

From Synthetic Worlds to Situated Action: Multi-Agent Grounding for Virtual Reality Training

Mike Kentros[†], Manos Kamarianakis, George Papagiannakis
University of Crete
FORTH – Institute of Computer Science
ORamaVR SA

Abstract—Virtual reality (VR) training systems already encode rich procedural, spatial, and performance state, yet conversational assistants commonly treat a session as an ordinary chat context. We present MAGES-ASSISTANT, an embodied multi-agent assistant for MAGES-based VR training that routes each trainee request to a specialized agent and grounds the answer in a compact runtime payload from the active simulation. This paper focuses on the assistant’s spatial layer, which is built on 3D Conformal Geometric Algebra (CGA). Using GA-Unity, user anchors and task objects are serialized as $\mathbb{R}^{4,1}$ conformal points, proximity spheres, and pose motors; spatial relations—proximity and user-relative front/behind/left/right—are computed as CGA distance, sphere-intersection, and oriented-plane predicates; and assistive object guidance is formulated as motor interpolation and dispatched as validated, payload-grounded transforms that Unity executes only when they match the runtime state. The same algebra represents object identity, proximity, relations, and pose, giving a single geometric substrate that bridges situated language and validated geometric action. We describe the representation and the action model, and evaluate the layer’s real-time feasibility on GA-Unity and the effect of its spatial grounding on situated question answering.

Index Terms—conformal geometric algebra, virtual reality training, embodied agents, motor interpolation, multi-agent systems, retrieval-augmented generation, MAGES, GA-Unity

I. INTRODUCTION

Virtual reality (VR) is the fully immersive end of the broader extended reality (XR) spectrum, which also spans augmented and mixed reality. Our contribution—grounding an assistant in the running state of a situated procedure—depends on the structured simulation rather than on the display device, so the system runs across this spectrum: it is realized in both VR and wider XR settings, and every example in this paper is shown in XR. We accordingly use XR as the general term throughout; VR appears in the title and abstract, naming the fully immersive instance from which the work originates.

XR training differs from conventional digital learning because the trainee is situated inside a procedure rather than merely reading instructions or watching a demonstration. In a MAGES scenario the learner interacts with simulated anatomy, tools, prompts, constraints, timing, and feedback, and MAGES already represents much of this structure through scene graphs, gamified actions, interaction events, analytics, and authored content [1]. Immersive-training research shows that such simulation can transfer procedural skill and that presence, agency,

and cognitive load are central design factors [2]–[4]. A useful conversational assistant for this setting should therefore answer from the active state of the simulation rather than from the utterance alone: “What now?” depends on the current and next actions, “How do I do this?” on the active action type and interaction cue, “What am I touching?” on hand-interaction state, and “How did I do?” on accumulated analytics.

We treat the XR simulation as a structured synthetic world that is converted into agent-specific context. Rather than asking a single model to infer everything from an utterance, MAGES-ASSISTANT collects relevant runtime state from MAGES, organizes it into compact typed payloads, routes the request to a specialized agent, and lets that agent answer from the correct slice of the simulation. Recent work has explored generative-AI assistants in immersive training, open XR virtual-human platforms, and RAG-based XR question answering [5]–[8]; our emphasis is runtime grounding and, in particular, a geometric-algebra spatial layer that grounds both situated spatial dialogue and assistive geometric action.

This work concentrates on that spatial layer. Its contributions are: (i) a conformal-geometric-algebra representation that serializes user and object state as $\mathbb{R}^{4,1}$ conformal points, proximity spheres, and pose motors through GA-Unity [9]; (ii) CGA relation predicates for proximity and user-relative direction that remain traceable to the symbolic scene; (iii) a motor-based assistive-action layer in which the rigid correction is a delta motor realized by versor interpolation, dispatched through a command envelope that constrains execution to validated, payload-grounded transforms; and (iv) an evaluation of the layer’s real-time feasibility on GA-Unity and of how its spatial grounding affects situated spatial question answering. The surrounding multi-agent assistant—embodiment, router, and procedural, analytics, and retrieval grounding—is summarized as the context in which this spatial layer operates.

II. RELATED WORK

A. Situated XR Assistance and Embodied XR Agents

Medical and procedural XR training places the learner inside an interactive task rather than beside a static page. MAGES 4.0 provides the authoring, networking, analytics, and replay substrate used here, including scene-graph structure and assessment instrumentation for medical XR scenarios [1], and surgical-training and XR-education evidence motivates treating timing, feedback, interaction state, and workload as

[†]: Corresponding Author, mike.kentros@oramavr.com

first-class variables [2]–[4]. Recent XR systems embed generative agents in the environment: Chheang et al. evaluate a generative-AI assistant for immersive anatomy education [5], Hemminki-Reijonen et al. study generative-AI-powered pedagogy for XR [6], and Shoa and Friedman release Milo, an open LLM virtual-human platform for XR [7]. Retrieval-augmented generation [10], [11] and agentic prompting [12] provide standard grounding mechanisms, and RAG-VR adapts retrieval to localized VR scenes [8]. MAGES-ASSISTANT differs by grounding answers in structured runtime state and, for spatial questions, in a geometric-algebra layer rather than in language alone.

