FROM PIXELS TO SYMBOLS: MACHINE-NATIVE SYMBOLIC WORLD MODELS FOR MULTIMODAL REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Multimodal large language models (MLLMs) are often trained and evaluated with visual inputs designed for human perception rather than machine reasoning. Inspired by the reverse of graphical user interface (GUI), we propose ASCII maps as a compact, discrete, symbolic world model that convert visual scenes into layouts encoding walls, free space, objects, targets, and action-relevant entities. To evaluate the performance of ASCII map as a symbolic visual representation, we introduce *SymbolicBench*, which covers short-range path planning, long-horizon navigation, and object-centered action planning across maze, floorplan, and agent environments using grid-like images. Across top-tier proprietary and open-source MLLMs, ASCII-based inputs substantially outperform image-only inputs, achieving from 2 to 83 times improvement, while reducing token usage. Further analysis shows that these gains persist across different task complexity levels and generalize beyond grid-like images to real-world images for spatial reasoning tasks with strong robustness. variations. Together, these findings support compact symbolic world representations as an effective interface between visual scene structure and MLLM reasoning, enabling more reliable spatial reasoning and planning with fewer input tokens. The source code will be fully released upon paper acceptance.

1 INTRODUCTION

Effective reasoning about a visual scene often depends on identifying its underlying structure. Human practices offer familiar examples: a portrait painter establishes axes, contours, and proportional landmarks before rendering texture and shading; architects use floor plans to express spatial organization; and maps encode routes and locations through lines and symbols. As illustrated in Fig. 1, these representations make selected aspects of a scene explicit while suppressing other details. Their value lies in this task-dependent abstraction: preserving the information needed for a decision can make a complex visual environment easier to reason about.

This observation motivates a closer examination of the visual representations used by multimodal large language models (MLLMs). Modern MLLMs are increasingly expected to reason, navigate, and plan from visual observations (Anthropic, 2026; Gemini Team, Google, 2026; Singh et al., 2026). Natural images provide rich appearance cues, including texture, color, and perspective (Simoncelli & Olshausen, 2001; Ge et al., 2022), but task-relevant entities and relations must still be extracted from this perceptual detail. For example, navigating a maze requires identifying obstacles, traversable paths, and the target location. Planning further requires reasoning about states, constraints, and possible transitions (Hafner et al., 2019; Ha & Schmidhuber, 2018; Schrittwieser et al., 2020). An effective representation should therefore make these structures accessible to the model.

We investigate *symbolic enhancement* as a principle for designing such representations: explicitly encoding task-relevant entities, spatial relations, and constraints in a form that models can readily process. Our central question is: *Can symbolic scheme make visual representations more effective for machine reasoning?* Here, effectiveness is measured by how well a representation supports accurate reasoning and successful planning. We organize this investigation around three research questions: **1) Effectiveness:** Can symbolic enhancement improve reasoning and planning compared with the original visual inputs? **2) Mechanism:** Which symbolic structures may contribute to model

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

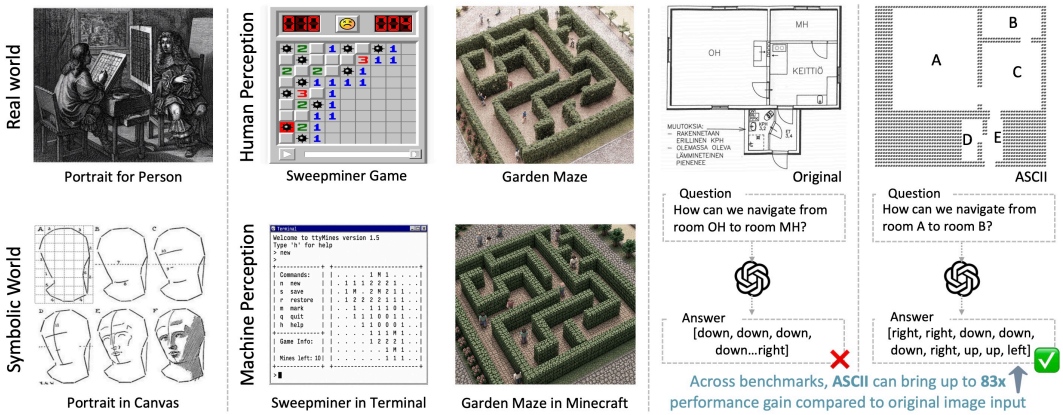

Figure 1: **Motivation for symbolic enhancement of visual representations.** Left: A grid-based drawing schema illustrates how explicit spatial structure supports visual depiction (Scott, 2013; Rousar; Chase, 1919). Middle: Examples from Sweepminer and Garden Maze. Right: An original visual observation and its corresponding ASCII map, which encodes task-relevant structure through spatially arranged symbols.

performance? **3) Scope:** Under what task conditions do the benefits of symbolic structure outweigh the loss of visual detail?

To study these questions, we propose *ASCII maps* as a concrete form of symbolic enhancement for MLLMs. An ASCII map represents a visual scene as a spatially organized character grid, using discrete symbols to encode entities such as walls, free spaces, targets, objects, and agents. The arrangement of these symbols preserves selected aspects of the scene's geometry and topology, making object locations, obstacles, and connectivity explicit in a textual format. Speficially, ASCII maps offer a useful setting for examining both the potential and the limits of symbolic enhancement. They expose task-relevant structure while reducing appearance-level detail, and their spatial organization supports reasoning about relations among entities. Their usefulness nevertheless depends on the information retained: texture, color, or fine-grained geometry may be essential in some tasks. We therefore use ASCII maps to investigate how the choice of representation shapes model performance, with the broader aim of understanding when explicit symbolic structure yields better representations for visual reasoning and planning.

To evaluate whether ASCII-based symbolic priors can improve MLLM reasoning, we introduce *SymbolicBench*, a benchmark built from grid-like images representing world environments. *SymbolicBench* covers tasks with different complexity: short-range path planning, long-horizon navigation, and object-centered action planning. Across several top-tier MLLMs, ASCII-based symbolic world models substantially outperform image-only inputs, achieving 2–83$\times$ improvements and generalizing across tasks, difficulty levels, and model families. Beyond grid-like environments, we further evaluate ASCII-based symbolic priors on real-world images using SIGBench (Wu et al., 2025), where they improve performance by approximately 2$\times$ to 5$\times$. Our contributions are:

- We propose ASCII map as a symbolic world-model representation for MLLMs, externalizing visual structure into discrete textual layouts that significantly enhance the visual reasoning of MLLMs.
- We demonstrate through extensive experiments that ASCII-based symbolic representations substantially outperform image-only inputs, with improvements ranging from 2–83$\times$ on grid-like image tasks and approximately 2$\times$ on real-world spatial reasoning tasks.
- We illustrate that ASCII-based symbolic priors generalize across task types, image domains, difficulty levels, and MLLM families with strong robustness, substantially improving average performance over image-only inputs from 19.09% to 64.82%.

2 Related Work

Symbolic world models. World models, which support prediction, simulation, and planning, have become a central paradigm in modern AI (Ha & Schmidhuber, 2018; Hafner et al., 2019). Recent studies show that LLMs can acquire structured internal world models from discrete sequence data. For example, transformers trained on legal move sequences can develop emergent representations of board states and spatial structure (Li et al., 2023); LLMs can encode abstract, goal-relevant world states beyond surface token statistics (Li et al., 2024b); and causal language models trained on logic puzzles learn decodable internal states consistent with search and reasoning (Shah et al., 2024). More broadly, language models can track entity properties and relations across discourse, suggesting their ability to form structured symbolic maps (Li et al., 2021). However, such symbolic world modeling does not naturally transfer to MLLMs. Prior works show that MLLMs remain brittle in image-grounded spatial reasoning (Wang et al., 2024; Liu et al., 2025; Yang et al., 2025) and structurally complex visual graph tasks (Zhu et al., 2025). A key bottleneck is the representational gap: LLMs operate over discrete symbolic tokens, whereas MLLMs rely on continuous visual encoders that may fail to preserve the topological constraints needed for coherent world-model construction (Chen et al., 2024b). As a result, MLLMs lack a naturally discrete medium for simulating environmental dynamics, limiting their robustness in image-grounded reasoning.

Visual representation for MLLM reasoning. Existing work enhances MLLM visual reasoning, particularly spatial reasoning, through structured intermediate representations with explicit relational or geometric priors. Early efforts like (Li et al., 2024a) and (Wang et al., 2025) utilize scene graphs to model semantic dependencies, while (Chen et al., 2024a) and (Sun et al., 2025) inject metric spatial descriptors and numerical poses to ground the model in physical dimensions. For more complex 3D environments, (Zemskova & Yudin, 2025) and (Cheng et al., 2024) leverage learnable 3D scene graphs and region-based depth plug-ins to capture quasi-3D spatiality. Furthermore, (Zhang et al., 2025) incorporates functional object relations to enhance utility-based reasoning and (Wu et al., 2025) proposes spatial intelligence grid (SIG), which is a physics-informed, grid-based symbolic schema that can be used to improve MLLMs' spatial reasoning. Collectively, these works suggest that structured intermediate representations can improve task-specific grounding. However, representations like scene graphs often focus on isolated object relationships, while SIG struggles to efficiently represent scenarios with complex background objects such as road or walls. This makes them difficult to use as comprehensive world models for long-horizon reasoning or navigation, where the agent needs a sparse, symbolic summary of the environment rather than a dense coordinate map.

3 Method

Preliminaries. For a trained MLLM with parameters θ, let g be the vision encoder, h the vision-language projector, and $\mathcal{T}$ the text embedding map. Following (Liu et al., 2023), an image I and text prompt Q yield image features $Z_v = g(I)$, visual embeddings $X_v = h(Z_v)$, and textual embeddings $X_t = \mathcal{T}(Q)$. The image-based maximum-likelihood prediction $\widehat{Y}_v$ is:

$$\widehat{Y}_v = \arg\max_{Y} \prod_{\ell=1}^{|Y|} p_\theta(y_\ell \mid y_{<\ell}, X_v, X_t)\,. \tag{1}$$

Here, Y is a complete output sequence including the end-of-sequence token, $|Y|$ is its length, y_ℓ is its ℓ-th token, $y_{<\ell} = (y_1, \ldots, y_{\ell-1})$ is the preceding prefix, and p_θ is the conditional token distribution.

