A Unified Framework for a Dynamically Embodied, Multimodal AI Agent

Introduction

This report presents a technical framework for a next-generation artificial intelligence, a generalist agent capable of fluidly perceiving, reasoning, and acting across both physical and digital domains. The central challenge addressed is that of *dynamic embodiment*: enabling a single AI to inhabit and control a robotic platform and a desktop Graphical User Interface (GUI), either sequentially or simultaneously. The objective is to move beyond specialized, single-domain agents and architect a unified intelligence that can leverage the unique affordances of different environments to achieve complex, open-ended goals. Such a system would represent a significant step toward Artificial General Intelligence (AGI), particularly its embodied form, which requires interaction with and understanding of the world in its multifaceted reality.[1]

The core architectural proposal put forth in this document is a novel system centered on a unified **Cognitive Core**, which houses the agent's general, embodiment-agnostic intelligence. This Core interfaces with its environment not directly, but through a sophisticated middleware layer termed the **Cognitive-Embodiment Abstraction Layer (CEAL)**. The CEAL is designed to decouple high-level, semantic reasoning from low-level, platform-specific execution. This abstraction allows the Cognitive Core to issue embodiment-agnostic intentions—abstract goals like ``—which the CEAL then translates into concrete, executable action sequences for the currently active "body," whether that is a physical robot or a digital desktop environment. This design is the foundational key to enabling dynamic embodiment switching, concurrent control of multiple platforms, and true cross-domain skill generalization.

The report is structured to provide a comprehensive blueprint for the research and development of this system. Section 1 defines the agent's "mind," the Cognitive Core, detailing its multimodal architecture and internal world model. Section 2 dissects the two distinct "bodies" the AI will inhabit—the robotic and desktop embodiments—specifying their unique perception and action subsystems. Section 3 presents the technical architecture of the CEAL, the critical abstraction layer that connects mind and body. Section 4 details the advanced mechanisms for control and arbitration, explaining how the agent decides which body to use and how it can manage them concurrently. Finally, Section 5 provides a strategic roadmap for implementation, outlines key research challenges, and situates the proposed framework within the broader context of the pursuit of Embodied AGI.

Section 1: The Cognitive Core: Architecting a Generalist Multimodal Intelligence

The heart of the proposed agent is its Cognitive Core, an embodiment-agnostic "brain" designed to process a rich tapestry of sensory information from any source and formulate high-level, goal-oriented plans. This core must be a generalist, capable of reasoning about concepts and tasks abstractly, independent of the specific physical or digital form it currently inhabits. Its architecture is therefore paramount, dictating the system's ability to learn, generalize, and scale.

1.1. Foundational Multimodal Architecture: A Hybrid Approach

The foundation of the Cognitive Core must be a powerful Multimodal Large Language Model (MLLM) capable of "any-to-any" modality processing.[3] The contemporary AI landscape has rapidly evolved from simple text-vision models to sophisticated systems that can seamlessly integrate a wide array of sensory inputs, including text, images, audio, video, depth, thermal, and inertial measurement unit (IMU) data.[5] The choice of MLLM architecture directly influences the system's scalability, training efficiency, and the ease with which new sensory modalities or even entirely new embodiments can be incorporated in the future.
A comprehensive 2024 survey systematically identifies and characterizes four prevalent architectural patterns for MLLMs, distinguished by their method of fusing multimodal inputs.[3] These types are:

- **Type-A (Standard Cross-Attention Deep Fusion):** Integrates modalities deep within the model's internal layers using standard cross-attention mechanisms. Models like Flamingo exemplify this approach, which allows for rich, interleaved fusion but can be computationally intensive.[3]
- **Type-B (Custom Layer Deep Fusion):** Also performs deep fusion but utilizes custom-designed layers instead of standard cross-attention, offering greater architectural flexibility to optimize the fusion process for specific modalities.
- **Type-C (Non-Tokenizing Early Fusion):** Employs modality-specific encoders to process each input stream independently. The outputs of these encoders are then fused at the input stage of the core LLM. This is a non-tokenizing approach that promotes modularity.[3]
- **Type-D (Tokenizing Early Fusion):** Leverages tokenizers to convert all input modalities into a unified sequence of discrete tokens. This allows the model to process diverse data within a single, shared framework but can introduce complexity in the tokenization process itself.[3]

For the proposed generalist agent, a **hybrid Type-C/D architecture** is recommended. This architecture leverages a powerful, frozen, pre-trained LLM (such as LLaMA-3 or a model from

the Gemini family) as the central cognitive engine, responsible for high-level reasoning and planning.[9] The hybrid nature of the architecture is realized through two key components:

1. **Type-C Component (Modular Modality Encoders):** In line with the modularity required for dynamic embodiment, each distinct sensory modality will be processed by a specialized, pre-trained encoder. This approach allows for optimal feature extraction for each data type and facilitates future expansion. For instance, vision will be handled by a CLIP-based Vision Transformer (ViT) encoder [10], audio by an encoder like CLAP [9], and more specialized data streams, such as proprioceptive and force-torque feedback from the robot, will have their own dedicated encoders. The desktop embodiment will similarly have "encoders" for processing structured data derived from screenshots and GUI element analysis.[11]
2. **Type-D Component (Unified Input Space):** The outputs from these diverse modality encoders are projected into a shared, high-dimensional embedding space. This creates a unified "language of thought" that the core LLM can understand and reason over, regardless of the originating sensory modality.[5] This projection is managed by a learnable "connector" module. The connector can range from a simple Multi-Layer Perceptron (MLP), as seen in the LLaVA series, to a more sophisticated query-based mechanism like the Q-Former from BLIP-2.[9]

The justification for this hybrid approach lies in its ability to combine the primary advantages of both Type-C and Type-D architectures. The modularity of Type-C is essential for a system designed to interface with different "bodies," making it straightforward to add, remove, or upgrade sensors or even integrate an entirely new embodiment in the future. Simultaneously, the unified reasoning space characteristic of Type-D is what empowers the LLM to perform complex, cross-modal reasoning—for example, relating a spoken command (audio) to an object seen by the robot's camera (vision) and an instruction manual displayed on the desktop (GUI screenshot). This combination provides the ideal balance of flexibility and reasoning power required for a truly generalist agent.[3]

Table 1: Comparative Analysis of MLLM Architectural Patterns

Architectural Type	Fusion Method	Scalability	Modularity	Data Requirements	Suitability for Dynamic Embodiment
Type-A (SCDF)	Deep Fusion via Standard Cross-Attention	Medium	Low	High	**Low.** The tight integration of modalities within the model's core layers makes it difficult to dynamically add or switch sensory inputs from different

					embodiments without significant retraining.
Type-B (CLDF)	Deep Fusion via Custom-Designed Layers	Medium	Low	High	**Low.** Similar to Type-A, the deep fusion approach creates a tightly coupled system that is not well-suited for the plug-and-play nature of dynamic embodiment.
Type-C (NTEF)	Early Fusion via Modality-Specific Encoders	High	High	High (for encoders)	**High.** The use of independent encoders for each modality is highly modular. A new sensor or an entire embodiment can be added by simply training a new encoder and a connector, without altering the core LLM.
Type-D (TEF)	Early Fusion via Unified Tokenization	High	Medium	Very High	**Medium.** While it creates a powerful unified representation, the need for a universal tokenizer that

					can handle all possible modalities (including novel sensor data) can be a bottleneck. It is less modular than Type-C.

