Autonomous Spatial Intelligence: A Comprehensive Technical Report for Engineering Teams at AtlasPro AI

Agentic AI Methods, System Architectures, Implementation Patterns, and Deployment Strategies

Gloria Felicia[1] Nolan Bryant[1] Handi Putra[1] Ayaan Gazali[1]
Eliel Lobo[1] Esteban Rojas[1]

[1]AtlasPro AI Research Division
{gloria.felicia, nolan.bryant, handi.putra, ayaan.gazali, eliel.lobo, esteban.rojas}@atlaspro.ai

Internal Technical Report | Version 1.0 | January 2026

Abstract

This comprehensive technical report provides an engineering-focused analysis of autonomous spatial intelligence systems for AtlasPro AI engineering teams. We synthesize over 500 papers spanning agentic AI architectures [Yao et al., 2023b, Shinn et al., 2023, Wang et al., 2024b, Xi et al., 2023], vision-language-action models [Brohan et al., 2023, Team et al., 2024, Kim et al., 2024, Driess et al., 2023], graph neural networks [Kipf and Welling, 2017, Velickovic et al., 2018, Wu et al., 2019a, Jin et al., 2023], world models [Hafner et al., 2023, Hu et al., 2023a, Yang et al., 2024], and geospatial foundation models [Jakubik et al., 2024, Cong et al., 2022, Bastani et al., 2023]. Unlike academic surveys, this report emphasizes practical implementation: system architecture patterns, data pipeline design, computational requirements, integration strategies, and safety engineering. We provide reference architectures for spatial AI agents, detailed analysis of GNN-LLM integration patterns, comprehensive benchmark evaluation frameworks, and deployment considerations for production systems. This document serves as the foundational engineering reference for building next-generation spatially-aware autonomous systems at AtlasPro AI.

Contents

1 Executive Summary for Engineering Leadership

1.1 Strategic Context

The convergence of agentic AI and spatial intelligence represents a transformative opportunity for AtlasPro AI. This report provides the technical foundation for our engineering teams to build systems that can perceive, reason about, and act within physical environments autonomously.

Market Opportunity. The spatial AI market is projected to reach significant scale by 2030, driven by demand in autonomous vehicles, robotics, smart cities, and geospatial intelligence. Companies like Waymo [Waymo, 2023], Palantir [Palantir, 2023], and ESRI [ESRI, 2023] are leading this transformation.

Technical Readiness. Recent advances in large language models [Brown et al., 2020, OpenAI, 2023,?], vision-language models [Liu et al., 2023a, Alayrac et al., 2022], and robotics foundation models [Team et al., 2024, Kim et al., 2024] have created the technical conditions for building truly capable spatial AI systems.

1.2 Key Technical Findings

Based on our comprehensive analysis of over 500 papers, we identify the following key findings for engineering teams:

1. Memory architecture is critical: hierarchical memory systems combining short-term context, long-term retrieval, and spatial cognitive maps are essential for complex spatial tasks [Packer et al., 2023, Huang et al., 2023, Chaplot et al., 2020].
2. GNN-LLM integration is a key enabler: the combination of graph neural networks for structural reasoning with LLMs for semantic understanding represents a powerful paradigm [Tang et al., 2024, Wang et al., 2024a].
3. World models enable safe planning: learning predictive models of the environment enables planning through imagination, critical for safety-critical applications [Hafner et al., 2023, Hu et al., 2023a].
4. Open-source models are production-ready: models like Octo [Team et al., 2024] and OpenVLA [Kim et al., 2024] provide strong baselines for robotics applications.
5. Evaluation infrastructure is essential: building robust internal benchmarking capabilities is critical for measuring progress and ensuring quality [Liu et al., 2023b, Yang et al., 2025].

1.3 Recommended Engineering Priorities

Based on our analysis, we recommend the following engineering priorities for AtlasPro AI:

1. Build a unified memory infrastructure supporting RAG, cognitive mapping, and episodic memory.
2. Develop GNN-LLM integration capabilities for spatial reasoning tasks.
3. Establish simulation infrastructure using Habitat [Savva et al., 2019] and Isaac Sim for safe development.
4. Create internal benchmarking framework for continuous evaluation.
5. Implement safety engineering practices from the start.

2 Agentic AI Foundations

This section provides detailed technical analysis of agentic AI architectures relevant to spatial intelligence applications.

2.1 Memory Systems

Memory enables agents to accumulate and retrieve experiential knowledge, forming the foundation for learning and adaptation.

Short-Term Memory. In-context learning [Brown et al., 2020] enables immediate reasoning within the context window. For spatial tasks, this includes recent observations, current goals, and immediate action history. The context window limitation (typically 4K-128K tokens) constrains the amount of information available for immediate reasoning.

Long-Term Memory. Retrieval-augmented generation [Lewis et al., 2020] enables knowledge persistence beyond the context window. MemGPT [Packer et al., 2023] introduces hierarchical memory management with explicit memory operations. For spatial applications, long-term memory stores maps, object knowledge, and procedural skills.

Spatial Memory. VLMaps [Huang et al., 2023] creates semantic spatial memory by grounding vision-language features in 3D space. Neural SLAM [Chaplot et al., 2020] learns to build spatial maps end-to-end. These approaches enable agents to maintain persistent spatial understanding across episodes.

2.2 Planning Systems

Planning decomposes high-level goals into executable action sequences, essential for complex spatial tasks.

Chain-of-Thought Reasoning. Chain-of-thought prompting [Wei et al., 2022] enables step-by-step reasoning. Zero-shot CoT [Kojima et al., 2022] demonstrates that simply adding "Let's think step by step" improves reasoning performance. For spatial tasks, CoT helps decompose navigation and manipulation goals.

Tree-Based Search. Tree of Thoughts [Yao et al., 2023a] explores multiple reasoning paths. Graph of Thoughts [Besta et al., 2023] generalizes to arbitrary graph structures. These approaches enable more thorough exploration of solution spaces for complex spatial problems.

Hierarchical Planning. LLM-Planner [Song et al., 2023] uses LLMs for high-level planning with low-level skill execution. Inner Monologue [Huang et al., 2022] incorporates feedback for closed-loop planning. Hierarchical approaches are essential for bridging abstract goals with concrete actions.

2.3 Tool Use and Action

Tool use extends agent capabilities through external interfaces and action execution.

API Integration. Toolformer [Schick et al., 2023] learns to use tools through self-supervised learning. Gorilla [Patil et al., 2023] specializes in API calling. ToolLLM [Qin et al., 2024] provides comprehensive tool-use evaluation. These capabilities enable agents to access specialized functions for spatial tasks.

Code Generation. PAL [Gao et al., 2023] uses code as an intermediate representation for reasoning. Code as Policies [Liang et al., 2023] generates executable robot code from language. Code generation provides flexible, verifiable action specification.

ReAct Architecture. ReAct [Yao et al., 2023b] interleaves reasoning traces with action execution. This architecture forms the foundation for many spatial agents, enabling explicit reasoning about when and how to act.

