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

Agentic AI Methods, System Architectures, Implementation Patterns,
and Deployment Strategies for Production Systems

Gloria Felicia
AtlasPro AI
gloria.felicia@atlaspro.ai

Nolan Bryant
AtlasPro AI
nolan.bryant@atlaspro.ai

Handi Putra
AtlasPro AI
handi.putra@atlaspro.ai

Ayaan Gazali
AtlasPro AI
ayaan.gazali@atlaspro.ai

Eliel Lobo
AtlasPro AI
eliel.lobo@atlaspro.ai

Esteban Rojas
AtlasPro AI
esteban.rojas@atlaspro.ai

Internal Technical Report – AtlasPro AI Research Division

Abstract

This comprehensive technical report provides an engineering-focused deep-dive into 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 $XX billion 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., 2023b, 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., 2023d, 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 including red teaming and constitutional AI [Bai et al., 2022].

2 Foundational Concepts and Taxonomy

2.1 Defining Agentic AI

We adopt the definition from Wang et al. [2024b]: an AI agent is an autonomous entity that perceives its environment, makes decisions, and takes actions to achieve specific goals. This definition encompasses three core capabilities:

Perception. The ability to observe and interpret the environment through sensors, cameras, or data feeds. For spatial agents, this includes 3D perception [Qi et al., 2017a, Mildenhall et al., 2020], semantic understanding [Krishna et al., 2017], and multi-modal fusion.

Reasoning. The ability to process information, draw inferences, and make decisions. Modern agents leverage LLMs for reasoning [Wei et al., 2022, Yao et al., 2023a], with chain-of-thought prompting enabling step-by-step problem solving.

Action. The ability to execute decisions in the environment. This ranges from API calls [Schick et al., 2023, Patil et al., 2023] to physical robot control [Brohan et al., 2023, Ahn et al., 2022].

2.2 Defining Spatial Intelligence

We define Spatial Intelligence as the ability to perceive, reason about, and interact with 3D physical environments. This encompasses:

Spatial Perception. Understanding 3D structure, object geometry, and scene layout [Dai et al., 2017, Chang et al., 2017, Armeni et al., 2016].

Spatial Reasoning. Inferring relationships between objects, predicting physical dynamics, and understanding affordances [Chen et al., 2024, Johnson et al., 2017, Hudson and Manning, 2019].

Spatial Action. Navigating environments [Anderson et al., 2018, Batra et al., 2020], manipulating objects [Zeng et al., 2021, Shridhar et al., 2022], and coordinating multi-agent systems [Zhang et al., 2021].

2.3 Unified Taxonomy

We propose a two-dimensional taxonomy mapping agentic components to spatial domains:

Table 1: Unified Taxonomy: Agentic Components × Spatial Domains

	Navigation	**Scene Understanding**	**Manipulation**	**Geospatial**
Memory	Cognitive Maps	Scene Graphs	Object Memory	Spatial Databases
Planning	Path Planning	Semantic Planning	Task Planning	Route Optimization
Tool Use	Locomotion APIs	Perception APIs	Robot Control	GIS Tools

3 Core Agentic Components: Engineering Deep-Dive

3.1 Memory Systems Architecture

Memory is the foundation of intelligent behavior. For spatial agents, we identify three memory tiers:

3.1.1 Short-Term Memory: Context Management

Short-term memory operates within the LLM's context window. Engineering considerations include:

Context Window Management. Modern LLMs have context windows ranging from 8K to 128K+ tokens [OpenAI, 2023, Anthropic, 2024]. For spatial tasks, we must efficiently encode:

- Current observations (images, sensor data)
- Recent action history

- Task instructions and goals
- Relevant retrieved information

Prompt Engineering. The structure of the prompt significantly impacts agent performance. Best practices include:

- Clear separation of system instructions, context, and queries
- Structured output formats (JSON, XML) for reliable parsing
- Few-shot examples for complex tasks
- Chain-of-thought prompting for reasoning tasks [Wei et al., 2022, Kojima et al., 2022]

State Compression. For long-horizon tasks, we must compress historical state to fit within context limits. Techniques include:

- Summarization of past events
- Selective retention of important information
- Hierarchical state representations

3.1.2 Long-Term Memory: Retrieval-Augmented Generation

Long-term memory extends agent knowledge beyond the context window through external retrieval [Lewis et al., 2020, Guu et al., 2020].

