitly stating the viewpoint type:
- · Wrist viewpoint: "From a wrist-mounted first-person view, . . . "
- · Agentview viewpoint: "From an external third-person view, . . . "

Reference example

One of several examples provided in context.

`concise`: "From a wrist-mounted first-person view, pull open the drawer, grasp the black bowl, and place it inside."

`intricate`: "From a wrist-mounted first-person view, a white bowl, a black bowl, a blue box, and a dark bottle rest on the wooden table beside a black drawer. Lower the black robotic gripper toward the gray drawer handle and pull the drawer open. Then raise the gripper, move toward the black bowl positioned above the white bowl, grasp it, lift it from the table, and place it inside the open drawer."

`landmark`: "From a wrist-mounted first-person view, move to the gray handle of the black drawer and pull it open. Then move toward the black bowl positioned above the white bowl, grasp and lift it, and place it inside the open drawer."

worlds. Both incorporate a vertical displacement z_t into the base parameterization, defining a 4-dimensional action space that accounts for jumping, falling, and terrain-following, alongside planar movement and yaw. Unlike other domains, the gaming split is partitioned by map rather than by dataset. Because excluding an entire dataset would eliminate one of the two unique embodiments from the training distribution, specific maps and locations are strictly held out for the Val Unseen split. This strategy preserves embodiment diversity during training while evaluating generalization to novel scene layouts and visual conditions.

Listing A3: Annotation prompt used to generate 3D gaming instructions (Minecraft source).

Role. You are a Minecraft first-person visual navigation data annotation assistant.

Inputs. Minecraft first-person screenshots in temporal order; a 2D top-down trajectory map, trajectory metadata, horizontal travel distance, elevation change, and a turn summary.

Task. Combine the scene visible in the screenshots, their temporal order, and the 2D trajectory geometry to produce three clearly distinct English navigation instructions for the same Minecraft movement trajectory. The 2D trajectory map is for judging direction changes only. Do not output your reasoning; output JSON only.

Coordinates. Each position frame is $[x, y, z]$, where x, y is the horizontal map plane and z is elevation; each delta frame is $[\Delta x, \Delta y, \Delta z, \Delta\text{yaw}]$, with yaw in radians. The trajectory map uses the (x, y) components of position.

Turn rule. Ignore negligible perturbations.

Output styles

`concise.` The shortest and most abstract style, like a tight chain of action commands. It retains only the core movement, left/right turns, necessary key positions, and the final stop. It does not list the blocks, trees, animals, or environmental details seen along the way, and avoids lengthy passing/past process descriptions.

`intricate.` The most detailed of the three styles, clearly longer than concise and landmark. It describes how the action unfolds strictly in temporal order, drawing as much as possible on Minecraft details genuinely visible in the screenshots (terrain, block textures, roads or paths, trees, vegetation, water, buildings, fences, bridges, torches, farmland, animals, sky, and lighting), tying this visual information to the action nodes.

`landmark.` The length and level of detail lie between concise and intricate. The instruction is anchored on the most recognizable Minecraft landmarks or path nodes (a distinctive building, stone tower, bridge, village entrance, fence, torch, farmland, pool, clear forest edge, or distinctive terrain), explicitly stating which action is performed on reaching each landmark. It is more structured and less exhaustive about roadside details than intricate, but provides more landmark-grounding information than concise.

Shared constraints

Each of the three styles must be no more than 100 words.

Reference example

One of several examples provided in context.

`concise:` "Walk straight, turn left at the stone tower, continue straight, and stop beside the oak fence."

`intricate:` "Move forward across the grassy clearing toward the stone tower, keeping the oak trees on your left and the water pool on your right. Continue along the dirt path until the tower fills the center of view, turn left around its corner, and follow the torch-lit path past the wooden fence and small wheat farm. Keep walking toward the bridge, cross it, and stop beside the birch fence at the far end."

`landmark:` "Head toward the stone tower, turn left at its corner, follow the torch-lit path to the bridge, and stop beside the birch fence."

CS:GO (Pearce & Zhu, 2022). This dataset comprises 5.5 million frames ($\sim$ 95 hours) of *Counter-Strike: Global Offensive* gameplay. The data was aggregated from public servers via spectator mode with inferred actions, supplemented by demonstrations. Trajectories exhibit high-speed, reactive dynamics such as strafing, sprinting, and abrupt view rotations, requiring instructions to be grounded in salient architectural references rather than slowly evolving scene content.

Minecraft (Lian et al., 2025). Derived from the LoopNav dataset, this source provides 250 hours (approximately 20 million frames) of loop-based navigation with recorded actions, collected across 147 diverse locations within the open-world environment of *Minecraft*.

The driving domain involves ego-vehicle motion on public roads and features the highest speeds in LCWM-Bench. In addition to the base parameterization, a scalar velocity v_t is introduced, which is critical for disambiguating maneuvers where identical planar displacements result from significantly different vehicle speeds. Across all datasets, multi-view recordings are filtered to retain only the front-facing camera as the agent-centric observation. Native resolutions, varying from 1408×376

Listing A4: Annotation prompt used to generate 3D gaming instructions (CS:GO source).

Role. You are a Counter-Strike: Global Offensive first-person visual navigation data annotation assistant.

Inputs. CS:GO first-person screenshots in temporal order; a 2D top-down trajectory map, with color from dark to light indicating time, a green point marking the start, a red point marking the end, and a red arrow indicating yaw heading; trajectory metadata, travel distance, elevation change, and a turn summary.

Task. Combine the scene genuinely visible in the screenshots, their temporal order, and the trajectory geometry to produce three clearly distinct English navigation instructions for the same CS:GO movement trajectory. The player's arms are typically visible at the bottom of the frame, a crosshair is typically visible at the center, and some segments also show reloading, holding a knife-shaped prop, teammates, or other in-game states. Do not output your reasoning; output JSON only.

CS:GO context requirement. All three instructions must preserve the first-person FPS context and include at least one CS:GO feature genuinely visible in the screenshots. CS:GO features should support the movement instruction, not replace the description of visible map structures such as roads, walls, crates, doors, and corridors.

Coordinates. Each position frame is $[x, y, z]$, where x, y is the horizontal map plane and z is elevation; each delta frame is $[\Delta x, \Delta y, \Delta z, \Delta\text{yaw}]$, with yaw in radians.

Turn rule. The red yaw arrows in the trajectory map are the primary direction signal; the position curve is only for judging the path and stopping location. Do not treat every arrow as a separate turn.

Output styles

`concise.` Minimal and command-like, retaining only the core movement action, direction change, at most one turn/stop positional cue when necessary, and the final stop. A very short CS:GO context phrase may be added, but the scene must not be expanded.

`intricate.` The most detailed of the three styles, clearly narrating in temporal order how the action unfolds, combining the crosshair, teammates, and CS:GO scene details genuinely visible in the screenshots (roads, walls, crates, containers, doors, windows, stairs, corners, corridors, railings, vehicles, vegetation, building edges, lighting, and sky). It states where movement continues, where a turn occurs, how movement continues after the turn, and where it stops. It should make full use of rich screen content but never fabricate anything that is not visible.

`landmark.` Organizes the instruction around the most recognizable CS:GO landmarks or key path nodes visible in the screenshots (crates, a building entrance, a long corridor, stairs, a door, a corner, a wall, a vehicle, or a road intersection), explicitly stating which action is performed on reaching each landmark, while retaining one natural crosshair or teammate context cue. It is more specific than concise and more structured and concise than intricate.

