Graph-Fused Vision-Language-Action Models for Semantically Safe Dual-Robot Control via Control Barrier Functions

Jiajun Gu[1,+], **Weihao Cheng**[2,+], **and Longsen Gao**[3,+,*]

[1]Institut Hohai-Lille, Hohai University, Nanjing, 210024, China
[2]School of Mechanical Engineering, Sichuan University, Chengdu, 610065, China
[3]Electrial and Computer Engineering Department, University of New Mexico, Albuquerque, 87106, USA
[*]corresponding.longsengao@outlook.com
[+]these authors contributed equally to this work

ABSTRACT

Deploying dual-arm robots in human-centric environments demands not only dexterous task execution but also strict adherence to common sense safety constraints. While recent advancements in Vision-Language-Action (VLA) models enable complex policy reasoning from human demonstrations, they typically lack the formal motion safeguards required to prevent semantically unsafe behaviors—such as manipulating liquids above electronics. In this work, we propose a unified framework that integrates a Graph-Fused VLA (GF-VLA) model with a semantic safety filter, enabling task-level reasoning and certified safe execution for dual-robot systems. To generate manipulation strategies, our approach extracts information-theoretic cues from visual inputs to construct temporal scene graphs that capture intricate hand-object interactions. A language-conditioned transformer leverages these graphs to output hierarchical behavior trees, interpretable Cartesian commands, and optimal cross-hand assignments. Concurrently, to ensure execution safety, the system builds a 3D semantic map and utilizes the contextual reasoning capabilities of large language models to identify semantically unsafe spatial relationships and poses. These semantic rules, alongside traditional geometric collision bounds, are rigorously enforced at the continuous control level via a Control Barrier Function (CBF) certification formulation. We evaluate the proposed framework across diverse dual-arm manipulation scenarios, encompassing complex spatial generalizations and practical real-world semantic constraints. Our results demonstrate that fusing information-theoretic scene representations with CBF-based motion safeguards yields highly reliable, human-readable task policies. Ultimately, this approach achieves high execution success rates while guaranteeing safe robot operation well beyond traditional collision avoidance.

Introduction

The deployment of dual-arm robotic systems in human-centric environments demands a delicate balance between advanced dexterous manipulation and rigorous safety compliance[1–3]. While recent developments in robot learning have enabled machines to acquire complex skills from human demonstrations[4], operating in unstructured domains requires more than just task execution[5]. Robots must understand and adhere to common-sense constraints recognized by humans, such as recognizing that moving a container of liquid over electronic devices is inherently risky[6]. Achieving this level of autonomy requires a framework that can simultaneously reason about task-level objectives, coordinate bimanual actions, and strictly enforce semantic-safety boundaries at the control level[7,8].

Current approaches to robotic manipulation largely treat high-level policy reasoning and low-level safety certification as isolated problems[9]. On one hand, Vision-Language-Action (VLA) models and imitation learning frameworks have shown immense promise in generating robotic behaviors from RGB-D inputs[10]. However, these methods frequently rely on low-level trajectory imitation or unconstrained neural network outputs, which struggle to generalize across novel spatial layouts and critically lack formal safety guarantees. On the other hand, established safe control methodologies, such as Control Barrier Functions (CBFs)[11], provide rigorous mathematical guarantees for collision avoidance. Yet, traditional CBF formulations are typically restricted to geometric distances and lack the contextual awareness[12] required to interpret semantic relationships, e.g., differentiating between a fragile glass and a sturdy block[13]. Consequently, there is a distinct gap in the literature regarding the integration of dynamic, multi-arm policy reasoning with semantically aware control safeguards.

To address these limitations, we propose a unified framework: Graph-Fused Vision-Language-Action Models for Semantically Safe Dual-Robot Control via Control Barrier Functions. Our approach bridges the gap between graph-grounded policy generation and semantically certified execution. To enable robust task reasoning, we introduce a Graph-Fused VLA (GF-VLA)

architecture that extracts task-critical spatial and relational features directly from visual demonstrations. These extracted representations are systematically encoded into temporally ordered scene graphs, explicitly capturing the intricate dynamics of hand-object and object-object interactions. A language-conditioned transformer subsequently processes these structured graphs to output interpretable Cartesian motion commands and dictate optimal cross-hand assignments for bimanual coordination.