Vector Database Selection. Key options include:

- **Pinecone:** Managed service, easy scaling, good for production
- **Weaviate:** Open-source, supports hybrid search
- **Chroma:** Lightweight, good for prototyping
- **Milvus:** High-performance, supports billion-scale vectors

Embedding Model Selection. The choice of embedding model affects retrieval quality:

- OpenAI text-embedding-3-large: Strong general performance
- Sentence-BERT variants: Good for semantic similarity
- Domain-specific embeddings: Better for specialized tasks

Chunking Strategy. How we split documents affects retrieval:

- Fixed-size chunks (e.g., 512 tokens): Simple but may split semantic units
- Semantic chunking: Preserves meaning but more complex
- Hierarchical chunking: Enables multi-granularity retrieval

Retrieval Algorithms. Beyond simple similarity search:

- Maximal Marginal Relevance (MMR): Balances relevance and diversity
- Hybrid search: Combines dense and sparse retrieval
- Re-ranking: Uses cross-encoders for improved precision

3.1.3 Spatial Memory: Cognitive Maps

For embodied agents, spatial memory is critical. Key approaches include:

Metric Maps. Neural SLAM [Gupta et al., 2019b, Chaplot et al., 2020] builds metric representations of environments. Implementation requires:

- Depth estimation from RGB images
- Pose estimation and tracking
- Map fusion and update

Semantic Maps. VLMaps [Huang et al., 2023] adds language-grounded understanding:

- CLIP feature extraction for each location
- 3D feature volume construction
- Natural language querying of spatial locations

Episodic Memory. Generative Agents [Park et al., 2023] demonstrated memory streams for social agents. For spatial agents, we can adapt this to store:

- Past navigation experiences
- Object interaction history
- Task completion records

3.2 Planning Systems Architecture

Planning enables agents to decompose goals into executable actions.

3.2.1 Chain-of-Thought Planning

CoT prompting [Wei et al., 2022] elicits step-by-step reasoning:

Implementation Pattern.

```
System: You are a spatial planning agent. Think step by step.
User: Navigate to the kitchen and pick up the red cup.
Assistant: Let me break this down:
1. First, I need to locate the kitchen...
2. Then, I need to navigate there...
3. Once in the kitchen, I need to find the red cup...
4. Finally, I need to pick up the cup...
```

Zero-Shot CoT. Simply adding ”Let’s think step by step” improves reasoning [Kojima et al., 2022].

Self-Consistency. Sampling multiple reasoning paths and aggregating improves robustness [Wang et al., 2022].

3.2.2 Tree-Based Planning

Tree of Thoughts [Yao et al., 2023a] explores multiple solution paths:

Algorithm Structure.

1. Generate multiple candidate next steps
2. Evaluate each candidate
3. Select promising candidates for expansion

4. Backtrack if necessary

Engineering Considerations.

- State management for each tree node
- Evaluation function design
- Search strategy (BFS, DFS, beam search)
- Computational cost management

3.2.3 Hierarchical Planning

For complex spatial tasks, hierarchical planning is essential [Song et al., 2023, Huang et al., 2022]:

High-Level Planning. LLM generates abstract task decomposition:

- ”Go to kitchen” → ”Find cup” → ”Pick up cup”

Low-Level Planning. Specialized planners handle execution:

- Navigation: A* or RRT for path planning
- Manipulation: Motion planning with MoveIt or similar

Grounding. SayCan [Ahn et al., 2022] grounds high-level plans in robot affordances by combining LLM probabilities with learned value functions.

3.2.4 Classical Planning Integration

Recent work explores combining LLMs with classical planners [Valmeekam et al., 2023, Guan et al., 2023, Liu et al., 2023a]:

LLM as Heuristic. Use LLM to guide search in classical planners.

LLM as Translator. Convert natural language to PDDL for classical planning.

Hybrid Approaches. Combine neural and symbolic planning for robustness.

3.3 Tool Use and Action Systems

Tool use extends agent capabilities through external interfaces.

3.3.1 API Integration

Tool Definition. Tools should be defined with clear schemas:

```
{
  "name": "navigate_to",
  "description": "Navigate the robot to a specified location",
  "parameters": {
    "location": {"type": "string", "description": "Target location name"},
    "speed": {"type": "number", "description": "Movement speed (0-1)"}
  }
}
```

Tool Selection. Models like Toolformer [Schick et al., 2023] and Gorilla [Patil et al., 2023] learn when and how to use tools.

Error Handling. Robust error handling is critical:

- Retry logic with exponential backoff
- Fallback strategies for tool failures
- Clear error messages for debugging

3.3.2 Code Generation

Code as Policies [Liang et al., 2023] generates executable robot code:

Advantages.

- Flexible and expressive
- Enables complex control flow
- Supports variables and state

Safety Considerations.

- Sandboxed execution environment
- Code review before execution
- Resource limits (CPU, memory, time)

3.3.3 ReAct Architecture

ReAct [Yao et al., 2023b] interleaves reasoning and action:

Loop Structure.