3.1 Image-to-ASCII Transformation

For grid-like images as shown in Fig. 2 top, we design an image transformation pipeline that can automatically transform it into the ASCII format. The transformation starts with the grid line detection. Our line detection module firstly localizes the dashed lines with specific color by first isolating pixels through a binary mask defined by histogram-derived intensity peaks and low RGB variance. Candidate line coordinates are identified by projecting this mask onto horizontal and vertical axes using a dynamic occupancy threshold. Each candidate undergoes structural validation via run-length analysis to ensure it meets specific geometric constraints, including minimum dash counts, individual dash/gap lengths, and coverage ratios. Since these lines divide the image into multiple

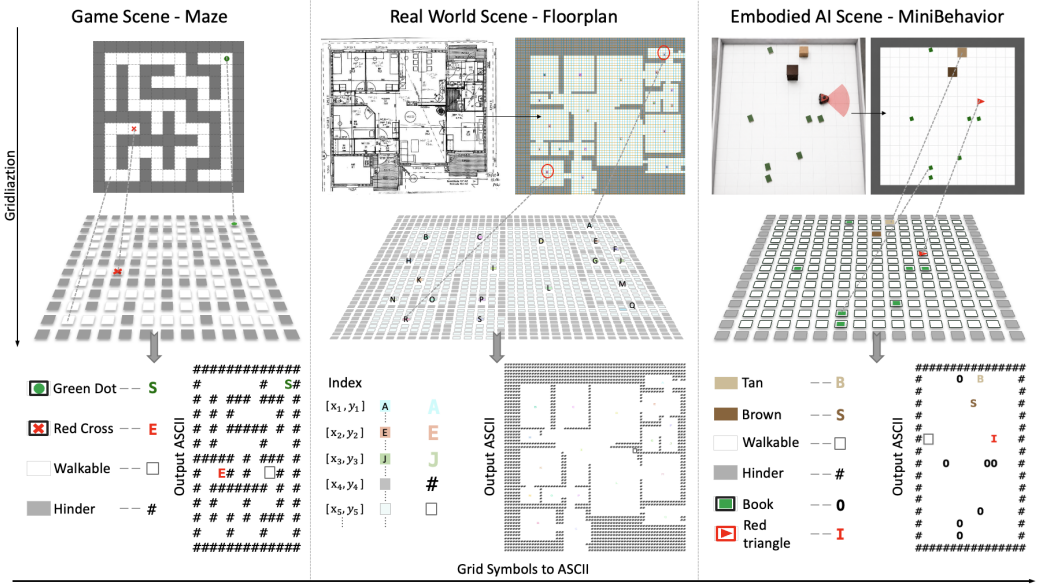

Figure 2: **Overview of our proposed transformation pipeline.** We provide three examples of converting visual images into ASCII-based symbolic world models, including a game maze, a real-world floorplan, and an embodied AI environment. For each scene, we first discretize the image into grid cells, identify traversable regions, obstacles, targets, and task-relevant objects, and then assign corresponding ASCII symbols to each cell.

grid patches, we design a patch-level detector to classify each grid cell as walkable, unwalkable, or object-occupied. Walkable and unwalkable regions are identified through color-based detection, for example, white and black cells respectively, while objects are detected using DINO-X (Ren et al., 2024). We then assign a specific ASCII symbol to each grid type, enabling the image to be systematically converted into an ASCII map. The detailed visualization of each step are illustrated in Fig. 2. We denote the designed pipeline as f_A and the transformed ASCII map of the image I as $A_I = f_A(I)$. Encoding the ASCII map as $X_a = \mathcal{T}(A_I)$ and replacing X_v by X_a gives the ASCII-based prediction $\widehat{Y}_a$:

$$\widehat{Y}_a = \arg\max_{Y} \prod_{\ell=1}^{|Y|} p_\theta(y_\ell \mid y_{<\ell}, X_a, X_t)\,. \tag{2}$$

3.2 Evaluation Metrics

We evaluate MLLM planning using metrics tailored to trajectory length, solution ambiguity, and interaction requirements. Short-range path planning emphasizes optimal-path recovery; long-horizon navigation accounts for partial progress and path efficiency; and object-centered action planning additionally assesses whether trajectories reach task-relevant entities and execute the correct actions at their locations. We also provide illustrative examples for evaluation metrics in Fig. 3.

Short-range path evaluation metrics. We adopt exact match (EM, η) and progressive ratio (PR, ρ), following (Xu et al., 2025). Let $P = (p_1, \ldots, p_T)$ denote the predicted sequence of T movement actions and $\mathcal{G} = \{G^{(1)}, \ldots, G^{(n)}\}$ a set of $n \geq 1$ shortest ground-truth paths, where each $G^{(m)} = (g_1^{(m)}, \ldots, g_M^{(m)})$ contains $M \geq 1$ movement actions. The metrics are defined as

$$\eta(P, \mathcal{G}) = \max_{1 \leq m \leq n} \mathbb{I}\left[P = G^{(m)}\right], \quad \rho(P, \mathcal{G}) = \frac{1}{M} \max_{1 \leq m \leq n} \sum_{t=1}^{\min\{T,M\}} \prod_{i=1}^{t} \mathbb{I}\left[p_i = g_i^{(m)}\right]. \tag{3}$$

EM requires an exact match to any shortest reference path, including its length. PR normalizes the longest matching prefix across reference paths by M, providing partial credit for consecutive correct actions from the start.

Long-horizon navigation evaluation metrics. We introduce three complementary metrics to assess task success and path efficiency, action-sequence similarity, and prefix consistency over long trajectories. All definitions assume nonempty predictions ($T \geq 1$).

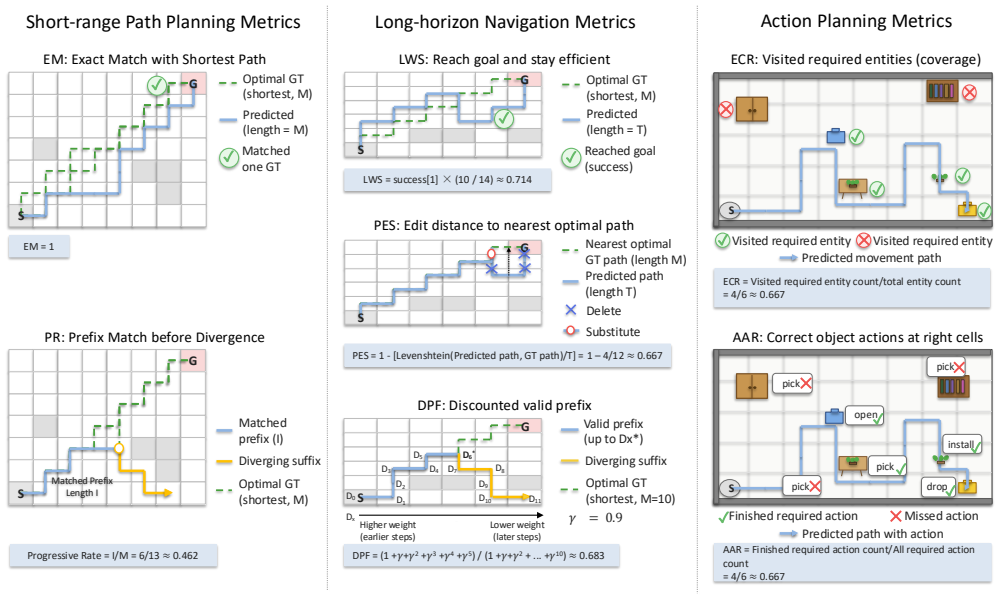

Figure 3: **Evaluation metric examples.** Calculations compare predicted paths and actions with their GT references.

Length-weighted success (LWS, λ). We define $\lambda(P, \mathcal{G}) = S(P)M/T$, where $S(P) = 1$ if P is a valid trajectory reaching the target and 0 otherwise. Since successful trajectories satisfy $T \geq M$, LWS rewards efficient task completion, attaining its maximum of 1 for a shortest successful path.

Path edit similarity (PES, σ). Let $d(P, G^{(m)})$ denote the action-level Levenshtein distance with unit insertion, deletion, and substitution costs. We have:

$$d_\star(P, \mathcal{G}) = \min_{1 \leq m \leq n} d(P, G^{(m)}), \qquad \sigma(P, \mathcal{G}) = 1 - \frac{d_\star(P, \mathcal{G})}{T}. \tag{4}$$

PES measures similarity to the closest shortest reference path, accommodating multiple optimal solutions and providing graded credit for action-sequence deviations. It equals 1 for an exact match and can be negative when $d_\star(P, \mathcal{G}) > T$.

Discounted prefix fidelity (DPF, δ_γ). Let ψ_t indicate whether the first t predicted actions match the prefix of any shortest reference path:

$$\psi_t(P, \mathcal{G}) = \begin{cases} \max\limits_{1 \leq m \leq n} \prod\limits_{i=1}^{t} \mathbb{I}\left[p_i = g_i^{(m)}\right], & 1 \leq t \leq \min\{T, M\}, \\ 0, & M < t \leq T. \end{cases} \tag{5}$$

DPF is the normalized discounted sum of these indicators:

$$\delta_\gamma(P, \mathcal{G}) = \frac{\sum_{t=1}^{T} \gamma^{t-1} \psi_t(P, \mathcal{G})}{\sum_{t=1}^{T} \gamma^{t-1}}, \qquad 0 < \gamma < 1, \tag{6}$$

which measures discounted prefix consistency, with smaller γ placing greater weight on earlier steps.

Action planning evaluation metrics. We partition predicted actions into movement actions ({`up`, `down`, `left`, `right`}) and object-related actions ({`pickup`, `drop`, `drop in`, `open`, `close`, `install`}). Entity coverage ratio (ECR, β) measures visitation of required entity locations, while admissible action ratio (AAR, α) measures coverage of required object-related actions at their specified locations.

For case k, let $\mathcal{E}_k$ denote the required entity locations in the ground-truth trajectory and $\mathcal{P}_k$ the grid locations visited by the prediction. Let $\mathcal{A}_k$ and $\widehat{\mathcal{A}}_k$ denote the required and predicted sets of action–location tuples (a, r, c), where a is an object-related action and (r, c) specifies its grid row

Table 1: **Quantitative performance comparison across different models on *SymbolicBench* and SIGBench-tiny.** The detailed version of evaluated MLLMs are GPT-5-Mini (GPT), Gemini-3-Flash (Gemini), Claude-Sonnet-4.6 (Claude), Grok-4-Fast (Grok), GLM-5-Turbo (GLM), Qwen-3.5-35B-A3B (Qwen-35B) and Qwen-3.5-397B-A17B (Qwen-397B). I, I+A and A denotes Image-only, Image+ASCII and ASCII-only. Acc and TU refers to accuracy and token usage. The superscript $\uparrow x$ in ASCII-only rows indicates the relative gain over Image-only. For cases that Image-only achieves 0, we consider its performance as 0.01 for calculating how many times improvement.