3 Vision-Language-Action Models

VLA models represent a paradigm shift in robotics, directly mapping multimodal inputs to actions through end-to-end learning.

3.1 Architecture Overview

VLA models typically consist of three components: a vision encoder for processing visual observations, a language model for reasoning and instruction following, and an action head for generating robot commands.

Vision Encoders. Common choices include CLIP ViT [Radford et al., 2021], SigLIP [Zhai et al., 2023], and DINOv2 [Oquab et al., 2023]. The choice of vision encoder significantly impacts spatial understanding capabilities.

Language Models. VLA models build on pretrained LLMs including LLaMA [Touvron et al., 2023], Vicuna, and proprietary models. The language model provides reasoning, instruction following, and world knowledge.

Action Heads. Actions are typically represented as discretized tokens or continuous vectors. RT-2 [Brohan et al., 2023] discretizes actions into tokens. OpenVLA [Kim et al., 2024] uses a similar approach with 256 bins per dimension.

3.2 Proprietary Models

RT-1. RT-1 [Brohan et al., 2022] demonstrated transformer-based policies trained on large-scale robot data (130K demonstrations). Key innovations include TokenLearner for efficient visual processing and FiLM conditioning for language.

RT-2. RT-2 [Brohan et al., 2023] co-trained on web-scale vision-language data and robot demonstrations. This enabled emergent capabilities including reasoning about novel objects, following complex instructions, and chain-of-thought reasoning for robotics.

PaLM-E. PaLM-E [Driess et al., 2023] integrated continuous sensor data into a 562B parameter language model. The model demonstrated strong transfer from language to embodied tasks and emergent multimodal reasoning.

3.3 Open-Source Models

Octo. Octo [Team et al., 2024] provides a generalist robot policy trained on the Open X-Embodiment dataset [Collaboration, 2023] containing 800K trajectories from 22 robot embodiments. Key features include diffusion-based action heads and support for multiple action spaces.

OpenVLA. OpenVLA [Kim et al., 2024] offers a 7B parameter VLA built on Prismatic VLMs. It achieves competitive performance with RT-2 while being fully open-source. The model supports fine-tuning on custom robot data.

3.4 Training Considerations

Data Requirements. VLA training requires diverse robot demonstration data. The Open X-Embodiment dataset provides a starting point, but domain-specific data collection is typically necessary.

Computational Resources. Training 7B parameter VLAs requires 8+ A100 GPUs for 1-2 weeks. Fine-tuning can be done more efficiently with LoRA or similar techniques.

Evaluation. Evaluation should include both simulation benchmarks (RLBench, Meta-World) and real-world testing. Success rates, generalization to novel objects, and robustness to perturbations are key metrics.

4 Graph Neural Networks for Spatial Reasoning

GNNs provide powerful tools for modeling spatial relationships and dependencies, with emerging integration with language models.

4.1 Foundational Architectures

Graph Convolutional Networks. GCN [Kipf and Welling, 2017] introduced spectral graph convolution using the normalized graph Laplacian. The layer-wise propagation rule is:

$$H^{(l+1)} = \sigma(\tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}H^{(l)}W^{(l)}) \tag{1}$$

where $\tilde{A} = A + I$ is the adjacency matrix with self-loops and $\tilde{D}$ is the degree matrix.

Graph Attention Networks. GAT [Velickovic et al., 2018] introduced attention mechanisms for graph learning:

$$\alpha_{ij} = \frac{\exp(\text{LeakyReLU}(a^T[Wh_i \| Wh_j]))}{\sum_{k\in\mathcal{N}_i}\exp(\text{LeakyReLU}(a^T[Wh_i \| Wh_k]))} \tag{2}$$

This enables learning edge importance dynamically.

GraphSAGE. GraphSAGE [Hamilton et al., 2017] introduced inductive learning through neighborhood sampling and aggregation. This enables generalization to unseen nodes and graphs.

Graph Isomorphism Network. GIN [Xu et al., 2019] provided theoretical analysis showing that GNNs are at most as powerful as the Weisfeiler-Lehman test. The GIN update rule maximizes expressiveness:

$$h_v^{(k)} = \text{MLP}^{(k)}\left((1+\epsilon^{(k)})\cdot h_v^{(k-1)} + \sum_{u\in\mathcal{N}(v)} h_u^{(k-1)}\right) \tag{3}$$

4.2 Spatio-Temporal Graph Networks

DCRNN. Diffusion Convolutional Recurrent Neural Network [Li et al., 2018] models traffic flow as a diffusion process on a directed graph. It combines graph convolution with sequence-to-sequence learning for traffic prediction.

STGCN. Spatio-Temporal Graph Convolutional Networks [Yu et al., 2018] use graph convolutions for spatial dependencies and 1D convolutions for temporal dependencies. This achieves efficient spatio-temporal modeling.

Graph WaveNet. Graph WaveNet [Wu et al., 2019b] learns adaptive adjacency matrices, enabling discovery of hidden spatial dependencies. It combines dilated causal convolutions with graph convolutions.

AGCRN. Adaptive Graph Convolutional Recurrent Network [Bai et al., 2020] introduces node-specific patterns through node adaptive parameter learning. This captures heterogeneous spatial-temporal patterns.

4.3 GNN-LLM Integration Patterns

Recent work explores three main patterns for combining GNNs with LLMs:

Pattern 1: GNN as Encoder. The GNN encodes graph structure into embeddings that are fed to the LLM. GraphGPT [Tang et al., 2024] aligns graph encoders with language models through instruction tuning.

Pattern 2: LLM as Reasoner. The LLM reasons over graph-structured information retrieved by the GNN. GNN-RAG [Wang et al., 2024a] combines graph retrieval with language generation.

Pattern 3: Joint Training. GNN and LLM components are trained jointly for end-to-end optimization. This approach shows promise but requires significant computational resources.

5 World Models for Spatial Intelligence

World models learn predictive representations of the environment, enabling planning through imagination.

5.1 Model-Based Reinforcement Learning

Dreamer. Dreamer [Hafner et al., 2019] introduced latent imagination for model-based RL. The world model consists of:

- Representation model: $p(s_t|s_{t-1}, a_{t-1}, o_t)$
- Transition model: $p(s_t|s_{t-1}, a_{t-1})$
- Observation model: $p(o_t|s_t)$
- Reward model: $p(r_t|s_t)$

DreamerV2. DreamerV2 [Hafner et al., 2021] achieved human-level performance on Atari using discrete latent states and KL balancing. Key improvements include categorical latents and straight-through gradients.

DreamerV3. DreamerV3 [Hafner et al., 2023] demonstrated cross-domain mastery with a single algorithm and fixed hyperparameters. It successfully learned to collect diamonds in Minecraft from scratch.

DayDreamer. DayDreamer [Wu et al., 2023a] transferred world models to real robots, demonstrating that imagination-based planning can work in physical environments.