B. Geometric Algebra for Graphics, XR, and AI

Geometric algebra (GA) and conformal geometric algebra (CGA) give a unified language for geometric objects and transformations [13]–[17], with recent surveys and libraries showing renewed activity [18]–[20]. GA-Unity is directly relevant because it operationalizes 3D CGA—multivectors, transformations, and interpolation—inside Unity [9]. Recent GA/XR work also couples CGA with LLMs for instruction-based 3D scene transformations [21], builds CGA-based neural rendering and real-time shading [22], and proposes GA as a neurosymbolic substrate for XR authoring and rendering [23]. MAGES-ASSISTANT contributes to this direction by using CGA as the spatial substrate for situated assistance during an active procedural training session, unifying the spatial predicates and the pose and assistive-transform representation in one algebra.

III. SYSTEM OVERVIEW

MAGES-ASSISTANT separates embodiment, application-service routing, and agent reasoning (Fig. 1). The Unity layer is built on MAGES [1], its Unity integration and embodied-interaction layer; MAGES-ASSISTANT is the assistant layered on top. On the Unity side the trainee speaks to an embodied character: speech is recognized, the configured behavior collects the needed payload, and the streamed response is synthesized to speech incrementally at punctuation boundaries (preserving UTF-8 boundaries) so the character can begin speaking before generation finishes, with the same chunks queued as subtitles. The application-service layer forwards requests and manages conversation history, agent metadata, associated-agent lookup, and libraries. The Python agent server loads agents from Hydra YAML configurations and exposes streaming, validation, library, and timing endpoints.

A router is the orchestration point: its prompt receives the available agent identifiers and descriptions and selects exactly one agent per request, so a query about the current procedural step is handled differently from one about interaction mechanics, spatial reference, background knowledge, or performance. Unity prepares a candidate payload for every sub-agent in one pass and sends them with the routing request; the server predicts the target identifier, selects the already-prepared payload, and delegates. An eager-start optimization constructs

the sub-agent behaviors at router initialization, keeping sub-agent startup and context construction off the critical path after the routing decision.

The implementation rule is that Unity builds the world state and the server reasons over a compact, typed payload rather than an unstructured dump of the simulation; this keeps each answer auditable against a bounded state slice. Each behavior owns a payload schema. The State Assistant receives current/previous/next scene-graph actions and task objects and explains what to do and why. The XR Gameplay Assistant receives the same scene-graph payload but maps action types to interaction conventions—for example, insert actions correspond to placing an object into a green hologram—and returns short practical guidance. The Debriefer receives analytics and returns supportive coaching rather than raw scores. The General Assistant uses retrieval over per-agent libraries for background and terminology questions [10], [11]. The Spatial Assistant receives the CGA spatial payload described next, which is the focus of this paper.

IV. CGA SPATIAL REPRESENTATION AND ACTION LAYER

For spatial questions and assistive-action requests, the Spatial Assistant grounds dialogue in a runtime scene snapshot built by `SpatialContextRetriever`: head and hand poses, task objects, touched/held/pointed/gazed objects, current/previous/next actions, and semantic labels. These symbolic fields are retained for traceability—object names, task links, and near/on/needed-for relations. `CgaSpatialMath` additionally converts Unity poses into GA-Unity $\mathbb{R}^{4,1}$ multivector snapshots: user and entity anchors become conformal points and pose motors, and each entity is assigned a CGA proximity sphere so local neighborhoods are described by sphere-intersection facts rather than ad-hoc object lists. When both symbolic and CGA facts are present, the server prefers the CGA-derived facts and uses the symbolic names as traceable labels (Fig. 2).

A. Conformal Encoding

Let $x \in \mathbb{R}^3$ be a Unity-space position after conversion into the scenario coordinate frame. We embed it as a normalized conformal point

$$X(x) = n_o + x + \frac{1}{2}\|x\|^2 n_\infty, \quad (1)$$

where n_o is the conformal origin and n_∞ is the point at infinity. For normalized conformal points, the squared Euclidean distance is recovered by

$$d^2(x_i, x_j) = -2 X_i \cdot X_j. \quad (2)$$

This lets the Spatial Assistant compute distances, thresholds, and proximity facts using the same representation used for transformations.

Each runtime entity e_i is represented by the record

$$E_i = (id_i, \ell_i, X_i, S_i, M_i, A_i), \quad (3)$$

where id_i is the Unity or MAGES object identifier, ℓ_i is the canonical semantic label, X_i is the conformal point for the