Type	Models	Input	Maze			Floorplan				Mini-Behavior			SIGBench-tiny	
			EM↑	PR↑	TU↓	LWS↑	PES↑	DPF↑	TU↓	ECR↑	AAR↑	TU↓	Acc.↑	TU↓
Proprietary	GPT	I	0.177	0.268	376	0.000	0.134	0.103	1933	0.138	0.041	1032	44.61	3477
		I+A	0.870	0.903	441	0.291	0.660	0.480	2385	0.893	0.810	1378	87.17	3910
		A	0.897$^{\uparrow 5x}$	0.919$^{\uparrow 3x}$	257	0.348$^{\uparrow 35x}$	0.694$^{\uparrow 5x}$	0.511$^{\uparrow 5x}$	650	0.880$^{\uparrow 6x}$	0.799$^{\uparrow 19x}$	789	86.99$^{\uparrow 2x}$	1367
	Gemini	I	0.333	0.403	1294	0.000	0.209	0.108	1295	0.586	0.442	1751	42.01	2320
		I+A	0.583	0.643	1363	0.762	0.865	0.624	1854	0.934	0.871	2116	85.87	2791
		A	0.723$^{\uparrow 2x}$	0.781$^{\uparrow 2x}$	264	0.826$^{\uparrow 83x}$	0.866$^{\uparrow 4x}$	0.647$^{\uparrow 6x}$	750	0.945$^{\uparrow 2x}$	0.897$^{\uparrow 2x}$	844	86.99$^{\uparrow 2x}$	1479
	Claude	I	0.143	0.250	415	0.000	0.176	0.134	1770	0.295	0.108	1176	36.80	2852
		I+A	0.513	0.587	499	0.156	0.637	0.403	2333	0.752	0.570	1550	86.99	3342
		A	0.550$^{\uparrow 4x}$	0.622$^{\uparrow 2x}$	290	0.197$^{\uparrow 20x}$	0.656$^{\uparrow 4x}$	0.415$^{\uparrow 3x}$	779	0.731$^{\uparrow 2x}$	0.572$^{\uparrow 5x}$	875	86.43$^{\uparrow 2x}$	1558
	GLM	I	0.197	0.291	405	0.000	0.160	0.113	2544	0.225	0.088	1132	46.28	3866
		I+A	0.320	0.413	469	0.029	0.420	0.318	2978	0.539	0.335	1478	83.46	4303
		A	0.310$^{\uparrow 2x}$	0.414$^{\uparrow 1x}$	313	0.059$^{\uparrow 6x}$	0.553$^{\uparrow 3x}$	0.336$^{\uparrow 3x}$	689	0.547$^{\uparrow 2x}$	0.319$^{\uparrow 4x}$	783	84.20$^{\uparrow 2x}$	1381
	Grok	I	0.017	0.145	623	0.000	0.000	0.121	1998	0.097	0.013	1210	37.92	3074
		I+A	0.867	0.899	674	0.348	0.764	0.547	2430	0.755	0.587	1555	85.69	3507
		A	0.900$^{\uparrow 53x}$	0.925$^{\uparrow 6x}$	407	0.421$^{\uparrow 42x}$	0.785$^{\uparrow 78x}$	0.544$^{\uparrow 4x}$	785	0.769$^{\uparrow 8x}$	0.592$^{\uparrow 46x}$	929	85.87$^{\uparrow 2x}$	1514
Open-source	Qwen-35B	I	0.223	0.344	363	0.000	0.014	0.017	1943	0.275	0.127	1015	39.96	3260
		I+A	0.210	0.312	442	0.030	0.249	0.189	2415	0.596	0.440	1385	84.57	3731
		A	0.597$^{\uparrow 3x}$	0.637$^{\uparrow 2x}$	281	0.098$^{\uparrow 10x}$	0.379$^{\uparrow 27x}$	0.305$^{\uparrow 18x}$	700	0.599$^{\uparrow 2x}$	0.434$^{\uparrow 3x}$	841	83.46$^{\uparrow 2x}$	1478
	Qwen-397B	I	0.223	0.372	363	0.000	0.121	0.070	1938	0.310	0.158	1014	44.80	3260
		I+A	0.287	0.383	437	0.145	0.450	0.392	2416	0.878	0.782	1384	86.62	3731
		A	0.840$^{\uparrow 4x}$	0.859$^{\uparrow 2x}$	281	0.526$^{\uparrow 53x}$	0.834$^{\uparrow 7x}$	0.610$^{\uparrow 9x}$	691	0.877$^{\uparrow 3x}$	0.756$^{\uparrow 5x}$	854	85.69$^{\uparrow 2x}$	1479

and column. ECR and AAR are defined for nonempty $\mathcal{E}_k$ and $\mathcal{A}_k$, respectively:

$$\beta_k = \frac{|\mathcal{E}_k \cap \mathcal{P}_k|}{|\mathcal{E}_k|}, \qquad \alpha_k = \frac{\left|\mathcal{A}_k \cap \widehat{\mathcal{A}}_k\right|}{|\mathcal{A}_k|}. \tag{7}$$

Both metrics provide partial credit without requiring an exact trajectory match. AAR complements ECR by requiring agreement in both action type and location.

4 Experiments

***SymbolicBench*.** We introduce *SymbolicBench*, a benchmark for evaluating MLLM planning in grid-like world environments. It contains three complementary subsets derived from MAZE (Ivanitskiy et al., 2023), CubiCasa5K (Kalervo et al., 2019), and Mini-BEHAVIOR (Jin et al., 2023), covering short-range path planning, long-horizon navigation, and object-centered action planning, respectively. More specifically, Maze contains 300 cases with grid sizes from 3×3 to 8×8, where models navigate from a start point to a target along an optimal path. Floorplan contains 300 cases constructed from CubiCasa5K floorplans using our annotation and preprocessing pipeline (detailes in Appendix A). Its realistic room-and-corridor layouts yield navigation trajectories of 15–45 steps across multiple target locations. Mini-Behavior contains 300 scenarios from five object-centered tasks in Mini-BEHAVIOR (Jin et al., 2023), requiring models to jointly plan movement and object interactions at task-relevant locations. Examples are shown in Fig. 4 top.

Datasets. We evaluate ASCII maps on both grid-like and real-world visual reasoning tasks. For grid-like environments, we use *SymbolicBench*. For real-world spatial reasoning, we use SIGBench-tiny (Wu et al., 2025), a visual question-answering benchmark based on autonomous-driving scenes. We adopt SIGBench-tiny because its GT Spatial Intelligence Grids (SIGs) can be directly converted into ASCII representations, enabling a controlled comparison between image and symbolic inputs.

Table 2: **Comparison of different visual inputs using GPT-5-Mini on SIGBench-tiny.** GT and Pred denote ground-truth and predicted ASCII maps, respectively. MLSM denotes Multi-level Spatial Matching, with precision (P), recall (R), F1-score (F1), and association accuracy (AssA). SRGS denotes Spatial Relation Graph Similarity; S and WS represent similarity with equal and unequal weighting of node and edge edit costs, respectively. MC denotes multiple-choice accuracy, with Dir. and Prox. referring to directional and proximal (relative-distance) relations. For details about evaluation metric, please refer to (Wu et al., 2025). Δ indicates absolute accuracy differences relative to the image baseline. ASCII (Pred) means using predicted ASCII map from our proposed pipeline.

Type	MLSM				SRGS		MC (Acc.)			
	P↑	R↑	F1↑	AssA↑	S↑	WS↑	Dir.↑	Δ Dir.	Prox.↑	Δ Prox.
Image Baseline	-	-	-	-	-	-	0.320	-	0.573	-
ASCII (GT)	1.000	1.000	1.000	1.000	1.000	1.000	0.751	+0.431	0.989	+0.416
ASCII (Pred)	0.778	0.773	0.775	0.652	0.757	0.743	0.524	+0.205	0.651	+0.078

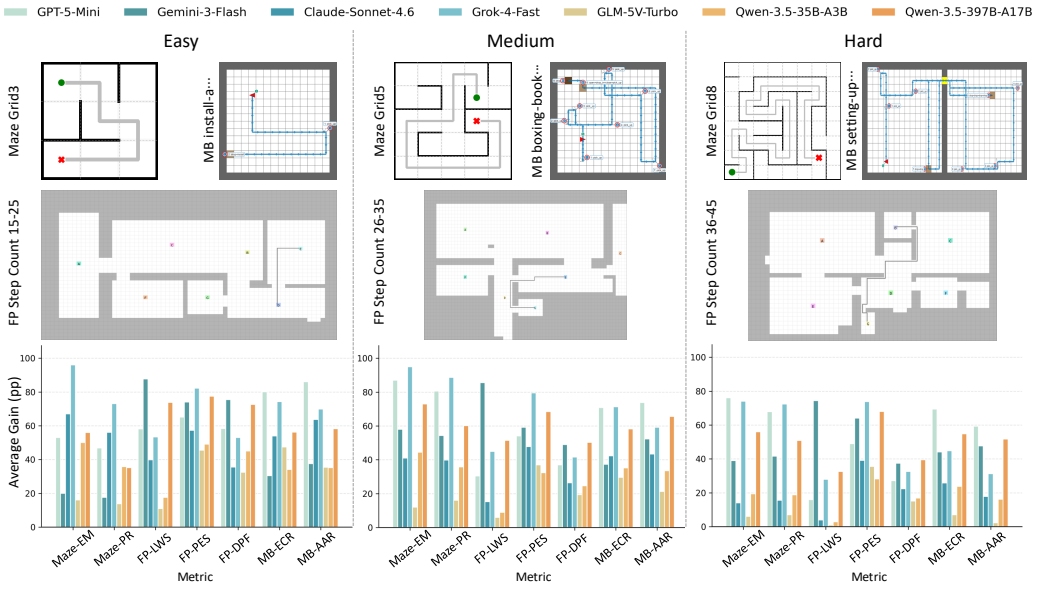

Figure 4: **Performance gain of ASCII-only over image-only.** FP and MB denote Floorplan and Mini-Behavior, respectively. For Maze, FP, and MB, we divide cases into three difficulty levels. The easy setting includes Maze grids of size 3 and 4, FP trajectories of 15-25 steps, and the two lowest-complexity MB tasks. The medium setting includes Maze grids of size 5 and 6, FP trajectories of 26-35 steps, and the two medium-complexity MB tasks. The hard setting includes Maze grids of size 7 and 8, FP trajectories of 36-45 steps, and the highest-complexity MB task.

4.1 Experimental Setup

Models and metrics. We evaluate MLLMs from six model families: GPT (Singh et al., 2026), Gemini (Gemini Team, Google, 2026), Claude (Anthropic, 2026), GLM (GLM-5-Team et al., 2026), Grok (xAI, 2025), and Qwen (Team, 2026); exact model versions are listed in Tab. 1. Following Sec. 3, we report EM and PR for Maze, LWS, PES, and DPF for Floorplan, ECR and AAR for Mini-Behavior, and accuracy for SIGBench-tiny.

4.2 Main Results

ASCII map improves grid-based planning. Tab. 1 shows substantial and consistent improvements with ASCII maps across models and tasks. On Maze, ASCII-only performs best for most models, with EM/PR gains of $5\times/3\times$ for GPT-5-Mini and $53\times/6\times$ for Grok-4-Fast, supporting its effectiveness for short-range grid navigation. On Floorplan, ASCII-only also improves long-range navigation: Gemini-3-Flash achieves LWS/PES/DPF gains of $83\times/4\times/6\times$, while Qwen-3.5-397B-A17B achieves $53\times/7\times/9\times$. Because all Image-only LWS scores are zero, these LWS ratios use a reference value of 0.01, following Tab. 1. On Mini-Behavior, ASCII-based inputs improve object-centered

action planning, yielding ECR/AAR gains of $8\times/46\times$ for Grok-4-Fast and $6\times/19\times$ for GPT-5-Mini. Together, these results suggest that explicitly encoding discrete spatial structure supports both navigation and location-dependent object interactions.

Table 3: **Performance under different ASCII noise levels.** Clean ASCII refers to using original ASCII map while Noise 10%, 30% and 50% refers to ASCII map with different level of added noise. The number shown for each model is the average results across 5 types of noise at this noise level. Blue superscripts $\uparrow x$ indicate performance relative to the original image input, expressed as a multiplier and rounded to the nearest integer. For zero original-image scores, a baseline of 0.01 is used to calculate the multiplier. GPT-5-Mini. Gemini-3-Flash. Qwen-3.5-397B.