Shared constraints

All three instructions must describe the same real trajectory and must not contradict one another in action order or main turns. The instructions should read like movement orders a CS:GO player would receive rather than generic robot-navigation sentences. Invisible target coordinates, metadata values, or trajectory-map text must not be written directly into the instructions. Elevation change should be mentioned only when stairs, a ramp, a platform, or a clear height difference is visible. If a landmark is unclear, fall back on a visible wall, road, crate, door, or corridor as a conservative reference rather than inventing map structure. Each of the three styles must be no more than 100 words.

Reference example

One of several examples provided in context.

`concise:` "Walk up the stairs, turn right at the top, continue straight, and stop."

`intricate:` "Keep the gun-shaped prop raised while steadily ascending the concrete stairs. At the top, continue alongside the gray textured wall and turn right onto the open platform. Move forward past the crouching player and another character on the right, then stop near the gray box ahead."

`landmark:` "Ascend the stairs, pass the gray wall at the top, and turn right onto the open platform. Continue straight and stop near the gray box ahead."

(KITTI-360) to 1920×1080 (DrivingDojo), are uniformly resized to 256×256. Because raw driving recordings are continuous and substantially exceed the duration of typical instructable maneuvers, scenes are segmented into clips of 8–60 frames, with segments containing reverse motion discarded.

Listing A5: Annotation prompt used to generate driving instructions.

Role. You are an autonomous-driving visual navigation instruction generation assistant.

Inputs. A sequence of evenly sampled, temporally ordered first-person driving images; a 2D top-down trajectory map (color from dark to light indicates time; a red arrow indicates the vehicle's yaw heading); trajectory metadata and a speed summary (for reference and must not be ignored).

Task. Combine the images, the 2D trajectory geometry, and the speed information to produce three English autonomous-driving navigation instructions. The 2D trajectory map is for judging direction changes only. Do not output your reasoning; output JSON only.

Rules. Focus on the overall geometric shape of the trajectory line, clear turns, intersection turns, continuous curves, lane changes, entering or leaving a main road, and the associated speed adjustments (accelerating, decelerating, or holding a steady speed).

If the trajectory contains a clear stopping segment (multiple consecutive frames with speed < 0.5 m/s and negligible displacement), all three styles must mention the stop. If the vehicle resumes travel after stopping, this must be described as continuing or proceeding after stopping. If the speed approaches zero and displacement stops increasing toward the end of the trajectory, all three styles must end with stop or an equivalent expression.

Output styles

> **`concise`.** The shortest and most abstract style, retaining only the core driving action, direction change, and speed action when necessary. It reads like a concise chain of driving commands, foregrounding core events such as go straight, turn left, turn right, merge, change lane, or stop. It does not expand on roadside scene details unless a positional cue is necessary for a turn or a stop.

> **`intricate`.** The most detailed style, narrating clearly in temporal order how the action unfolds and describing as richly as possible the vehicles, pedestrians, road infrastructure, buildings, vegetation, sky, road surface, and lane environment visible in the images. Speed changes should be woven naturally into the narration (e.g., proceeding smoothly, decelerating while approaching a turn, accelerating through an intersection, or holding a moderate speed along the main road). Stopping or stop-and-go behavior should be described with a natural cause (waiting at a red light, yielding, or traffic congestion), together with the process of resuming travel.

> **`landmark`.** Organizes the path around the most salient key landmarks or road nodes (a traffic light, intersection, lane split, crosswalk, construction barrier, gas station, distinctive building, or cluster of parked vehicles), explicitly stating which action is performed at each landmark. It should be more concise and structured than intricate. A stop at a key landmark must be explicitly stated.

Shared constraints

> Do not invent any details that do not appear in the images.
> Each of the three styles must be no more than 100 words.

Reference example: no stop

One of several examples provided in context.

> **`concise`:** "Drive straight ahead, turn right at the intersection, and continue forward."

> **`intricate`:** "Proceed straight along the busy multi-lane road at a steady speed, passing a row of parked cars and storefronts on your right, while several vehicles travel in the adjacent lanes and pedestrians remain visible on the sidewalks. As you approach the large signalized intersection where multiple cars are waiting, decelerate smoothly, turn right onto the side street, and accelerate gently as you continue past the houses and trees."

> **`landmark`:** "Move forward down the main avenue at a steady speed, passing the gas station on your right. At the signalized intersection beside the tall glass building, decelerate, turn right, and accelerate gently along the tree-lined street."

Reference example: with stop

One of several examples provided in context.

> **`concise`:** "Continue toward the traffic light ahead, stop there, then proceed through the intersection and turn right."

> **`intricate`:** "A two-lane residential road extends ahead, with buildings and parked vehicles lining both sides. Continue at a moderate speed toward the intersection, where a red traffic light and a line of waiting cars come into view. Decelerate gradually and stop behind the queued vehicles. When the light turns green, accelerate gently, turn right onto the side street, and continue forward past a row of trees and a construction barrier on your left."

> **`landmark`:** "Approach the traffic light at the upcoming intersection, stop behind the queued vehicles, then turn right once the signal changes and proceed past the construction barrier."

DrivingDojo (Wang et al., 2024b). As the largest dataset in the driving split, DrivingDojo distills approximately 7,500 hours of daily driving video using planning-and-control commands, manually defined rules, and model-assisted labeling. The dataset specifically targets maneuver completeness,

encompassing longitudinal actions (e.g., acceleration, emergency braking, stop-and-go) and lateral actions (e.g., U-turns, overtaking, lane changes), alongside dense multi-agent interactions such as cut-ins and merging (Chi et al., 2026b;a). Planar positions are derived from per-frame camera-to-world extrinsics, yaw is extracted from the horizontal projection of the camera axis, and world-frame displacements are projected onto the current heading to compute signed forward velocity.

Waymo (Sun et al., 2020). The Waymo Open Dataset provides 1,150 scenes (20 seconds each) recorded with a synchronized multi-sensor suite across San Francisco, Phoenix, and Mountain View. Planar position and yaw are extracted directly from the provided vehicle-pose matrices. The onboard sensor-reported vehicle velocity is rotated into the ego frame to isolate the forward component.

KITTI-360 (Liao et al., 2022). KITTI-360 comprises over 300k images of suburban driving sequences, featuring precise georegistered vehicle and camera poses obtained via GNSS/IMU fusion. As the only dataset providing pre-computed positions, headings, per-step increments, and velocities, its released increments are verified against consecutive pose differences and adopted directly.

nuScenes (Caesar et al., 2020). nuScenes consists of 1,000 scenes (20 seconds each) recorded in Boston and Singapore. Notably, it introduces severe environmental variability, including nighttime and rainy conditions, as well as left-hand traffic conventions. Scene boundaries and frame orderings are reconstructed from the released metadata. Translations and rotation quaternions from annotated keyframes are converted into yaw angles, with velocities derived via finite differences. Given its significant disparities in appearance, geography (pairing Asian and North American cities), and driving conventions (where left-hand traffic geometrically inverts turn instructions), nuScenes is excluded from training and reserved for the Val Unseen and test partitions to assess robust generalization.

A.2 Annotation Pipeline

To initialize the automated annotation pipeline, we manually annotated all three instruction styles for a seed set of 150 trajectories, spanning representative scenes across all domains. These expert-curated triplets serve as few-shot exemplars within the domain-specific prompts. During generation, the model is conditioned on multiple contextually relevant references drawn from structurally similar scenes, extending beyond the single illustrative examples provided in the subsequent tables.

Conditioned on these exemplars, Qwen3.6-27B (Qwen Team, 2026) generates all three instruction styles simultaneously for each remaining episode (prompts for each domain are provided in Listings A1–A5). The model input integrates evenly sampled first-person frames with a compact geometric trajectory summary. Specifically, navigation, driving, and gaming domains utilize a 2D top-down path map paired with distance, heading-change, and speed statistics. For robotics, the input comprises a storyboard of concatenated frames alongside the 7-DoF action sequence and gripper states, facilitating continuous object tracking during the manipulation.