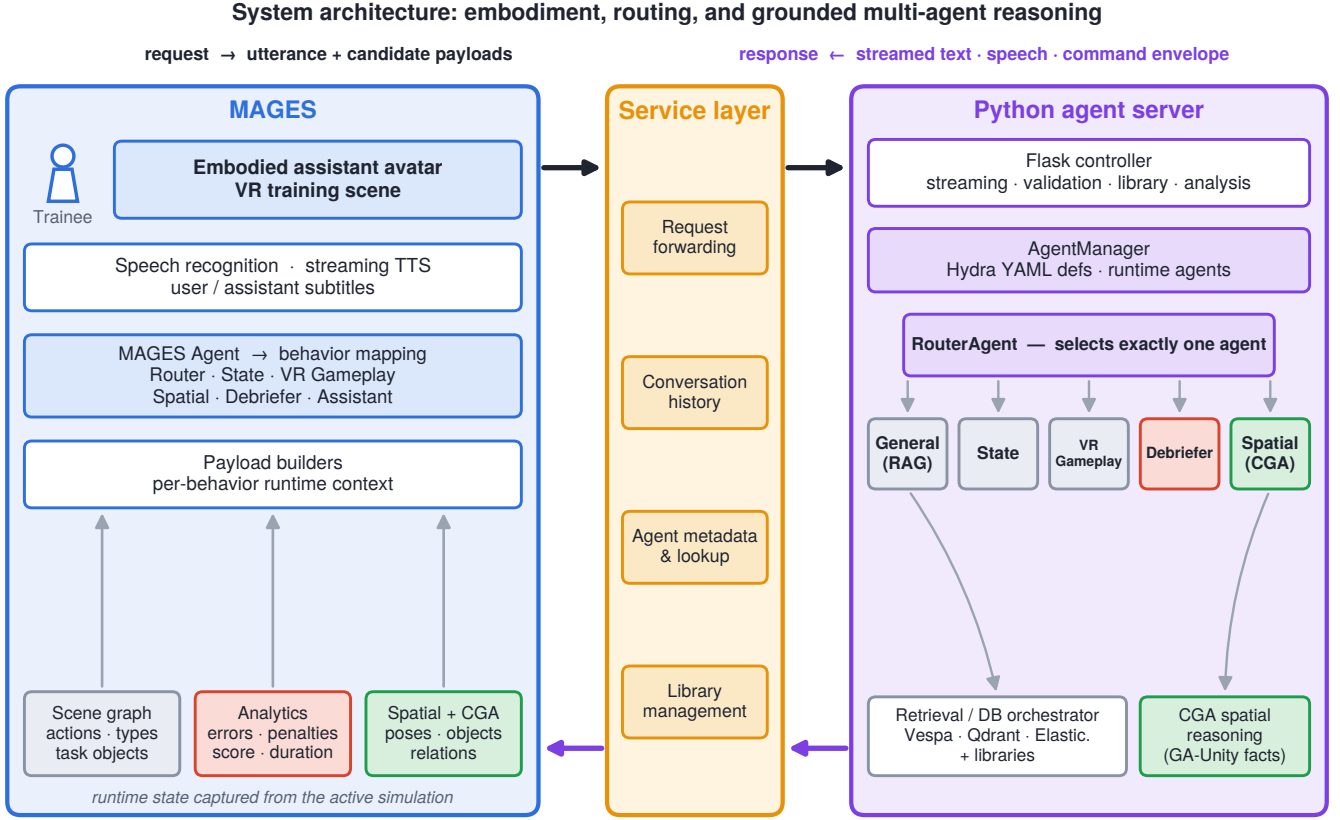


Fig. 1. Overall MAGES-ASSISTANT architecture. The MAGES client captures speech, builds per-behavior runtime payloads from the active simulation (scene graph, analytics, and CGA spatial context), and presents an embodied assistant with streaming speech and subtitles. The application service layer forwards requests and manages conversation history, agent metadata, and associated libraries. The CGA spatial context (conformal points, spheres, and relation predicates) is computed Unity-side through GA-Unity; the Python agent server runs the router, the five specialized agents, and retrieval, and its Spatial Assistant consumes the precomputed CGA facts to select grounded commands. The request path carries the utterance and candidate payloads; the response path returns streamed text, speech, and an optional hidden command envelope.

object anchor, S_i is a proximity or task-relevant bounding sphere, M_i is the current pose motor, and A_i stores links to current or next MAGES actions. The proximity sphere is represented in inner-product-null-space form as

$$S_i = X_i - \frac{1}{2}r_i^2 n_\infty, \quad (4)$$

where r_i is derived from the object’s authored interaction radius, bounding volume, or task-specific tolerance.

B. Spatial Relation Predicates

CGA-derived relation facts are serialized with their source operands and thresholds. A near relation is computed from point distance and proximity radii:

$$\text{near}(E_i, E_j) \Leftrightarrow -2(X_i \cdot X_j) \leq (r_i + r_j + \tau_n)^2, \quad (5)$$

which is equivalent to the sign of a point-sphere geometric product. For user-relative relations, the user frame defines oriented CGA planes through the head anchor. If x_u is the user head position and f_u is the (unit) user forward vector, the front plane is

$$\Pi_f = f_u - (f_u \cdot x_u)n_\infty. \quad (6)$$

With a fixed sign convention in Unity, an entity is in front of the user when

$$\text{inFront}(E_i, u) \Leftrightarrow X_i \cdot \Pi_f > \tau_f. \quad (7)$$

Analogous planes are constructed from the user right and up vectors for left/right and vertical checks. These CGA predicates are serialized next to the symbolic relation names, so the server can answer in ordinary language while the relation remains traceable to a geometric computation.

C. Rigid Body Motions

In conformal geometric algebra (CGA), rigid body transformations can be represented as *screw motions*, consisting of a rotation about an axis $\vec{a}$ by an angle θ , followed by a translation by a vector $\vec{t}$.