1.2. The World Model: Enabling Predictive Reasoning and Imagination

While a state-of-the-art MLLM provides formidable capabilities in semantic understanding and contextual reasoning, it operates primarily by recognizing patterns in its vast training data. It lacks a genuine, predictive understanding of physical dynamics and causal consequences.[15] An MLLM can reason that dropping a glass will likely cause it to break because this pattern is prevalent in text and video data. However, it cannot
simulate the physics of the fall to predict the exact outcome. To achieve robust, safe, and efficient long-horizon planning, particularly in the unpredictable physical world, the agent requires an internal, learned simulation of its environment—a **world model**.[16]
The proposed architecture for the Cognitive Core therefore includes a dedicated World Model module that operates in concert with the MLLM. This module will be architected based on a recurrent state-space model (RSSM), a design proven effective in agents like the Dreamer series.[17] The World Model's function is to learn the temporal dynamics of the environment. It takes the compressed latent representations from the various modality encoders as input and is trained to predict the next state of the environment given the current state and a proposed action.
This capability allows the agent to "imagine" the potential outcomes of different action sequences within a compressed latent space. This has two profound benefits. First, it dramatically improves sample efficiency for learning, as the agent can explore many possibilities in its "imagination" without costly and slow real-world trial and error.[17] Second, it enables more robust and safer planning. The MLLM is responsible for high-level, semantic planning (e.g., "To clean the table, I must first move the cup"). It can then query the World Model to check the feasibility and predict the outcome of the low-level plans derived from this high-level strategy (e.g., "Is it possible to grasp the cup from this angle without knocking it over? What is the predicted state of the world if I execute this grasp?").[16]
This creates a powerful internal validation loop, a synergy between the MLLM's semantic reasoning and the World Model's predictive, physics-based reasoning. The MLLM might generate a plan that is semantically plausible but physically impossible—a common form of

hallucination in embodied contexts.[20] For example, given the command "put the large book on the small shelf," an MLLM might generate a step-by-step plan because the concept of placing books on shelves is common. However, when this plan is passed to the World Model for validation, the model, having learned from visual data about relative sizes and physical constraints, would predict a failure state or an extremely low-probability outcome. This feedback acts as a strong corrective signal, forcing the MLLM to replan, perhaps by concluding the task is impossible and reporting this to the user. This internal conflict resolution is a critical mechanism for enhancing the safety and reliability of the agent's actions. For the desktop environment, the World Model's role is analogous; it learns the "physics" of the GUI, such as predicting that clicking a specific button will transition the screen to a new, predictable state.[21]

1.3. Training Strategy for Generalization

Training an agent of this complexity requires a carefully orchestrated, multi-stage strategy. This approach is designed to first build foundational knowledge and capabilities, then fine-tune the agent to follow instructions across its different embodiments, and finally align its behavior with human preferences and safety constraints.[9]

Stage 1: Foundational Pre-training

- **Objective:** The primary goals of this initial stage are twofold: first, to align the representations of all modality-specific encoders with the LLM's shared embedding space, and second, to train the World Model on the fundamental dynamics of both physical and digital environments.
- **Data:** This stage requires massive and diverse datasets. For multimodal alignment, this includes large-scale image-text pairs (e.g., LAION-5B, COYO-700M), video-text data, and audio-text corpora.[9] For training the World Model and grounding the agent in real-world interaction, large robotics datasets featuring sensorimotor trajectories are essential (e.g., the Open-X-Embodiment dataset).[9] For the desktop embodiment, datasets like ScreenAgent, Mind2Web, and other GUI interaction datasets are critical for teaching the agent how to perceive and act in digital environments.[25]
- **Method:** During this phase, the core LLM and the modality encoders are typically kept frozen to preserve their powerful pre-trained knowledge. The training focuses on the connector module, which learns to project the encoded features from different modalities into the LLM's space. This is often achieved using a standard cross-entropy loss, where the model learns to predict text (e.g., captions, descriptions) associated with the multimodal inputs.[9] Concurrently, the World Model is trained on a predictive loss objective, learning to forecast future latent states based on current states and actions.

Stage 2: Cross-Domain Instruction Tuning

- **Objective:** This stage moves beyond simple representation alignment to explicitly teach the agent how to follow complex, high-level instructions and generalize its skills across

both robotic and desktop tasks.

- **Data:** A high-quality, meticulously curated dataset of multimodal instruction-following examples is required. This dataset is the cornerstone of creating a true generalist. It must contain a rich mixture of tasks from both domains, formatted as instructions. Examples would range from "Pick up the red block and place it in the blue bin" for the robot, to "Find the latest quarterly sales report on the shared drive and email it to my manager" for the desktop.[9] Creating such a dataset will likely require significant investment in manual annotation or the use of powerful teacher models (e.g., GPT-4V, Gemini Ultra) to generate high-quality instruction-response pairs.
- **Method:** The agent undergoes supervised fine-tuning (SFT) on this cross-domain instruction dataset. This process updates the connector module and, crucially, fine-tunes the core LLM itself. To prevent the LLM from losing its general world knowledge (a phenomenon known as catastrophic forgetting), parameter-efficient fine-tuning (PEFT) techniques like Low-Rank Adaptation (LoRA) are employed. These methods introduce a small number of new, trainable parameters while keeping the bulk of the original LLM weights frozen, allowing for efficient adaptation without compromising the model's foundational capabilities.[10]

The very structure of this cross-domain instruction tuning phase is what enables the agent to develop a higher level of abstract reasoning. By training on an interleaved dataset of robotic and desktop tasks, the model is forced to learn the underlying concepts that unite them. For example, it might learn that "navigating to a file within a nested folder structure" on a desktop and "navigating to a can inside a cupboard" in a physical kitchen share an abstract structure of hierarchical search and retrieval. This cross-pollination of concepts is what will ultimately allow the agent to exhibit zero-shot or few-shot generalization to novel tasks that may even blend the two domains (e.g., "Find the assembly manual for this chair on the manufacturer's website and then guide my arms to build it"). This capability is a direct and intended outcome of this specific training strategy.[28]

Stage 3: Alignment Tuning

- **Objective:** The final stage of training is dedicated to ensuring the agent's behavior is safe, reliable, and aligned with nuanced human preferences. This is especially critical for an embodied agent that has the capacity to directly affect the physical and digital worlds.
- **Data:** This stage requires human preference data. This typically consists of pairs of agent responses or action sequences to a given prompt, where human annotators have ranked which outcome is better. Datasets like VLFeedback provide a starting point, but custom data collection focusing on safety and helpfulness in embodied scenarios will be necessary.[9]
- **Method:** The agent's policy is fine-tuned using techniques based on human feedback. The two leading methods are Reinforcement Learning from Human Feedback (RLHF) and Direct Preference Optimization (DPO). In RLHF, the preference data is first used to train a separate "reward model" that learns to predict human preferences. Then, reinforcement learning is used to optimize the agent's policy to maximize the score from

this reward model. DPO is a more recent technique that simplifies this pipeline by learning from the preference data directly without needing to train an explicit reward model.[9] This final alignment stage is crucial for mitigating harmful behaviors, reducing both factual and physical hallucinations, and ensuring the agent acts as a helpful, trustworthy, and safe assistant.[20]

Section 2: The Embodiment Layer: Perception and Action in Physical and Digital Worlds

The Cognitive Core, while powerful, is an ungrounded intelligence. It requires a "body" through which to perceive and act upon the world. This framework proposes two distinct embodiments—one physical and one digital—each with a specialized layer of hardware and software interfaces. The design of this layer is governed by the principle that while the specific tools of each embodiment differ, their fundamental operational cycle—the perception-action loop—is the same.

2.1. The Robotic Embodiment: Interfacing with the Physical World

The robotic embodiment is the agent's physical presence, allowing it to interact with, manipulate, and navigate the tangible world. Its capabilities are defined by its perception and action subsystems.

2.1.1. Perception Subsystems

A robot's intelligence is fundamentally constrained by the richness and fidelity of its sensory input.[6] A comprehensive sensor suite is therefore non-negotiable for robust performance in unstructured environments.