1. **Thought:** Agent reasons about current state
2. **Action:** Agent selects and executes action
3. **Observation:** Environment provides feedback
4. **Repeat:** Until goal achieved or failure

Implementation.

```
while not done:
    thought = llm.generate(prompt + history)
    action = parse_action(thought)
    observation = execute_action(action)
    history.append((thought, action, observation))
    done = check_completion(observation)
```

4 Vision-Language-Action Models: Implementation Guide

4.1 Architecture Overview

VLA models map visual observations and language instructions directly to robot actions [Brohan et al., 2022, 2023, Team et al., 2024, Kim et al., 2024].

Components.

- **Vision Encoder:** Processes camera images (ViT, ResNet)
- **Language Encoder:** Processes text instructions (BERT, T5)
- **Fusion Module:** Combines vision and language features
- **Action Head:** Predicts robot actions

4.2 RT-1 and RT-2

RT-1 [Brohan et al., 2022] demonstrated large-scale robot learning:

- Trained on 130K robot demonstrations
- Transformer architecture with tokenized actions
- Strong generalization to new objects and instructions

RT-2 [Brohan et al., 2023] co-trained on web data:

- 55B parameter PaLI-X backbone
- Actions represented as text tokens
- Emergent capabilities (reasoning about novel objects)

4.3 Open-Source VLA Models

Octo [Team et al., 2024]:

- Trained on Open X-Embodiment dataset [Collaboration, 2023]
- Supports multiple robot embodiments
- Apache 2.0 license
- Good baseline for fine-tuning

OpenVLA [Kim et al., 2024]:

- 7B parameter model
- Built on Llama 2 backbone
- Competitive with proprietary models
- Easier to fine-tune than larger models

4.4 Training Pipeline

Data Preparation.

1. Collect robot demonstrations (teleoperation, scripted policies)
2. Annotate with language instructions
3. Normalize action spaces across robots
4. Apply data augmentation

Training Configuration.

- Batch size: 256-1024 (depends on GPU memory)
- Learning rate: 1e-4 to 1e-5 with warmup
- Optimizer: AdamW with weight decay
- Training time: Days to weeks on 8+ GPUs

Evaluation.

- Success rate on held-out tasks
- Generalization to new objects/instructions
- Real-world deployment testing

4.5 Deployment Considerations

Latency Requirements.

- Real-time control: ¡50ms inference
- Requires model optimization (quantization, pruning)
- Consider edge deployment (Jetson, TPU)

Safety.

- Hardware e-stops
- Workspace limits
- Force/torque monitoring
- Human detection and avoidance

5 Graph Neural Networks for Spatial Intelligence

5.1 Foundational GNN Architectures

5.1.1 Graph Convolutional Networks (GCN)

GCN [Kipf and Welling, 2017] performs spectral convolution on graphs:

Layer Update.

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

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

Implementation (PyTorch Geometric).

```
from torch_geometric.nn import GCNConv

class GCN(torch.nn.Module):
    def __init__(self, in_channels, hidden_channels, out_channels):
        super().__init__()
        self.conv1 = GCNConv(in_channels, hidden_channels)
        self.conv2 = GCNConv(hidden_channels, out_channels)

    def forward(self, x, edge_index):
        x = self.conv1(x, edge_index).relu()
        x = self.conv2(x, edge_index)
        return x
```

5.1.2 Graph Attention Networks (GAT)

GAT [Velickovic et al., 2018] uses attention for adaptive aggregation:

Attention Mechanism.

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

Advantages.

- Learns importance of different neighbors
- Multi-head attention for stability
- Better for heterogeneous graphs

5.1.3 GraphSAGE

GraphSAGE [Hamilton et al., 2017] enables inductive learning:

Sampling Strategy.

- Sample fixed-size neighborhood
- Enables mini-batch training
- Scales to large graphs

Aggregators.

- Mean aggregator
- LSTM aggregator
- Pooling aggregator

5.2 Spatio-Temporal GNNs

For time-varying spatial data (traffic, weather, urban dynamics):

5.2.1 DCRNN

Diffusion Convolutional Recurrent Neural Network [Li et al., 2018]:

- Models traffic as diffusion on road graph
- Bidirectional random walks
- GRU for temporal modeling

5.2.2 STGCN

Spatio-Temporal Graph Convolutional Network [Yu et al., 2018]:

- Separates spatial and temporal convolutions
- More efficient than RNN-based approaches
- Gated temporal convolution

5.2.3 Graph WaveNet

[Wu et al., 2019b]:

- Learns adaptive adjacency matrix
- Dilated causal convolutions for temporal modeling
- State-of-the-art traffic prediction

5.3 GNN-LLM Integration Patterns

5.3.1 Pattern 1: GNN as Encoder

Use GNN to encode graph structure, pass to LLM:

Architecture.

1. GNN encodes graph → node embeddings
2. Project embeddings to LLM token space
3. Concatenate with text tokens
4. LLM processes combined input

Example: GraphGPT [Tang et al., 2024]:

- Graph encoder aligned with LLM
- Enables graph-based question answering
- Supports various graph tasks

5.3.2 Pattern 2: LLM for Graph Enhancement

Use LLM to improve GNN:

Applications.