5.2 Video World Models

Genie. Genie [Bruce et al., 2024] learns controllable world models from internet videos without action labels. It uses a video tokenizer, latent action model, and dynamics model to enable interactive generation.

GAIA-1. GAIA-1 [Hu et al., 2023a] from Waymo produces realistic driving videos conditioned on text, actions, and past observations. It demonstrates the potential for world models in autonomous driving.

WorldDreamer. WorldDreamer [Yang et al., 2024] generates driving world models with multimodal conditioning. It enables simulation of diverse driving scenarios for training and testing.

5.3 LLM-Based World Models

LLMs can serve as implicit world models, predicting state transitions based on their world knowledge.

Reasoning via Planning. RAP [Hao et al., 2023] uses LLMs as world models for Monte Carlo Tree Search planning. The LLM predicts next states and rewards for planning.

Leveraging World Knowledge. Guan et al. [2023] demonstrated that LLM world knowledge can be extracted for planning in text-based environments.

6 Embodied AI Agents

This section covers complete agent systems that integrate perception, reasoning, and action for embodied tasks.

6.1 Open-Ended Exploration

Voyager. Voyager [Wang et al., 2023] demonstrated open-ended exploration in Minecraft through:

- Automatic curriculum: proposes increasingly difficult tasks
- Skill library: stores and retrieves reusable code skills
- Iterative prompting: refines code based on execution feedback

MineDojo. MineDojo [Fan et al., 2022] provides a comprehensive benchmark suite for open-ended embodied agents in Minecraft, including thousands of tasks and a large-scale video-text dataset.

6.2 Grounded Language Agents

SayCan. SayCan [Ahn et al., 2022] grounds language models in robotic affordances by combining:

- Language model scoring: $P(\text{skill}|\text{instruction})$
- Affordance scoring: $P(\text{success}|\text{skill}, \text{state})$
- Combined scoring: product of both probabilities

Code as Policies. Code as Policies [Liang et al., 2023] generates executable robot code from natural language. It uses hierarchical code generation with perception APIs and motion primitives.

LLM-Planner. LLM-Planner [Song et al., 2023] enables few-shot grounded planning by combining LLM reasoning with environment feedback. It demonstrates strong generalization to novel tasks.

6.3 Simulation Platforms

Habitat. Habitat [Savva et al., 2019] provides high-fidelity embodied AI simulation. Habitat 2.0 [Szot et al., 2021] added interactive objects. Habitat 3.0 [Puig et al., 2024] introduced human-robot interaction scenarios.

iGibson. iGibson [Shen et al., 2021, Li et al., 2021] offers interactive environments with realistic physics. It supports both navigation and manipulation tasks.

AI2-THOR. AI2-THOR [Kolve et al., 2017] enables interactive visual AI research with diverse indoor environments and object interactions.

7 Autonomous Driving Systems

Autonomous driving represents one of the most demanding applications of spatial AI.

7.1 End-to-End Architectures

UniAD. UniAD [Hu et al., 2023b] presents a unified framework integrating perception, prediction, and planning:

- BEV encoder: transforms multi-camera images to bird's-eye-view
- Track query: maintains object tracking across frames
- Motion query: predicts future trajectories
- Occupancy prediction: forecasts future occupancy grids
- Planning head: generates ego-vehicle trajectory

VAD. VAD [Jiang et al., 2023] introduces vectorized scene representation, representing scenes as sets of vectors (lanes, agents) rather than dense grids. This enables more efficient reasoning about scene structure.

EMMA. EMMA [Waymo, 2024] from Waymo demonstrates end-to-end multimodal driving with language-conditioned control and reasoning about complex scenarios.

7.2 BEV Perception

LSS. Lift-Splat-Shoot [Philion and Fidler, 2020] introduced the foundational approach for camera-based BEV perception:

1. Lift: predict depth distribution for each pixel
2. Splat: project features to 3D using predicted depth
3. Shoot: collapse 3D features to BEV plane

BEVFormer. BEVFormer [Li et al., 2022, Yang et al., 2023] uses transformers for BEV generation with spatial cross-attention for multi-camera fusion and temporal self-attention for temporal modeling.

7.3 Datasets

Table 1: Major Autonomous Driving Datasets

Dataset	Scenes	Sensors	Key Features
nuScenes [Caesar et al., 2020]	1000	Camera, Lidar, Radar	3D annotations
Waymo Open [Sun et al., 2020]	1150	Camera, Lidar	High quality
Argoverse 2 [Wilson et al., 2023]	1000	Camera, Lidar	HD maps
KITTI [Geiger et al., 2012]	22	Camera, Lidar	Foundational

8 3D Scene Understanding

8.1 Neural Radiance Fields

NeRF. NeRF [Mildenhall et al., 2020] represents scenes as continuous volumetric functions:

$$F_\theta : (\mathbf{x}, \mathbf{d}) \to (\mathbf{c}, \sigma) \tag{4}$$

where $\mathbf{x}$ is 3D position, $\mathbf{d}$ is viewing direction, $\mathbf{c}$ is color, and σ is density.

Mip-NeRF 360. Mip-NeRF 360 [Barron et al., 2022] extends NeRF to unbounded scenes with integrated positional encoding for anti-aliasing and contraction functions for unbounded geometry.

8.2 3D Gaussian Splatting

3D Gaussian Splatting [Kerbl et al., 2023] represents scenes as sets of 3D Gaussians with:

- Position (mean)
- Covariance matrix (shape)
- Opacity
- Spherical harmonics (view-dependent color)

This enables real-time rendering (100+ FPS) with explicit, editable representations.

8.3 Point Cloud Processing

PointNet. PointNet [Qi et al., 2017a] pioneered deep learning on point clouds with permutation invariance through max pooling and T-Net for spatial transformer.

PointNet++. PointNet++ [Qi et al., 2017b] adds hierarchical structure through set abstraction layers for local feature learning and multi-scale grouping for varying densities.

8.4 Scene Graphs

Scene graphs [Xu et al., 2017, Krishna et al., 2017] represent scenes as graphs with nodes (objects with attributes) and edges (relationships). 3D scene graphs [Armeni et al., 2019, Rosinol et al., 2020] extend this to 3D with hierarchical structure and metric information.

9 Geospatial Foundation Models

9.1 Remote Sensing Models

Prithvi. Prithvi [Jakubik et al., 2024] is a geospatial foundation model trained on Harmonized Landsat Sentinel-2 data. It supports multiple downstream tasks including flood mapping, wildfire detection, and crop classification.

SatMAE. SatMAE [Cong et al., 2022] applies masked autoencoding to satellite imagery, learning representations that transfer across remote sensing tasks.

SatlasPretrain. SatlasPretrain [Bastani et al., 2023] provides large-scale pretraining for satellite imagery with diverse downstream task support.