- **Core Sensors:** The primary sensory apparatus should include:
    - **High-resolution stereo cameras:** To provide rich visual data and enable depth perception.
    - **LiDAR:** For generating accurate, dense 3D point clouds of the environment, crucial for mapping and obstacle avoidance.[31]
    - **Multiple microphones:** Arranged in an array to enable sound source localization and capture clear audio for speech recognition and environmental sound analysis.
    - **Force-torque sensors:** Located in the robot's joints and end-effectors (wrists) to provide feedback on physical interactions, enabling compliant and safe manipulation.
- **Advanced Sensors:** To further enhance perceptual capabilities, the suite can be

augmented with:
    - **Tactile sensors:** A "skin" for the gripper or hand, providing fine-grained data about contact pressure, texture, and slip detection, which is vital for delicate object manipulation.
    - **Inertial Measurement Units (IMUs):** To provide data on acceleration and orientation, aiding in state estimation and stabilizing the mobile base.
- **Real-Time Data Processing:** Raw sensor data is high-bandwidth and noisy. It must be processed in real-time to be useful for decision-making.[33] This necessitates a hierarchical processing pipeline. At the lowest level, running on edge processors close to the sensors, data undergoes filtering and noise reduction. The cleaned data streams are then fused at a higher level to create a coherent, unified model of the environment's state. For example, visual data from cameras can be fused with depth data from LiDAR using techniques like Kalman filters to produce a more accurate and robust 3D representation than either sensor could provide alone.[35]

2.1.2. Action Subsystems

The agent exerts its will on the physical world through a set of actuators, coordinated by a sophisticated control middleware.

- **Hardware:** The physical platform should consist of:
    - **High-Degree-of-Freedom (DoF) arms:** At least one, preferably two, 7-DoF arms to mimic human-like dexterity and reach.
    - **Mobile base:** An omnidirectional or differential drive base for navigation in complex indoor spaces.
    - **Adaptive gripper:** An end-effector capable of both strong power grasps for heavy objects and delicate precision grasps for small or fragile items.
- **Control Middleware (ROS 2):** The Robot Operating System (ROS) is the de facto industry and research standard for building robotic applications.[37] This framework will utilize
  ROS 2 for its enhanced real-time capabilities, improved security, and more robust communication architecture compared to ROS 1. ROS provides a modular, message-passing system where each sensor, actuator, and processing algorithm can be run as an independent "node".[39] These nodes communicate by publishing and subscribing to named data streams called "topics".[41] This distributed, tool-based philosophy aligns perfectly with the proposed modular architecture of the Cognitive Core and the CEAL, allowing the system to treat each hardware component as a distinct, addressable tool.
- **APIs for Control:** At the lowest level, ROS nodes must translate abstract commands (e.g., "move gripper to coordinates") into hardware-specific signals. This is handled by standardized robotics APIs. While ROS provides the messaging framework, high-speed, low-latency control commands may be sent via protocols like gRPC, while less

time-critical configuration could use RESTful APIs. These APIs form the final link in the chain, converting ROS messages into the electrical signals that drive the motors.[43]

2.1.3. The Sim-to-Real Challenge

Training a robotic agent, especially one using reinforcement learning, directly and exclusively in the real world is prohibitively slow, expensive, and potentially dangerous.[45] Therefore, a significant portion of the agent's training, particularly for learning the World Model's physical dynamics and for initial policy optimization, must occur in a high-fidelity physics simulation (e.g., NVIDIA Isaac Sim, MuJoCo). This introduces the "sim-to-real" problem: policies trained in simulation often fail when transferred to the real robot due to the **reality gap**—the inevitable discrepancies between the simulated and real worlds in terms of physics (friction, contact dynamics), sensor noise, and visual appearance.[45]
To bridge this gap, a two-pronged strategy is essential:

1. **Domain Randomization:** During the simulation phase, the parameters of the environment are intentionally and continuously randomized. This includes randomizing physical properties like mass and friction coefficients of objects, as well as visual properties like lighting conditions, textures, and camera positions.[45] By exposing the agent to a wide variety of simulated conditions, this technique forces the learned policy to become robust to these variations and to focus on learning features that are invariant across domains. The goal is that, from the policy's perspective, the real world appears as just another variation of the randomized simulation it has already seen.[48]
2. **Domain Adaptation:** This technique aims to make the simulated data look more like real-world data. For visual data, this can be achieved using generative models like CycleGAN, which learn to translate images from the simulation domain to the real-world domain without requiring paired examples.[49] By training the agent on these adapted, more realistic images, the visual discrepancy between sim and real is minimized. Combining domain randomization with domain adaptation provides a powerful and robust methodology for achieving successful sim-to-real transfer.[49]

2.2. The Desktop Embodiment: Interfacing with the Digital World

The desktop embodiment allows the agent to perceive and act within a standard computer GUI. While it lacks physical form, it possesses its own set of "senses" and "actuators" that are conceptually parallel to its robotic counterpart. The fundamental operational principle remains the perception-action loop: the agent perceives the screen, decides on an action, executes it, and perceives the resulting new screen state.[51]

2.2.1. Perception Subsystems

The primary perceptual challenge in the digital realm is **Visual Screen Comprehension**. This is not merely about taking a screenshot; it requires a deep, structured understanding of the GUI's content and affordances. A state-of-the-art perception stack for this purpose would be multi-layered, inspired by recent models like Microsoft's Magma and OmniParser, and projects like ScreenAgent.[11]

- **Layer 1: Element Detection and Segmentation:** A computer vision model, such as a fine-tuned YOLO or a vision transformer, is used to identify and draw bounding boxes around all interactive and non-interactive GUI elements on the screen. This includes buttons, text fields, icons, menus, sliders, and images.[54]
- **Layer 2: Text and Icon Recognition:** An Optical Character Recognition (OCR) engine extracts all textual content from the screen, associating the text with the GUI elements detected in the previous layer (e.g., the text label on a button).[55] Specialized icon recognition models can classify common icons (e.g., save, print, trash).
- **Layer 3: Contextual Understanding:** The structured data from the first two layers (element locations, types, and text) is fed into a Vision Question Answering (VQA) model. This allows the agent to reason about the screen at a higher level of abstraction. It can answer internal queries like "Where is the 'Submit' button?", "Is the 'Save' icon currently active?", or "What is the value in the text field labeled 'Total Price?'".[57]

The final output of this perception stack is not a flat pixel image, but a rich, structured representation of the GUI—a scene graph or JSON object detailing every element, its properties, and its relationships. This structured data is what gets passed to the Cognitive Core's encoders, providing a much more informative input than a raw screenshot.

2.2.2. Action Subsystems

The agent "acts" within the digital world by simulating human input through software.

- **Low-Level Control:** This is achieved via operating system-level APIs that provide programmatic control over the mouse cursor (moving to coordinates, clicking, scrolling) and the keyboard (typing text, pressing individual or combination keys).[59] Libraries such as PyAutoGUI in Python offer a cross-platform foundation for these basic actions.
- **Automation Frameworks:** For more robust and targeted interaction, especially within web browsers or specific enterprise applications, higher-level automation frameworks are used. Tools like Selenium or Playwright allow the agent to interact with web elements via their underlying code (e.g., HTML DOM) rather than just visual coordinates, which is often more reliable.[61] The agent's Policy Translation Engine (detailed in Section 3) will be trained to decide whether to use low-level coordinate-based actions (necessary for custom applications or games) or higher-level framework-based actions when available.

The entire process of perception and action in the desktop environment is encapsulated by frameworks like **ScreenAgent**.[25] ScreenAgent provides a complete pipeline that includes a

plan-act-reflect loop, where the agent decomposes a task, executes a GUI action, observes the new screen state, and reflects on the outcome to decide the next step. This iterative loop is the digital equivalent of the physical perception-action loop and will be a core component of the desktop embodiment's control flow.