Models	Type	Floorplan			Mini-BEHAVIOR		SIGBench-tiny
		LWS↑	PES↑	DPF↑	ECR↑	AAR↑	Acc.↑
GPT	Image	0.000	0.179	0.158	0.142	0.040	0.483
	Clean ASCII	$0.347^{\uparrow 35x}$	$0.760^{\uparrow 4x}$	$0.622^{\uparrow 4x}$	$0.888^{\uparrow 6x}$	$0.793^{\uparrow 20x}$	$0.883^{\uparrow 2x}$
	Noise 10%	$0.297^{\uparrow 30x}$	$0.720^{\uparrow 4x}$	$0.592^{\uparrow 4x}$	$0.840^{\uparrow 6x}$	$0.755^{\uparrow 19x}$	$0.833^{\uparrow 2x}$
	Noise 30%	$0.217^{\uparrow 22x}$	$0.726^{\uparrow 4x}$	$0.600^{\uparrow 4x}$	$0.685^{\uparrow 5x}$	$0.570^{\uparrow 14x}$	$0.783^{\uparrow 2x}$
	Noise 50%	$0.230^{\uparrow 23x}$	$0.645^{\uparrow 4x}$	$0.552^{\uparrow 3x}$	$0.565^{\uparrow 4x}$	$0.497^{\uparrow 13x}$	$0.767^{\uparrow 2x}$
Gemini	Image	0.000	0.206	0.123	0.603	0.440	0.417
	Clean ASCII	$0.798^{\uparrow 80x}$	$0.939^{\uparrow 5x}$	$0.884^{\uparrow 7x}$	$0.944^{\uparrow 2x}$	$0.901^{\uparrow 2x}$	$0.883^{\uparrow 2x}$
	Noise 10%	$0.723^{\uparrow 72x}$	$0.936^{\uparrow 5x}$	$0.831^{\uparrow 7x}$	$0.861^{\uparrow 1x}$	$0.808^{\uparrow 2x}$	$0.833^{\uparrow 2x}$
	Noise 30%	$0.610^{\uparrow 61x}$	$0.922^{\uparrow 4x}$	$0.772^{\uparrow 6x}$	$0.753^{\uparrow 1x}$	$0.666^{\uparrow 2x}$	$0.767^{\uparrow 2x}$
	Noise 50%	$0.576^{\uparrow 58x}$	$0.903^{\uparrow 4x}$	$0.750^{\uparrow 6x}$	$0.616^{\uparrow 1x}$	$0.521^{\uparrow 1x}$	$0.783^{\uparrow 2x}$
Qwen-397B	Image	0.000	0.075	0.089	0.285	0.134	0.417
	Clean ASCII	$0.630^{\uparrow 63x}$	$0.920^{\uparrow 12x}$	$0.827^{\uparrow 9x}$	$0.881^{\uparrow 3x}$	$0.750^{\uparrow 6x}$	$0.867^{\uparrow 2x}$
	Noise 10%	$0.513^{\uparrow 51x}$	$0.880^{\uparrow 12x}$	$0.822^{\uparrow 9x}$	$0.792^{\uparrow 3x}$	$0.717^{\uparrow 5x}$	$0.817^{\uparrow 2x}$
	Noise 30%	$0.447^{\uparrow 45x}$	$0.850^{\uparrow 11x}$	$0.775^{\uparrow 9x}$	$0.652^{\uparrow 2x}$	$0.497^{\uparrow 4x}$	$0.767^{\uparrow 2x}$
	Noise 50%	$0.365^{\uparrow 37x}$	$0.833^{\uparrow 11x}$	$0.698^{\uparrow 8x}$	$0.580^{\uparrow 2x}$	$0.492^{\uparrow 4x}$	$0.783^{\uparrow 2x}$

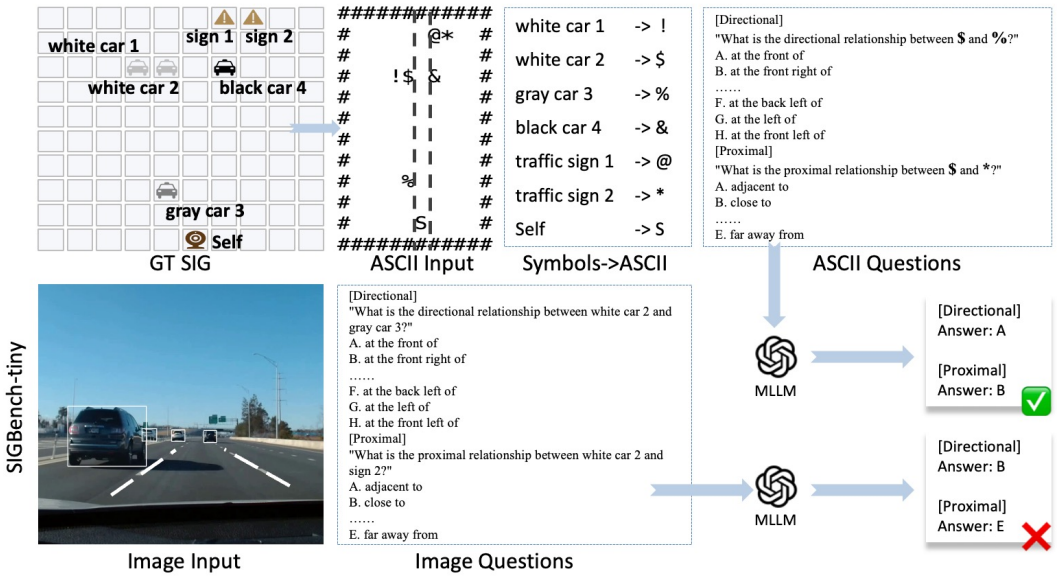

Figure 5: Illustration example of one case from SIGBench-tiny.

ASCII map improves real-world spatial reasoning. Beyond grid-like images in *SymbolicBench*, ASCII maps improve spatial reasoning on real-world images in SIGBench-tiny (Tab. 1, right; see Fig. 5 for an example). GT ASCII-only inputs consistently outperform Image-only inputs across all evaluated MLLMs, increasing average accuracy from 41.77% to 85.66%. These gains suggest that structured symbolic layouts support spatial reasoning beyond synthetic environments. As an end-to-end illustration, we further develop an image-to-ASCII pipeline tailored to SIGBench samples (Appendix C.2). As shown in Tab. 2, even with predicted ASCII maps, GPT-5-Mini's directional and proximal accuracies increase from 32.0% and 57.3% to 52.4% and 65.1%, respectively.

4.3 Robustness Analysis

Robustness of ASCII map as visual representation. We evaluate robustness to ASCII corruption with different noise types, symbol choices, grid resolution, and image degradation, please refer to Appendix B for details. As shown in Tab. 3, ASCII inputs outperform original images across all tested models and metrics on Floorplan, Mini-Behavior, and SIGBench-tiny even with 50% injected noise. On Floorplan, reversing or substituting symbols preserves the advantage over images, although scores vary with the encoding (Tab. 5). Coarser grids generally improve planning scores while reducing the number of cells, suggesting that finer spatial resolution does not yield better reasoning (Tab. 6).

Robustness of grid-like image transformation pipeline. For the Image-to-ASCII pipeline on Maze, cell and object detection remain perfect under all tested Gaussian blur and edge-softening settings (Appendix B.2, Tab. 7). Gaussian noise progressively degrades object detection, with cell and object detection accuracies falling to 62.5% and 27.5% at the highest severity. These findings support robustness within the tested settings while identifying strong pixel noise as a limitation of the conversion pipeline.

4.4 Findings

Token efficiency of ASCII map. ASCII maps improve reasoning performance while reducing token usage. On Floorplan, ASCII-only averages 721 tokens across models, compared with 1,917 for Image-only and 2,402 for Image+ASCII, corresponding to reductions of approximately 62% and 70%, respectively. Similar reductions on Maze and Mini-BEHAVIOR further demonstrate the compactness of ASCII representations for grid-based reasoning.

Limited benefits of combining images with ASCII map. Interestingly, adding images to ASCII does not consistently improve performance. Although Image+ASCII is competitive in some cases, such as GPT-5-Mini on Mini-Behavior, it often underperforms ASCII-only while requiring considerably more tokens. This suggests that once the spatial and semantic information has been explicitly encoded in ASCII form, the additional visual input may introduce redundant details or increase cross-modal alignment difficulty. Therefore, ASCII-only offers a strong balance between accuracy, robustness, and efficiency, making it an effective visual representation.

Generalizability across task difficulty and model families. Fig. 4 highlights the strong generalizability of ASCII-only input across different task difficulties. The performance gain of ASCII-only over Image-only remains consistently positive from easy to medium and hard settings, showing that the advantage of ASCII map is not restricted to simple cases. As task difficulty increases, such as larger Maze grids, longer Floorplan trajectories, and more complex Mini-Behavior tasks, ASCII still provides clear benefits across most models and metrics. This suggests that ASCII representations preserve task-relevant structure in a scalable way: they make spatial layouts, paths, and object locations explicit even when the environment becomes more complex. These gains also appear across both proprietary and open-source models. Although the magnitude varies by model and metric, the overall trend is shared across architectures, showing that ASCII is not tailored to a single model but serves as a generally effective representation for planning-oriented reasoning.

5 Conclusion and Future Work

In this paper, we proposed ASCII maps as compact symbolic visual representations for improving spatial intelligence in MLLMs. Through *SymbolicBench*, we demonstrated their effectiveness across short-range path planning, long-horizon navigation, and object-centered action planning. Across the evaluated models and task difficulties, ASCII-based inputs substantially improve performance while reducing token usage. Robustness analyses further show that these advantages persist under the tested map corruptions and symbol variations, although performance remains sensitive to representation quality and grid resolution. Additional experiments on SIGBench-tiny support the usefulness of symbolic layouts for spatial reasoning in driving scenes. Together, these findings establish a proof of concept for symbolic maps as an effective interface between visual scene structure and language-model reasoning. Our central contribution is the introduction and empirical validation of this representational approach, with the dataset-specific image-to-ASCII pipeline serving as a supporting illustration. Extending symbolic representations to more complex environments and developing reliable conversion pipelines across image domains remain important directions for future work.

AI USE STATEMENT

In this work, we used generative AI tools for assist in the writing of proofs, design or provide feedback on research methodology or experiments, implement methods, assist with translation, support qualitative and thematic data analysis, interpret results. We have not used generative AI tools for generate synthetic data sets, help develop theoretical models or conceptual frameworks, propose or refine hypotheses, clean and reformat dataset and formulate mathematical claims, provide critical ingredients for proving mathematical claims are not applicable to this work. We have reviewed all AI-assisted work. We thoroughly checked all LLM-generated texts inside this paper and LLM-generated code was verified and tested for correctness. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This paper has no ethical concerns.

REPRODUCIBILITY STATEMENT

The code will be open sourced upon acceptance.

REFERENCES

Anthropic. Claude sonnet 4.6 system card. Technical report, Anthropic, February 2026. URL https://www.anthropic.com/news/claude-sonnet-4-6. Technical Report.

Joseph Cummings Chase. *Drawing Made Easy*. EJ Clode, 1919.

Boyuan Chen, Zhuo Xu, Sean Kirmani, Brian Ichter, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14455–14465, 2024a.