A translation by $\vec{t} = (t_1, t_2, t_3)$ is represented by a *translator*, defined as the multivector

$$T := 1 - \frac{1}{2}(t_1 e_1 + t_2 e_2 + t_3 e_3) n_\infty. \quad (8)$$

Spatial Assistant pipeline: Unity snapshot → CGA encoding → spatial facts → grounded assistive action

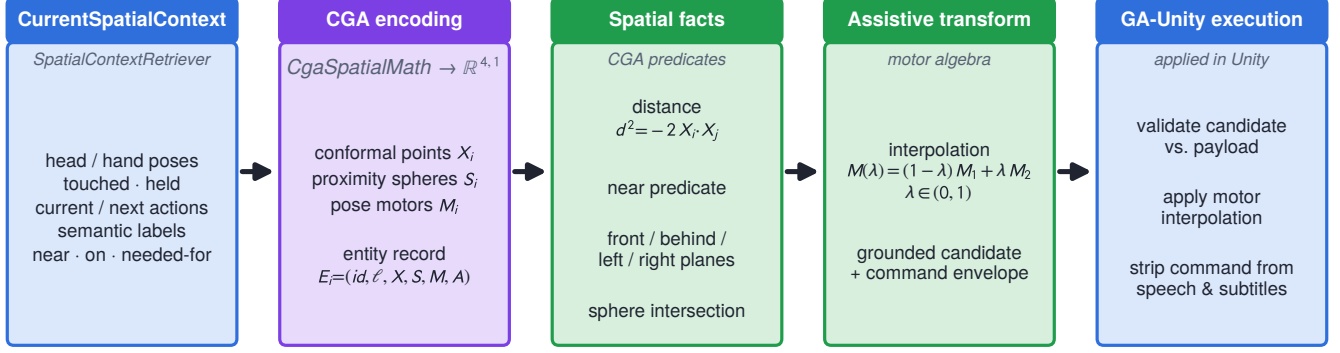


Fig. 2. CGA spatial and action pipeline inside the Spatial Assistant. *SpatialContextRetriever* captures the Unity scene snapshot; *CgaSpatialMath* encodes user and entity anchors as $\mathbb{R}^{4,1}$ conformal points, proximity spheres, and pose motors; CGA predicates yield distance, proximity, and user-relative relation facts; and a grounded assistive transform is dispatched through a hidden command envelope. Unity validates each candidate by *transformId* against the runtime payload before applying it through GA-Unity’s *LerpTR*. Symbolic relations are retained for traceability, while CGA-derived facts are preferred for grounding.

Similarly, a rotation is commonly expressed by a quaternion $q = a + bi + cj + dk$, which admits an equivalent representation as a *rotor*

$$R := a - de_{12} + ce_{13} - be_{23}. \quad (9)$$

A rigid body transformation is then represented by the geometric product of a translator and a rotor, yielding a *motor*

$$M = TR. \quad (10)$$

Under this convention, the rotor is applied first, followed by the translation encoded by the translator.

D. Assistive Action Through Multivector Interpolation

The action layer realizes object guidance as a rigid transformation from the object’s current pose to a grounded target pose. Let M_1 and M_2 denote the source and target pose motors, respectively. If these motors are not directly available, they can be constructed from the corresponding translators T_1 , T_2 and rotors R_1 , R_2 as

$$M_1 = T_1 R_1, \quad M_2 = T_2 R_2. \quad (11)$$

Guidance is implemented through direct linear interpolation between the two motors:

$$M(\lambda) = (1 - \lambda)M_1 + \lambda M_2, \quad \lambda \in [0, 1]. \quad (12)$$

Using the GA-Unity framework [9], the interpolated motor can be factorized into a translator and a rotor,

$$M(\lambda) = T(\lambda)R(\lambda), \quad (13)$$

where $T(\lambda)$ and $R(\lambda)$ encode the interpolated translational and rotational components, respectively.

The resulting translator and rotor may be used to recover conventional transformation parameters, such as the translation vector and the rotation axis–angle representation (or, equivalently, a quaternion representation). More importantly, they can be directly converted into a homogeneous 4×4 transformation

matrix, enabling seamless integration with modern real-time rendering and simulation engines, including Unity.

From an implementation perspective, the use of direct linear interpolation (LERP) is computationally efficient and produces satisfactory results in the present application, as interpolation is performed between consecutive object poses exhibiting only small translational and rotational differences [24]. Such conditions are typical in real-time simulation environments, where pose updates occur at millisecond timescales. Consequently, the interpolated transformations remain sufficiently close in the motor manifold, allowing LERP to provide smooth and visually accurate guidance. This approach avoids the substantially higher computational cost associated with spherical interpolation methods, which require the evaluation of multivector logarithms and exponentials to implement motor-based SLERP. As a result, the proposed method achieves an effective balance between computational efficiency and interpolation quality, making it well suited for real-time execution within interactive environments such as Unity.

E. Grounded Command Execution

The assistant does not generate a transformation. For a spatial-action request the server returns a natural-language answer plus a hidden to the user, typed command envelope identifying a pre-existing object candidate from *CurrentSpatialContext*, with fields *actionType*, *transformId*, *sourceEntityId*, and *targetEntityId*. *MAGES* strips the envelope from speech and subtitles and executes the candidate only when its *transformId* matches a transform present in the runtime payload, then applies it through GA-Unity’s *LerpTR*. This makes the geometry/action boundary explicit: the language model chooses among grounded candidates, while Unity executes only payload-matched commands, so the assistant can answer situated questions and trigger assistive motion without introducing unseen objects or speculative actions.