This dual-embodiment approach is unified by treating each perception and action capability as a "tool." The ROS architecture naturally supports this view for the robot, where each sensor and actuator is a distinct node.[39] This concept can be extended to the desktop, where the screenshot API is a perception tool, and the mouse and keyboard control APIs are action tools. By defining each embodiment as a registry of available tools, the Cognitive Core's task is simplified to selecting and orchestrating the right sequence of tools to achieve a goal, a paradigm that is inherently more scalable and generalizable than learning monolithic behaviors for each body.[53]

Table 2: Mapping of Sensory Modalities to Embodiment Environments

Abstract Modality	Robotic Embodiment Source	Desktop Embodiment Source	Data Format
Vision	Stereo Camera Feed, LiDAR Point Cloud	Desktop Screenshot, Application Window Capture	RGB Video Stream, 3D Point Cloud, PNG/JPEG Image
Audio	Microphone Array	System Audio Output, Microphone Input	WAV Audio Stream, Text Transcript
Spatial Layout	LiDAR-based 3D Map, Proprioception	GUI Element Bounding Boxes, DOM Tree	Occupancy Grid, Joint Angles, JSON/XML Scene Graph
User Input	Speech Recognition (Microphone)	Keyboard/Chat Interface, Mouse Clicks	Text String, Key Press Events, Click Coordinates
Haptic Feedback	Force-Torque Sensors, Tactile Skin	(N/A - a primary difference)	Force Vector, Pressure Map
Action Result	Change in Camera/LiDAR View	Change in Screenshot/GUI State	New Perceptual Data Stream

This table clarifies how abstract perceptual concepts are instantiated differently in each domain. "Vision" for the robot is a 3D, dynamic stream, while for the desktop it is a 2D, static image. This fundamental difference in data structure and semantics is precisely the challenge that the Cognitive-Embodiment Abstraction Layer is designed to solve.

Section 3: The Cognitive-Embodiment Abstraction Layer (CEAL): A Framework for Dynamic Instantiation

The most novel architectural component of this framework is the Cognitive-Embodiment Abstraction Layer (CEAL). This sophisticated middleware serves as the crucial interface between the agent's "mind" (the Cognitive Core) and its "body" (the active embodiment). Its primary function is to decouple high-level, abstract reasoning from low-level, platform-specific implementation details, thereby enabling the fluid, dynamic embodiment that is the central goal of this project.

3.1. Rationale and Analogy

Conceptually, the CEAL is analogous to a Hardware Abstraction Layer (HAL) in a modern operating system or the physics engine abstraction layer in a video game engine. A HAL provides a consistent, standardized API to software applications, allowing them to run on a wide variety of different hardware configurations without being rewritten for each one. The HAL is responsible for translating the application's generic requests (e.g., "write data to disk") into the specific, low-level commands required by the particular hard drive controller installed in the machine.
Similarly, the CEAL provides a consistent interface to the Cognitive Core, shielding it from the immense complexity and heterogeneity of its potential embodiments. The Cognitive Core formulates plans using abstract concepts, and the CEAL is responsible for translating those abstractions into the concrete sensor data and actuator commands relevant to the robot or the desktop GUI.[65] This modularity is the bedrock of the system's flexibility, scalability, and, most importantly, its ability to transfer learned knowledge across domains.

3.2. CEAL Architecture

The CEAL is architected as a modular interface composed of four key components. Together, these components manage the bidirectional flow of information in the perception-action loop, translating sensory feedback into a common language for the mind and translating the mind's intentions into specific actions for the body.[51]

3.2.1. Feedback Normalization Unit

This component constitutes the "input" pathway of the CEAL, responsible for processing all incoming perceptual data.

- **Function:** It receives raw, embodiment-specific perceptual data streams—such as a 3D point cloud from the robot's LiDAR, a force-feedback vector from its gripper, or the structured GUI representation from the desktop perception stack—and normalizes them into a standardized, abstract format that the Cognitive Core can ingest.
- **Process:** This unit takes the outputs from the various modality-specific encoders (as

defined by our hybrid Type-C architecture) and projects them into the unified, high-dimensional embedding space that the core MLLM is trained on. A critical function of this unit is to ensure representational consistency. For example, it must ensure that the abstract concept of a "cup" is represented by a similar vector in the embedding space, whether that concept originates from the robot's camera feed or from an image of a cup displayed on the desktop screen. This is achieved through techniques like contrastive learning during the foundational pre-training stage, which explicitly trains the encoders and the projection layers to map semantically similar inputs from different modalities to nearby points in the embedding space.[5]

3.2.2. Intent Abstraction Module

This component is the "output" interface for the Cognitive Core. To maintain embodiment-agnosticism, the MLLM and World Model do not generate low-level motor commands or pixel coordinates. Instead, they produce high-level, structured *intentions*.

- **Function:** It provides a formalized structure for the Cognitive Core's decisions. These intentions represent the "what" of a desired action, abstracting away the "how."
- **Format:** Intentions are formatted as structured objects, akin to an API call, containing the core action and its relevant parameters. For example:
    - {action: 'GRASP', target_id: 'red_cup__01', constraint: 'gentle'}
    - {action: 'NAVIGATE', target_id: 'Login_Button'}
    - {action: 'TYPE', content: 'hello world', target_id: 'text_field_username'}
      This approach, inspired by how LLMs can be trained to generate code or function calls, allows the Cognitive Core to focus on strategic, sequential planning without being burdened by the implementation details of each potential body.65

3.2.3. Embodiment Schema Registry

This component acts as the CEAL's dynamic knowledge base, maintaining a real-time, structured representation of the agent's currently available embodiment(s). It is the system's source of truth for what actions are possible at any given moment.

- **Function:** The registry is a dynamic database that is continuously updated to reflect the status of the agent's physical and digital bodies.
- **Content:** For each registered embodiment, the schema stores critical information:
    - **embodiment_id**: A unique identifier (e.g., 'Robot_Arm_7DoF', 'Desktop_Windows11').
    - **status**: The current state of the embodiment (e.g., 'idle', 'active', 'error', 'offline').
    - **available_tools**: A list of all perception and action tools currently available through that embodiment (e.g., ['camera', 'gripper', 'microphone'] for the robot, or ['screenshot_api', 'mouse_api', 'keyboard_api'] for the desktop).
    - **tool_signatures**: The specific parameters and formats that each tool accepts

(e.g., gripper.close(force: 0-100N, velocity: 0.1-1.0m/s)).

This registry is not static. If a robot's gripper malfunctions, its status in the registry changes to 'error', and the gripper tool becomes unavailable. This allows for graceful failure handling and dynamic adaptation. If a new USB camera is plugged into the desktop, a new perception tool can be registered, becoming immediately available for the agent to use without requiring a system restart. This makes the entire system highly adaptive and resilient, a core tenet of advanced embodied intelligence.[2]

3.2.4. Policy Translation Engine

This is the operational heart of the CEAL, responsible for the crucial step of grounding abstract thought into concrete action.

- **Function:** The Policy Translation Engine is a learned module that translates the abstract *intention* received from the Cognitive Core into a concrete, executable sequence of tool calls for a specific embodiment. It uses the Embodiment Schema Registry to understand the available tools and their parameters.
- **Implementation:** This engine is itself a sophisticated multi-task policy, likely a transformer-based model trained using a combination of imitation learning (from human demonstrations) and reinforcement learning. It learns the complex mapping: f(intention, embodiment_schema) -> action_sequence.
    - For example, it learns that the intention {action: 'GRASP', target_id: 'red_cup_01'} for the robotic embodiment translates into a complex sequence of robot_arm.move_to(coordinates), robot_arm.orient_wrist(orientation), and gripper.close(force) commands.
    - Conversely, it learns that the intention {action: 'NAVIGATE', target_id: 'Login_Button'} for the desktop embodiment translates into a mouse_api.move_to(coordinates) command followed by a mouse_api.click() command.
      State-of-the-art Vision-Language-Action (VLA) models like Microsoft's Magma, which are pre-trained to generate action proposals from visual and language inputs, provide a strong technical precedent for the feasibility of such a translation engine.11