Gongwei Chen, Leyang Shen, Rui Shao, Xiang Deng, and Liqiang Nie. LION: Empowering multimodal large language model with dual-level visual knowledge. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024b. URL https://openaccess.thecvf.com/content/CVPR2024/html/Chen_LION_Empowering_Multimodal_Large_Language_Model_with_Dual-Level_Visual_Knowledge_CVPR_2024_paper.html.

An-Chieh Cheng, Hongxu Yin, Yang Fu, Qiushan Guo, Ruihan Yang, Jan Kautz, Xiaolong Wang, and Sifei Liu. Spatialrgpt: Grounded spatial reasoning in vision-language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Yunhao Ge, Yao Xiao, Zhi Xu, Xingrui Wang, and Laurent Itti. Contributions of shape, texture, and color in visual recognition. In *Lecture Notes in Computer Science*, pp. 369–386. Springer Nature Switzerland, 2022. doi: 10.1007/978-3-031-19775-8_22.

Gemini Team, Google. Gemini 3.1 pro - model card. Google DeepMind Technical Report, 2026. URL https://deepmind.google/models/model-cards/gemini-3-1-pro/.

GLM-5-Team, :, Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, Qinkai Zheng, Bin Chen, Da Yin, Chendi Ge, Chenghua Huang, Chengxing Xie, Chenzheng Zhu, Congfeng Yin, Cunxiang Wang, Gengzheng Pan, Hao Zeng, Haoke Zhang, Haoran Wang, Huilong Chen, Jiajie Zhang, Jian Jiao, Jiaqi Guo, Jingsen Wang, Jingzhao Du, Jinzhu Wu, Kedong Wang, Lei Li, Lin Fan, Lucen Zhong, Mingdao Liu, Mingming Zhao, Pengfan Du, Qian Dong, Rui Lu, Shuang-Li, Shulin Cao, Song Liu, Ting Jiang, Xiaodong Chen, Xiaohan Zhang, Xuancheng Huang, Xuezhen Dong, Yabo Xu, Yao Wei, Yifan An, Yilin Niu, Yitong Zhu, Yuanhao Wen, Yukuo Cen, Yushi Bai, Zhongpei Qiao, Zihan Wang, Zikang Wang, Zilin Zhu, Ziqiang Liu, Zixuan Li, Bojie Wang, Bosi Wen, Can Huang, Changpeng Cai, Chao Yu, Chen Li, Chengwei Hu, Chenhui Zhang, Dan Zhang, Daoyan Lin, Dayong Yang, Di Wang, Ding Ai, Erle Zhu, Fangzhou Yi, Feiyu Chen, Guohong Wen, Hailong Sun, Haisha Zhao, Haiyi Hu, Hanchen Zhang, Hanrui Liu, Hanyu Zhang, Hao Peng, Hao Tai,

Haobo Zhang, He Liu, Hongwei Wang, Hongxi Yan, Hongyu Ge, Huan Liu, Huanpeng Chu, Jia'ni Zhao, Jiachen Wang, Jiajing Zhao, Jiamin Ren, Jiapeng Wang, Jiaxin Zhang, Jiayi Gui, Jiayue Zhao, Jijie Li, Jing An, Jing Li, Jingwei Yuan, Jinhua Du, Jinxin Liu, Junkai Zhi, Junwen Duan, Kaiyue Zhou, Kangjian Wei, Ke Wang, Keyun Luo, Laiqiang Zhang, Leigang Sha, Liang Xu, Lindong Wu, Lintao Ding, Lu Chen, Minghao Li, Nianyi Lin, Pan Ta, Qiang Zou, Rongjun Song, Ruiqi Yang, Shangqing Tu, Shangtong Yang, Shaoxiang Wu, Shengyan Zhang, Shijie Li, Shuang Li, Shuyi Fan, Wei Qin, Wei Tian, Weining Zhang, Wenbo Yu, Wenjie Liang, Xiang Kuang, Xiangmeng Cheng, Xiangyang Li, Xiaoquan Yan, Xiaowei Hu, Xiaoying Ling, Xing Fan, Xingye Xia, Xinyuan Zhang, Xinze Zhang, Xirui Pan, Xu Zou, Xunkai Zhang, Yadi Liu, Yandong Wu, Yanfu Li, Yidong Wang, Yifan Zhu, Yijun Tan, Yilin Zhou, Yiming Pan, Ying Zhang, Yinpei Su, Yipeng Geng, Yong Yan, Yonglin Tan, Yuean Bi, Yuhan Shen, Yuhao Yang, Yujiang Li, Yunan Liu, Yunqing Wang, Yuntao Li, Yurong Wu, Yutao Zhang, Yuxi Duan, Yuxuan Zhang, Zezhen Liu, Zhengtao Jiang, Zhenhe Yan, Zheyu Zhang, Zhixiang Wei, Zhuo Chen, Zhuoer Feng, Zijun Yao, Ziwei Chai, Ziyuan Wang, Zuzhou Zhang, Bin Xu, Minlie Huang, Hongning Wang, Juanzi Li, Yuxiao Dong, and Jie Tang. Glm-5: from vibe coding to agentic engineering, 2026. URL https://arxiv.org/abs/2602.15763.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, pp. 2455–2467, Red Hook, NY, USA, 2018. Curran Associates Inc.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2555–2565. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/hafner19a.html.

Michael Igorevich Ivanitskiy, Rusheb Shah, Alex F. Spies, Tilman Räuker, Dan Valentine, Can Rager, Lucia Quirke, Chris Mathwin, Guillaume Corlouer, Cecilia Diniz Behn, and Samy Wu Fung. A configurable library for generating and manipulating maze datasets, 2023. URL https://arxiv.org/abs/2309.10498.

Emily Jin, Jiaheng Hu, Zhuoyi Huang, Ruohan Zhang, Jiajun Wu, Li Fei-Fei, and Roberto Martín-Martín. Mini-behavior: A procedurally generated benchmark for long-horizon decision-making in embodied ai. *arXiv preprint 2310.01824*, 2023.

Ahti Kalervo, Juha Ylioinas, Markus Häikiö, Antti Karhu, and Juho Kannala. Cubicasa5k: A dataset and an improved multi-task model for floorplan image analysis. In Michael Felsberg, Per-Erik Forssén, Ida-Maria Sintorn, and Jonas Unger (eds.), *Image Analysis*, pp. 28–40, Cham, 2019. Springer International Publishing. ISBN 978-3-030-20205-7.

Belinda Z. Li, Maxwell Nye, and Jacob Andreas. Implicit representations of meaning in neural language models. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 1813–1827, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.143. URL https://aclanthology.org/2021.acl-long.143/.

Kenneth Li, Aspen K. Hopkins, David Bau, Fernanda B. Viégas, Hanspeter Pfister, and Martin Wattenberg. Emergent world representations: Exploring a sequence model trained on a synthetic task. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=DeG07_TcZvT.

Rongjie Li, Songyang Zhang, Dahua Lin, Kai Chen, and Xuming He. From pixels to graphs: Open-vocabulary scene graph generation with vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 28076–28086, 2024a.

Zichao Li, Yanshuai Cao, and Jackie C.K. Cheung. Do LLMs build world representations? probing through the lens of state abstraction. In *Advances in Neural Information Processing Systems*, volume 37, 2024b. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/b1b16c4b875eb84d3585cb70d23970ca-Abstract-Conference.html.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023.

Jingping Liu, Ziyan Liu, Zhedong Cen, Yan Zhou, Yinan Zou, Weiyan Zhang, Haiyun Jiang, and Tong Ruan. Can multimodal large language models understand spatial relations? In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 620–632, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.31. URL https://aclanthology.org/2025.acl-long.31/.

Tianhe Ren, Yihao Chen, Qing Jiang, Zhaoyang Zeng, Yuda Xiong, Wenlong Liu, Zhengyu Ma, Junyi Shen, Yuan Gao, Xiaoke Jiang, Xingyu Chen, Zhuheng Song, Yuhong Zhang, Hongjie Huang, Han Gao, Shilong Liu, Hao Zhang, Feng Li, Kent Yu, and Lei Zhang. Dino-x: A unified vision model for open-world object detection and understanding, 2024. URL https://arxiv.org/abs/2411.14347.

Darren Rousar. Alberti's veil. URL https://www.sightsize.com/albertis-veil/.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, Timothy Lillicrap, and David Silver. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588 (7839):604–609, 2020. doi: 10.1038/s41586-020-00305-8.

Katie Scott. Screen wise, screen play: Jacques de lajoue and the ruses of rococo. *Art History*, 36(3): 568–607, 2013.

Kulin Shah, Nishanth Dikkala, Xin Wang, and Rina Panigrahy. Causal language modeling can elicit search and reasoning capabilities on logic puzzles. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://openreview.net/pdf?id=i5PoejmWoC.

Eero P. Simoncelli and Bruno A. Olshausen. Natural image statistics and neural representation. *Annual Review of Neuroscience*, 24(1):1193–1216, 2001. doi: 10.1146/annurev.neuro.24.1.1193.

Aaditya Singh et al. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*, 2026. URL https://arxiv.org/abs/2601.03267.

Fan-Yun Sun, Weiyu Liu, Siyi Gu, Dylan Lim, Goutam Bhat, Federico Tombari, Manling Li, Nick Haber, and Jiajun Wu. Layoutvlm: Differentiable optimization of 3d layout via vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Qwen Team. Qwen3.5-omni technical report, 2026. URL https://arxiv.org/abs/2604.15804.

Jiayu Wang, Yifei Ming, Zhenmei Shi, Vibhav Vineet, Xin Wang, Yixuan Li, and Neel Joshi. Is a picture worth a thousand words? delving into spatial reasoning for vision language models. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/89cc5e613d34f90de90c21e996e60b30-Abstract-Conference.html.

Jingyi Wang, Jianzhong Ju, Jian Luan, and Zhidong Deng. LLaVA-SG: Leveraging scene graphs as visual semantic expression in vision-language models. In *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2025.

Guanlin Wu, Boyan Su, Yang Zhao, Pu Wang, Yichen Lin, and Hao Frank Yang. Towards physics-informed spatial intelligence with human priors: An autonomous driving pilot study. In *The Thirty-Ninth Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2025.

xAI. Grok 4 fast model card. Technical Report, September 2025. URL https://data.x.ai/2025-09-19-grok-4-fast-model-card.pdf.

Yi Xu, Chengzu Li, Han Zhou, Xingchen Wan, Caiqi Zhang, Anna Korhonen, and Ivan Vulić. Visual planning: Let's think only with images, 2025. URL https://arxiv.org/abs/2505.11409.

Jihan Yang, Shusheng Yang, Anjali W. Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. Thinking in space: How multimodal large language models see, remember, and recall spaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Yang_Thinking_in_Space_How_Multimodal_Large_Language_Models_See_Remember_CVPR_2025_paper.html.

Tatiana Zemskova and Dmitry Yudin. 3dgraphllm: Combining semantic graphs and large language models for 3d scene understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

Chenyangguang Zhang, Alexandros Delitzas, Fangjinhua Wang, Ruida Zhang, Xiangyang Ji, Marc Pollefeys, and Francis Engelmann. Open-vocabulary functional 3d scene graphs for real-world indoor spaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19401–19413, 2025.

Yingjie Zhu, Xuefeng Bai, Kehai Chen, Yang Xiang, Jun Yu, and Min Zhang. Benchmarking and improving large vision-language models for fundamental visual graph understanding and reasoning. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 30678–30701, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1482. URL https://aclanthology.org/2025.acl-long.1482/.