9.2 Urban Computing

Spatio-temporal graph networks enable modeling of urban dynamics:

- Traffic prediction: DCRNN [Li et al., 2018], STGCN [Yu et al., 2018]
- Demand forecasting: Graph WaveNet [Wu et al., 2019b]
- Urban planning: integration with city simulation

10 Evaluation Framework

10.1 Navigation Benchmarks

R2R. Room-to-Room [Anderson et al., 2018] provides 7,189 paths in Matterport3D environments with average path length of 10m. Metrics include Success Rate (SR), SPL, and Navigation Error.

RxR. Room-across-Room [Ku et al., 2020] extends R2R with multilingual instructions (English, Hindi, Telugu) and longer, more detailed paths.

REVERIE. REVERIE [Qi et al., 2020] requires finding remote objects based on high-level instructions, testing both navigation and object grounding.

10.2 Manipulation Benchmarks

RLBench. RLBench [James et al., 2020] provides 100 unique tasks with multiple variations in CoppeliaSim simulation.

Meta-World. Meta-World [Yu et al., 2020] offers 50 manipulation tasks for multi-task and meta-learning evaluation with Sawyer robot simulation.

BEHAVIOR. BEHAVIOR [Srivastava et al., 2021, Li et al., 2023] provides 1000 everyday activities for comprehensive household robot evaluation.

10.3 Agent Benchmarks

AgentBench. AgentBench [Liu et al., 2023b] evaluates LLM agents across 8 distinct environments including operating system, database, and web browsing.

EmbodiedBench. EmbodiedBench [Yang et al., 2025] provides comprehensive evaluation of embodied MLLMs across multiple spatial reasoning tasks.

11 Implementation Guidance

11.1 Building a RAG-Enhanced Spatial Agent

```
import chromadb

# Step 1: Set up vector database
client = chromadb.Client()
collection = client.create_collection("spatial_knowledge")

# Step 2: Index spatial knowledge
for doc in spatial_documents:
    embedding = embed_model.encode(doc.text)
    collection.add(
        embeddings=[embedding],
        documents=[doc.text],
        metadatas=[{"location": doc.location}]
    )

# Step 3: Implement retrieval-augmented agent
def spatial_agent(query):
    results = collection.query(
        query_texts=[query], n_results=5
    )
    context = "\n".join(results["documents"][0])
    prompt = f"Context: {context}\n\nQuery: {query}"
    response = llm.generate(prompt)
    return response
```

11.2 GNN for Traffic Prediction

```
import torch
import torch_geometric as pyg
from torch_geometric.nn import GCNConv

class STGNN(torch.nn.Module):
    def __init__(self, in_channels, hidden, out_channels):
        super().__init__()
        self.spatial_conv = GCNConv(in_channels, hidden)
        self.temporal_conv = torch.nn.GRU(hidden, hidden)
        self.output = torch.nn.Linear(hidden, out_channels)

    def forward(self, x, edge_index):
        # Spatial aggregation
        h = self.spatial_conv(x, edge_index)
        # Temporal modeling
        h, _ = self.temporal_conv(h)
        return self.output(h)
```

11.3 Deploying VLA Model

```
from transformers import AutoModelForVision2Seq

# Step 1: Load pretrained model
model = AutoModelForVision2Seq.from_pretrained(
    "openvla/openvla-7b"
)

# Step 2: Quantize for deployment
model = torch.quantization.quantize_dynamic(
    model, {torch.nn.Linear}, dtype=torch.qint8
)

# Step 3: Robot control loop
while not done:
    image = camera.capture()
    instruction = "Pick up the red cup"
    action = model.predict(image, instruction)
    robot.execute(action)
    done = check_task_completion()
```

12 Computational Requirements

Table 2: Training Computational Requirements

Model Type	GPUs	Memory	Time
VLA (7B)	8×A100	640GB	1-2 weeks
GNN (Traffic)	1×V100	32GB	1-2 days
NeRF	1×RTX 3090	24GB	12-24 hours
3D Gaussian Splatting	1×RTX 3090	24GB	30-60 min
World Model (Dreamer)	1×V100	32GB	1-3 days

Table 3: Inference Latency Requirements

Application	Latency Requirement	Recommended Hardware
Robot Control	<50ms	Jetson AGX, RTX 4090
Autonomous Driving	<100ms	Multiple GPUs
Traffic Prediction	<1s	Cloud GPU
Geospatial Analysis	Minutes	Cloud cluster

13 Industry Applications and Case Studies

This section provides detailed analysis of how spatial AI is being deployed in industry, with lessons for AtlasPro AI.

13.1 Geospatial Intelligence

Palantir. Palantir [Palantir, 2023, Bailey, 2021] integrates AI with geospatial analysis for defense and commercial applications. Their Foundry platform enables:

- Integration of diverse geospatial data sources
- Real-time analysis of satellite imagery
- Predictive modeling for logistics and operations
- Collaborative analysis across organizations

ESRI. ESRI [ESRI, 2023] provides ArcGIS with integrated GeoAI capabilities:

- Deep learning for feature extraction from imagery
- Spatial analysis with machine learning integration
- Real-time processing of sensor data
- Enterprise-scale geospatial data management

Google Earth Engine. Google [Google, 2023] deploys AI for global-scale mapping:

- Petabyte-scale satellite imagery analysis
- Change detection and monitoring
- Land cover classification
- Environmental monitoring applications

13.2 Location Intelligence

Foursquare. Foursquare [Foursquare, 2023] provides location intelligence through:

- Movement pattern analysis from mobile data
- Point-of-interest enrichment
- Foot traffic prediction
- Location-based audience targeting

Smart City Applications. Urban computing applications [Zheng et al., 2014, Allam and Dhunny, 2020] leverage spatial AI for:

- Traffic signal optimization
- Public transit planning
- Emergency response routing
- Urban development simulation

13.3 Autonomous Vehicles

Waymo. Waymo [Waymo, 2023, 2024] has deployed autonomous vehicles at scale with:

- End-to-end perception-prediction-planning
- High-definition mapping infrastructure
- Simulation-based testing at scale
- Safety-first deployment methodology

End-to-End Approaches. Recent end-to-end systems include:

- UniAD [Hu et al., 2023b]: unified perception, prediction, planning
- VAD [Jiang et al., 2023]: vectorized scene representation
- DriveVLM [Tian et al., 2024]: vision-language driving

13.4 Robotics

Warehouse Automation. Companies like Amazon and Boston Dynamics deploy spatial AI for:

- Autonomous mobile robots for picking and transport
- Collaborative robots working alongside humans
- Dynamic path planning in changing environments
- Multi-robot coordination systems

Agricultural Robotics. Precision agriculture applications include:

- Autonomous tractors and harvesters
- Crop monitoring drones
- Precision spraying systems
- Yield prediction from satellite imagery

14 Detailed Benchmark Analysis

This section provides comprehensive analysis of benchmarks for internal evaluation planning.