Crucially, to ensure that the generated policies are contextually safe, these Cartesian commands are passed through a novel semantic safety filter before execution. Utilizing 3D point-cloud perceptions, our system constructs a continuous semantic map of the environment and leverages the reasoning capabilities of Large Language Models (LLMs) to infer context-specific safety rules. These rules are translated into spatial, behavioral, and pose-based constraints, which are formally embedded into a quadratic programming (QP) formulation using CBFs. This allows the system to filter and modify the VLA-generated commands in real-time, ensuring strict adherence to both geometric and semantic safety bounds.

The primary contributions of this work are summarized as follows:

- A novel architecture that integrates information-theoretic scene graph representations and VLA-based policy generation with a formal semantic safety filter for dual-arm robotic systems.
- A robust method for extracting task-relevant cues from human demonstrations to generate hierarchical behavior trees and Cartesian commands, improving execution efficiency and cross-hand coordination in bimanual tasks.
- A real-time certification mechanism that maps LLM-inferred "common sense" constraints (including spatial relationships and behavioral limits) into mathematically rigorous CBFs, expanding robot safety beyond traditional collision avoidance.
- Extensive empirical validation in complex spatial and semantic environments, demonstrating high task success rates, improved generalization, and strict adherence to context-aware safety constraints.

1 Related Work

Recent breakthroughs in robot learning have increasingly shifted away from task-specific architectures[14] toward general-purpose foundation models[15]. Traditional imitation learning methods, such as Behavioral Cloning (BC)[16] and imitation via low-level trajectory matching[17], often struggle to generalize across diverse object instances and spatial layouts. To address this, VLA models have emerged as a powerful paradigm, integrating visual perception, natural language instructions, and action generation into unified end-to-end architectures. Furthermore, the integration of Large Language Models (LLMs) has enabled embodied agents to perform complex, multi-step reasoning through prompt-based Chain-of-Thought planning. However, while VLAs excel at generating high-level task plans and heuristic Cartesian commands, they exhibit two critical limitations: they frequently fail to capture the high-precision geometric constraints required for dual-arm coordination, and their end-to-end nature inherently lacks formal mathematical guarantees for motion safety. Our work leverages the reasoning power of VLA architectures but grounds their outputs in both structured relational graphs and rigorous control theory.

To overcome the spatial generalization bottlenecks of purely pixel-based VLA models, literature has increasingly turned to object-centric and relational abstractions[18]. Scene graphs, which represent environments as nodes (objects/hands) and edges (spatial or semantic relationships), have been widely adopted to encode the hierarchical structure of a workspace. In dual-arm manipulation, capturing temporal dynamics, specifically hand-object and object-object interactions, is crucial for efficient task execution and optimal cross-hand assignment[2]. Recent studies have demonstrated that utilizing information-theoretic metrics, such as Shannon entropy and mutual information, can effectively filter out redundant background data from human demonstrations, isolating the most task-relevant interaction cues[19]. By embedding these information-theoretic abstractions into temporally ordered scene graphs, our framework provides the LLM planner with a highly accurate, human-readable representation of the workspace, thereby mitigating the spatial ambiguities that plague direct image-to-action models[20].

Guaranteeing collision-free motion is a foundational requirement in robotic manipulation[21]. Classical approaches often utilize artificial potential fields[22], sampling-based planners[23], or reactive optimization frameworks[24] to navigate around obstacles. In recent years, Control Barrier Functions (CBFs) have become the gold standard for ensuring safety in continuous-time dynamical systems[11]. By defining a safe operating region as the forward-invariant superlevel set of a continuously differentiable function, CBFs allow safety constraints to be strictly enforced via real-time Quadratic Programming (QP) optimization[25]. In dual-arm manipulation, CBFs are often used to handle self-collision, joint limits, and static environmental boundaries[26]. However, traditional CBF formulations are strictly geometric; they rely on Euclidean distances and bounding volumes, rendering them blind to the context or semantics of the objects being manipulated. They cannot distinguish between a permissible close-proximity interaction (e.g., placing a lid on a cup) and a hazardous one (e.g., spilling liquid on a laptop[27].