From Pixels to Symbols: Machine-Native Symbolic World Models for Multimodal Reasoning

Supplementary Material

CONTENTS

A *SymbolicBench* **15**
A.1 Data Preprocessing. 15
A.2 Data Sample Illustration 15

B Robustness Analysis **17**
B.1 Robustness Analysis of ASCII Map as Representation 17
B.2 Robustness Analysis of Proposed Transformation Pipeline 17

C SIG Prediction Pipeline for Real-world Images **19**
C.1 Results 19
C.2 Overview and perception features 19
C.3 Learning general and appearance-based spatial predictors 20
C.4 Scene-to-grid Transformer 20
C.5 Tree predictors and training protocol 21
C.6 Ensemble construction 22
C.7 Inference and spatial-map composition 23

A *SymbolicBench*

A.1 Data Preprocessing.

Since the original image inside CubiCasa5K does not contain grid as background, which makes it hard for evaluating the model's performance, we develop a preprocessing pipeline that converts each original floorplan as shown in Fig. 6 into a uniform rasterized grid. Specifically, we first parse the original SVG annotations and assign room-level semantic labels by modifying the tags associated with corresponding spatial regions. We then apply Otsu's thresholding to convert the floorplan into a grayscale or binary representation. After thresholding, the image is partitioned into non-overlapping patches with a fixed patch size of 30 pixels. Grid lines are overlaid along patch boundaries to obtain the final grid-like floorplan representation. Finally, we implement an annotation tool to place A–Z tags on the rasterized grid, where different colors are used to distinguish room-level labels. This process transforms realistic floorplans into structured navigation maps while preserving their spatial layout.

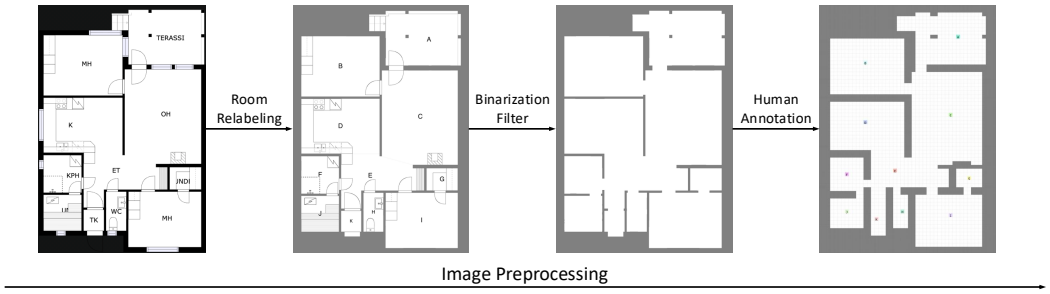

Figure 6: **Floorplan image preprocessing pipeline.**

A.2 Data Sample Illustration

We provide the visualization of data sample illustration of *SymbolicBench* shown in Fig. 7.

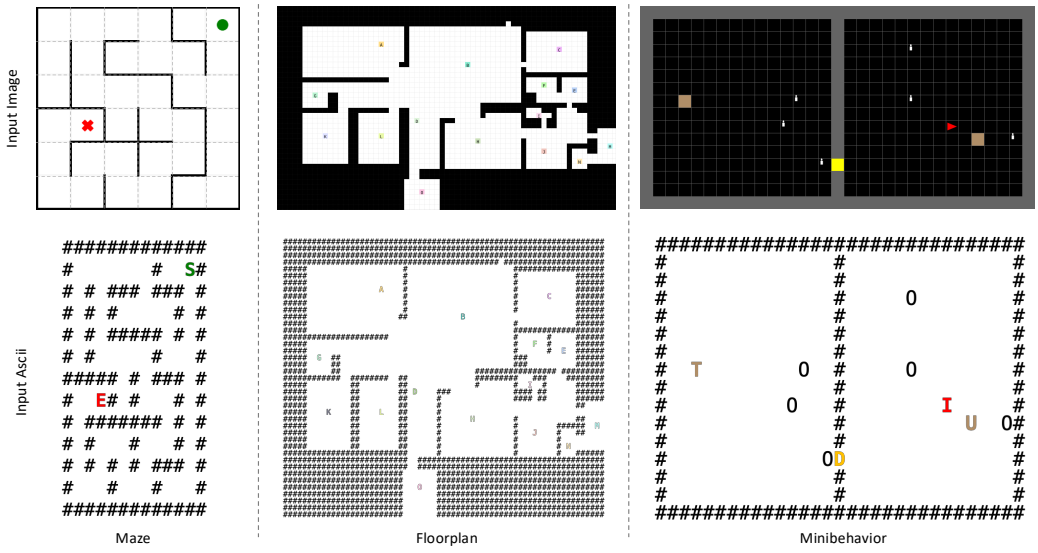

Figure 7: Examples from three subsets in *SymbolicBench*.

Table 4: Noise types and their manifestations across datasets. FP, MB and SIG means floorplan, Mini-BEHAVIOR and SIGBench-tiny dataset.

	Noise type 1: Entity Localization Noise	**Noise type 2: Spurious OOV Injection Noise**	**Noise type 3: Symbol Dropout Noise**	**Noise type 4: In-Vocabulary Misclassification Noise**	**Noise type 5: Structural Topology Noise**
FP	Room label shifted to a nearby cell.	Unknown symbol inserted into empty space.	Room label is missed.	Floor or doorway is mistaken for a wall.	Missing wall creates a false opening.
MB	Correct symbol at the wrong location.	False detection inserts an unknown symbol.	Task-relevant object is missed.	Object is mistaken for an obstacle.	Segmentation alters scene connectivity.
SIG	Object displaced by localization or alignment error.	Artifact interpreted as an unknown symbol.	Non-query object is missed.	Object is mistaken for a boundary.	Grid shift distorts the scene layout.

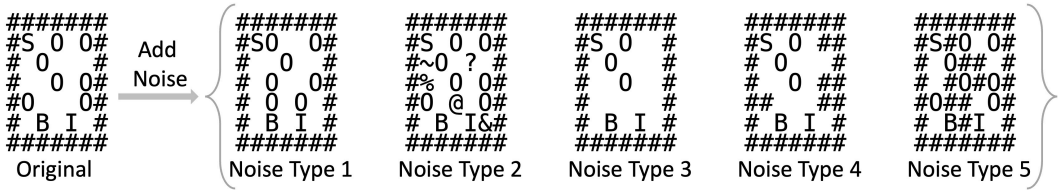

Figure 8: **Visualization of different noise types.** We provide the visualized illustration of different added noise types, corresponding to the noise types inside Tab. 4.

B Robustness Analysis

B.1 Robustness Analysis of ASCII Map as Representation

Noise injection. We conduct experiment by injecting noise. The following compact example illustrates the five noise types on a reduced 5×5 Mini-BEHAVIOR grid. For concision, we show the shared original map and only the 50% variant of each noise type; the ellipsis denotes the omitted 10% and 30% levels. The actual evaluation scenes are substantially larger. Noise types 1–5 are defined in the Tab. **??** and further illustrated in Fig. 8. We also conduct experiments on datasets with added noise. We also conduct experiments on datasets with added noise. Specifically, we evaluate three models on Floorplan, Mini-BEHAVIOR, and SIGBench-tiny at noise levels of 10%, 30%, and 50%, averaging results across five noise types. As shown in Table 3, performance generally declines as noise increases, yet noisy ASCII inputs consistently outperform original image inputs across all evaluated models and metrics, even at 50% noise. For example, GPT-5-Mini achieves an AAR of 0.497 on Mini-BEHAVIOR at 50% noise, compared with 0.040 using original images, while Gemini-3-Flash maintains a PES of 0.903 on Floorplan, compared with 0.206. On SIGBench-tiny, accuracy remains between 0.767 and 0.783 at the highest noise level, substantially exceeding the original-image range of 0.417–0.483. These results suggest that the benefits of ASCII representations persist under substantial corruption, although their sensitivity to noise varies across models and tasks.

Symbol sensitivity and grid size ablation. We examine sensitivity to symbol choices using two ASCII variants on Floorplan with GPT-5-Mini (Table 5). In Reverse ASCII, we swap the symbols for interior walls and walkable space: interior `#` cells become blank, blank floor cells become `#`, and the outer boundary is represented by $\cdot$. In Substitute ASCII, blank walkable cells are replaced with `$`, while wall cells retain `#`. We further evaluate the effect of grid resolution using GPT-5-Mini and Gemini-3-Flash at 20, 30, and 40 pixels per cell on the same 30 cases (Table 6). We report the mean grid cell count, calculated as the average number of ASCII cells (rows $\times$ columns) across these cases and rounded to the nearest integer.

Table 5: **Performance comparison of different input modes on Floorplan using GPT-5-Mini.** Bold values indicate the best performance.

Input mode	LWS↑	PES↑	DPF↑
Image	0.000	0.133	0.118
ASCII	0.324	0.633	0.519
Reverse ASCII	**0.339**	**0.724**	**0.650**
Substitute ASCII	0.169	0.667	0.615

Table 6: **Performance comparison across different grid resolutions.** Grid size is measured in pixels per cell. Scores are reported on a 0–1 scale. Mean grid cell count is the average number of ASCII cells (rows $\times$ columns) across the same 30 cases, rounded to the nearest integer.

Grid size	Model	LWS↑	PES↑	DPF↑	Mean grid cell count
20	GPT-5-Mini	0.033	0.609	0.677	9,404
	Gemini-3-Flash	0.566	0.930	0.920	9,404
30	GPT-5-Mini	0.067	0.741	0.723	4,190
	Gemini-3-Flash	0.831	0.959	0.958	4,190
40	GPT-5-Mini	0.230	0.841	0.776	2,374
	Gemini-3-Flash	0.895	0.960	0.939	2,374

B.2 Robustness Analysis of Proposed Transformation Pipeline

Degraded images. We apply the following three types of image degradation to evaluate the robustness of our image-to-ASCII conversion pipeline on Maze dataset. Gaussian noise independently perturbs RGB pixel intensities with zero-mean random values, whereas Gaussian blur suppresses

high-frequency image details through isotropic smoothing. Edge softening applies Gaussian smoothing only around Canny-detected boundaries, weakening grid lines and object contours while largely preserving uniform regions. Here, cell detection accuracy measures whether grid lines are correctly identified and cells are accurately segmented and assigned to their corresponding grid positions. Object detection accuracy measures whether special object cells are correctly recognized and localized.

Table 7: **Robustness of the image-to-ASCII conversion pipeline to image degradation on Maze.** We evaluate Gaussian noise, Gaussian blur, and edge softening around Canny-detected boundaries at four severity levels. Cell detection accuracy measures correct grid-line identification, cell segmentation, and grid-position assignment. Object detection accuracy measures correct recognition and localization of special object cells. Accuracies are reported on a 0–1 scale.