The CEAL's architecture enables a powerful capability: **Cross-Embodiment Skill Transfer**. Because the Cognitive Core operates on abstract intentions, it can learn the high-level structure of a task in one embodiment and potentially apply that abstract knowledge to perform a similar task in the other. Consider the high-level task "clear the workspace." On the desktop, this might involve the agent generating an abstract plan like -> ->. In the physical world, the same high-level task might result in the plan -> ->. The Cognitive Core learns the abstract, syntactic structure of the plan: IDENTIFY -> ACQUIRE -> RELOCATE. If the agent has only ever performed this task on the desktop, it has still learned this abstract plan. When faced with the task in the real world for the first time, it can propose the same abstract plan. The Policy Translation Engine, which has been separately trained on basic robotic skills, can

then translate this abstract plan into concrete robot actions. This constitutes a form of zero-shot or few-shot task transfer across embodiments, a powerful emergent capability that arises directly from the CEAL's decoupling of "what" from "how".[50]

Table 3: Abstract Intentions vs. Embodiment-Specific Commands

High-Level Intention	Robotic Embodiment Command Sequence (Pseudo-code)	Desktop Embodiment Command Sequence (Pseudo-code)
``	plan = motion_planner.plan_path(target='shelf_A') mobile_base.execute_path(plan) arm.move_to(item_location) gripper.grasp(item='report.pdf') arm.move_to(delivery_location)	file_explorer.open() file_explorer.search(query='report.pdf') mouse.right_click(target='report.pdf') mouse.select_option('Copy')
``	arm.move_to(tool_rack) gripper.grasp(item='screwdriver') arm.orient_wrist(orientation='use_screwdriver')	mouse.move_to(target='start_menu') mouse.click() keyboard.type('Photoshop') keyboard.press('Enter')
``	speaker.play_audio(file='status_ok.wav') led_strip.set_color('green')	chat_window.focus() keyboard.type('Task complete. All systems normal.') keyboard.press('Enter')
``	camera.set_pan_tilt(pan=0, tilt=-45) vision_system.find_object(color='red', shape='round') # returns coordinates if found	gui_parser.analyze_screenshot() gui_parser.find_element(label='red', type='button') # returns coordinates and properties if found

This table makes the function of the CEAL concrete. It shows the one-to-many mapping from a single, abstract intention generated by the Cognitive Core to the diverse, platform-specific command sequences executed by the Policy Translation Engine. This translation is the mechanism that allows a single "thought" to manifest as either a physical or a digital action, directly fulfilling the user's core requirement for a unified, dually-embodied agent.

Section 4: Arbitration and Concurrent Control

With a unified Cognitive Core and two distinct embodiments connected by the CEAL, the final architectural challenge is to manage these resources effectively. When a complex task is presented and both the robot and the desktop are available, the system requires a sophisticated mechanism for **arbitration**: deciding which embodiment is best suited for a given sub-task, whether they should operate in parallel, or if one should assist the other.[69] The architecture must be designed to support switched, concurrent, and even collaborative control paradigms to unlock the full potential of its dual embodiment.[65]

4.1. The Arbitration Challenge

The problem of arbitration is one of control allocation. A simple, rule-based system (e.g., "use the desktop for file tasks, use the robot for physical tasks") would be brittle and fail to handle tasks that bridge both domains, such as "Find the recipe for this dish online, then gather the ingredients from the pantry." A more intelligent and dynamic approach is required. This problem is analogous to those studied in multi-robot systems, where coordination is often framed as a decentralized control problem aimed at optimizing a collective objective.[71]

4.2. Proposed Solution: Dynamic Attention for Arbitration

Attention mechanisms, which are fundamental to modern transformer architectures, provide a powerful tool for dynamically weighing the importance of different information sources.[74] In particular,
cross-attention allows one stream of information (the query) to selectively attend to another (the key/value pairs), calculating relevance scores to create a contextually weighted representation.[13] This mechanism can be repurposed from its typical role in modality fusion to serve as the core of a dynamic arbitration module.
We propose an **Arbitration Module** located within the Cognitive Core that utilizes a dedicated cross-attention layer to perform control allocation.

- **Query:** The current high-level task goal or sub-task intention (e.g., ``) generated by the MLLM planner is encoded to form the query vector. This vector represents "what needs to be done."
- **Keys and Values:** The available_tools and current status for each active embodiment, as listed in the Embodiment Schema Registry, are encoded to form the key and value vectors. These vectors represent "what tools are available to do it."
- **Arbitration via Attention:** The cross-attention mechanism computes attention scores by comparing the task query to the embodiment keys. These scores represent the "relevance" or "utility" of each embodiment's toolset for the current sub-task. For the intention , the desktop embodiment's tools (`file_explorer.search`, `mouse.click`) would receive a high attention score, while the robot's tools (`gripper.grasp`) would receive a low score. The system would thus allocate control to the desktop. Conversely, an

intention like would cause attention to focus squarely on the robot.
This arbitration process is not a static, one-time decision. It is re-evaluated dynamically for each step in the MLLM's generated plan. This allows for fluid switching between embodiments as a task progresses.[78] For example, after the desktop finds the assembly manual, the next sub-task in the plan might be ``, which would cause the attention mechanism to shift control to the robot. This transforms the arbiter from a simple switch into a dynamic resource manager, continuously optimizing the allocation of its physical and digital actuators to achieve the overall goal most efficiently.

4.3. Control Paradigms

This architecture supports three increasingly sophisticated modes of control.

4.3.1. Switched Control

This is the default and most fundamental operational mode. The Arbitration Module assigns a single embodiment as "active" for a given sub-task based on the attention scores. The CEAL then routes the Cognitive Core's intention exclusively to that embodiment's Policy Translation Engine for execution. This allows the agent to seamlessly switch between controlling the desktop and the robot to complete a sequential task.

4.3.2. Concurrent Control

For tasks that are decomposable into independent sub-goals, the system can operate its embodiments in parallel. For example, given the command "Sort these red blocks into the red bin and delete all temporary files on my desktop," the MLLM planner can generate two independent sub-plans. The Arbitration Module can assign one plan to the robot and the other to the desktop. The CEAL then manages two separate perception-action loops concurrently, one for each embodiment.
Enabling this capability introduces challenges from the field of multi-agent systems.[70] When both embodiments act simultaneously, the state of the world is being changed by two independent actors. From the robot's perspective, the desktop's actions are part of a non-stationary environment, and vice-versa. Therefore, training for concurrent control cannot use simple single-agent reinforcement learning. It requires the application of **Multi-Agent Reinforcement Learning (MARL)** techniques. Frameworks from Multi-Task Reinforcement Learning (MTRL) are directly applicable, where the two embodiments can be treated as two "tasks" being learned simultaneously by a single, shared policy network (the Cognitive Core).[28] The training process must be designed to handle this concurrency, for example by using a centralized training scheme (where a central critic evaluates the joint

action of both embodiments) with decentralized execution (where each embodiment acts based on its local perception and the shared policy).

4.3.3. Collaborative Control

This is the most advanced and powerful mode of operation, where the embodiments work together in a tightly coupled manner. In this paradigm, the actions of one embodiment provide the perceptual input for the other, creating a **cross-embodiment perception-action loop**. Consider the complex instruction: "Find the video tutorial for installing this graphics card online and guide my arm to plug it in correctly."

1. **Arbitration:** The initial sub-task, "find the video tutorial," is assigned to the Desktop embodiment.
2. **Action (Desktop):** The agent navigates the web, finds the video, and begins playing it.
3. **Cross-Embodiment Perception:** The screen content—the video showing how to install the card—now becomes a primary perceptual input for the *entire system*. This visual stream is processed by the desktop's perception stack and fed into the Cognitive Core's unified embedding space.
4. **Reasoning (Cognitive Core):** The MLLM now reasons based on a combination of the original user command, its own knowledge, the robot's camera view of the physical computer case, and the visual instructions from the desktop's screen.
5. **Action (Robot):** Based on this fused, cross-embodiment understanding, the Cognitive Core generates intentions for the Robot arm (e.g., [ALIGN, 'card_connector', 'motherboard_slot']), which are translated and executed by the robotic embodiment.

This creates a feedback loop where the digital world directly guides action in the physical world, mediated by the agent's unified cognitive process. Achieving this level of collaboration represents a significant step towards truly general-purpose, helpful embodied agents.