Every generated instruction undergoes rigorous human verification against the corresponding video. Annotators manually refine the text to ensure factual correctness and linguistic naturalness, finalizing the three instruction variants released per episode. Furthermore, a random subset of these finalized instructions is independently evaluated by human raters, as detailed in Sec. A.3.

A.3 Human Evaluation of Instruction Quality

To quantify the quality of the released instructions, we conduct a human study on a random subset of LCWM-Bench. We draw 1,200 episodes, approximately balanced across the four domains and the three instruction styles, and recruit volunteer raters from our research group who took no part in the annotation pipeline. For each item, a rater watches the agent-view video, inspects the accompanying 2D or 3D trajectory map, and scores the paired instruction on five dimensions along a six-point scale, where the top score denotes an instruction indistinguishable from one written by human annotators. The dimensions are: (1) *Trajectory–Instruction Alignment*, whether the described path matches the executed one in its turns, distances, and stopping point; (2) *Style Adherence*, whether the instruction stays within its assigned style instead of drifting toward another; (3) *Geometric Correctness*, whether directional references and spatial relations hold from the agent's own viewpoint; (4) *Landmark Accuracy*, whether the referenced visual landmarks exist in the scene and are correctly placed; and (5) *Naturalness & Fluency*, whether the wording reads like language a human guide would actually use. Every episode–instruction pair receives three independent ratings, which we average over raters

and dimensions. The mean over all 1,200 samples is 5.37 out of 6.00, indicating that the released instructions sit close to human-written quality.

A.4 Multi-Style Instruction Examples

Table A1 presents one representative episode per domain, each annotated in all three styles. Because the three styles describe the same episode, they differ only in specificity and emphasis, never in the underlying trajectory. Concise keeps only what is needed to reproduce the path; Landmark-based anchors each maneuver to a named visual feature, such as the yellow archway in navigation or the black sedan in driving; and Intricate embeds the same path in extended scene description, much of which is not required for execution. The three styles are not arbitrary partitions of a single distribution but correspond to distinct grounding demands, and they follow how people actually give directions, which range from terse imperatives to landmark-anchored guidance to running commentary depending on the speaker and on how familiar the listener is assumed to be with the scene (Kolagar & Zarcone, 2024; Kong et al., 2024; Wu et al., 2025). Accordingly, Concise probes whether a model can follow minimal directional language, Landmark-based whether it can ground words in specific visual features, and Intricate whether it can recover control-relevant content from verbose description.

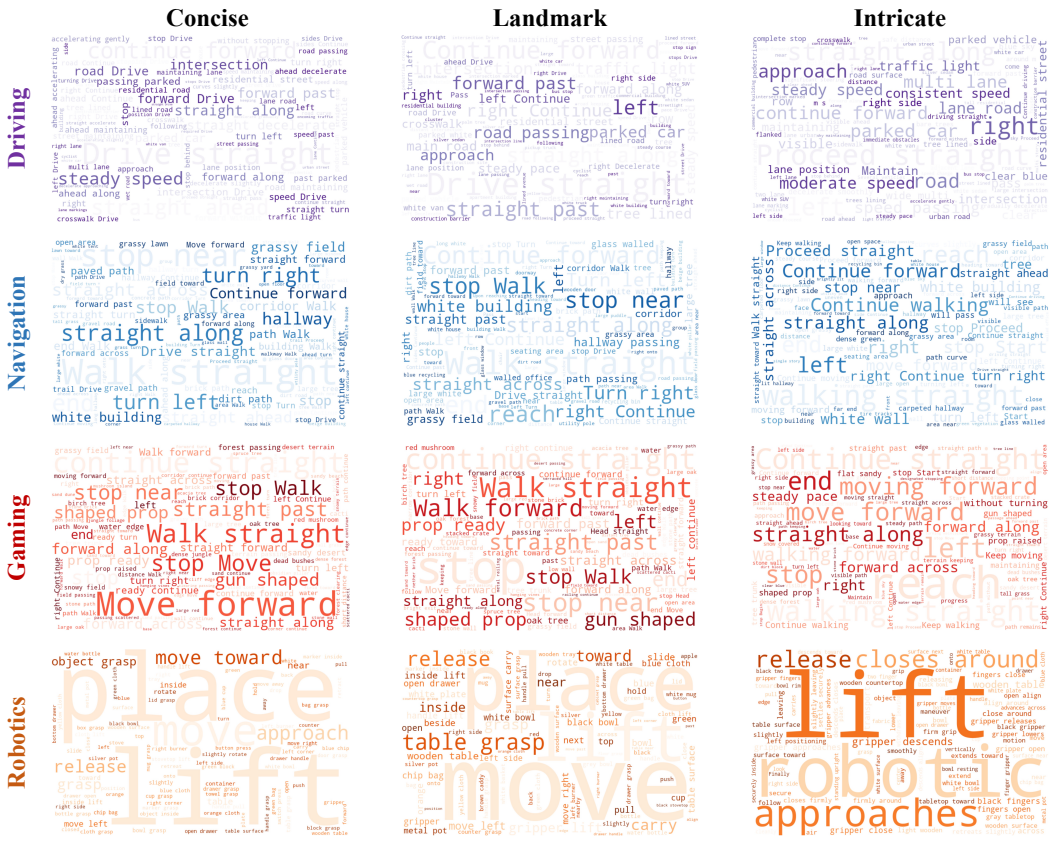


Figure A1: **Word clouds of the released instructions, by domain and instruction style.** Word size reflects frequency within the corresponding style-domain subset of LCWM-Bench.

A.5 More Benchmark Statistics

Table A2 summarizes the full composition of LCWM-Bench: the source dataset, real-vs-simulated status, action dimensionality, and episode/instruction counts for each of the 15 datasets across the four domains. Fig. A1 aggregates the released instructions into per-style word clouds for each domain, complementing the single-episode examples of Table A1 with a corpus-level view of the

Table A1: **Multi-style instruction samples from LCWM-Bench.** One episode per domain, each paired with all three instruction styles. The styles share the same trajectory and differ only in the level of specificity and the degree of visual anchoring, enabling a controlled evaluation of language grounding across domains.

	Style	Instruction
Navigation	Concise	Move forward, turn left and continue straight until you reach the open area, then stop.
	Intricate	Start by moving forward through the open space with a yellow archway on the left and a blue seating area ahead. Follow the path as it curves gently to the left, then continue past the blue seating area and a white pillar. The floor transitions from carpet to a shiny white surface. Proceed into the larger open area with tables and chairs visible in the distance, then stop.
	Landmark	Walk forward past the yellow archway and blue seating area. Follow the path as it curves left, then continue past the white pillar onto the shiny white floor. Proceed into the open area with tables and stop.
Robotics	Concise	From an external third-person view, grasp and lift one corner of the blue cloth, then fold it diagonally.
	Intricate	From an external third-person view, the robot arm approaches a light blue cloth spread flat on a wooden table near several colorful toy vegetables. The gripper targets the bottom-right corner, closes around the fabric, and lifts it upward. The arm then moves diagonally across the table, drawing the corner toward the top-left side. As the cloth is pulled, it folds over itself into a triangular shape. The gripper releases the fabric, leaving the cloth partially folded on the table.
	Landmark	From an external third-person view, move to the bottom-right corner of the blue cloth on the wooden table, grasp it, and pull it diagonally toward the top-left corner to fold the fabric.
3D Gaming	Concise	Move straight along the jungle path and stop when the foliage opens up.
	Intricate	Begin by moving forward through the thick, green jungle canopy, navigating past the large, vine-covered tree trunk immediately on your left. Keep walking straight along the narrow, leafy path, maintaining a steady pace as the dense foliage surrounds you. As you progress, the path remains clear directly ahead, with hanging vines and lush vegetation on both sides. Continue forward until the immediate path ahead becomes slightly more open, then stop in the clearing.
	Landmark	Walk forward past the large vine-covered tree on the left, continue straight through the dense jungle path, and stop when the foliage opens up slightly ahead.
Driving	Concise	Drive straight ahead, accelerating gently through the intersection and continuing up the hill.
	Intricate	Start by driving straight through the wide intersection under green traffic lights, accelerating gently behind a black sedan. Continue uphill along the straight, tree-lined residential street, with parked cars and buildings lining both sides. Maintain a steady, moderate speed and a safe distance from the sedan as you proceed under the clear blue sky.
	Landmark	Proceed straight through the intersection beneath the green traffic lights, then continue up the steep hill between rows of parked cars and residential buildings at a steady pace.