Noise family	Level	Configured setting	Cell detection accuracy	Object detection accuracy
Gaussian noise	1	$\sigma \sim \mathcal{U}[0.0, 20.4]$	1.000	1.000
	2	$\sigma \sim \mathcal{U}[20.4, 45.9]$	1.000	0.817
	3	$\sigma \sim \mathcal{U}[45.9, 66.3]$	1.000	0.558
	4	$\sigma \sim \mathcal{U}[66.3, 96.9]$	0.625	0.275
Gaussian blur	1	$\sigma = 0.6$	1.000	1.000
	2	$\sigma = 1.2$	1.000	1.000
	3	$\sigma = 2.4$	1.000	1.000
	4	$\sigma = 4.0$	1.000	1.000
Edge softening	1	$\sigma = 0.75$, blend $= 0.25$, radius $= 1$	1.000	1.000
	2	$\sigma = 1.5$, blend $= 0.50$, radius $= 1$	1.000	1.000
	3	$\sigma = 3.0$, blend $= 0.75$, radius $= 2$	1.000	1.000
	4	$\sigma = 5.0$, blend $= 1.00$, radius $= 3$	1.000	1.000

C SIG PREDICTION PIPELINE FOR REAL-WORLD IMAGES

C.1 RESULTS

Instead of directly use GT SIG and transform it to ASCII map, we also design an end-to-end pipeline that can directly transform an input image from SIGBench into SIG and then transform it into an ASCII map. We evaluate the quality of predicted SIG and the performance of using the predicted ASCII map as input. The results are shown in Tab. 2. As illustrated in the results, although the automatically predicted ASCII maps remain less accurate than the ground-truth ones, they nevertheless improve MLLM performance relative to the image-only setting. These findings suggest that the benefits of ASCII-based representations are not limited to manually constructed or idealized inputs and may persist even when the representations are generated through an imperfect automated pipeline.

C.2 OVERVIEW AND PERCEPTION FEATURES

Our pipeline learns to predict spatial intelligence grid (SIG) from frozen perception features and annotated spatial maps. It combines three vehicle predictors: a geometry-aware ensemble P, an appearance-enriched ensemble H, and a scene-to-grid Transformer T. A two-vote consensus determines vehicle coordinates, while an all-category regressor B supplies traffic-sign and traffic-light coordinates. This appendix describes how perception outputs become learning inputs, how the predictors are trained, and how their predictions are combined.

Data and supervision. Training uses 1,323 non-tiny SIGBenchmark scenes with 4,411 vehicles; B uses the 1,318 scenes with usable proposal–target pairs. SIGBench-tiny evaluation contains 90 scenes, 302 vehicles, 112 traffic signs, and 80 traffic lights, with no overlap in training and evaluation scene identifiers. The setting provides vehicle boxes and counts, ego position, lane and attention information, and scene metadata. Vehicle proposals are annotation-conditioned, and B uses annotation-assisted object association at both training and evaluation.

Frozen perception models. SAM 3 produces object masks, using vehicle box prompts and category text prompts for signs and lights. Depth Anything 3 supplies the principal dense depth map; Depth Pro adds metric-depth cues for a small/far vehicle specialist. DINOv1 ViT-B/16 provides 768-dimensional object-crop and scene descriptors, denoted v_i and v_s. DINOv3 ViT-B/16 supplies patch features for pooled object representations and local appearance tokens. All perception models remain frozen; spatial labels supervise the downstream networks and tree models.

From perception outputs to learning inputs. Numerical features combine normalized box geometry, box/mask depth statistics, mask shape, ground-contact contours, and relative object positions within a scene. Depth statistics use valid positive samples and scene-relative normalization. For DINOv3, we concatenate mask-weighted mean, foreground maximum, surrounding-ring mean, and box mean features into $r_i \in \mathbb{R}^{3072}$. The scene-to-grid model additionally receives up to 24 foreground and eight surrounding-ring tokens, together with relative position, mask occupancy, and token-type metadata. Table 8 summarizes the inputs; numerical standardization and PCA are fitted using training data.

Predictor	Numerical feature groups	Dimension
General network / box trees	Box geometry and depth	34
Mask trees	Box/depth and mask features	65
Appearance network	Box/depth, masks, and scene relations	83
Scene-to-grid network	Appearance-network groups and ground contact	113
Ground trees / single-vehicle experts	Box/depth, masks, contact, and neighbor relations	109
Small/far expert	Ground-tree groups and Depth Pro cues	159

Table 8: Numerical inputs. Neural networks also receive DINOv1 descriptors; the appearance and scene-to-grid networks additionally use DINOv3 features.

C.3 Learning general and appearance-based spatial predictors

Coordinate representation and matching prior. For target (x, y) and supplied ego lateral position s_x, we use 15 lateral classes, ten forward classes, and 150 joint cells:

$$\ell = \mathrm{clip}(\mathrm{round}(x - s_x) + 7, 0, 14), \qquad f = \mathrm{clip}(\mathrm{round}(y), 0, 9), \qquad c = 10\ell + f. \tag{8}$$

Only the lateral coordinate is ego-relative; canonical output coordinates are $(\ell-2, f)$ with ego position $(5, 0)$. An initial proxy discretizes horizontal box center and mean box depth into a 10×10 grid, after normalizing by image width and maximum scene depth. Its shifted coordinates $\pi_i = (g_i^x + 2, g_i^y)$ provide a matching prior, rather than a metric bird's-eye-view reconstruction.

General spatial set network. The general network jointly processes vehicles, signs, and lights. Projected numerical features and DINOv1 crop descriptors form object tokens; a category embedding identifies each object class, and a projected DINOv1 scene descriptor supplies scene context. Four width-256 Transformer layers with eight attention heads model interactions among these tokens. Object heads predict lateral logits a_i, forward logits b_i, and joint residuals j_i:

$$J_i(\ell, f) = j_i(10\ell + f) + a_i(\ell) + b_i(f). \tag{9}$$

Softmax gives axis distributions p_i^ℓ, p_i^f and joint distribution p_i^J. Auxiliary heads predict objectness, vehicle type/color, category counts, and ego lateral position. Category-wise Hungarian matching uses

$$C_{ij} = -\log p_i^J(t_j) + 0.20\|\widehat{\mu}_i - t_j\|_1 + 0.12\rho_e\|\pi_i - t_j\|_1, \quad \rho_e = \max\left(0, 1 - \frac{e-1}{30}\right), \tag{10}$$

where $t_j = (\ell_j, f_j)$ and $\widehat{\mu}_i$ is the expected coordinate. Training combines axis/joint cross-entropy, SmoothL1 coordinate loss, axis cumulative-distribution loss, auxiliary classification, and a pairwise term that preserves relative positions:

$$\mathcal{L}_{\mathrm{pair}} = \mathrm{mean}_{i,j}\,\mathrm{SmoothL1}\big[(\widehat{\mu}_i - \widehat{\mu}_j) - (t_i - t_j)\big]. \tag{11}$$

The mean is over matched pairs and coordinate components, and is zero for fewer than two matches. Coordinate and attribute labels use smoothing 0.02. The general network supplies coordinate probabilities to the ensembles and vehicle attributes to the final map.

Vehicle appearance network. A vehicle-specific adaptation of the general network initializes the appearance model. Adaptation pairs left-to-right proposals with targets sorted by lateral then forward coordinate, while freezing objectness, count, and ego heads. The appearance model expands the numerical input to 83 dimensions and adds a gated projection of r_i to each object token:

$$h_i^0 = P_n(\widetilde{x}_i) + P_v(v_i) + \mathrm{sigmoid}(\gamma_p)P_r(r_i) + e_{\mathrm{veh}}. \tag{12}$$

Here $\widetilde{x}_i$ is standardized, P_r maps $3072 \to 512 \to 256$, and γ_p starts at -2.5. Four width-256, eight-head Transformer layers produce the heads in Equation equation 9. Training uses dynamic Hungarian matching with cost $-\log p_i^\ell(\ell_j) - \log p_i^f(f_j) - 0.5\log p_i^J(t_j) + \lambda_e\|\pi_i - t_j\|_1$, where λ_e decreases from 1.1 to 0.1, and minimizes

$$\mathcal{L}_{\mathrm{app}} = \mathrm{CE}_\ell^{(0.02)} + \mathrm{CE}_f^{(0.02)} + \mathrm{CE}_J^{(0.02)} + 0.35\,\mathrm{SmoothL1}(\widehat{\mu}, t) + 0.10\mathcal{L}_{\mathrm{pair}}. \tag{13}$$

For ensemble use, joint marginals $q_i^\ell = \sum_f p_i^J$ and $q_i^f = \sum_\ell p_i^J$ are combined with the corresponding axis heads as $\widetilde{p}_i^a = \mathrm{softmax}(\log p_i^a + \log q_i^a)$.

C.4 Scene-to-grid Transformer

The scene-to-grid predictor T learns object coordinates together with scene-level cell occupancy. It uses the 113-dimensional numerical input, DINOv1 crop/scene descriptors, and DINOv3 local tokens. Unlike the pooled appearance representation, local tokens retain spatial variation within and around each object.

Object and scene encoding. Numerical and crop projections initialize width-192 object features h_i^0; DINOv3 tokens and their positional metadata are projected to the same width and processed by one six-head Transformer layer. An object queries its valid local tokens t_{ik} through gated attention:

$$\alpha_{ik} = \mathrm{softmax}_k\left(\frac{(h_i^0)^\top t_{ik}}{\sqrt{192}}\right), \qquad \widetilde{h}_i = h_i^0 + \mathrm{sigmoid}(\gamma_z)\sum_k \alpha_{ik} t_{ik}. \tag{14}$$

The gate starts at $\gamma_z = -1.5$. Three further width-192, six-head Transformer layers process the object tokens with a projected scene token, yielding object features h_i and scene summary h_s.

Scene-dependent forward calibration. Lateral logits are predicted from h_i. Forward prediction uses ordered anchors derived from the scene. Positive increments $\delta_k = \mathrm{softplus}(w_k^\top h_s + b_k) + 0.05$ define cumulative positions $s_k = \sum_{j=0}^{k} \delta_j$ and anchors

$$\mu_k = 9\frac{s_k - s_0}{s_9 - s_0 + 10^{-6}}, \qquad k = 0, \ldots, 9. \tag{15}$$

With object coordinate $z_i = 9\,\mathrm{sigmoid}(w_z^\top h_i + b_z)$ and scene scale $\sigma_s = 0.35 + \mathrm{softplus}(w_\sigma^\top h_s + b_\sigma)$, forward logits are

$$b_i(k) = b_i^{\mathrm{residual}}(k) + b_s(k) - \frac{\mathrm{sigmoid}(\gamma_f)}{2}\left(\frac{z_i - \mu_k}{\sigma_s}\right)^2. \tag{16}$$

The ordered anchors let the scene determine the mapping from continuous forward evidence to discrete classes; γ_f starts at -1.5.

Cell queries and joint scores. Each of the 150 cells has a learned query augmented with its grid position. Cross-attention to scene/object features produces contextual cell features g_c and occupancy logits o_c. Joint residuals combine a linear object head and normalized object–cell similarity:

$$j_i(c) = w_c^\top h_i + b_c + 4\left\langle \frac{W_o h_i}{\|W_o h_i\|_2}, \frac{W_g g_c}{\|W_g g_c\|_2} \right\rangle. \tag{17}$$

The full joint logits follow Equation equation 9.