Operating in human-centric environments requires robots to adhere to “common sense" behavioral rules that go beyond mere obstacle avoidance[28]. The advent of open-vocabulary vision models, such as CLIP[29] and the Segment Anything Model (SAM)[30], has revolutionized semantic scene understanding, allowing robots to construct dense, 3D semantic maps from RGB-D

data. While previous research has successfully utilized semantic mapping for object-goal navigation and heuristic task execution, integrating semantic constraints directly into low-level control loops remains relatively underexplored. Emerging work has begun using LLMs to infer context-specific safety rules from spatial relationships, object states, and desired poses. However, translating these discrete, language-based logical rules into continuous control signals is challenging. Our framework addresses this gap by mapping LLM-derived semantic rules—encompassing unsafe spatial relationships, behavioral constraints, and rotational limits—into differentiable distance functions. By synthesizing these semantic functions with geometric boundaries within a unified CBF-QP framework, we ensure that the task policies generated by the VLA are executed with rigorous, context-aware safety guarantees.

2 Methodology

2.1 Kinematics and Dynamics of Manipulators

Kinematics defines the relationship between coordinates in the joint space $\mathbf{q}$ and those in the task space x. The forward kinematics (FK) problem maps joint coordinates to task space coordinates as $FK : \mathbf{q} \mapsto \mathbf{x}$, while the inverse kinematics (IK) problem defines the mapping in the opposite direction as $IK : \mathbf{x} \mapsto \mathbf{q}$.

Several techniques exist for solving kinematics, such as geometric formulations and the Denavit–Hartenberg (DH) convention[31]. Although the DH approach provides a systematic representation, the associated closed-form equations may exhibit singularities and nonlinearities, which increase the computational cost of IK solutions in complex configurations.

In this study, two robotic platforms are used for experimental validation: a 6-DOF UR5e (`developed by` Universal Robots). manipulator and a 7-DOF WAM manipulator (`developed by` Barrett Technology). The IK and ID models for UR5e and UR10e robot are derived using the DH parameterization as shown in Table. 1 and Table. 2, respectively. Fig. 1 depict the coordinate frame for the UR series robot arm assignments that covers both UR10e and UR5e, based on the DH convention. For each robot, the head pan and tilt joints are excluded, as their motions are independent of the primary manipulation tasks considered here.

Table 1. 6-DOF UR5e DH and inertial parameters (units: m, rad; CoM relative to frame i).

Kinematics	θ [rad]	d [m]	a [m]	α [rad]	Dynamics	Mass [kg]	Center of Mass [m]
Joint1	θ_1	0.089159	0	$\pi/2$	Link1	3.7	[0, −0.02561, 0.00193]
Joint2	θ_2	0	−0.425	0	Link2	8.393	[0.2125, 0, 0.11336]
Joint3	θ_3	0	−0.39225	0	Link3	2.33	[0.11993, 0, 0.0265]
Joint4	θ_4	0.10915	0	$\pi/2$	Link4	1.219	[0, −0.0018, 0.01634]
Joint5	θ_5	0.09465	0	$-\pi/2$	Link5	1.219	[0, 0.0018, 0.01634]
Joint6	θ_6	0.0823	0	0	Link6	0.1879	[0, 0, −0.001159]

Let ${}^{i}\mathcal{T}_{j}$ denote the homogeneous transformation matrix from frame F_i to frame F_j. The transformation from the base frame to the end-effector frame, ${}^{0}\mathcal{T}_{E}$, is obtained by recursively multiplying the transformations between consecutive frames:

$$ {}^{0}\mathcal{T}_{E} = \prod_{i=0}^{n} {}^{i}\mathcal{T}_{i+1} = \begin{bmatrix} {}^{0}R_{E} & {}^{0}p_{E} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix}, \tag{1} $$

where ${}^{0}R_{E} \in \mathbb{R}^{3\times3}$ is the rotation matrix from the base to the end-effector frame, and ${}^{0}p_{E} \in \mathbb{R}^{3}$ is the position vector of the end-effector expressed in the base frame. The integer n is the number of intermediate frames in the kinematic chain.