same three-way separation. Across all four domains, Concise instructions are dominated by the motion and control verbs that name a maneuver directly, such as *turn right*, *move forward*, and *grasp*, *lift*, *release* in robotics: the vocabulary needed to reproduce the trajectory and little else. Landmark-based instructions carry the same verbs but pair them with frequent, concrete object nouns, such as *white building*, *parked car*, and *wooden table*, reflecting the explicit tie between each maneuver and a recognizable anchor that the style is designed to enforce. Intricate instructions retain this vocabulary as well, but surround it with descriptive modifiers and scene qualifiers that are largely absent from the other two styles, such as *consistent speed*, *carpeted hallway*, and *gripper closes around*, consistent with the extended descriptions of surrounding layout and appearance. This pattern shows that the three styles differ not only in length, as Fig. 2d already reports, but in the composition of vocabulary.

Table A2: Composition of LCWM-Bench. Action variant follows the base parameterization (u_t, ϕ_t) of Sec. 3.1, with domain-specific channels (z_t, v_t, or ρ_t, ϑ_t, g_t) appended where applicable. Every episode is paired with three instruction styles (concise, intricate, landmark), yielding three instruction-conditioned samples per episode.

Domain	Dataset	Source	Action variant	Dim.	#Episodes	#Instructions
Navigation	GoStanford	Real	(u, ϕ)	3	10,080	30,240
	RECON	Real	(u, ϕ)	3	10,392	31,176
	SCAND	Real	(u, ϕ)	3	5,427	16,281
	HuRoN	Real	(u, ϕ)	3	9,346	28,038
	TartanDrive	Real	(u, ϕ)	3	5,896	17,688
Robotics	BridgeData V2	Real	$+ z, \rho, \vartheta, g$	7	10,000	30,000
	DROID	Real	$+ z, \rho, \vartheta, g$	7	25,172	75,516
	LIBERO	Sim	$+ z, \rho, \vartheta, g$	7	11,776	35,328
	RT-1	Real	$+ z, \rho, \vartheta, g$	7	10,000	30,000
3D Gaming	Minecraft (LoopNav)	Sim	$+ z$	4	24,930	74,790
	CS:GO	Sim	$+ z$	4	5,366	16,098
Driving	DrivingDojo	Real	$+ v$	4	6,805	20,415
	KITTI-360	Real	$+ v$	4	1,984	5,952
	Waymo	Real	$+ v$	4	4,096	12,288
	nuScenes	Real	$+ v$	4	1,717	5,151
Total	15 datasets	12 real / 3 sim	–	3–7	142,987	428,961

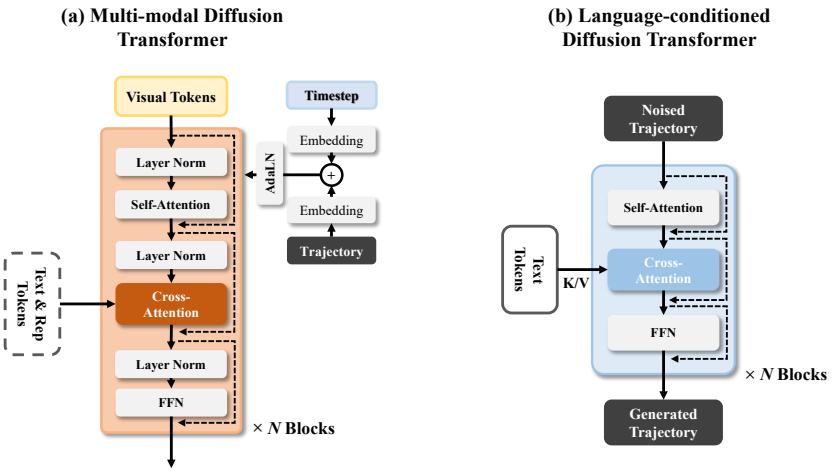


Figure A2: **Architectures of two trained modules of *Tiresias*. (a)** The world model's multimodal diffusion Transformer (Sec. B.3). **(b)** LDP's language-conditioned diffusion Transformer (Sec. B.2).

The domain-specific vocabulary within each style further reflects the action space of Sec. 3.1 together with each domain's typical scene content. Driving instructions cluster around traffic elements such as intersections, crosswalks, and parked cars, alongside speed-related terms tied to the velocity channel unique to that domain. Navigation vocabulary is architectural, dominated by terms like hallway, corridor, and building. Gaming instructions reference terrain and vegetation, such as grassy field, oak tree, and birch tree. Robotics instructions stand apart from other three domains at the vocabulary level: they are dominated by contact verbs tied to gripper, such as grasp and release.

Table A3: **System prompts used to condition the language-conditioned world model.** For each domain, the system prompt and the episode instruction are concatenated into one text sequence, encoded by the frozen umT5 encoder, and supplied to the world model. Prompts are fixed per domain and identical at training and inference time.

Navigation	**System**: You are a language-conditioned navigation world model. One initial observation is provided together with the route instruction and the full action sequence. Generate the complete future agent-view video in one shot. Treat the instruction as the route semantics and the action sequence as transition control for timing, direction, and motion scale. Keep motion physically plausible, temporally continuous, and faithful to both instruction and action. **Instruction**: {language instruction}
Robotics	**System**: You are a language-conditioned robot-manipulation world model. One initial observation is provided together with the manipulation instruction and the full 7-DoF end-effector action sequence. Generate the complete future video in one shot. Preserve the viewpoint stated at the beginning of the instruction: a wrist-mounted first-person view must remain wrist-mounted first-person, and an external third-person view must remain external third-person. In an external third-person view, camera extrinsics are fixed: never pan, tilt, zoom, dolly, orbit, or track the robot. In a wrist-mounted first-person view, the camera moves only because it is rigidly attached to the robot wrist. The actor is a robotic arm with a mechanical gripper; never replace it with a human hand, human arm, skin, or fingers. Preserve whether the initial scene is real or simulated rather than changing visual domain. Treat the instruction as task and contact semantics, and the action sequence as transition control for end-effector translation, rotation, and gripper opening or closing. Maintain robot identity, geometry, kinematic continuity, object permanence, contact, grasp, lift, transport, placement, and release. **Instruction**: {language instruction}
3D Gaming	**System**: You are a language-conditioned first-person gaming world model. One initial observation is provided together with the gameplay instruction and the full ego-action sequence. Generate the complete future first-person gaming video in one shot. Treat the instruction as task semantics and the action sequence as transition control for forward motion, lateral strafing, yaw rotation (looking left/right), and vertical movement (jumping, falling, stairs). Keep scene geometry, player motion, lighting, and object appearance physically plausible and temporally continuous. Be faithful to both instruction and action. **Instruction**: {language instruction}
Driving	**System**: You are a language-conditioned self-driving world model. One initial front-facing observation is provided together with the route instruction and the full ego-action sequence. Generate the complete future driving video in one shot. Treat the instruction as route semantics and the action sequence as transition control for speed, direction, and motion scale. Keep road geometry, ego-motion, traffic, and lighting physically plausible and temporally continuous. Be faithful to both instruction and action. **Instruction**: {language instruction}

B *Tiresias* Details

B.1 System Prompts

The world model is conditioned on a domain-specific system prompt prepended to the episode instruction. Following the template in Table A3, the concatenated text sequence is encoded by the frozen umT5 encoder and injected into the world model via cross-attention as the instruction embedding. This prompt is fixed per domain across all episodes and remains strictly identical during training and inference.