User: “Is the Ligasure in front of me?”

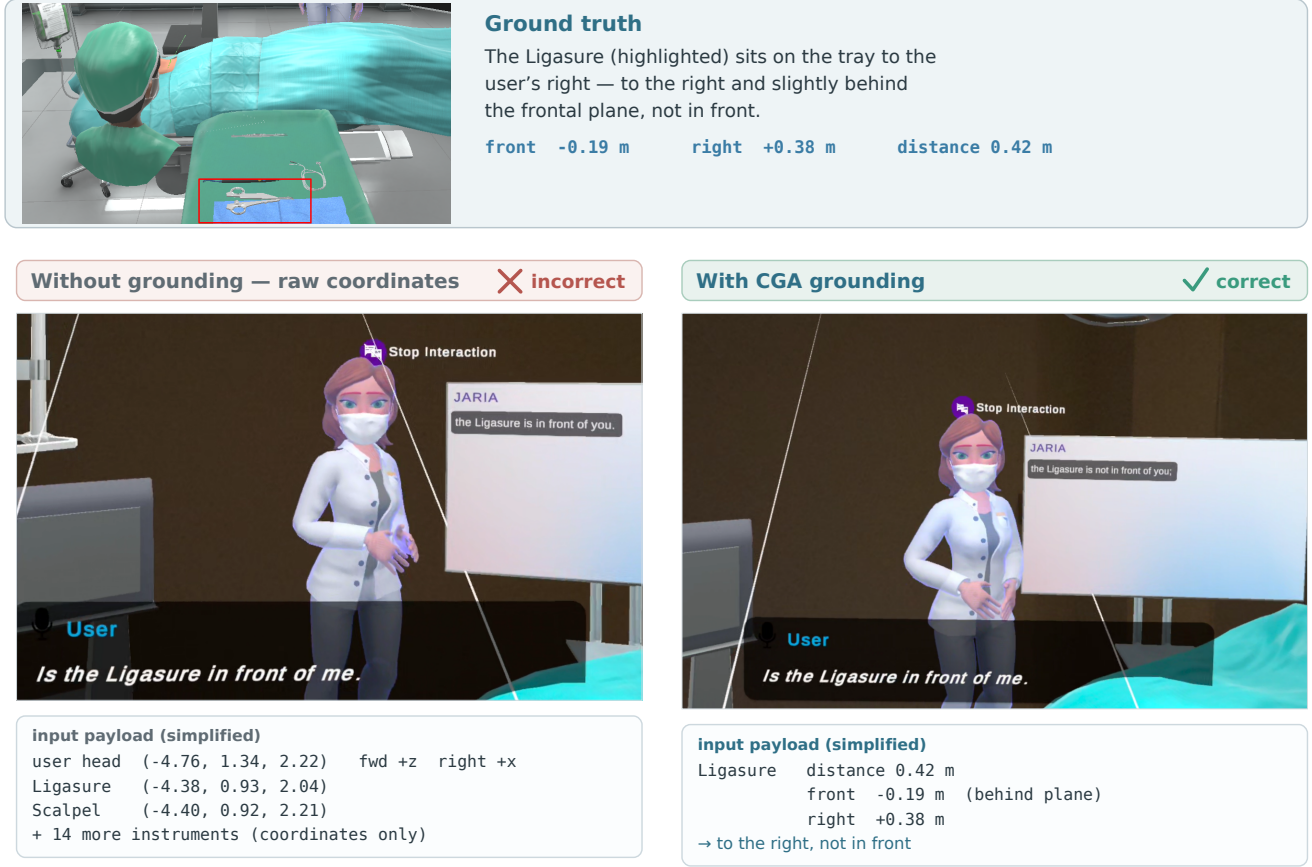


Fig. 3. Representative in-headset spatial-QA case from a Neuromonitoring procedure. For the question “Is the Ligasure in front of me?”, the ground truth (top) places the Ligasure on the tray to the user’s right and slightly behind the frontal plane (front -0.19 m, right $+0.38$ m, distance 0.42 m). Supplied the same scene as unstructured coordinates, the Raw model answers *yes* (incorrect); supplied the CGA relation facts, the deployed Grounded model answers *no* (correct). The simplified payload given in each condition appears beneath the corresponding response.

V. EVALUATION

We evaluate the CGA spatial layer on two axes: its real-time cost, and the effect of grounding situated answers in it. The surrounding assistant’s routing accuracy and end-to-end latency are logged separately; here we isolate the geometric layer. Measurements use controlled MAGES and synthetic Unity scenes with known object poses.