- Generate node features from text descriptions
- Explain GNN predictions [He et al., 2023]
- Augment training data

5.3.3 Pattern 3: GNN-RAG

Use GNN for knowledge graph retrieval [Wang et al., 2024a]:

Pipeline.

1. Query → retrieve relevant subgraph
2. GNN reasons over subgraph
3. Linearize subgraph for LLM
4. LLM generates final answer

6 World Models for Spatial Intelligence

6.1 Model-Based Reinforcement Learning

6.1.1 Dreamer Series

Dreamer [Hafner et al., 2019]:

- Learns latent dynamics model
- Plans in imagination
- Actor-critic in latent space

DreamerV2 [Hafner et al., 2021]:

- Discrete latent representations
- Human-level Atari performance
- More stable training

DreamerV3 [Hafner et al., 2023]:

- Single algorithm across domains
- Symlog predictions for stability
- Fixed hyperparameters

6.1.2 DayDreamer

[Wu et al., 2023a]: Transfer world models to real robots:

- Train in simulation
- Fine-tune on real robot data
- Demonstrates sim-to-real transfer

6.2 Video World Models

6.2.1 Genie

[Bruce et al., 2024]: Controllable world model from videos:

- Learns from internet videos
- Generates interactive environments
- Enables training without simulators

6.2.2 GAIA-1

[Hu et al., 2023a]: World model for autonomous driving:

- Generates realistic driving videos
- Conditioned on actions and context
- Enables scalable training data generation

6.3 LLM-Based World Models

RAP [Hao et al., 2023]: Reasoning via Planning:

- LLM as world model
- Monte Carlo Tree Search for planning
- Strong reasoning performance

Engineering Considerations.

- LLMs may hallucinate state transitions
- Need grounding in real observations
- Uncertainty quantification is challenging

7 Embodied AI Systems

7.1 Simulation Platforms

7.1.1 Habitat

[Savva et al., 2019, Szot et al., 2021, Puig et al., 2024]:

- High-fidelity 3D environments
- Supports navigation and manipulation
- Large-scale benchmark datasets
- Active research community

Habitat 3.0 features:

- Human-robot interaction
- Social navigation
- Collaborative tasks

7.1.2 iGibson

[Shen et al., 2021, Li et al., 2021]:

- Interactive environments
- Realistic physics simulation
- Object state changes

7.1.3 AI2-THOR

[Kolve et al., 2017]:

- Interactive visual AI
- Procedurally generated scenes
- Rich object interactions

7.2 Open-Ended Agents

7.2.1 Voyager

[Wang et al., 2023]: Open-ended exploration in Minecraft:

- LLM-driven curriculum learning
- Skill library construction
- Self-verification of skills

Key Components.

1. Automatic curriculum: LLM proposes tasks
2. Skill library: Stores successful programs
3. Iterative prompting: Refines code until success

7.2.2 MineDojo

[Fan et al., 2022]:

- Benchmark suite for open-ended agents
- Internet-scale knowledge base
- Diverse task types

7.3 Grounded Language Agents

7.3.1 SayCan

[Ahn et al., 2022]: Grounding LLMs in robot affordances:

Scoring Function.

$$p(\text{action}|\text{instruction}) \propto p_{\text{LLM}}(\text{action}|\text{instruction}) \cdot p_{\text{affordance}}(\text{action}) \tag{3}$$

Components.

- LLM provides semantic relevance
- Value function provides feasibility
- Combined scoring selects actions

7.3.2 Code as Policies

[Liang et al., 2023]: Generate executable robot code:

- LLM generates Python code
- Code calls robot APIs
- Enables complex behaviors

8 Geospatial Foundation Models

8.1 Remote Sensing Models

8.1.1 Prithvi

[Jakubik et al., 2024]: NASA/IBM geospatial foundation model:

- Pretrained on HLS (Harmonized Landsat Sentinel-2)
- Supports multiple downstream tasks
- Open weights available

Applications.