From the dynamics perspective, the motion of the manipulator relates joint torques/forces to joint positions, velocities, accelerations, and external wrenches applied to the end-effector. Common formulations include the Newton–Euler method, the Lagrangian method, and the Hamiltonian method[32]. In this paper, the Newton–Euler recursive algorithm is adopted with DH parameters to obtain the inverse dynamics model.

The general inverse dynamics equation for an N-DOF manipulator is:

$$ \tau = \mathbf{M}(\mathbf{q})\,\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\,\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) + \mathbf{J}^{\mathsf{T}}(q)\,\mathcal{F}, \tag{2} $$

where $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{N\times N}$ is the generalized inertia matrix,$\mathbf{C}(\mathbf{q},\dot{\mathbf{q}}) \in \mathbb{R}^{N\times N}$ is the Coriolis/centrifugal/friction matrix, $\mathbf{G}(\mathbf{q}) \in \mathbb{R}^{n}$ is the gravitational torque vector, $\mathbf{J}(\mathbf{q}) \in \mathbb{R}^{6\times N}$ is the Jacobian matrix, $\mathcal{F} \in \mathbb{R}^{N}$ is the external wrench (forces and moments) applied at the end-effector.

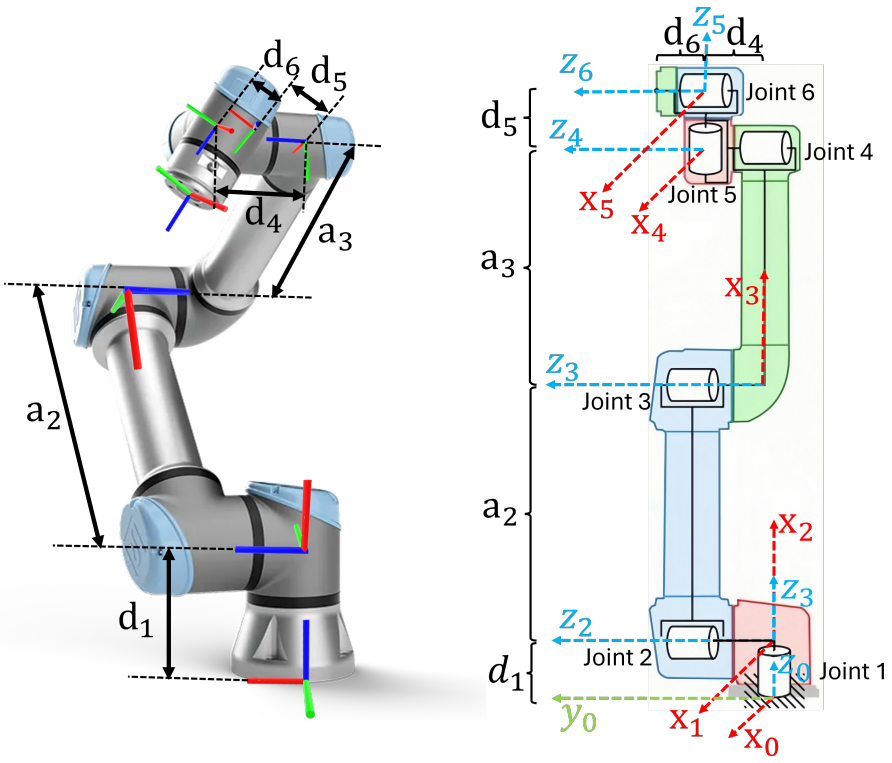

Figure 1. 6 DoF UR series robot manipulator with coordinate frame assignment.

The geometric and inertial parameters of each link are obtained from the CAD models of the UR5e and WAM manipulators. The moments of inertia are evaluated at each link's center of mass (CoM) and expressed in its local frame. The state variables $\mathbf{q}$, $\dot{\mathbf{q}}$, $\ddot{\mathbf{q}}$, and τ denote joint position, velocity, acceleration, and generalized torque/force, respectively.

A special case is the inverse statics model, obtained by setting

$$\dot{\mathbf{q}} = \ddot{\mathbf{q}} = \mathbf{0}_{\in \mathbb{R}^N}. \tag{3}$$

In this work, no external forces or torques are applied to the end-effector, implying

$$\mathcal{F} = \mathbf{0}_{\in \mathbb{R}^N}. \tag{4}$$

Table 2. 6-DOF UR10e DH and inertial parameters (units: m, rad; CoM relative to frame i).