A. Real-Time Feasibility

We profiled the GA-Unity CGA layer on-device on Meta Quest 3—the constrained standalone target rather than a development PC—separating a one-time per-query snapshot cost from a per-frame guidance cost. We report managed allocation alongside wall-clock time because, on a mobile-class garbage collector, per-frame allocation is the dominant frame-time risk. For a representative scene of 20 task-relevant entities, building the conformal points, proximity spheres, and pose-motor representations and evaluating the full predicate set costs 1.27 ms on average (1.38 ms at the 99th percentile)—

0.04% of a representative 2000 ms language-model-and-speech response, i.e. negligible against the turn it precedes. Encoding scales linearly with entity count while the pairwise predicates scale quadratically and dominate at large scenes (Fig. 4); because the assistant grounds only task-relevant entities, the practical count stays small and the snapshot remains near a millisecond. During guidance, a single transform-interpolation step costs $13.0\ \mu\text{s}$ ($18.2\ \mu\text{s}$ at p99), or 0.16% of the 11.11 ms budget at 90 Hz ; the one-time delta construction costs $79\ \mu\text{s}$. Every measured operation is allocation-free (0 bytes per call across encoding, predicates, and interpolation), so the layer introduces no garbage-collection pauses. CGA spatial reasoning through GA-Unity is therefore comfortably within the budget for live embodied assistance. As a correctness check, the R410 predicates reproduce reference Euclidean classifications exactly over $79,440$ predicate evaluations, confirming the implementation.

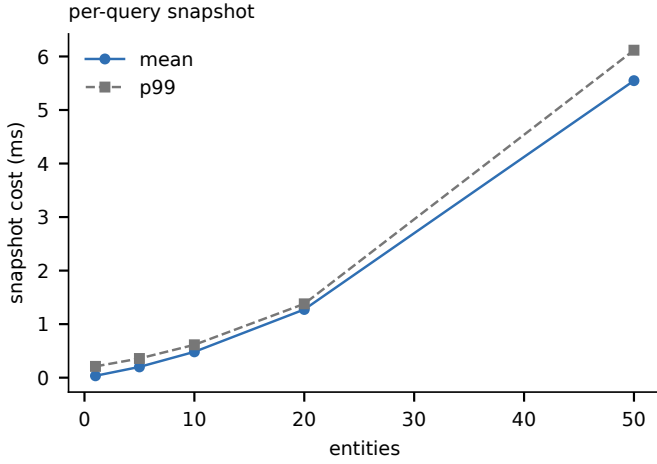


Fig. 4. Real-time cost of the GA-Unity CGA layer. Per-query snapshot cost (encode + predicates) versus entity count; the pairwise predicates dominate at scale.

B. Spatial-QA Grounding

To test whether the spatial layer is useful end-to-end, we measured situated spatial question-answering accuracy under two conditions that hold the scene information constant and vary only how it is delivered: *Grounded*, the deployed system, whose answers are grounded in the CGA layer’s typed relation facts; and *Raw*, in which the same geometry—user head basis, entity positions, and radii—is supplied as unstructured coordinates with no precomputed relations, so the model must infer the relations itself. We used gpt-4o at temperature 0 over 30 scenes and 300 questions spanning proximity and user-relative direction, with ground truth computed from the known poses and scored automatically; questions are balanced yes/no (chance 50.00%) and three-way multiple choice (chance 33.33%).

Grounding the answers in the CGA relation facts raises accuracy from 69.67% (Raw) to 95.67% (Grounded), a 26-percentage-point improvement that is statistically decisive (McNemar, $n = 300$, $p < 10^{-19}$; of the discordant questions, 82 go to Grounded and only 4 to Raw). The gain is stable across scene difficulty—Grounded holds at 95.33% on the hard, dense, near-boundary scenes where the Raw model falls to 68.00%—and is largest for relations that require comparing entities or resolving handedness: *closest* (93.33% vs 46.67%) and *left/right*, where the Raw model is near or below chance (e.g. 40.00% on *left*); Grounded is weakest on *right* (66.67%). The error profiles differ in kind: Grounded errors are wrong-entity selections with no hallucinations or refusals, whereas the Raw model fails mainly by misjudging boolean relations (57 cases) (Fig. 5). Figure 3 shows one such case in-headset: asked whether the Ligasure is in front, the Raw model answers *yes* from the raw coordinates, while the deployed Grounded model uses the CGA relation facts—front -0.19 m (behind the frontal plane) and right $+0.38$ m—to correctly answer *no*.

This isolates the effect of *structured spatial grounding*, not of CGA over Euclidean math: the conditions vary how the

TABLE I
REPRESENTATIONAL UNITY: SPATIAL OPERATIONS OF MAGES-ASSISTANT IN THE CONVENTIONAL DECOMPOSED UNITY PIPELINE VERSUS THE CGA PIPELINE, WHICH KEEPS EVERY OPERATION IN ONE REPRESENTATION ($\mathbb{R}^{4,1}$ MULTIVECTORS), CONVERTING TO UNITY TYPES ONLY AT THE BOUNDARY. THE MOTOR DELTA AND SCREW ARE THE ACTION *model*; THE DEPLOYED INTERPOLATION IS GA-UNITY’S `LerpTR`.

Operation	Decomposed (Unity)	CGA (this work)
Object anchor	<code>Vector3</code>	conformal point X
Distance / proximity	<code>Vector3.Distance</code>	inner product $X_i \cdot X_j$
Proximity region	radius + distance test	dual sphere $X - \frac{1}{2}r^2n_\infty$
Direction relations	dots with basis axes	oriented planes Π
Pose	<code>Vector3+Quaternion</code>	motor M
Pose delta	quat. mult + vec. sub	$M_t \bar{M}_s$
Interpolation	<code>Vector3.Lerp</code> , <code>Quaternion.Slerp</code>	<code>R410_Helper.LerpTR</code>
Representations	<code>Vector3</code> , <code>Quaternion</code> , <code>Matrix4x4</code>	one ($\mathbb{R}^{4,1}$)

geometry is delivered, not the geometry itself, and we believe a Euclidean layer could supply the same relations. The result shows that precomputing the relations—which the CGA layer does cheaply and deterministically, as the feasibility measurement above confirms—makes situated spatial answering substantially more reliable than leaving the model to infer them from raw coordinates.