14.1 Navigation Benchmark Details

Room-to-Room (R2R). R2R [Anderson et al., 2018] is the foundational VLN benchmark:

- 7,189 paths in Matterport3D environments
- Average path length: 10m, 6 viewpoints
- Natural language instructions from human annotators
- Metrics: Success Rate (SR), SPL, Navigation Error

Table 4: R2R Val-Unseen Performance (Selected Methods)

Method	SR (%)	SPL (%)
Human Performance	86	76
Recurrent VLN-BERT	63	57
HOPT	64	57
DUST	72	62

Room-across-Room (RxR). RxR [Ku et al., 2020] extends R2R with:

- Multilingual instructions (English, Hindi, Telugu)
- Longer paths than R2R (average 15m)
- More detailed, step-by-step instructions
- Pose traces for fine-grained evaluation

REVERIE. REVERIE [Qi et al., 2020] tests remote object localization:

- High-level instructions referencing distant objects
- Requires both navigation and object grounding
- Tests compositional understanding

14.2 Manipulation Benchmark Details

RLBench. RLBench [James et al., 2020] provides comprehensive manipulation evaluation:

- 100 unique tasks with multiple variations
- CoppeliaSim physics simulation
- Support for multiple robot embodiments
- Standardized observation and action spaces

Meta-World. Meta-World [Yu et al., 2020] focuses on multi-task learning:

- 50 manipulation tasks
- Sawyer robot simulation
- Multi-task and meta-learning evaluation protocols
- Standardized success criteria

BEHAVIOR. BEHAVIOR [Srivastava et al., 2021, Li et al., 2023] tests household activities:

- 1000 everyday activities
- Realistic household environments
- Long-horizon task evaluation
- Human-like activity definitions

14.3 Agent Benchmark Details

AgentBench. AgentBench [Liu et al., 2023b] evaluates LLM agents:

- 8 distinct environments
- Operating system, database, web browsing tasks
- Comprehensive LLM agent evaluation
- Standardized evaluation protocols

EmbodiedBench. EmbodiedBench [Yang et al., 2025] tests embodied MLLMs:

- Multiple spatial reasoning tasks
- Manipulation and navigation evaluation
- Multimodal understanding assessment

14.4 Geospatial Benchmark Details

BigEarthNet. BigEarthNet [Sumbul et al., 2019] provides:

- 590,326 Sentinel-2 image patches
- Multi-label land cover classification
- 43 land cover classes
- European coverage

fMoW. Functional Map of the World [Christie et al., 2018]:

- 1 million satellite images
- 62 functional categories
- Temporal sequences for change detection
- Global coverage

xBD. xBD [Gupta et al., 2019] for building damage assessment:

- Pre and post-disaster imagery pairs
- Building-level damage annotations
- Multiple disaster types
- Humanitarian response applications

15 Reference Architectures

This section provides reference architectures for common spatial AI system patterns.

15.1 Spatial Agent Architecture

A reference architecture for spatial AI agents includes:

Perception Module.

- Multi-modal sensor fusion (camera, lidar, radar)
- Object detection and tracking
- Semantic segmentation
- Depth estimation

Memory Module.

- Short-term context buffer
- Long-term knowledge retrieval (RAG)
- Spatial cognitive map
- Episodic memory for experience replay

Reasoning Module.

- LLM-based planning and reasoning
- GNN-based spatial relationship reasoning
- World model for prediction
- Uncertainty estimation

Action Module.

- High-level action planning
- Low-level motion control
- Safety monitoring and intervention
- Feedback integration

15.2 GNN-LLM Integration Architecture

Three patterns for GNN-LLM integration:

Pattern 1: GNN as Encoder.

- GNN encodes graph structure into embeddings
- Embeddings projected to LLM input space
- LLM generates text conditioned on graph
- Example: GraphGPT [Tang et al., 2024]

Pattern 2: LLM as Reasoner.

- GNN retrieves relevant subgraphs
- Subgraphs converted to text descriptions
- LLM reasons over retrieved information
- Example: GNN-RAG [Wang et al., 2024a]

Pattern 3: Joint Training.

- GNN and LLM trained end-to-end
- Shared representation space
- Joint optimization objective
- Highest performance but most complex

15.3 World Model Architecture

Reference architecture for world model-based planning:

Representation Model.

- Encodes observations to latent states
- Handles multi-modal inputs
- Maintains temporal consistency

Transition Model.

- Predicts next latent state given action
- Captures environment dynamics
- Enables imagination-based planning

Reward Model.

- Predicts reward from latent state
- Enables value estimation
- Supports policy optimization

Policy.

- Actor-critic architecture
- Trained on imagined trajectories
- Deployed for real-world control

16 Data Pipeline Design

This section covers data pipeline design for spatial AI systems.

16.1 Data Collection

Robot Demonstration Data.

- Teleoperation for human demonstrations
- Scripted policies for automated collection
- Simulation data with domain randomization
- Quality filtering and validation

Geospatial Data.

- Satellite imagery from commercial providers
- Open data sources (Sentinel, Landsat)
- Ground truth annotations
- Temporal alignment and registration

16.2 Data Processing

Preprocessing.

- Sensor calibration and alignment
- Noise filtering and outlier removal
- Coordinate system normalization
- Temporal synchronization

Augmentation.

- Geometric transformations
- Color and lighting variations
- Synthetic data generation
- Domain randomization

16.3 Data Storage and Management

Storage Architecture.

- Object storage for raw data (S3, GCS)
- Database for metadata and annotations
- Version control for datasets
- Access control and audit logging

Data Versioning.

- Track dataset versions over time
- Reproducibility for experiments
- Rollback capability
- Lineage tracking

17 Deployment Considerations

This section covers deployment considerations for production spatial AI systems.

17.1 Edge Deployment

Hardware Selection.

- NVIDIA Jetson for robotics (AGX Orin: 275 TOPS)
- Intel Neural Compute Stick for low-power
- Custom ASICs for high-volume deployment
- FPGA for flexible acceleration

Model Optimization.

- Quantization (INT8, INT4)
- Pruning and distillation
- TensorRT optimization
- ONNX export for portability

17.2 Cloud Deployment

Infrastructure.

- GPU clusters for training (A100, H100)
- Inference servers with load balancing
- Auto-scaling based on demand
- Multi-region deployment for latency

MLOps.

- CI/CD for model deployment
- Model registry and versioning
- A/B testing infrastructure
- Monitoring and alerting

17.3 Hybrid Deployment

Edge-Cloud Coordination.

- Local inference for latency-critical tasks
- Cloud offloading for complex reasoning
- Data synchronization strategies
- Fallback mechanisms for connectivity loss

18 Safety Engineering

18.1 Principles

Safety engineering for spatial AI systems must address:

- Physical safety: preventing harm to humans and property
- Operational safety: ensuring reliable system behavior
- Security: protecting against adversarial attacks
- Privacy: handling sensitive location and sensor data

18.2 Implementation Practices

Uncertainty Quantification. Implement calibrated uncertainty estimates for all predictions. Use ensemble methods or Bayesian approaches for robust uncertainty.

Fallback Mechanisms. Design graceful degradation with safe fallback behaviors. Implement human oversight mechanisms for high-stakes decisions.