B.2 LDP Details

As illustrated in Fig. A2b, the Language-conditioned Diffusion Policy (LDP) denoises a sampled action sequence into a valid trajectory guided solely by language. Each of its N blocks sequentially applies self-attention over the action tokens, cross-attention using the frozen umT5 (Chung et al., 2023) instruction embeddings as keys and values, and a feed-forward layer, with residual

connections throughout. Notably, apart from the diffusion timestep, this cross-attention mechanism serves as the exclusive conditioning source; visual inputs are strictly excluded. This scene-agnostic design ensures that sampling the LDP with independent noise seeds generates diverse, instruction-compliant trajectories rather than observation-constrained motions. Consequently, this produces a sufficiently diverse candidate pool for the subsequent ranking stage, effectively deferring the scene-dependent evaluation to the world model and V-Ranker. LDP is trained from scratch by denoising expert trajectories sampled from the LCWM-Bench training split.

B.3 Language-Conditioned World Model Details

As illustrated in Fig. A2(a), the world model adapts the Wan2.2 (Wan et al., 2025) multimodal diffusion Transformer while preserving its native interfaces. Frames are encoded into visual tokens by a frozen VAE (Wan et al., 2025) and subsequently noised. Each transformer block sequentially applies self-attention, cross-attention, and a feed-forward layer, interleaved with layer normalization and residual connections. Only transformer parameters are updated during training.

The two conditioning signals described in Sec. 4.2 are injected via distinct mechanisms, reflecting their respective structural roles. The candidate trajectory is embedded by a lightweight MLP, added to the timestep embedding, and injected via AdaLN, enabling it to modulate all visual tokens uniformly at each denoising step. In contrast, the instruction and scene features serve as keys and values in the cross-attention layers, formed by concatenating the prompted umT5 (Chung et al., 2023) tokens with the DINOv3 (Siméoni et al., 2025) representation tokens of the initial frame.

This architectural decoupling aligns with the distinct nature of the signals: the trajectory is a low-dimensional, temporally aligned control signal that governs macroscopic ego-motion, which is effectively captured by uniform modulation. Conversely, grounding fine-grained semantics (e.g., *the yellow archway*) necessitates dense, patch-level alignment between text tokens and visual features, naturally facilitated by cross-attention. The model is trained using the flow-matching objective detailed in Sec. 4.2.

B.4 Tr-Ranker Details

The Tr-Ranker evaluates a candidate trajectory based solely on language and motion, operating without visual rendering. It processes the frozen umT5 instruction tokens alongside a candidate action sequence generated by the LDP. The actions are first projected into the model dimension and augmented with temporal positional embeddings to preserve maneuver ordering. These action tokens then query the instruction embeddings via cross-attention. Subsequently, a temporal Transformer aggregates the outputs into a unified representation per candidate, from which four distinct heads predict the evaluation metrics: SR, nDTW, ATE, and RPE.

Training labels are generated offline by evaluating each training-time candidate against its corresponding ground-truth trajectory. The classification head (SR) is optimized using cross-entropy, while the three continuous metric heads employ a Huber loss to mitigate the impact of severe outliers. During inference, the predictions from these heads are normalized to a common scale and fused into the final language-side score $S_{\mathrm{tr}}^{(m)}$, where SR and nDTW are incorporated as positive rewards, and ATE and RPE as penalties.

B.5 V-Ranker Details

The V-Ranker evaluates each candidate through the future states predicted by the frozen world model, without directly accessing the candidate action sequence. For candidate m, the world model generates a sequence of video latents $\hat{X}^{(m)}$. The anchor frame is excluded from the rollout descriptor so that it summarizes the imagined future. For each latent channel, we compute the temporal mean, standard deviation, and mean absolute frame-to-frame difference, summarizing appearance and temporal variation. The resulting descriptor is projected to a shared dimension together with three contextual features: the instruction embedding e_I, the scene semantic features, and a learned domain-specific embedding. The fusion MLP receives the concatenation of these representations, augmented with the element-wise products between the rollout descriptor and the projected

instruction and scene features. Within an episode, contextual features are shared across candidates; candidate-specific evidence enters V-Ranker only through the imagined rollout.

Finally, four heads predict an overall score and three auxiliary scores measuring language consistency, semantic consistency, and motion validity. Their outputs are combined with weights of 1:0.5:0.5:0.25, respectively, to obtain the final visual score $S_{\mathrm{v}}^{(m)}$, assigning smaller coefficients to the auxiliary scores than to the overall score.

B.6 Pseudo-Code for Inference

Alg. 1 outlines the selective-imagination pipeline of Sec. 4.4. Because all four modules are trained independently and kept frozen, inference is a direct composition without joint fine-tuning.

The two stages process the language instruction differently. Stage one encodes the instruction alone; as the LDP and Tr-Ranker are scene-agnostic, they evaluate only whether a trajectory aligns with the text command. In contrast, stage two prepends a domain-specific system prompt to the instruction, providing the world model with the embodiment context required to simulate domain-appropriate futures. This pipeline relies on a structural asymmetry in computational cost. The Tr-Ranker, operating without visual inputs, efficiently scores the entire pool of N candidates. The world model is then invoked exclusively for the top-k candidates. Restricting simulation to this shortlist ensures that ranked imagination remains computationally efficient compared to evaluating every proposal.

Algorithm 1: *Tiresias*: Inference with Selective Imagination

Require: observation o_s, instruction I, domain d
Require: frozen LDP π, world model p_θ, Tr-Ranker, V-Ranker
Require: pool size N, shortlist size k, weights $\lambda_{\mathrm{tr}}, \lambda_{\mathrm{v}}$
Ensure: executed trajectory $\hat{\mathcal{A}}^*$

$x_s \leftarrow \mathrm{VAE}(o_s), \quad f_s \leftarrow \mathrm{DINOv3}(o_s)$
{Stage one: propose and rank from the instruction alone}
$T_I \leftarrow \mathrm{umT5}(I)$
for $m = 1$ to N **do**
  $\hat{\mathcal{A}}^{(m)} \sim \pi(\cdot \mid T_I;\, \epsilon_m)$
  $S_{\mathrm{tr}}^{(m)} \leftarrow$ Tr-Ranker$(\hat{\mathcal{A}}^{(m)}, T_I)$
end for
$\mathcal{S} \leftarrow$ indices of the top-k scores in $\{S_{\mathrm{tr}}^{(m)}\}_{m=1}^{N}$
{Stage two: imagine only the shortlisted candidates}
$e_I \leftarrow \mathrm{umT5}(\mathrm{prompt}_d \oplus I)$
for $m \in \mathcal{S}$ **do**
  $\hat{X}^{(m)} \sim p_\theta\big(\cdot \mid x_s,\, [f_s; e_I],\, \hat{\mathcal{A}}^{(m)}\big)$
  $S_{\mathrm{v}}^{(m)} \leftarrow$ V-Ranker$(\hat{X}^{(m)}, e_I, d)$
end for
normalize S_{tr} and S_{v} within $\mathcal{S}$
$m^* \leftarrow \arg\max_{m \in \mathcal{S}} \; \lambda_{\mathrm{tr}} S_{\mathrm{tr}}^{(m)} + \lambda_{\mathrm{v}} S_{\mathrm{v}}^{(m)}$
return $\hat{\mathcal{A}}^* = \hat{\mathcal{A}}^{(m^*)}$