C. Representational Unification

Beyond cost, the practical argument for CGA here is representational: the perception predicates and the pose representation live in a single algebra, so spatial facts and the assistive-transform candidate share one representation and one audit boundary rather than being spread across `Vector3`, `Quaternion`, and `Matrix4x4` with conversions between them (Table I).

VI. DISCUSSION AND CONCLUSION

MAGES-ASSISTANT applies controlled grounding: each agent receives compact context for its role, which reduces irrelevant context and makes answers traceable to a specific channel—retrieved documents, scene-graph state, analytics, or the CGA spatial layer. The geometric layer keeps the spatial predicates and the pose representation in one algebra, so relation facts and the assistive-transform candidate share a representation and a single audit boundary. The main limitations are that the spatial model is object-level rather than anatomy-aware—it does not yet model deformable constraints or continuous hand-object contact required for autonomous procedural intervention—and that the router selects a single agent per request, which keeps prompts compact and traceable but precludes multi-agent synthesis for complex questions.

We presented the CGA spatial layer of MAGES-ASSISTANT, an embodied multi-agent assistant for MAGES XR training. The layer serialises user and object state as conformal points, proximity spheres, and pose motors, computes proximity and user-relative relations as CGA predicates, and dispatches assistive guidance as payload-grounded transforms

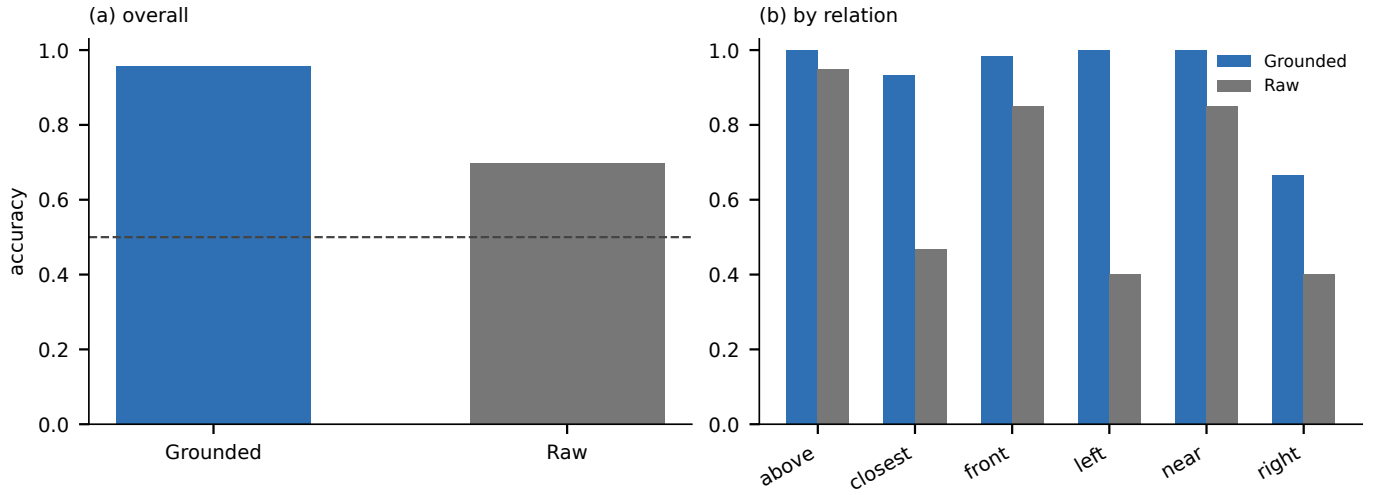


Fig. 5. Situated spatial-QA accuracy by condition: *Grounded* (CGA relation facts, the deployed system) and *Raw* (the same geometry as unstructured coordinates). (a) Overall; (b) by relation type. Chance is 50.00% for yes/no and 33.33% for multiple-choice questions.

validated before execution, with a motor-based screw as the geometric action model. We showed the layer to be real-time-feasible and allocation-free through GA-Unity, and that grounding situated answers in it substantially improves spatial question answering over raw-coordinate context. The result is a CGA spatial substrate that grounds situated language in the running state of the simulation—a practical step toward XR assistants that understand not only language but the training situation in which it is spoken.