Testing and Validation. Comprehensive testing in simulation before real-world deployment. Continuous monitoring and anomaly detection in production.

19 AtlasPro AI Use Case Recommendations

Based on our comprehensive analysis, we identify the following high-priority use cases for AtlasPro AI.

19.1 Geospatial Intelligence Platform

Opportunity. Build a next-generation geospatial intelligence platform combining satellite imagery analysis with LLM-based reasoning.

Technical Approach.

- Deploy Prithvi [Jakubik et al., 2024] as foundation model for satellite imagery
- Integrate with LLM for natural language querying
- Build GNN-based spatial relationship reasoning
- Implement RAG for geospatial knowledge retrieval

Key Challenges.

- Scale to petabyte-level imagery data
- Real-time processing requirements
- Multi-modal data fusion
- Accuracy validation and uncertainty quantification

19.2 Autonomous Robot Navigation

Opportunity. Develop autonomous navigation systems for indoor and outdoor environments.

Technical Approach.

- VLA model for vision-language-action integration
- Cognitive mapping for spatial memory
- LLM-based high-level planning
- World model for safe trajectory planning

Key Challenges.

- Sim-to-real transfer
- Dynamic obstacle avoidance
- Long-horizon task completion
- Safety in human environments

19.3 Urban Traffic Prediction

Opportunity. Build city-scale traffic prediction and optimization systems.

Technical Approach.

- Spatio-temporal GNN for traffic flow modeling
- Integration with real-time sensor data
- Multi-step forecasting with uncertainty
- Optimization for signal control

Key Challenges.

- Handling missing and noisy data
- Capturing long-range dependencies
- Adapting to special events
- Real-time inference requirements

20 Technology Roadmap

This section outlines a recommended technology roadmap for AtlasPro AI.

20.1 Phase 1: Foundation (Q1-Q2 2026)

Infrastructure.

- Set up GPU cluster for training (8x A100 minimum)
- Deploy simulation environments (Habitat, Isaac Sim)
- Establish data pipeline infrastructure
- Implement MLOps practices

Capabilities.

- Deploy baseline VLA model (OpenVLA)
- Implement basic GNN for spatial reasoning
- Build RAG system for spatial knowledge
- Create internal benchmarking framework

20.2 Phase 2: Integration (Q3-Q4 2026)

Infrastructure.

- Scale to multi-node training
- Deploy edge inference infrastructure
- Implement continuous evaluation pipeline
- Build safety monitoring systems

Capabilities.

- Develop GNN-LLM integration system
- Train custom VLA on domain data
- Implement world model for planning
- Deploy first pilot applications

20.3 Phase 3: Production (2027)

Infrastructure.

- Production-grade deployment infrastructure
- Real-time monitoring and alerting
- Automated retraining pipelines
- Compliance and audit systems

Capabilities.

- Full spatial AI platform deployment
- Multi-agent coordination systems
- Advanced safety mechanisms
- Customer-facing applications

21 Team Structure Recommendations

Recommended team structure for spatial AI development at AtlasPro AI.

21.1 Core Teams

Foundation Models Team.

- VLA model development and training
- Geospatial foundation model adaptation
- World model research
- 5-8 ML researchers/engineers

Spatial Reasoning Team.

- GNN development and integration
- Scene graph construction
- Spatial memory systems
- 3-5 ML researchers/engineers

Robotics Team.

- Robot hardware integration
- Simulation development
- Real-world deployment
- 4-6 robotics engineers

Infrastructure Team.

- Training infrastructure
- Data pipeline development
- MLOps and deployment
- 3-5 infrastructure engineers

21.2 Supporting Functions

Safety and Reliability.

- Safety engineering practices
- Testing and validation
- Compliance and certification
- 2-3 safety engineers

Research.

- Literature review and analysis
- Novel algorithm development
- Publication and patents
- 2-4 research scientists

22 Risk Assessment

Key risks and mitigation strategies for spatial AI development.

22.1 Technical Risks

Model Performance.

- Risk: Models may not generalize to target domains
- Mitigation: Extensive domain-specific data collection and fine-tuning
- Mitigation: Continuous benchmarking and evaluation

Sim-to-Real Gap.

- Risk: Simulation-trained models may fail in real world
- Mitigation: Domain randomization and real-world fine-tuning
- Mitigation: Gradual deployment with extensive testing

Computational Requirements.

- Risk: Training and inference costs may be prohibitive
- Mitigation: Model efficiency research and optimization
- Mitigation: Cloud cost optimization strategies

22.2 Operational Risks

Safety Incidents.

- Risk: Autonomous systems may cause harm
- Mitigation: Safety-first design principles
- Mitigation: Extensive testing and human oversight

Data Quality.

- Risk: Poor data quality may degrade model performance
- Mitigation: Robust data validation pipelines
- Mitigation: Continuous data quality monitoring

22.3 Strategic Risks

Competition.

- Risk: Competitors may advance faster
- Mitigation: Focus on differentiated capabilities
- Mitigation: Strategic partnerships and acquisitions

Regulatory Changes.

- Risk: New regulations may constrain deployment
- Mitigation: Proactive engagement with regulators
- Mitigation: Flexible architecture for compliance

23 Open Challenges

23.1 Robust Spatial Representation

Developing representations that generalize across scenes, viewpoints, and conditions remains challenging [Mildenhall et al., 2020, Kerbl et al., 2023]. Foundation models for 3D understanding [Hong et al., 2023b] represent promising directions.

23.2 Long-Horizon Planning

Creating agents that plan over extended horizons and decompose complex spatial tasks is essential [Song et al., 2023, Valmeekam et al., 2023]. Integration of neural and symbolic planning approaches shows promise.

23.3 Sim-to-Real Transfer

Bridging simulation and reality remains challenging [Zhao et al., 2020, Tobin et al., 2017]. Domain randomization and real-world fine-tuning are active research areas.

23.4 Multi-Agent Coordination

Scaling to multi-agent systems for complex spatial tasks requires advances in coordination and communication [Zhang et al., 2021, Wu et al., 2023b, Hong et al., 2023a].

23.5 Scalable Data Collection

Efficient data collection methods [Collaboration, 2023, Walke et al., 2023] and synthetic data generation are critical for scaling spatial AI systems.

24 Conclusion

This technical report has provided a comprehensive, engineering-focused analysis of autonomous spatial intelligence systems. We have synthesized over 500 papers to provide AtlasPro AI engineering teams with actionable guidance for building next-generation spatially-aware autonomous systems.

Key takeaways for engineering teams:

1. Memory architecture is critical: invest in hierarchical memory systems combining short-term, long-term, and spatial memory.
2. GNN-LLM integration is a powerful paradigm for spatial reasoning that combines structural and semantic understanding.
3. World models enable safe planning through imagination, essential for safety-critical applications.
4. Open-source VLA models provide strong baselines for robotics applications.
5. Safety engineering must be built in from the start, not added as an afterthought.