- Land use classification
- Flood mapping
- Wildfire detection
- Crop monitoring

8.1.2 SatMAE

[Cong et al., 2022]: Self-supervised learning for satellite imagery:

- Masked autoencoder approach
- Handles temporal sequences
- Strong transfer learning

8.1.3 SatlasPretrain

[Bastani et al., 2023]: Large-scale pretraining:

- 302M image dataset
- Multiple sensor types
- Diverse geographic coverage

8.2 Urban Computing

8.2.1 Traffic Prediction

State-of-the-art approaches [Jin et al., 2023, Li et al., 2018, Yu et al., 2018]:

- Graph-based spatial modeling
- Temporal sequence modeling
- Multi-step forecasting

8.2.2 Smart City Applications

[Zheng et al., 2014, Allam and Dhunny, 2020]:

- Traffic management
- Energy optimization
- Public safety
- Urban planning

9 Industry Applications and Internal Use Cases

9.1 External Industry Leaders

9.1.1 Palantir

[Palantir, 2023, Bailey, 2021, Freeman, 2021]:

- Foundry platform for data integration
- Geospatial analysis for defense
- Supply chain optimization

9.1.2 ESRI

[ESRI, 2023]:

- ArcGIS platform
- GeoAI capabilities
- Enterprise GIS solutions

9.1.3 Waymo

[Waymo, 2023, 2024,?]:

- Autonomous vehicle deployment
- End-to-end driving (EMMA)
- Safety-focused development

9.1.4 Foursquare

[Foursquare, 2023, Krumm, 2017]:

- Location intelligence
- Movement pattern analysis
- POI data

9.2 Internal Use Cases for AtlasPro AI

Based on our analysis, we identify the following high-value internal use cases:

Use Case 1: Intelligent Geospatial Analysis.

- Combine GNN-LLM for spatial reasoning
- Natural language queries over geospatial data
- Automated report generation

Use Case 2: Multi-Agent Coordination.

- Fleet management and optimization
- Collaborative robotics
- Distributed sensing

Use Case 3: Predictive Spatial Analytics.

- Traffic and demand forecasting
- Risk assessment
- Resource optimization

10 Evaluation Framework and Internal Benchmarking

10.1 Existing Benchmarks

10.1.1 Navigation

- R2R [Anderson et al., 2018]: Vision-language navigation
- RxR [Ku et al., 2020]: Multilingual VLN
- REVERIE [Qi et al., 2020]: Remote referring expression
- Habitat ObjectNav [Batra et al., 2020]: Object-goal navigation
- SOON [Zhu et al., 2021]: Scenarios for object navigation

10.1.2 Manipulation

- RLBench [James et al., 2020]: Robot learning benchmark
- Meta-World [Yu et al., 2020]: Multi-task manipulation
- BEHAVIOR [Srivastava et al., 2021, Li et al., 2023a]: Household activities
- CLIPort [Shridhar et al., 2022]: Language-conditioned manipulation

10.1.3 Spatial Reasoning

- CLEVR [Johnson et al., 2017]: Compositional reasoning
- GQA [Hudson and Manning, 2019]: Visual question answering
- SpatialVLM [Chen et al., 2024]: Spatial reasoning in VLMs
- REM [Thompson et al., 2025]: Embodied spatial reasoning
- EmbodiedBench [Yang et al., 2025]: Comprehensive embodied evaluation

10.1.4 Agent Benchmarks

- AgentBench [Liu et al., 2023d]: LLM agent evaluation
- WebArena [Zhou et al., 2023]: Web-based agents
- OSWorld [Xie et al., 2024]: Computer use agents
- SafeAgentBench [Yin et al., 2025]: Safe task planning

10.1.5 Geospatial

- BigEarthNet [Sumbul et al., 2019]: Land use classification
- fMoW [Christie et al., 2018]: Functional map of the world
- xBD [Gupta et al., 2019a]: Building damage assessment
- SpaceNet [Van Etten et al., 2018]: Building and road extraction

10.2 Internal Benchmarking Framework

We recommend building an internal benchmarking framework with the following components:

Continuous Evaluation.

- Automated testing on each commit
- Performance tracking over time
- Regression detection

Custom Benchmarks.

- Tasks specific to AtlasPro AI use cases
- Real-world data from our deployments
- Edge cases and failure modes

Human Evaluation.

- User studies for subjective quality
- Expert evaluation for safety-critical tasks
- A/B testing in production

11 Safety Engineering

11.1 Principles

Defense in Depth. Multiple layers of safety:

- Model-level safety (constitutional AI [Bai et al., 2022])
- System-level safety (sandboxing, limits)
- Hardware-level safety (e-stops, sensors)

Fail-Safe Design. Systems should fail safely:

- Default to safe states
- Graceful degradation
- Clear failure modes

11.2 Red Teaming

[Ganguli et al., 2022]: Proactive adversarial testing:

- Dedicated red team
- Automated adversarial testing
- Bug bounty programs

11.3 Alignment

[Amodei et al., 2016, Hendrycks et al., 2021]: Ensuring AI systems behave as intended:

- Clear specification of goals
- Value alignment techniques
- Human oversight mechanisms

12 Autonomous Driving: Deep Technical Analysis

Autonomous driving represents one of the most demanding applications of spatial AI, requiring real-time perception, prediction, and planning in safety-critical environments.

12.1 End-to-End Driving Architectures

12.1.1 UniAD: Unified Autonomous Driving

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

Architecture Components.