C Experiments Details

C.1 Evaluation Metric Details

A predicted action sequence is forward-integrated into a position trajectory by composing each planar displacement with the accumulated yaw, and the reference trajectory is obtained the same way from the ground-truth actions. Navigation and driving are compared in the plane, $p_t = (x_t, y_t) \in \mathbb{R}^2$; 3D gaming and robotics are compared in space, $p_t = (x_t, y_t, z_t) \in \mathbb{R}^3$. $P = (p_1, \ldots, p_{T_{\mathrm{gt}}})$ and $\hat{P} = (\hat{p}_1, \ldots, \hat{p}_{\hat{T}})$ denote the reference and predicted trajectories, and $\delta_{\mathcal{D}}$ represents average step size of domain $\mathcal{D}$.

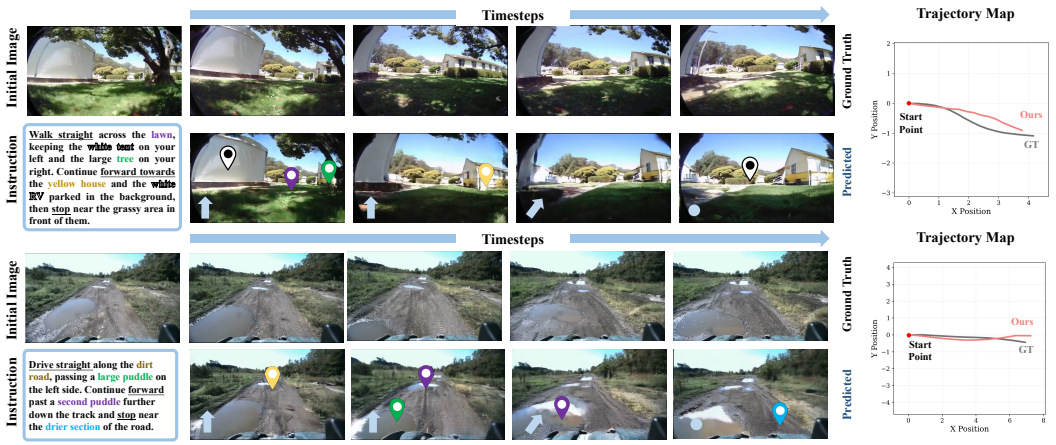


Figure A3: **More qualitative results on val_seen and val_unseen navigation scenarios.** For each case, we show the initial observation and instruction, the imagined rollout of the selected plan against the ground-truth future, and the predicted trajectory against the reference.

SR. Success Rate (SR) is a binary outcome at the terminal state: the episode succeeds if the Euclidean distance between the predicted and reference endpoints is within the domain's average step size. In robotics, success additionally requires the terminal gripper state to match the reference, since reaching the right position while releasing at the wrong time or place does not complete the manipulation task.

nDTW (Ilharco et al., 2019). We align $\hat{P}$ and P with dynamic time warping and take the minimum cumulative Euclidean cost $DTW(\hat{P}, P)$ over order-preserving matches:

$$\text{nDTW} = \exp\left(-\frac{DTW(\hat{P}, P)}{\delta_{\mathcal{D}} T_{\text{gt}}}\right), \tag{A1}$$

normalized to 1 for a perfect match and decaying toward 0 as the paths diverge, in units of average steps per reference step so it stays comparable across domains with very different motion scales. Because the alignment is monotonic but not one-to-one, nDTW tolerates a different speed or step count along the same route while still penalizing the wrong route.

ATE and RPE (Sturm et al., 2012). Both are computed over the temporally corresponding steps of the two trajectories, with no start-point, end-point, Procrustes, or DTW alignment applied beforehand. ATE averages the position error between corresponding steps and reflects global drift; RPE averages the error between corresponding step-to-step displacements and reflects local motion error. The two metrics are complementary: a plan with accurate per-step motion can still accumulate large ATE, and reporting both separates local control error from the long-horizon drift.

Imagination Metrics. To evaluate simulation performance, SSIM, PSNR, and LPIPS (Zhang et al., 2018), and DreamSim (Fu et al., 2023), are reported at two horizons: @1 compares the single predicted next frame to the ground truth, and @10 evaluates a ten-step open-loop rollout. FVD (Unterthiner et al., 2019) compares distributions of complete sequences and is reported at the horizon of @32.

C.2 Implementation Details

Diamond (Alonso et al., 2024). We select Diamond as a world-model baseline representative of UNet-based diffusion architectures, adopting its best-performing configuration following the official implementation. Continuous actions are incorporated through a linear embedding layer for conditioning. To jointly support action prediction and imagination, we pair Diamond with LCVN-AC (Dong et al., 2026), training LCVN-AC from scratch on the LCWM-Bench train split, with Diamond as the world model.



Figure A4: **More qualitative results on driving scenarios.** For each case, we show the initial observation and instruction, the imagined rollout of the selected plan against the ground-truth future, and the predicted trajectory against the reference.



Figure A5: **More qualitative results on driving scenarios.** For each case, we show the initial observation and instruction, the imagined rollout of the selected plan against the ground-truth future, and the predicted trajectory against the reference.

NWM (Bar et al., 2025). We select NWM as a second world-model baseline, representative of DiT-based diffusion architectures. We adopt NWM's best reported configuration, a context length of 4 with a CDiT-XL backbone (1B parameters), training it from scratch on the LCWM-Bench train split. At inference we follow the original paper and use 250 diffusion denoising steps. As with Diamond, we pair NWM with LCVN-AC (Dong et al., 2026) to jointly support planning and simulation, training LCVN-AC from scratch on the LCWM-Bench train split with NWM as the world model.

LCVN-WM (Dong et al., 2026). We adopt the official implementation and training recipe of LCVN-WM, retraining it from scratch on the LCWM-Bench train split under our shared action interface. Following their configuration, an ImageVAE (Kingma & Welling, 2013) first compresses input images into 32×32 latent representations, and a DiT-XL backbone (approximately 1B parameters, hidden size 1152) is trained with diffusion forcing (Chen et al., 2024) so that each frame retains an independent noise level. We use the AdamW optimizer (Loshchilov & Hutter, 2017) with linear warmup and a constant learning rate of 1×10^{-4}, and sample with the deterministic DDIM sampler (Song et al., 2020) for 50 steps, following (Dong et al., 2026).

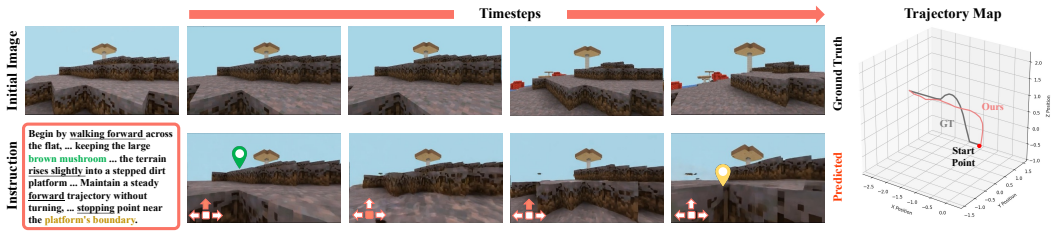


Figure A6: **More qualitative results on 3D gaming scenarios.** For each case, we show the initial observation and instruction, the imagined rollout of the selected plan against the ground-truth future, and the predicted trajectory against the reference.