Section 5: Implementation Roadmap and Future Directions

The framework detailed in this report is ambitious, representing a significant research and development effort. Its realization requires a strategic, phased approach that builds foundational capabilities before tackling the more complex aspects of the system. This section outlines a practical development roadmap and acknowledges the key open research challenges that must be addressed.

5.1. Phased Development Plan

A multi-year, four-phase plan is proposed to manage the complexity of the project.

- **Phase 1: The Core and The Bodies (Years 1-2):** This phase focuses on developing the fundamental components in parallel.
    - **Cognitive Core:** Develop and pre-train the hybrid Type-C/D MLLM and the parallel World Model. This involves curating the massive datasets required for Stage 1 pre-training and establishing the computational infrastructure.
    - **Embodiments:** In a separate track, develop the two embodiment platforms. For the robot, this involves integrating the sensor suite and actuators with ROS 2. For the desktop, it involves building the GUI perception stack (element detection, OCR, VQA) and the input simulation action stack. At the end of this phase, there will be a powerful but ungrounded AI and two fully instrumented but non-intelligent platforms.
- **Phase 2: The Connection (Year 3):** This phase focuses on integrating the mind and bodies via the CEAL.
    - **CEAL Development:** Architect and implement the four core components of the CEAL: the Feedback Normalization Unit, Intent Abstraction Module, Embodiment Schema Registry, and Policy Translation Engine.
    - **Instruction Tuning:** Curate the cross-domain instruction-following dataset and perform Stage 2 SFT. The primary goal is to train the Policy Translation Engine to reliably translate abstract intentions into correct, single-embodiment action sequences.
    - **Goal:** The milestone for this phase is achieving robust **switched control**. The agent should be able to reliably complete complex, sequential tasks that require it to alternate between the desktop and the robot, but not use them at the same time.
- **Phase 3: The Collaboration (Year 4):** This phase focuses on advanced control paradigms.
    - **Arbitration Module:** Implement the attention-based Arbitration Module within the Cognitive Core.
    - **MARL Training:** Develop the multi-agent reinforcement learning training pipeline. This involves designing reward functions for concurrent and collaborative tasks and implementing a MARL algorithm (e.g., a multi-agent variant of PPO or SAC).
    - **Goal:** The milestone for this phase is demonstrating robust **concurrent and collaborative control**. The agent should be able to successfully execute tasks that require both embodiments to act in parallel or in a tightly coupled, cross-perceptual loop.
- **Phase 4: The Evolution (Year 5 and beyond):** This phase shifts focus from initial capability development to long-term autonomy and improvement.
    - **Lifelong Learning:** Implement mechanisms for lifelong and continual learning, allowing the agent to continuously update its knowledge and skills from its ongoing interactions with the world.[82] This requires tackling the challenge of catastrophic forgetting, potentially through techniques like memory replay or dynamic network expansion.
    - **Alignment and Safety:** Continuously perform Stage 3 alignment tuning

(RLHF/DPO) with new data to ensure the agent's behavior remains safe, reliable, and aligned with human values as it evolves.

5.2. Key Research Challenges

This project exists at the frontier of AI research, and several significant challenges must be overcome.

- **Cross-Embodiment Generalization:** While the architecture is designed to facilitate skill transfer, the true extent to which high-level plans learned in a digital environment can generalize to the noisy, unpredictable physical world remains a major open research question.[82] The gap between the "physics" of a GUI and the physics of the real world is substantial.
- **Data Scarcity for Instruction Tuning:** The success of Phase 2 hinges on the availability of a large, high-quality, cross-domain instruction-following dataset. Creating this dataset will be a monumental undertaking, as off-the-shelf datasets with the required breadth and structure do not currently exist.[66]
- **Computational Cost:** The proposed system is exceptionally demanding. It involves running multiple large models (MLLM, World Model, Policy Translation Engine) and processing multiple real-time sensor streams concurrently. The computational and memory requirements will necessitate a distributed computing architecture and significant hardware investment.[27]
- **Physical and Digital Safety:** An autonomous agent with the power to manipulate both a physical robot and a user's desktop presents significant safety risks. Ensuring that the agent cannot be prompted or tricked into causing physical harm, deleting critical files, or leaking private information is a paramount concern. While alignment tuning is the primary defense, additional architectural safeguards, such as hard-coded constraints within the CEAL and a human-in-the-loop confirmation step for potentially destructive actions, will be essential.[20]

5.3. The Path to Embodied AGI

This framework is not merely an engineering proposal; it is a structured research program aimed at advancing the state of the art in embodied, generalist intelligence.[1] Its progress can be benchmarked against systematic taxonomies of Embodied AGI, such as the proposed five-level (L1-L5) roadmap which evaluates agents on dimensions of modality, cognition, responsiveness, and generalization.[87]

- A successful **Phase 2** implementation, demonstrating robust switched control across a wide variety of tasks, would result in an agent that meets the criteria for **L3 Embodied AGI (Conditional General-Purpose Task Completion)**. Such an agent could handle a diverse range of task categories across both physical and digital domains, adapting

dynamically to instructions, but would still struggle with entirely novel, open-ended tasks.[87]

- A successful **Phase 3** implementation, demonstrating reliable concurrent and collaborative control, would be pushing the boundaries toward **L4 Embodied AGI (Highly General-Purpose Robots)**. This level of capability requires the agent to have a deeply internalized model of the world and its own affordances, enabling it to reason about how its different "bodies" can be orchestrated to achieve complex goals with near-human accuracy and minimal intervention.[88]

Conclusion

The framework presented in this report outlines a coherent and comprehensive architecture for a single, multimodal AI agent capable of dynamic embodiment in both robotic and desktop environments. The core innovations of this framework—the hybrid **Cognitive Core** combining a reasoning MLLM with a predictive World Model, and the **Cognitive-Embodiment Abstraction Layer (CEAL)** that decouples mind from body—are designed to directly address the fundamental challenges of cross-domain generalization and control.

By leveraging a modular, tool-based approach to perception and action, and by employing an attention-based mechanism for dynamic arbitration, the proposed system is architected for flexibility, scalability, and emergent intelligence. The ability to perform switched, concurrent, and truly collaborative control across physical and digital realms represents a qualitative leap beyond current single-domain agents. The phased implementation plan provides a practical roadmap for development, while acknowledging the significant research hurdles that lie ahead, particularly in data curation, safety, and proving the hypothesis of cross-embodiment skill transfer.

Ultimately, this framework is more than a design for a single product; it is a research program aimed at tackling the core scientific questions of what it means to build a generalist intelligence. By grounding AI in the rich, interactive, and multimodal reality of both our physical and digital worlds, this work charts a deliberate course toward more capable, adaptable, and truly embodied artificial intelligence.

Works cited

1. A Comprehensive Survey on Embodied Intelligence: Advancements, Challenges, and Future Perspectives - SciOpen, accessed July 3, 2025, https://www.sciopen.com/article/10.26599/AIR.2024.9150042
2. Embodied Intelligence: The Key to Unblocking Generalized Artificial Intelligence - arXiv, accessed July 3, 2025, https://arxiv.org/html/2505.06897v1
3. The Evolution of Multimodal Model Architectures - arXiv, accessed July 3, 2025, https://arxiv.org/html/2405.17927v1
4. Demystifying Multi-modal AI - IEEE Computer Society, accessed July 3, 2025, https://www.computer.org/publications/tech-news/trends/demystifying-multi-mo