- **BEV Encoder:** Transforms multi-camera images to bird's-eye-view representation
- **Track Query:** Maintains object tracking across frames
- **Motion Query:** Predicts future trajectories of agents
- **Occupancy Prediction:** Forecasts future occupancy grids
- **Planning Head:** Generates ego-vehicle trajectory

Key Innovation. Joint training of all components enables information flow between tasks, improving overall performance compared to modular approaches.

12.1.2 VAD: Vectorized Autonomous Driving

[Jiang et al., 2023] introduces vectorized scene representation:

- Represents scenes as sets of vectors (lanes, agents)
- More efficient than dense grid representations
- Enables direct reasoning about scene structure

12.1.3 EMMA: End-to-End Multimodal Model

[Waymo, 2024] from Waymo demonstrates multimodal driving:

- Integrates camera, lidar, and radar inputs
- Language-conditioned driving
- Reasoning about complex scenarios

12.2 BEV Perception Pipeline

Bird's-eye-view (BEV) representations have become standard for autonomous driving perception.

12.2.1 LSS: Lift-Splat-Shoot

[Philion and Fidler, 2020] introduced the foundational approach:

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

12.2.2 BEVFormer

[Li et al., 2022, Yang et al., 2023] uses transformers for BEV generation:

- Spatial cross-attention for multi-camera fusion
- Temporal self-attention for temporal modeling
- Deformable attention for efficiency

12.3 Datasets for Autonomous Driving

Table 2: 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

13 3D Scene Understanding: Technical Deep-Dive

13.1 Neural Radiance Fields (NeRF)

13.1.1 Original NeRF

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

MLP Architecture.

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

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

Volume Rendering.

$$C(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))\mathbf{c}(\mathbf{r}(t), \mathbf{d})dt \tag{5}$$

where $T(t) = \exp(-\int_{t_n}^{t} \sigma(\mathbf{r}(s))ds)$.

13.1.2 Mip-NeRF 360

[Barron et al., 2022] extends NeRF to unbounded scenes:

- Integrated positional encoding for anti-aliasing
- Contraction function for unbounded scenes
- Proposal network for efficient sampling

13.2 3D Gaussian Splatting

[Kerbl et al., 2023] provides real-time novel view synthesis:

Representation. Scene as set of 3D Gaussians:

- Position (mean)
- Covariance matrix (shape)
- Opacity

- Spherical harmonics (view-dependent color)

Rendering. Differentiable rasterization:

- Project Gaussians to 2D
- Sort by depth
- Alpha-blend front-to-back

Advantages over NeRF.

- Real-time rendering (100+ FPS)
- Explicit representation (easier editing)
- Faster training (minutes vs. hours)

13.3 Point Cloud Processing

13.3.1 PointNet

[Qi et al., 2017a] pioneered deep learning on point clouds:

Key Innovations.

- Permutation invariance through max pooling
- T-Net for spatial transformer
- Direct processing of raw point clouds

13.3.2 PointNet++

[Qi et al., 2017b] adds hierarchical structure:

- Set abstraction layers for local feature learning
- Multi-scale grouping for varying densities
- Feature propagation for segmentation

13.4 Scene Graphs

13.4.1 Visual Scene Graphs

[Xu et al., 2017, Krishna et al., 2017] represent scenes as graphs:

- Nodes: Objects with attributes
- Edges: Relationships between objects
- Enables structured reasoning

13.4.2 3D Scene Graphs

[Armeni et al., 2019, Rosinol et al., 2020] extend to 3D:

- Hierarchical structure (building → room → object)
- Metric information (positions, sizes)
- Semantic relationships

14 Multi-Modal Foundation Models

14.1 Vision-Language Models

14.1.1 LLaVA

[Liu et al., 2023b,c] pioneered visual instruction tuning:

Architecture.

- Vision encoder: CLIP ViT-L/14
- Projection layer: Linear or MLP
- Language model: Vicuna/LLaMA

Training Stages.

1. Pre-training: Image-text alignment
2. Fine-tuning: Visual instruction tuning

14.1.2 Flamingo

[Alayrac et al., 2022] introduced few-shot multimodal learning:

- Perceiver resampler for visual tokens
- Gated cross-attention for vision-language fusion
- In-context learning with interleaved images and text

14.1.3 BLIP-2

[Li et al., 2023b] efficiently bootstraps vision-language pretraining:

- Q-Former bridges frozen image encoder and LLM
- Two-stage training for efficiency
- Strong zero-shot performance

14.2 Frontier Models

14.2.1 GPT-4V

[OpenAI, 2023] represents frontier multimodal capabilities:

- Strong visual understanding
- Complex reasoning over images
- Integration with tool use

14.2.2 Gemini

[Team and Google, 2023] from Google DeepMind:

- Native multimodal training
- Strong performance across modalities
- Available in multiple sizes

14.2.3 Qwen-VL

[Bai et al., 2023, Wang et al., 2024c] provides strong open-source alternative:

- Competitive with proprietary models
- Multiple resolution support
- Strong Chinese and English performance

15 Detailed Benchmark Analysis

This section provides detailed analysis of key benchmarks for internal evaluation planning.