Training objective. Hungarian matching uses the appearance-model cost, with proxy weight decreasing from 0.80 to 0.05. For each scene, the coordinate objective is

$$\begin{aligned}\mathcal{L}_{\mathrm{coord}} = {} & \mathrm{CE}_\ell^{(0.01)} + 1.35\,\mathrm{CE}_f^{(0.01)} + 0.65\,\mathrm{CE}_J + 0.30\,\mathrm{SmoothL1}(\widehat{\mu}, t) \\ & + 0.20\,\mathrm{SmoothL1}(z_i, f_i^*) + 0.14\mathcal{L}_{\mathrm{pair}}.\end{aligned} \tag{18}$$

Cells containing matched targets have occupancy label $y_c = 1$. With m matches and positive weight $w_+ = \min(20, (150 - m)/\max(1, m))$, occupancy loss is

$$\mathcal{L}_{\mathrm{occ}} = 0.45\,\mathrm{BCE}(o, y; w_+) + 0.20\left(1 - \frac{2\sum_c \mathrm{sigmoid}(o_c) y_c + 1}{\sum_c \mathrm{sigmoid}(o_c) + \sum_c y_c + 1}\right). \tag{19}$$

Training minimizes the batch mean of $\mathcal{L}_{\mathrm{coord}} + \mathcal{L}_{\mathrm{occ}}$. With probability 0.5, geometric features and token positions are horizontally reflected and lateral targets become $14 - \ell$. DINO descriptors remain unchanged, so this is an approximate reflection in feature space.

C.5 Tree predictors and training protocol

Complementary coordinate classifiers. ExtraTrees and XGBoost learn separate lateral and forward classifiers from box/depth features (34 dimensions) and box/depth/mask features (65 dimensions). Their targets are assigned by category-wise Hungarian matching with cost $\|\pi_i - t_j\|_1$. For P, additional classifiers use the 109-dimensional ground-contact representation. A further ExtraTrees model appends 96 whitened principal components of pooled DINOv3 features, combining geometry with compressed appearance in a 205-dimensional input. ExtraTrees uses 600 trees for box/mask models in H and 700 for those in P and ground models; XGBoost uses 250 depth-three rounds at learning rate 0.04.

Geometric specialists. Three types of experts provide conditional corrections to P. A forward-coordinate bag averages five ExtraTrees models on ground features, varying leaf size and feature subsampling. A small/far expert uses the 159-dimensional input, adding Depth Pro range, contact-depth, and apparent-size cues. It is trained on vehicles whose normalized box area is below the training median and whose median mask depth is above the training median, with equality included at both thresholds. Separate lateral and forward experts handle scenes containing exactly one selected vehicle proposal. All experts use class-balanced ExtraTrees; their corrections are gated as described in Section C.6.

All-category coordinate regression. The regressor B uses 43 features describing category, object order and count, ego position, box/depth geometry, and participant, route, and frame metadata. Category-wise annotation-assisted matching associates proposals with supplied spatial targets using $|9u_i/W - x_j| + 0.20|g_i^y - y_j|$, where u_i is horizontal box center. Two histogram-gradient-boosted regressors minimize absolute coordinate error, with learning rate 0.06 and at most 220 boosting iterations. Model selection uses a 20% scene holdout, followed by refitting on all 1,318 eligible scenes. Predictions are rounded and clipped to $[0, 9]$. The final composition retains this branch's sign/light coordinates and the supplied identities, ego position, lanes, and attention information.

Neural optimization. Neural predictors use AdamW, cosine decay, gradient clipping, and seed 2026 (Table 9). The general and appearance networks share a 1,058/265 scene fitting/validation split; 1,020 fitting scenes have usable vehicle sets. General-model selection maximizes assignment recall minus 0.01 times matched Manhattan distance. The one-epoch vehicle adaptation initializes the appearance network. Both retained networks remain trained on their fitting subsets.

Network	Learning rate	Weight decay	Batch	Selected epoch
General	7×10^{-4}	2×10^{-4}	48	75
Appearance	1.5×10^{-4}	5×10^{-4}	48	22
Scene-to-grid	1.5×10^{-4}	8×10^{-4}	16	8 (final fit)

Table 9: Optimization settings. Maximum selection runs are 100, 50, and 24 epochs, respectively. Scene-to-grid selection uses five participant folds before the final eight-epoch fit.

Validation and final fitting. The 265-scene holdout selects ensemble weights and specialist settings. During this selection, trees, PCA, and eligibility thresholds use only fitting scenes; these components are then refitted on the eligible full training pool. For T, five folds keep each participant in one validation fold. Checkpoints are selected by exact-cell set F1, breaking ties by matched Manhattan distance. Out-of-fold predictions select its decoding weights; a newly initialized final model is trained on all eligible scenes for eight epochs, the rounded median selected epoch. Both learning-rate and matching-prior schedules are recomputed for this duration.

C.6 Ensemble construction

Validation criterion and probability pooling. Selection measures exact-cell agreement between vehicle multisets, independently of type/color. Each scene/cell contributes the smaller of its predicted and target counts to precision, recall, and F1; optimal matched Manhattan distance measures coordinate error. For axis $a \in \{\ell, f\}$, model probabilities are combined into log scores:

$$\mathcal{P}_i^a(k; w) = \frac{\sum_m w_m^a \log \max(p_{im}^a(k), 10^{-8})}{\sum_m w_m^a}. \tag{20}$$

Common-source weights are searched over $\{0, 0.5, 1, 2, 3\}$, excluding the all-zero vector. Axis-level recall screens candidates; full-cell agreement under the inference assignment rule selects lateral/forward combinations.

Appearance and geometry ensembles. H directly pools six common sources. P first pools the same source families and applies softmax; that distribution is then pooled with ground ExtraTrees, ground XGBoost, and ground-plus-PCA ExtraTrees. Table 10 gives the selected weights. Each pooling stage retains the normalization in Equation equation 20, so the outer pool treats the normalized inner ensemble as one source.

Confidence-gated specialist corrections. Starting from the outer scores of P, an eligible expert modifies an axis only when it disagrees with the current prediction and is sufficiently confident. With eligibility g_i, expert distribution p_i^{E}, and current score S_i, the update is

$$I_i = \mathbf{1}[g_i = 1]\mathbf{1}[\arg\max p_i^{\mathrm{E}} \neq \arg\max S_i]\mathbf{1}[\max_k p_i^{\mathrm{E}}(k) \geq \tau],$$
$$S_i'(k) = S_i(k) + I_i \omega \log \max(p_i^{\mathrm{E}}(k), 10^{-8}). \tag{21}$$

The corrections in Table 11 are sequential; each disagreement test uses the updated scores. The confidence cutoff τ is the listed quantile among eligible, disagreeing predictions in the current

Pool	Lateral weights	Forward weights
H (common sources)	$(0, 3, 3, 0, 0, 0.5)$	$(0.5, 2, 3, 1, 0, 0.5)$
P (inner/common)	$(1, 3, 1, 0.5, 0.5, 0)$	$(0, 2, 2, 0, 0.5, 0.5)$
P (outer/ground)	$(3, 0, 0.5, 2)$	$(2, 0.5, 1, 0.5)$

Table 10: Common-source order: general network, box ExtraTrees, mask ExtraTrees, box XGBoost, mask XGBoost, and appearance network. Outer-pool order: normalized inner pool, ground ExtraTrees, ground XGBoost, and ground-plus-PCA ExtraTrees.

evaluation collection. It is computed from evaluation-time predictions; small/far area and depth eligibility thresholds remain fitted on training data.

Expert / eligibility	Axis	Weight ω	Quantile
Five-model bag / all vehicles	Forward	0.25	0.85
Small/far Depth Pro expert	Forward	1.00	0.85
Single-vehicle expert	Lateral	0.25	0.92
Single-vehicle expert	Forward	0.25	0.50

Table 11: Ordered corrections applied to P.

C.7 Inference and spatial-map composition

Feature extraction and branch predictions. For each evaluation scene, the frozen perception models produce the same mask, depth, and appearance features used in training. Training-fitted standardization and the full-training PCA transform are applied. The general and appearance networks and tree classifiers produce coordinate distributions, which form H and P through the pools and specialist corrections in Section C.6. No model parameters are updated during inference.

For T, predictions are averaged over original and approximately reflected features. The reflected lateral class axis is reversed:

$$\bar{p}_i^{\ell}(k) = \tfrac{1}{2}\big[p_{i,\mathrm{orig}}^{\ell}(k) + p_{i,\mathrm{refl}}^{\ell}(14-k)\big], \qquad \bar{p}_i^{f}(k) = \tfrac{1}{2}\big[p_{i,\mathrm{orig}}^{f}(k) + p_{i,\mathrm{refl}}^{f}(k)\big]. \tag{22}$$

The same procedure applies to joint marginals q_i^a and axis marginals u_s^a of the sigmoid occupancy map. Reflection changes geometric features and token positions while preserving DINO appearance vectors. The averaged quantities form the scene-to-grid scores:

$$S_{T,i}^{a} = \log \bar{p}_i^{a} + 0.25 \log \bar{q}_i^{a} + 0.50 \log \bar{u}_s^{a}. \tag{23}$$

The coefficients are selected on out-of-fold predictions and fixed at inference. Occupancy marginals sum sigmoid cell probabilities along the other axis, without first normalizing the map to sum one.

Distinct-cell assignment within each branch. For each of P, H, T, Hungarian assignment maps selected vehicle proposals to distinct cells in the 150-cell internal grid:

$$\widehat{A}_m = \operatorname*{arg\,max}_{A \text{ injective}} \sum_i \Big[S_{m,i}^{\ell}(\ell_{A(i)}) + S_{m,i}^{f}(f_{A(i)})\Big], \qquad m \in \{P, H, T\}. \tag{24}$$

Assigned cells are converted to canonical coordinates $(\ell - 2, f)$. All branches use a common vehicle identity list derived from the general network's type/color predictions, with an objectness threshold of 0.25. Cells are ordered by proposal horizontal center and paired with this shared list; the counts must agree. Coordinate scoring includes all selected proposals, independently of the identity threshold.

Vehicle consensus. For vehicle identity k, write the three predicted cells as $\widehat{c}_{P,k}$, $\widehat{c}_{H,k}$, and $\widehat{c}_{T,k}$. A cell receives one vote from each branch that predicts it:

$$n_k(c) = \sum_{m \in \{P,H,T\}} \mathbf{1}[\widehat{c}_{m,k} = c], \qquad \widehat{c}_{V,k} = \begin{cases} c, & n_k(c) \geq 2, \\ \widehat{c}_{P,k}, & \text{otherwise.} \end{cases} \tag{25}$$

Two agreeing branches determine the output; if all three disagree, the geometry-aware prediction is retained. A missing alternative identity contributes no vote. This combines the geometry, pooled-appearance, and scene-to-grid predictions at the level of complete spatial cells.

Complete SIG. The regressor B predicts coordinates for annotation-associated objects. Its sign/light coordinates and supplied ego, lane, and attention fields are retained. Vehicle consensus coordinates replace its vehicle component, after translating from canonical ego position $(5, 0)$ to the supplied ego position s^B:

$$c_{F,k} = \widehat{c}_{V,k} + s^B - (5, 0). \tag{26}$$

The resulting map combines learned object coordinates with the supplied contextual fields. Distinct-cell assignment is performed within each vehicle branch before voting; the final composition does not apply another uniqueness assignment or additional coordinate clipping.

Transform predicted SIG to ASCII Map. With the predicted SIG, we are able to transform it to ASCII map using our designed pipeline as mentioned inside the method.