dal-ai
5. What are the latest advancements in multimodal AI? - Milvus, accessed July 3, 2025, https://milvus.io/ai-quick-reference/what-are-the-latest-advancements-in-multimodal-ai
6. Recent advancements in multimodal human–robot interaction - PMC - PubMed Central, accessed July 3, 2025, https://pmc.ncbi.nlm.nih.gov/articles/PMC10210148/
7. The Evolution of Multimodal Model Architectures - ResearchGate, accessed July 3, 2025, https://www.researchgate.net/publication/380935647_The_Evolution_of_Multimodal_Model_Architectures
8. The Evolution of Multimodal Model Architectures - Emergent Mind, accessed July 3, 2025, https://www.emergentmind.com/papers/2405.17927
9. survey on multimodal large language models | National Science ..., accessed July 3, 2025, https://academic.oup.com/nsr/article/11/12/nwae403/7896414
10. The Revolution of Multimodal Large Language ... - ACL Anthology, accessed July 3, 2025, https://aclanthology.org/2024.findings-acl.807.pdf
11. Magma: A foundation model for multimodal AI agents across digital ..., accessed July 3, 2025, https://www.microsoft.com/en-us/research/blog/magma-a-foundation-model-for-multimodal-ai-agents-across-digital-and-physical-worlds/
12. OmniParser: Microsoft's Breakthrough in AI-Powered UI Interaction | by Malyaj Mishra | Data Science in Your Pocket | Medium, accessed July 3, 2025, https://medium.com/data-science-in-your-pocket/omniparser-microsofts-breakthrough-in-ai-powered-ui-interaction-08c7c2cc28d5
13. How Multimodal Learning is Used in Generative AI - DigitalOcean, accessed July 3, 2025, https://www.digitalocean.com/community/tutorials/multimodal-learning-generative-ai
14. The (R)Evolution of Multimodal Large Language Models: A Survey - arXiv, accessed July 3, 2025, https://arxiv.org/html/2402.12451v1
15. [Discussion] What exactly are World Models in AI? What problems do they solve, and where are they going? : r/MachineLearning - Reddit, accessed July 3, 2025, https://www.reddit.com/r/MachineLearning/comments/1kf3pes/discussion_what_exactly_are_world_models_in_ai/
16. arxiv.org, accessed July 3, 2025, https://arxiv.org/html/2506.22355v1#:~:text=Reasoning%20and%20planning%3A%20A%20world,and%20execute%20tasks%20more%20effectively.
17. Topic 35: What are World Models?, accessed July 3, 2025, https://www.turingpost.com/p/topic-35-what-are-world-models
18. [2506.22355] Embodied AI Agents: Modeling the World - arXiv, accessed July 3, 2025, https://arxiv.org/abs/2506.22355
19. Embodied AI Agents: Modeling the World - arXiv, accessed July 3, 2025, https://arxiv.org/html/2506.22355v1

20. The Revolution of Multimodal Large Language Models: A Survey - ResearchGate, accessed July 3, 2025, https://www.researchgate.net/publication/384217973_The_Revolution_of_Multimodal_Large_Language_Models_A_Survey
21. Learning 4D Embodied World Models - OpenReview, accessed July 3, 2025, https://openreview.net/forum?id=mnwlhvmKMN
22. [2402.12451] The Revolution of Multimodal Large Language Models: A Survey - arXiv, accessed July 3, 2025, https://arxiv.org/abs/2402.12451
23. A Survey on Multimodal Large Language Models - OpenReview, accessed July 3, 2025, https://openreview.net/pdf?id=2iwozOs6YB
24. Magma: Microsoft Research's new foundation model for multimodal AI agents - Wandb, accessed July 3, 2025, https://wandb.ai/byyoung3/ml-news/reports/Magma-Microsoft-Research-s-new-foundation-model-for-multimodal-AI-agents---VmlldzoxMTQ0MDU5Ng
25. ScreenAgent: A Vision Language Model-driven Computer ... - IJCAI, accessed July 3, 2025, https://www.ijcai.org/proceedings/2024/0711.pdf
26. [2402.07945] ScreenAgent: A Vision Language Model-driven Computer Control Agent, accessed July 3, 2025, https://arxiv.org/abs/2402.07945
27. arXiv:2403.04866v1 [cs.AI] 7 Mar 2024, accessed July 3, 2025, https://arxiv.org/pdf/2403.04866
28. Efficient Multi-Task Reinforcement Learning with Cross-Task Policy Guidance - NIPS, accessed July 3, 2025, https://proceedings.neurips.cc/paper_files/paper/2024/file/d5cd70b708f726737e2ebace18c3f71b-Paper-Conference.pdf
29. Projected Task-Specific Layers for Multi-Task Reinforcement Learning, accessed July 3, 2025, https://arxiv.org/pdf/2309.08776
30. Multimodal AI in Robotics: Simplifying Automation Complexity, accessed July 3, 2025, https://www.akira.ai/blog/multimodal-ai-in-robotics
31. The Future of AI: Multimodal Models Leading the Way - Rapid Innovation, accessed July 3, 2025, https://www.rapidinnovation.io/post/the-future-of-ai-how-multimodal-models-are-leading-the-way
32. How is multimodal AI used in robotics? - Milvus, accessed July 3, 2025, https://milvus.io/ai-quick-reference/how-is-multimodal-ai-used-in-robotics
33. What Are Real-Time Control Systems In Robotics? Key Insights - Indmall Automation, accessed July 3, 2025, https://www.indmallautomation.com/faq/what-are-real-time-control-systems-in-robotics/
34. How do robots handle real-time sensor data processing? - Zilliz ..., accessed July 3, 2025, https://zilliz.com/ai-faq/how-do-robots-handle-realtime-sensor-data-processing
35. How do robots handle real-time sensor data processing? - Milvus, accessed July 3, 2025, https://milvus.io/ai-quick-reference/how-do-robots-handle-realtime-sensor-data-processing

36. How do robots process real-time sensor data for adaptive behaviors? - Milvus, accessed July 3, 2025, https://milvus.io/ai-quick-reference/how-do-robots-process-realtime-sensor-data-for-adaptive-behaviors
37. Top Robotics APIs for Developers - IndustryWired, accessed July 3, 2025, https://industrywired.com/top-robotics-apis-for-developers/
38. APIs for Robotic Programming - HEBI Robotics, accessed July 3, 2025, https://www.hebirobotics.com/apis
39. ROS Architecture and Concepts - Packt, accessed July 3, 2025, https://www.packtpub.com/en-us/learning/how-to-tutorials/ros-architecture-and-concepts
40. Mastering Robot Operating System (ROS) - Number Analytics, accessed July 3, 2025, https://www.numberanalytics.com/blog/mastering-robot-operating-system-ros
41. Robot Operating System - ROS - RoboPI, accessed July 3, 2025, https://robopi.ece.ufl.edu/files/AuRu_Materials/Lecture_2.pdf
42. Intro to ROS — ROS Tutorials 0.5.2 documentation, accessed July 3, 2025, https://www.clearpathrobotics.com/assets/guides/melodic/ros/Intro%20to%20the%20Robot%20Operating%20System.html
43. The Ultimate Guide to Robotics APIs - Number Analytics, accessed July 3, 2025, https://www.numberanalytics.com/blog/ultimate-guide-robotics-apis
44. Mastering APIs for Robotics - Number Analytics, accessed July 3, 2025, https://www.numberanalytics.com/blog/mastering-apis-robotics-innovation
45. The Ultimate Guide to Sim-to-Real Transfer - Number Analytics, accessed July 3, 2025, https://www.numberanalytics.com/blog/ultimate-guide-sim-to-real-transfer
46. What exactly makes sim to real transfer a challenge in reinforcement learning? : r/robotics, accessed July 3, 2025, https://www.reddit.com/r/robotics/comments/1j99vrt/what_exactly_makes_sim_to_real_transfer_a/
47. Virtual to Real-World Transfer Learning: A Systematic Review - MDPI, accessed July 3, 2025, https://www.mdpi.com/2079-9292/10/12/1491
48. Using Transfer Learning to solve the simulator-to-real problem in the Duckietown environment | by SmartLab AI, accessed July 3, 2025, https://smartlabai.medium.com/using-transfer-learning-to-solve-the-simulator-to-real-problem-in-the-duckietown-environment-9f66c80db2ce
49. Toward Generalized Sim-to-Real Transfer for Robot Learning - Google Research, accessed July 3, 2025, https://research.google/blog/toward-generalized-sim-to-real-transfer-for-robot-learning/
50. Transfer Learning in Robotics: An Upcoming Breakthrough? A Review of Promises and Challenges - arXiv, accessed July 3, 2025, https://arxiv.org/html/2311.18044v2
51. Embodied AI: Giving Intelligence a Physical Presence | by Anirudh ..., accessed July 3, 2025, https://medium.com/@anirudhsekar2008/embodied-ai-giving-intelligence-a-phy