Kinematics	θ **[rad]**	d **[m]**	a **[m]**	α **[rad]**	**Dynamics**	**Mass [kg]**	**Center of Mass [m]**
Joint1	θ_1	0.1807	0	$\pi/2$	Link1	7.369	[0.021, 0.000, 0.027]
Joint2	θ_2	0	-0.6127	0	Link2	13.051	[0.38, 0.000, 0.158]
Joint3	θ_3	0	-0.57155	0	Link3	3.989	[0.24, 0.000, 0.068]
Joint4	θ_4	0.17415	0	$\pi/2$	Link4	2.1	[0.000, 0.007, 0.018]
Joint5	θ_5	0.11985	0	$-\pi/2$	Link5	1.98	[0.000, 0.007, 0.018]
Joint6	θ_6	0.11655	0	0	Link6	0.615	[0, 0, -0.026]

2.2 Impedance Control

The proposed impedance control strategy is implemented on a single hydraulic manipulator link as shown in Fig. 2. At its core, the architecture relies on a target impedance model to synthesize Cartesian positional setpoints and force feedback signals, generating a dynamically modified reference position, x_s. This updated setpoint, alongside system feedback, is fed into an inner-loop position controller—specifically, a nonlinear proportional-integral (NPI) controller—which generates the final

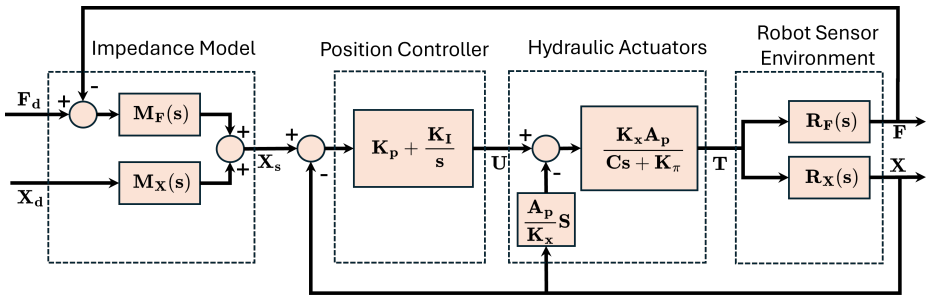

Figure 2. Block diagram of the impedance control.

hydraulic control signal. Expanding this framework to an n-dimensional operational space inherently requires a task-space to joint-space transformation within the position control loop.

A standard linear, second-order system is adopted to define the target impedance due to its widespread applicability and established reliability in contact tasks. The target dynamics are governed by the following relationship:

$$m\ddot{x}_s + c(\dot{x}_s - \dot{x}_d) + k(x_s - x_d) = F_d - F \tag{5}$$

where m, c, and k represent the desired mass, damping, and stiffness parameters. The variables x_s and x_d denote the modified and nominal position trajectories, while F and F_d correspond to the measured and desired interaction forces, respectively.This formulation yields the impedance transfer functions $\mathbf{M_X}$ and $\mathbf{M_F}$. Notably, unlike prior methodologies that assume a unity transfer function for $\mathbf{M_X}$—effectively superimposing the force response directly onto the reference trajectory—the present approach derives $\mathbf{M_X}$ strictly from the target impedance relation. Consequently, it explicitly encapsulates the dynamics associated with an actively changing reference trajectory, x_d.

The hydraulic system is represented using linearized transfer functions derived from standard nonlinear fluid flow equations. The dominant constants resulting from the linearization of the spool valve dynamics are defined as:

$$\begin{aligned} K_x &= \frac{\partial}{\partial U}\left(K_A U\sqrt{\Delta P}\right) = K_A\sqrt{\Delta P} \\ K_\pi &= \frac{\partial}{\partial(\Delta P)}\left(K_A U\sqrt{\Delta P}\right) = \frac{K_A U}{2\sqrt{\Delta P}} \end{aligned} \tag{6}$$

In these expressions, A_p is the piston area, U is the control input signal, ΔP is the mean pressure differential across the spool valve, and K_A is a proportionality constant linking the input signal to the open valve area. Within the broader model, C represents the hydraulic fluid compressibility, and T signifies the generalized output torque.