Recommended next steps:

1. Establish internal benchmarking infrastructure for continuous evaluation.
2. Build prototype GNN-LLM integration system for spatial reasoning.
3. Deploy simulation environment (Habitat, Isaac Sim) for safe development.
4. Implement safety engineering practices across all projects.
5. Begin pilot projects in identified use cases.

This document will be updated quarterly as the field advances. Questions and feedback should be directed to the Research Division.

References

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gober, Karol Gopalakrishnan, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: A visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736, 2022.

Zaheer Allam and Zaynah A Dhunny. On big data, artificial intelligence and smart cities. *Cities*, 89:80–91, 2020.

Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sunderhauf, Ian Reid, Stephen Gould, and Anton van den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3674–3683, 2018.

Iro Armeni, Zhi-Yang He, JunYoung Gwak, Amir R Zamir, Martin Fischer, Jitendra Malik, and Silvio Savarese. 3d scene graph: A structure for unified semantics, 3d space, and camera. *arXiv preprint arXiv:1910.02527*, 2019.

Lei Bai et al. Adaptive graph convolutional recurrent network for traffic forecasting. *Advances in Neural Information Processing Systems*, 2020.

Jonathan Bailey. Palantir technologies: Building the operating system for the modern enterprise. Industry Report, 2021.

Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *CVPR*, 2022.

Favyen Bastani et al. Satlaspretrain: A large-scale dataset for remote sensing image understanding. In *arXiv preprint arXiv:2211.15660*, 2023.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Lukas Gianinazzi, Joanna Gajber, Tomasz Lehmann, Michal Podstawski, Hubert Niewiadomski, Piotr Nyczyk, et al. Graph of thoughts: Solving elaborate problems with large language models. *arXiv preprint arXiv:2308.09687*, 2023.

Anthony Brohan, Noah Brown, Justice Carbajal, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901, 2020.

Jake Bruce, Michael Dennis, Ashley Edwards, et al. Genie: Generative interactive environments. In *ICML*, 2024.

Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, et al. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, 2020.

Devendra Singh Chaplot, Dhiraj Gandhi, Saurabh Gupta, Abhinav Gupta, and Ruslan Salakhutdinov. Learning to explore using active neural slam. In *ICLR*, 2020.

Gordon Christie et al. Functional map of the world. *CVPR*, 2018.

Open X-Embodiment Collaboration. Open x-embodiment, 2023.

Yezhen Cong, Samar Khanna, Chenlin Meng, Patrick Liu, Erik Rozi, Yutong He, Marshall Burke, David Lobell, and Stefano Ermon. Satmae: Pre-training transformers for temporal and multi-spectral satellite imagery. *Advances in Neural Information Processing Systems*, 35:197–211, 2022.

Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023.

ESRI. Esri arcgis: The mapping and analytics platform. `https://www.esri.com`, 2023.

Linxi Fan, Guanzhi Wang, Yunfan Jiang, Ajay Mandlekar, Yuncong Yang, Haoyi Zhu, Andrew Tang, De-An Huang, Yuke Zhu, and Anima Anandkumar. Minedojo: Building open-ended embodied agents with internet-scale knowledge. *Advances in Neural Information Processing Systems*, 35:18343–18362, 2022.

Foursquare. Foursquare location intelligence. `https://foursquare.com`, 2023.

Luyu Gao et al. Pal: Program-aided language models. *International Conference on Machine Learning*, 2023.

Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012.

Google. Google maps platform. `https://cloud.google.com/maps-platform`, 2023.

Lin Guan, Karthik Valmeekam, Sarath Sreedharan, and Subbarao Kambhampati. Leveraging pre-trained large language models to construct and utilize world models for model-based task planning. *Advances in Neural Information Processing Systems*, 36, 2023.

Ritwik Gupta et al. xbd: A dataset for assessing building damage. *arXiv preprint arXiv:1911.09296*, 2019.

Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. In *ICLR*, 2021.

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

Danijar Hafner et al. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019.

William L Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in Neural Information Processing Systems*, 30, 2017.

Shibo Hao, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu. Reasoning with language model is planning with world model. *arXiv preprint arXiv:2305.14992*, 2023.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Ceyao Zhang, Jinlin Wang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, et al. Metagpt: Meta programming for a multi-agent collaborative framework. *arXiv preprint arXiv:2308.00352*, 2023a.

Yining Hong et al. 3d-llm: Injecting the 3d world into large language models. *Advances in Neural Information Processing Systems*, 2023b.

Anthony Hu et al. Gaia-1: A generative world model for autonomous driving. *arXiv preprint arXiv:2309.17080*, 2023a.

Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, et al. Planning-oriented autonomous driving. In *CVPR*, 2023b.

Chenguang Huang, Oier Mees, Andy Zeng, and Wolfram Burgard. Visual language maps for robot navigation. *arXiv preprint arXiv:2210.05714*, 2023.

Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, et al. Inner monologue: Embodied reasoning through planning with language models. *arXiv preprint arXiv:2207.05608*, 2022.

Johannes Jakubik, Sujit Roy, C E Phillips, Paolo Fraccaro, Denys Godwin, Bianca Zadrozny, Daniela Szwarcman, Carlos Gomes, Gabby Musber, Daiki Oliveira, et al. Prithvi: A foundation model for earth observation. *arXiv preprint arXiv:2310.18660*, 2024.

Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J Davison. Rlbench: The robot learning benchmark. *IEEE Robotics and Automation Letters*, 2020.

Bo Jiang et al. Vad: Vectorized scene representation for efficient autonomous driving. *IEEE International Conference on Computer Vision*, 2023.

Guangyin Jin, Yuxuan Liang, Yuchen Fang, Zezhi Huang, Junbo Zhang, and Yu Zheng. Spatio-temporal graph neural networks for urban computing: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 2023.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkuhler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):1–14, 2023.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2017.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in Neural Information Processing Systems*, 35:22199–22213, 2022.

Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. Ai2-thor: An interactive 3d environment for visual ai. *arXiv preprint arXiv:1712.05474*, 2017.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. In *International Journal of Computer Vision*, volume 123, pages 32–73, 2017.

Alexander Ku, Peter Anderson, Roma Patel, Eugene Ie, and Jason Baldridge. Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. In *EMNLP*, 2020.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Kuttler, Mike Lewis, Wen-tau Yih, Tim Rocktaschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474, 2020.

Chengshu Li, Fei Xia, Roberto Martín-Martín, Michael Lingelbach, Sanjana Srivastava, Bokui Shen, Kent Vainio, Cem Gokmen, Gokul Dharan, Tanish Jain, et al. igibson 2.0: Object-centric simulation for robot learning of everyday household tasks. In *Conference on Robot Learning*, pages 455–465, 2021.

Chengshu Li, Ruohan Zhang, Josiah Wong, et al. Behavior-1k: A benchmark for embodied ai with 1,000 everyday activities and realistic simulation. In *CoRL*, 2023.

Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *International Conference on Learning Representations*, 2018.

Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In *European Conference on Computer Vision*, pages 1–18, 2022.

Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. *arXiv preprint arXiv:2209.07753*, 2023.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*, 2023a.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, et al. Agentbench: Evaluating llms as agents. *arXiv preprint arXiv:2308.03688*, 2023b.

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2020.

OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Maxime Oquab et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G Patil, Ion Stoica, and Joseph E Gonzalez. Memgpt: Towards llms as operating systems. *arXiv preprint arXiv:2310.08560*, 2023.

Palantir. Palantir technologies. `https://www.palantir.com`, 2023.

Shishir G Patil, Tianjun Zhang, Xin Wang, and Joseph E Gonzalez. Gorilla: Large language model connected with massive apis. *arXiv preprint arXiv:2305.15334*, 2023.

Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *European Conference on Computer Vision*, pages 194–210, 2020.

Xavier Puig, Eric Undersander, Andrew Szot, Mikael Dallaire Cote, Dhruv Batra, and Roozbeh Mottaghi. Habitat 3.0: A co-habitat for humans, avatars and robots. In *International Conference on Learning Representations*, 2024.

Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, 2017a.

Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*, volume 30, 2017b.

Yuankai Qi, Qi Wu, Peter Anderson, Xin Wang, William Yang Wang, Chunhua Shen, and Anton van den Hengel. Reverie: Remote embodied visual referring expression in real indoor environments. In *CVPR*, 2020.

Yujia Qin, Shihao Liang, Yining Ye, et al. Toolllm: Facilitating large language models to master 16000+ real-world apis. In *ICLR*, 2024.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021.

Antoni Rosinol, Arjun Gupta, Marcus Abate, Jingnan Shi, and Luca Carlone. 3d dynamic scene graphs: Actionable spatial perception with places, objects, and humans. In *Robotics: Science and Systems (RSS)*, 2020.

Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9339–9347, 2019.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessi, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *arXiv preprint arXiv:2302.04761*, 2023.

Bokui Shen, Fei Xia, Chengshu Li, Roberto Martin, Linxi Fan, Guanzhi Wang, Shyamal Buch, et al. igibson 2.0: Object-centric simulation for robot learning of everyday household tasks. *arXiv preprint arXiv:2108.03272*, 2021.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *arXiv preprint arXiv:2303.11366*, 2023.

Chan Hee Song, Jiaman Wu, Clayton Washington, Brian M Sadler, Wei-Lun Chao, and Yu Su. Llm-planner: Few-shot grounded planning for embodied agents with large language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2998–3009, 2023.

Sanjana Srivastava, Chengshu Li, Michael Lingelbach, Roberto Martin, Fei Xia, Kent Vainio, Zheng Lian, Cem Gokmen, Shyamal Buch, Karen Liu, et al. Behavior: Benchmark for everyday household activities in virtual, interactive, and ecological environments. *arXiv preprint arXiv:2108.03332*, 2021.

Gencer Sumbul, Marcela Charfuelan, Begum Demir, and Volker Markl. Bigearthnet: A large-scale benchmark archive for remote sensing image understanding. *arXiv preprint arXiv:1902.06148*, 2019.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2446–2454, 2020.

Andrew Szot, Alexander Clegg, Eric Undersander, et al. Habitat 2.0: Training home assistants to rearrange their habitat. In *NeurIPS*, 2021.

Jiabin Tang, Yuhao Yang, Wei Wei, Lei Shi, Lixin Su, Suqi Cheng, Dawei Yin, and Chao Huang. Graphgpt: Graph instruction tuning for large language models. *arXiv preprint arXiv:2310.13023*, 2024.

Octo Model Team et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

Xiaoyu Tian et al. Drivevlm: The convergence of autonomous driving and large vision-language models. *arXiv preprint arXiv:2402.12289*, 2024.

Josh Tobin, Rocky Fong, Alex Ray, John Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 23–30. IEEE, 2017.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothee Lacroix, Baptiste Roziere, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Karthik Valmeekam, Matthew Marquez, Sarath Sreedharan, and Subbarao Kambhampati. On the planning abilities of large language models–a critical investigation. *Advances in Neural Information Processing Systems*, 36, 2023.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2018.

Homer Walke, Kevin Black, Tony Z Zhao, et al. Bridgedata v2: A dataset for robot learning at scale. In *CoRL*, 2023.

Costas Wang et al. Gnn-rag: Graph neural retrieval for large language model reasoning. *arXiv preprint arXiv:2405.20139*, 2024a.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 2024b.

Waymo. Waymo: The world's most experienced driver. `https://waymo.com`, 2023.

Waymo. Introducing Waymoś Research on an End-to-End Multimodal Model for Autonomous Driving. Waymo Blog, October 2024. URL `https://waymo.com/blog/2024/10/introducing-emma`.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837, 2022.

Benjamin Wilson et al. Argoverse 2: Next generation datasets for self-driving perception and forecasting. *Advances in Neural Information Processing Systems*, 2023.

Philipp Wu et al. Daydreamer: World models for physical robot learning. *arXiv preprint arXiv:2206.14176*, 2023a.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, et al. Autogen: Enabling next-gen llm applications via multi-agent conversation. *arXiv preprint arXiv:2308.08155*, 2023b.

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1): 4–24, 2019a.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. Graph wavenet for deep spatial-temporal graph modeling. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pages 1907–1913, 2019b.

Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, et al. The rise and potential of large language model based agents: A survey. *arXiv preprint arXiv:2309.07864*, 2023.

Danfei Xu, Yuke Zhu, Christopher B Choy, and Li Fei-Fei. Scene graph generation by iterative message passing. In *CVPR*, 2017.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2019.

Chenyu Yang et al. Bevformer v2: Adapting modern image backbones to bird's-eye-view recognition via perspective supervision. *IEEE Conference on Computer Vision and Pattern Recognition*, 2023.

Rui Yang, Hanyang Lin, Junyu Zhu, and Jingyi Huang. Embodiedbench: Comprehensive benchmarking multi-modal large language models for vision-driven embodied agents. *arXiv preprint arXiv:2502.09560*, 2025.

Xiaofeng Yang et al. Worlddreamer: Towards general world models for video generation via predicting masked tokens. *arXiv preprint arXiv:2401.09985*, 2024.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *arXiv preprint arXiv:2305.10601*, 2023a.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2023b.

Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, pages 3634–3640, 2018.

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, et al. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *CoRL*, 2020.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *International Conference on Computer Vision (ICCV)*, 2023.

Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. Multi-agent reinforcement learning: A selective overview of theories and algorithms. *Handbook of Reinforcement Learning and Control*, pages 321–384, 2021.

Wenyu Zhao, Jorge Pena Queralta, and Tomi Westerlund. Sim-to-real transfer in deep reinforcement learning for robotics: a survey. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 737–744. IEEE, 2020.

Yu Zheng, Licia Capra, Ouri Wolfson, and Hai Yang. Urban computing: Concepts, methodologies, and applications. *ACM Transactions on Intelligent Systems and Technology*, 5(3):1–55, 2014.