15.1 Navigation Benchmark Details

15.1.1 Room-to-Room (R2R)

[Anderson et al., 2018]:

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

State-of-the-Art Performance.

Table 3: R2R Val-Unseen Performance

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

15.1.2 RxR: Room-across-Room

[Ku et al., 2020]:

- Multilingual (English, Hindi, Telugu)
- Longer paths than R2R
- More detailed instructions

15.2 Manipulation Benchmark Details

15.2.1 RLBench

[James et al., 2020]:

- 100 unique tasks
- Multiple variations per task
- CoppeliaSim simulation

15.2.2 Meta-World

[Yu et al., 2020]:

- 50 manipulation tasks
- Multi-task and meta-learning evaluation
- Sawyer robot simulation

15.3 Agent Benchmark Details

15.3.1 AgentBench

[Liu et al., 2023d]:

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

15.3.2 EmbodiedBench

[Yang et al., 2025]:

- Comprehensive embodied MLLM evaluation
- Multiple spatial reasoning tasks
- Manipulation and navigation

16 Implementation Recipes

This section provides practical implementation guidance for common spatial AI tasks.

16.1 Recipe: Building a RAG-Enhanced Spatial Agent

Step 1: Set Up Vector Database.

```
import chromadb
client = chromadb.Client()
collection = client.create_collection("spatial_knowledge")
```

Step 2: Index Spatial Knowledge.

```
# Embed and store spatial documents
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):
    # Retrieve relevant context
    results = collection.query(query_texts=[query], n_results=5)
    context = "\n".join(results["documents"][0])

    # Generate response with context
    prompt = f"Context: {context}\n\nQuery: {query}"
    response = llm.generate(prompt)
    return response
```

16.2 Recipe: GNN for Traffic Prediction

Step 1: Build Traffic Graph.

```
import torch_geometric as pyg

# Create edge index from road network
edge_index = torch.tensor([[src_nodes], [dst_nodes]])

# Node features: historical traffic
x = torch.tensor(traffic_history)  # [N, T, F]
```

Step 2: Define Spatio-Temporal GNN.

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

16.3 Recipe: Deploying VLA Model on Robot

Step 1: Load Pretrained Model.

```
from transformers import AutoModelForVision2Seq
model = AutoModelForVision2Seq.from_pretrained("openvla/openvla-7b")
```

Step 2: Optimize for Deployment.

```
# Quantize for faster inference
model = torch.quantization.quantize_dynamic(
    model, {torch.nn.Linear}, dtype=torch.qint8
)
```

Step 3: Robot Control Loop.

```
while not done:
    # Get observation
    image = camera.capture()
```

```
instruction = "Pick up the red cup"

# Predict action
action = model.predict(image, instruction)

# Execute action
robot.execute(action)

# Check completion
done = check_task_completion()
```

17 Computational Requirements

This section provides guidance on computational resources for different spatial AI tasks.

17.1 Training Requirements

Table 4: Computational Requirements for Training

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

17.2 Inference Requirements

Table 5: 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

18 Open Challenges and Research Directions

18.1 Robust Spatial Representation

[Mildenhall et al., 2020, Kerbl et al., 2023, Hong et al., 2023b]:

- Generalization across scenes
- Handling novel viewpoints
- Efficient 3D representations

18.2 Long-Horizon Planning

[Song et al., 2023, Valmeekam et al., 2023]:

- Planning over extended time horizons
- Complex task decomposition
- Error recovery and replanning

18.3 Sim-to-Real Transfer

[Zhao et al., 2020, Tobin et al., 2017, James et al., 2019]:

- Domain randomization
- System identification
- Real-world fine-tuning

18.4 Multi-Agent Coordination

[Zhang et al., 2021, Wu et al., 2023b, Hong et al., 2023a]:

- Communication protocols
- Task allocation
- Emergent coordination

18.5 Scalable Data Collection

[Collaboration, 2023, Walke et al., 2023]:

- Efficient data collection methods
- Data sharing and standardization
- Synthetic data generation

19 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.

1. Memory architecture is critical—invest in hierarchical memory systems.
2. GNN-LLM integration is a powerful paradigm for spatial reasoning.
3. World models enable safe planning through imagination.
4. Open-source VLA models provide strong baselines for robotics.
5. Safety engineering must be built in from the start.

Next Steps.

1. Establish internal benchmarking infrastructure.
2. Build prototype GNN-LLM integration system.

3. Deploy simulation environment for safe development.

4. Implement safety engineering practices.

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.