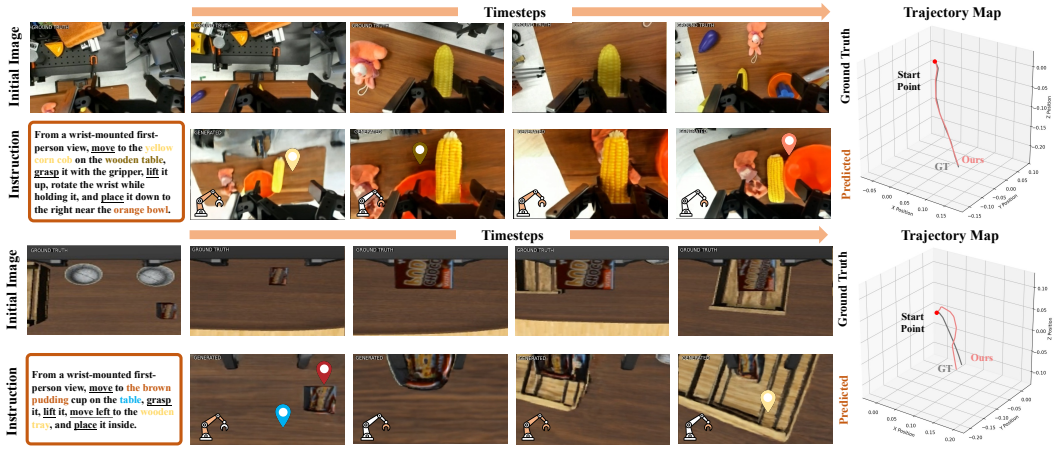


Figure A7: **More qualitative results on first-person-view robotics scenarios.** For each case, we show the initial observation and instruction, the imagined rollout of the selected plan against the ground-truth future, and predicted trajectory against reference.

LCVN-Uni (Dong et al., 2026). We adopt the official implementation of LCVN-Uni, which fine-tunes GAIR Anole-7B (Chern et al., 2024) with a 4096-token context while keeping the bin, text, and image tokenizers frozen. We retrain it on the LCWM-Bench train split under their reported configuration: input images are resized to 448×448 (height $\times$ width) and discretized into 784 visual tokens for context sizes of 1 or 2, and 625 tokens for a context size of 4. Following their setup, only the LoRA (Hu et al., 2022) adapters (rank 16) in the Transformer's *qkv* projections are updated (Liu et al., 2023b). Optimization uses AdamW for 20 epochs at a learning rate of 2×10^{-4}, on 8×NVIDIA A100 GPUs (80GB each) with a global batch size of 8.

Tiresias. The world model is initialized from Wan2.2 and the LDP is trained from scratch; umT5, DINOv3, and the VAE remain frozen throughout training. Visual encoders process all observations at a uniform 256×256 resolution, and scene features are extracted using a frozen DINOv3 ViT-L/16 distilled encoder (300M parameters). Training and evaluation run on Ubuntu 24.04.4 LTS with Python 3.10.20, PyTorch 2.4.1, and CUDA 11.8, on 8×NVIDIA A100 80GB PCIe GPUs (driver 580.173.02). The world model is trained for 10 epochs on LCWM-Bench. At inference, the LDP samples N=128 candidates using fixed noise seeds, the Tr-Ranker shortlists k=8 candidates, and the two evaluation scores are fused with a ratio of λ_{tr}:$\lambda_{\text{v}} = 2$:1.

C.3 More Experimental Results

Full Results of the Candidate Funnel Ablation. Table A4 reports the complete numerical results behind Fig. 6(b–d) in the main paper. The table lists all four planning metrics (SR, nDTW, ATE, and RPE) with the standard deviation over runs as a subscript. The top block enumerates the full grid of funnel shapes $N \to k \to 1$ over pool sizes $N \in \{16, \ldots, 512\}$ and shortlist sizes $k \in \{1, \ldots, 64\}$; the middle block compares Tr-Ranker against random shortlisting at matched imagination budgets



Figure A8: **More qualitative results on third-person-view robotics scenarios.** For each case, we show the initial observation and instruction, the imagined rollout of the selected plan against ground-truth future, and predicted trajectory against reference.

with $N{=}128$; and the bottom block sweeps the fusion weights $\lambda_{\text{tr}}{:}\lambda_{\text{v}}$ under the default $128 \to 8$ funnel. The $N \to 1$ rows correspond to selection with the Tr-Ranker alone (equivalently, fusion weight 1:0), and the $128 \to k$ rows together with the Random $\to k$ rows provide the Final SR columns of Table 6. The trends are consistent with those discussed in Sec. 5: performance improves with larger pools and shortlists but saturates around $N{=}128$ and $k{=}8$, Tr-Ranker shortlisting outperforms random shortlisting even at twice the imagination budget, and the 2:1 fusion yields the highest SR among all weightings. Beyond SR, the additional metrics show that the improvements in goal completion are accompanied by consistent reductions in trajectory error: ATE and RPE decrease as the funnel widens, confirming that ranked imagination improves the executed path as a whole rather than only the terminal position.

Performance–Latency Trade-off. Table A5 reports planning performance and inference latency. The default $128 \to 8 \to 1$ pipeline achieves an SR of 0.60 with a latency of 36 s. Reducing the shortlist to two candidates lowers latency to 9 s, with an SR of 0.55, illustrating the trade-off between planning performance and imagination cost. Results for NoMaD (lang) and LCVN-Uni provide additional reference points under their respective configurations.

Text Encoder. To assess sensitivity to the text encoder, we replace all umT5 text encoders in *Tiresias* with CLIP text encoders. This variant achieves an overall LCWM-Bench score of 62 on val seen, compared with 63 for default configuration. The small performance gap suggests that our framework remains effective with an alternative text encoder and that its gains do not depend solely on umT5.

C.4 More Qualitative Results

Figs. A3–A8 provide additional cases covering all four domains, both robotics viewpoints, and both real and simulated sources. Each panel includes the initial observation and instruction, the imagined rollout against the ground-truth future, and the predicted trajectory against the reference.

Navigation. Fig. A3 pairs a wide-angle campus scene with an off-road track held out entirely from training (Sec. 3.2). The rollouts place named landmarks on the sides the instruction specifies and preserve the source appearance, including mud and standing water characteristic of unseen domain.

Table A4: Ablation on the **candidate funnel** on val seen. *Top*: $N \to k \to 1$ samples N candidates, imagines the Tr-Ranker top-k, and selects one trajectory. *Middle*: Tr-Ranker versus random shortlisting at matched imagination budgets. *Bottom*: ranker score fusion with the default $128 \to 8$ funnel. Bold marks the default configuration.