REFERENCES

- [1] P. Zikas, A. Protopsaltis, N. Lydatakis, M. Kentros, S. Geronikolakis, S. Kateros, M. Kamarianakis, G. Evangelou, A. Filippidis, E. Grigoriou, D. Angelis, M. Tamiolakis, M. Dodis, G. Kokiadis, J. Petropoulos, M. Pateraki, and G. Papagiannakis, “Mages 4.0: Accelerating the world’s transition to vr training and democratizing the authoring of the medical metaverse,” *IEEE Computer Graphics and Applications*, vol. 43, no. 2, pp. 43–56, 2023.
- [2] N. E. Seymour, A. G. Gallagher, S. A. Roman, M. K. O’Brien, V. K. Bansal, D. K. Andersen, and R. M. Satava, “Virtual reality training improves operating room performance: Results of a randomized, double-blinded study,” *Annals of Surgery*, vol. 236, no. 4, pp. 458–464, 2002.
- [3] L. Jensen and F. Konradsen, “A review of the use of virtual reality head-mounted displays in education and training,” *Education and Information Technologies*, vol. 23, pp. 1515–1529, 2018.
- [4] G. Makransky and G. B. Petersen, “The cognitive affective model of immersive learning (CAMIL): A theoretical research-based model of learning in immersive virtual reality,” *Educational Psychology Review*, vol. 33, pp. 937–958, 2021.
- [5] V. Chheang, S. Sharmin, R. Márquez-Hernández, M. Patel, D. Rajasekaran, G. Caulfield, B. Kiafar, J. Li, P. Kullu, and R. L. Barmaki, “Towards anatomy education with generative AI-based virtual assistants in immersive virtual reality environments,” in *2024 IEEE International Conference on Artificial Intelligence and eXtended and Virtual Reality (AIxVR)*, 2024, pp. 21–30.
- [6] U. Hemminki-Reijonen, N. M. A. M. Hassan, M. Huottilainen, J.-M. Koivisto, B. U. Cowley *et al.*, “Design of generative AI-powered pedagogy for virtual reality environments in higher education,” *npj Science of Learning*, vol. 10, no. 31, p. 31, 2025.
- [7] A. Shoa and D. Friedman, “Milo: An LLM-based virtual human open-source platform for extended reality,” *Frontiers in Virtual Reality*, vol. 6, p. 1555173, 2025.
- [8] S. Ding and Y. Chen, “RAG-VR: Leveraging retrieval-augmented generation for 3d question answering in VR environments,” in *Proceedings - 2025 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*. IEEE, 2025, pp. 131–136.
- [9] M. Kamarianakis, N. Lydatakis, and G. Papagiannakis, “GA-Unity: A production-ready unity package for seamless integration,” in *Advances in Computer Graphics*, ser. Lecture Notes in Computer Science, vol. 15340. Cham: Springer, 2025, pp. 349–363.
- [10] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Kuttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, “Retrieval-augmented generation for knowledge-intensive NLP tasks,” in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 9459–9474.
- [11] V. Karpukhin, B. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W.-t. Yih, “Dense passage retrieval for open-domain question answering,” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 6769–6781.
- [12] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafraan, K. Narasimhan, and Y. Cao, “ReAct: Synergizing reasoning and acting in language models,” in *International Conference on Learning Representations*, 2023.
- [13] D. Hestenes and G. Sobczyk, *Clifford Algebra to Geometric Calculus: A Unified Language for Mathematics and Physics*. Dordrecht: Springer, 1984.
- [14] L. Dorst, D. Fontijne, and S. Mann, *Geometric Algebra for Computer Science: An Object-Oriented Approach to Geometry*, revised ed. Morgan Kaufmann, 2009.
- [15] H. Li, D. Hestenes, and A. Rockwood, “Generalized homogeneous coordinates for computational geometry,” in *Geometric Computing with Clifford Algebras*, G. Sommer, Ed. Berlin, Heidelberg: Springer, 2001, pp. 27–59.
- [16] C. Perwass, *Geometric Algebra with Applications in Engineering*. Berlin, Heidelberg: Springer, 2009.
- [17] D. Hildenbrand, *Foundations of Geometric Algebra Computing*. Berlin, Heidelberg: Springer, 2013.
- [18] E. Hitzer, C. Lavor, and D. Hildenbrand, “Current survey of clifford geometric algebra applications,” *Mathematical Methods in the Applied Sciences*, vol. 47, no. 3, pp. 1331–1361, 2024.
- [19] E. Hitzer, M. Kamarianakis, G. Papagiannakis, and P. Vasik, “Survey of new applications of geometric algebra,” *Mathematical Methods in the Applied Sciences*, vol. 47, no. 14, pp. 11 368–11 384, 2024.
- [20] A. H. Eid and F. G. Montoya, “Developing GA-FuL: A generic wide-purpose library for computing with geometric algebra,” *Mathematics*, vol. 12, no. 14, p. 2272, 2024.
- [21] D. Angelis, P. Kolyvakis, M. Kamarianakis, and G. Papagiannakis, “Geometric algebra meets large language models: Instruction-based transformations of separate meshes in 3d, interactive and controllable scenes,” 2024.

- [22] E. Geronikolakis, M. Kamarianakis, A. Protopsaltis, and G. Papagiannakis, "Neural-GASh: A CGA-based neural radiance prediction pipeline for real-time shading," in *Empowering Novel Geometric Algebra for Graphics and Engineering*. Cham: Springer, 2026, pp. 3–24.
- [23] M. Kamarianakis, A. Protopsaltis, and G. Papagiannakis, "One algebra for all: Geometric algebra methods for neurosymbolic XR scene authoring, animation and neural rendering," 2025.
- [24] M. Kamarianakis, I. Chrysovergis, N. Lydatakis, M. Kentros, and G. Papagiannakis, "Less is more: Efficient networked vr transformation handling using geometric algebra," vol. 33, no. 1, p. 6.