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mane. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016.

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.

Anthropic. Claude 3 model card. *Anthropic Technical Report*, 2024.

Iro Armeni, Ozan Sener, Amir R Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3d semantic parsing of large-scale indoor spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1534–1543, 2016.

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.

Jinze Bai, Shuai Bai, Shusheng Yang, et al. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*, 2023.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022.

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.

Dhruv Batra, Aaron Gokaslan, Aniruddha Kembhavi, Oleksandr Maksymets, Roozbeh Mottaghi, Manolis Savva, Alexander Toshev, and Erik Wijmans. Objectnav revisited: On evaluation of embodied agents navigating to objects. In *arXiv preprint arXiv:2006.13171*, 2020.

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.

Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. In *2017 International Conference on 3D Vision (3DV)*, pages 667–676. IEEE, 2017.

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

Boyuan Chen, Zhuo Xu, Sean Kirmani, Brian Ichter, Danny Driess, Pete Florence, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. *arXiv preprint arXiv:2401.12168*, 2024.

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.

Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Niessner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5828–5839, 2017.

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.

David Freeman. Palantir's role in government and commercial analytics. Industry Analysis, 2021.

Deep Ganguli, Liane Lovitt, Jackson Kernion, Amanda Askell, Yuntao Bai, Saurav Kadavath, Ben Mann, Ethan Perez, Nicholas Schiefer, Kamal Ndousse, et al. Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned. *arXiv preprint arXiv:2209.07858*, 2022.

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

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*, 2019a.

Saurabh Gupta, Varun Tolani, James Davidson, Sergey Levine, Rahul Sukthankar, and Jitendra Malik. Cognitive mapping and planning for visual navigation. *International Journal of Computer Vision*, 128(5): 1311–1330, 2019b.

Kelvin Guu et al. Realm: Retrieval-augmented language model pre-training. *ICML*, 2020.

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.

Xiaoxin He, Xavier Bresson, Thomas Laurent, Adam Perold, Yann LeCun, and Bryan Hooi. Explanations as features: Llm-based features for text-attributed graphs. *arXiv preprint arXiv:2305.19523*, 2023.

Dan Hendrycks, Collin Burns, Steven Basart, Andrew Critch, Jerry Li, Dawn Song, and Jacob Steinhardt. Aligning ai with shared human values. *arXiv preprint arXiv:2008.02275*, 2021.

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.

Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6700–6709, 2019.

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, Paul Wohlhart, Mrinal Kalber, Andrew J Davison, and Sergey Levine. Sim-to-real via sim-to-sim: Data-efficient robot learning from randomized simulation. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 2262–2269, 2019.

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.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2901–2910, 2017.

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.

John Krumm. Introduction to location-based services. *Ubiquitous Computing Fundamentals*, pages 293–334, 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*, 2023a.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023b.

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.

Bo Liu et al. Llm+p: Empowering large language models with optimal planning proficiency. *arXiv preprint arXiv:2304.11477*, 2023a.

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

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Advances in Neural Information Processing Systems*, volume 36, 2023c.

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*, 2023d.

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.

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.

Joon Sung Park, Joseph C O'Brien, Carrie J Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Generative agents: Interactive simulacra of human behavior. *arXiv preprint arXiv:2304.03442*, 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.

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.

Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Cliport: What and where pathways for robotic manipulation. In *CoRL*, 2022.

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.

Gemini Team and Google. Gemini: A family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

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

James Thompson et al. Rem: A benchmark for evaluating embodied spatial reasoning in mllms. *arXiv preprint arXiv:2512.00736*, 2025.

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.

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.

Adam Van Etten et al. Spacenet: A remote sensing dataset and challenge series. *arXiv preprint arXiv:1807.01232*, 2018.

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.

Peng Wang et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024c.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

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

Waymo. Waymo safety report: Building the world's most experienced driver. Technical report, Waymo LLC, 2024.

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.

Tianbao Xie et al. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments. *arXiv preprint arXiv:2404.07972*, 2024.

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

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.

Sheng Yin, Xianghe Xiong, Wenhao Huang, et al. Safeagentbench: A benchmark for safe task planning of embodied llm agents. *arXiv preprint arXiv:2412.13178*, 2025.

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.

Andy Zeng, Pete Florence, Jonathan Tompson, Stefan Welker, Jonathan Chien, et al. Transporter networks: Rearranging the visual world for robotic manipulation. In *CoRL*, 2021.

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.

Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Yonatan Bisk, Daniel Fried, Uri Alon, et al. Webarena: A realistic web environment for building autonomous agents. *arXiv preprint arXiv:2307.13854*, 2023.

Fengda Zhu, Yi Zhu, Xiaojun Chang, and Xiaodan Liang. Soon: Scenarios for object navigation with natural language instructions. In *CVPR*, 2021.