Setting	SR↑	nDTW↑	ATE↓	RPE↓
Funnel shape $N \to k \to 1$ (stage one: Tr-Ranker)				
$16 \to 1$	$0.508_{\pm 0.022}$	$0.604_{\pm 0.005}$	$0.797_{\pm 0.010}$	$0.155_{\pm 0.001}$
$16 \to 4$	$0.545_{\pm 0.016}$	$0.622_{\pm 0.005}$	$0.791_{\pm 0.011}$	$0.152_{\pm 0.003}$
$16 \to 8$	$0.560_{\pm 0.032}$	$0.636_{\pm 0.002}$	$0.766_{\pm 0.001}$	$0.146_{\pm 0.001}$
$32 \to 1$	$0.516_{\pm 0.002}$	$0.610_{\pm 0.007}$	$0.786_{\pm 0.013}$	$0.153_{\pm 0.002}$
$32 \to 4$	$0.552_{\pm 0.005}$	$0.628_{\pm 0.006}$	$0.783_{\pm 0.009}$	$0.150_{\pm 0.002}$
$32 \to 8$	$0.565_{\pm 0.011}$	$0.641_{\pm 0.004}$	$0.762_{\pm 0.008}$	$0.145_{\pm 0.002}$
$32 \to 16$	$0.570_{\pm 0.012}$	$0.646_{\pm 0.006}$	$0.757_{\pm 0.013}$	$0.143_{\pm 0.002}$
$64 \to 1$	$0.524_{\pm 0.005}$	$0.620_{\pm 0.004}$	$0.778_{\pm 0.005}$	$0.148_{\pm 0.001}$
$64 \to 4$	$0.556_{\pm 0.005}$	$0.641_{\pm 0.004}$	$0.770_{\pm 0.006}$	$0.145_{\pm 0.001}$
$64 \to 8$	$0.578_{\pm 0.017}$	$0.648_{\pm 0.005}$	$0.731_{\pm 0.007}$	$0.139_{\pm 0.002}$
$64 \to 16$	$0.584_{\pm 0.010}$	$0.652_{\pm 0.004}$	$0.726_{\pm 0.007}$	$0.136_{\pm 0.001}$
$64 \to 32$	$0.588_{\pm 0.012}$	$0.654_{\pm 0.005}$	$0.722_{\pm 0.010}$	$0.134_{\pm 0.001}$
$128 \to 1$	$0.530_{\pm 0.020}$	$0.640_{\pm 0.004}$	$0.770_{\pm 0.007}$	$0.132_{\pm 0.002}$
$128 \to 4$	$0.580_{\pm 0.019}$	$0.648_{\pm 0.005}$	$0.726_{\pm 0.007}$	$0.131_{\pm 0.002}$
$\mathbf{128 \to 8}$	$\mathbf{0.602_{\pm 0.013}}$	$\mathbf{0.658_{\pm 0.004}}$	$\mathbf{0.685_{\pm 0.005}}$	$\mathbf{0.123_{\pm 0.001}}$
$128 \to 16$	$0.608_{\pm 0.010}$	$0.660_{\pm 0.004}$	$0.682_{\pm 0.006}$	$0.121_{\pm 0.001}$
$128 \to 32$	$0.606_{\pm 0.026}$	$0.659_{\pm 0.004}$	$0.684_{\pm 0.007}$	$0.122_{\pm 0.002}$
$128 \to 64$	$0.603_{\pm 0.022}$	$0.658_{\pm 0.003}$	$0.686_{\pm 0.008}$	$0.123_{\pm 0.002}$
$256 \to 1$	$0.534_{\pm 0.026}$	$0.645_{\pm 0.005}$	$0.765_{\pm 0.010}$	$0.130_{\pm 0.002}$
$256 \to 8$	$0.606_{\pm 0.025}$	$0.662_{\pm 0.005}$	$0.683_{\pm 0.006}$	$0.120_{\pm 0.002}$
$256 \to 16$	$0.607_{\pm 0.018}$	$0.666_{\pm 0.004}$	$0.680_{\pm 0.005}$	$0.117_{\pm 0.001}$
$256 \to 32$	$0.611_{\pm 0.022}$	$0.667_{\pm 0.005}$	$0.678_{\pm 0.006}$	$0.117_{\pm 0.001}$
$512 \to 1$	$0.535_{\pm 0.028}$	$0.646_{\pm 0.005}$	$0.764_{\pm 0.010}$	$0.129_{\pm 0.002}$
Stage-one selector at matched imagination budget (N=128)				
Random $\to 4$	$0.270_{\pm 0.025}$	$0.550_{\pm 0.002}$	$1.050_{\pm 0.007}$	$0.160_{\pm 0.001}$
Random $\to 8$	$0.390_{\pm 0.016}$	$0.590_{\pm 0.003}$	$0.980_{\pm 0.016}$	$0.140_{\pm 0.001}$
Random $\to 16$	$0.460_{\pm 0.013}$	$0.620_{\pm 0.002}$	$0.930_{\pm 0.002}$	$0.130_{\pm 0.001}$
$\mathbf{Tr \to 8}$ (default)	$\mathbf{0.602_{\pm 0.013}}$	$\mathbf{0.658_{\pm 0.004}}$	$\mathbf{0.685_{\pm 0.005}}$	$\mathbf{0.123_{\pm 0.001}}$
Fusion weights $\lambda_{\mathrm{tr}} : \lambda_{\mathrm{v}}$ ($128 \to 8$)				
$1:0$	$0.530_{\pm 0.020}$	$0.640_{\pm 0.004}$	$0.770_{\pm 0.007}$	$0.132_{\pm 0.002}$
$\mathbf{2:1}$	$\mathbf{0.602_{\pm 0.013}}$	$\mathbf{0.658_{\pm 0.004}}$	$\mathbf{0.685_{\pm 0.005}}$	$\mathbf{0.123_{\pm 0.001}}$
$1:1$	$0.570_{\pm 0.018}$	$0.650_{\pm 0.004}$	$0.730_{\pm 0.006}$	$0.120_{\pm 0.001}$
$1:2$	$0.520_{\pm 0.020}$	$0.640_{\pm 0.005}$	$0.810_{\pm 0.008}$	$0.130_{\pm 0.002}$
$0:1$	$0.470_{\pm 0.011}$	$0.610_{\pm 0.006}$	$0.950_{\pm 0.010}$	$0.150_{\pm 0.003}$

Table A5: **Planning performance and inference latency** on val seen. Latency measures the time to produce the final action trajectory, including candidate generation and selection where applicable. For *Tiresias*, $N \to k \to 1$ imagines k shortlisted candidates before selecting one.

Method	Configuration	SR↑	Latency (s)↓
NoMaD (lang)	–	0.32	0.7
LCVN-Uni	–	0.47	85
Tiresias	$128 \to 8 \to 1$ (default)	0.60	36
Tiresias	$128 \to 2 \to 1$	0.55	9

3D Gaming. Fig. A6 shows an episode whose terrain rises into a stepped platform, exercising the vertical channel that this domain adds in Sec. 3.1. The elevation change is visible in both the rollout and the 3D trajectory, and block-level geometry stays consistent as the camera advances.

Driving. Figs. A4 and A5 span daylight residential streets, fog, and night. Adverse conditions are preserved rather than normalized away, with fog density, nighttime illumination, and road markings

all remaining plausible across the horizon, and the speed directives in each instruction are reflected in the pace of the rollout.

Robotics. Figs. A7 and A8 separate the two viewpoints. The wrist-mounted cases show the camera moving only with the end-effector, including a commanded wrist rotation while holding the object, while the third-person cases keep the camera fixed across the horizon as only the arm moves. Both figures contain a real and a simulated episode, and in neither does the viewpoint or the visual domain drift, consistent with the robotics system prompt. The full contact sequence is realized, up to releasing an object inside an open drawer.

D LIMITATIONS

Our evaluation is conducted on LCWM-Bench, which includes data from both real and simulated environments, but does not include experiments deploying *Tiresias* on physical robots. Its effectiveness on real hardware therefore remains to be validated. We leave deployment and evaluation on physical robotic platforms for navigation and manipulation to future work.