sical-presence-c7a584e25cd4
52. Embodied Intelligence: Grounding AI in the Physical World for Enhanced Capability and Adaptability - Alphanome.AI, accessed July 3, 2025, https://www.alphanome.ai/post/embodied-intelligence-grounding-ai-in-the-physical-world-for-enhanced-capability-and-adaptability
53. Computer Use and AI Agents: A New Paradigm for Screen ... - Medium, accessed July 3, 2025, https://medium.com/data-science/computer-use-and-ai-agents-a-new-paradigm-for-screen-interaction-b2dcbea0df5b
54. Activities - Computer Vision activities - UiPath Documentation, accessed July 3, 2025, https://docs.uipath.com/activities/other/latest/ui-automation/computer-vision-activities
55. How to Automate UI Testing with Visual Verification - Keysight, accessed July 3, 2025, https://www.keysight.com/us/en/solutions/automate-ui-testing-with-visual-verification.html
56. AI Computer Vision - Introduction - UiPath Documentation, accessed July 3, 2025, https://docs.uipath.com/ai-computer-vision/automation-cloud-public-sector/latest/user-guide/introduction
57. How AI Agents is Redefining Visual Question Answering in Real-Time, accessed July 3, 2025, https://www.akira.ai/blog/ai-agents-in-visual-question-answering
58. How Visual AI Agents Can Be a Game-Changer for Your Mobile App - Zco Corporation, accessed July 3, 2025, https://www.zco.com/blog/visual-ai-agents/
59. Comprehensive Guide to User Input Simulation on Any Device - Adapta Robotics, accessed July 3, 2025, https://www.adaptarobotics.com/blog/comprehensive-guide-to-user-input-simulation-on-any-device/
60. How to simulate user interaction in WinUI3 desktop application - Microsoft Q&A, accessed July 3, 2025, https://learn.microsoft.com/en-gb/answers/questions/2180042/how-to-simulate-user-interaction-in-winui3-desktop
61. Automate Form Filling without code - Axiom.ai, accessed July 3, 2025, https://axiom.ai/automate/form-filling
62. Top Free Automation Tools for Testing Desktop Applications (2025) - Test Guild, accessed July 3, 2025, https://testguild.com/automation-tools-desktop/
63. Introduction to ROS | CS-STEM Network, accessed July 3, 2025, https://www.cs2n.org/u/mp/badge_pages/3574
64. Large Model Agents: State-of-the-Art, Cooperation Paradigms, Security and Privacy, and Future Trends - arXiv, accessed July 3, 2025, https://arxiv.org/html/2409.14457v1
65. Embodied AI with Two Arms: Zero-shot Learning, Safety and Modularity - arXiv, accessed July 3, 2025, https://arxiv.org/html/2404.03570v3
66. Building Computing Systems for Embodied Artificial Intelligence -

Communications of the ACM, accessed July 3, 2025, https://cacm.acm.org/blogcacm/building-computing-systems-for-embodied-artificial-intelligence/
67. Unlocking Embodied Cognition - Number Analytics, accessed July 3, 2025, https://www.numberanalytics.com/blog/ultimate-guide-action-perception-cycle
68. Magma: A Foundation Model for Multimodal AI Agents - arXiv, accessed July 3, 2025, https://arxiv.org/html/2502.13130v1
69. [2003.05097] A General Arbitration Model for Robust Human-Robot Shared Control with Multi-Source Uncertainty Modeling - arXiv, accessed July 3, 2025, https://arxiv.org/abs/2003.05097
70. (PDF) Multi-agent Embodied AI: Advances and Future Directions - ResearchGate, accessed July 3, 2025, https://www.researchgate.net/publication/391575549_Multi-agent_Embodied_AI_Advances_and_Future_Directions
71. The box-pushing robot's control architecture. Behavior arbitration is handled using a xed priority sub- sumption network. - ResearchGate, accessed July 3, 2025, https://www.researchgate.net/figure/The-box-pushing-robots-control-architecture-Behavior-arbitration-is-handled-using-a-xed_fig2_2440067
72. Coordinated Control of Multi-Robot Systems: A Survey - University of California San Diego, accessed July 3, 2025, http://terrano.ucsd.edu/jorge/publications/data/2017_CoEg-jcmsi.pdf
73. [2203.12416] A Framework for Controlling Multi-Robot Systems Using Bayesian Optimization and Linear Combination of Vectors - arXiv, accessed July 3, 2025, https://arxiv.org/abs/2203.12416
74. How do attention mechanisms work in multimodal AI models? - Milvus, accessed July 3, 2025, https://milvus.io/ai-quick-reference/how-do-attention-mechanisms-work-in-multimodal-ai-models
75. Multi-Modal AI Models: Expand AI Capabilities | Ultralytics, accessed July 3, 2025, https://www.ultralytics.com/blog/multi-modal-models-and-multi-modal-learning-expanding-ais-capabilities
76. A multimodal educational robots driven via dynamic attention - Frontiers, accessed July 3, 2025, https://www.frontiersin.org/journals/neurorobotics/articles/10.3389/fnbot.2024.1453061/full
77. Why Cross-Attention is the Secret Sauce of Multimodal Models | by Jakub Strawa | Medium, accessed July 3, 2025, https://medium.com/@jakubstrawadev/why-cross-attention-is-the-secret-sauce-of-multimodal-models-f8ec77fc089b
78. Knowledge-Guided Dynamic Modality Attention Fusion Framework for Multimodal Sentiment Analysis - ACL Anthology, accessed July 3, 2025, https://aclanthology.org/2024.findings-emnlp.865.pdf
79. Multi-agent Embodied AI: Advances and Future Directions - arXiv, accessed July 3, 2025, https://arxiv.org/html/2505.05108v1

80. Multi-Task Reinforcement Learning for Quadrotors - Robotics and Perception Group, accessed July 3, 2025, https://rpg.ifi.uzh.ch/docs/RAL25_Xing.pdf
81. [2412.12442] Multi-Task Reinforcement Learning for Quadrotors - arXiv, accessed July 3, 2025, https://arxiv.org/abs/2412.12442
82. [Literature Review] Embodied AI with Foundation Models for Mobile Service Robots: A Systematic Review - Moonlight | AI Colleague for Research Papers, accessed July 3, 2025, https://www.themoonlight.io/en/review/embodied-ai-with-foundation-models-for-mobile-service-robots-a-systematic-review
83. [2506.24019] Ella: Embodied Social Agents with Lifelong Memory - arXiv, accessed July 3, 2025, https://arxiv.org/abs/2506.24019
84. Survey of Large Multimodal Model Datasets, Application Categories and Taxonomy - arXiv, accessed July 3, 2025, https://arxiv.org/abs/2412.17759
85. Embodied Artificial Intelligence: Advancing the Frontiers of Robot Sensing and Interaction, accessed July 3, 2025, https://www.frontiersin.org/research-topics/67109/embodied-artificial-intelligence-advancing-the-frontiers-of-robot-sensing-and-interaction
86. Exploring Embodied Intelligence in Soft Robotics: A Review - MDPI, accessed July 3, 2025, https://www.mdpi.com/2313-7673/9/4/248
87. Toward Embodied AGI: A Review of Embodied AI and the Road Ahead - arXiv, accessed July 3, 2025, https://arxiv.org/html/2505.14235v1
88. [Literature Review] Toward Embodied AGI: A Review of Embodied AI and the Road Ahead, accessed July 3, 2025, https://www.themoonlight.io/en/review/toward-embodied-agi-a-review-of-embodied-ai-and-the-road-ahead