To analyze the interaction dynamics, a fourth-order model characterizing the coupled robot-sensor-environment system is utilized, adapting the framework established by Volpe and Khosla. The transfer functions mapping the actuator torque T to the resulting position X ($\mathbf{R_X}$) and force F ($\mathbf{R_F}$) are expressed as:

$$\begin{aligned} \mathbf{R_X} &= \frac{m_B s^2 + (c_s + c_e)\, s + (k_s + k_e)}{\Delta_R(s)} \\ \mathbf{R_F} &= \frac{k_s\left(m_B s^2 + c_e s + k_e\right)}{\Delta_R(s)} \end{aligned} \tag{7}$$

where the characteristic denominator is defined as:

$$\Delta_R(s) = [m_B s^2 + (c_s + c_e)s + (k_s + k_e)][m_A s^2 + c_r s + k_r] + [m_B s^2 + c_e s + k_e](c_s s + k_s) \tag{8}$$

where the subscripts r, s, and e correspond to the physical parameters (mass, damping, stiffness) of the robot, sensor, and environment, respectively, while the subscripts A and B designate the distinct interfaces between these coupled elements.

2.3 Visual-Language-Action models

Consider a visual backbone, $\mathcal{F}$, pre-trained extensively on human demonstration data. The input consists of a batch of M triplets, $\{H_i, R_i, L_i\}_{i=1}^{M}$, which represent paired human videos, robot videos, and their corresponding natural language task

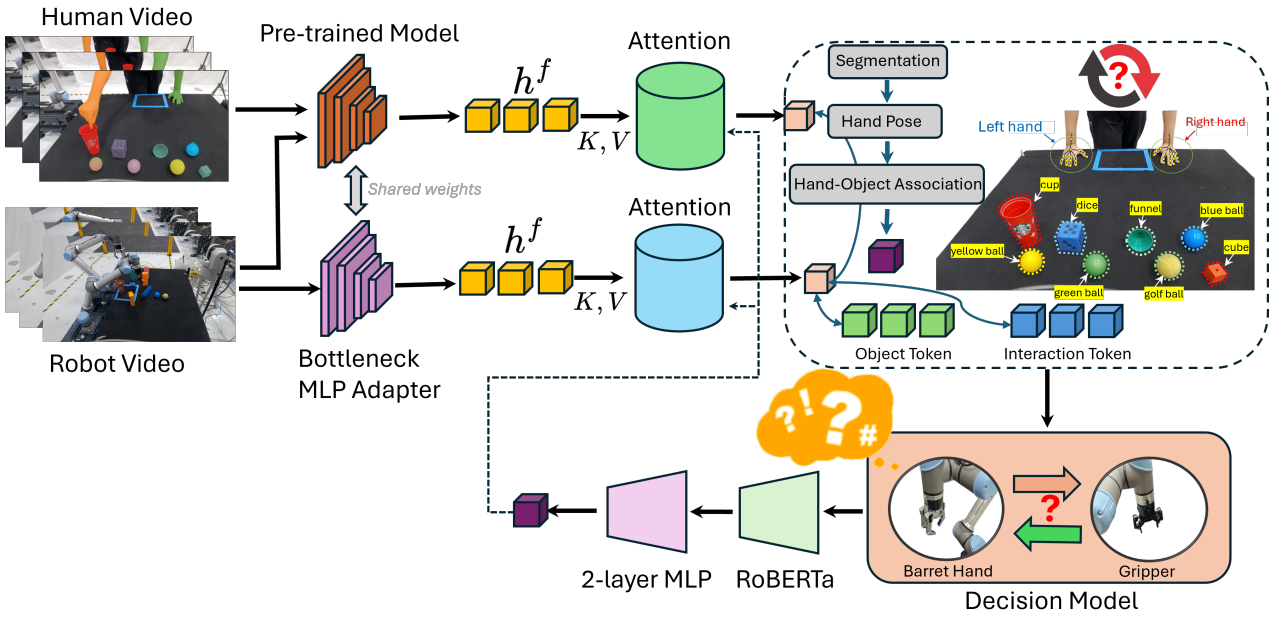

Figure 3. Architecture of the proposed Graph-Fused Vision-Language-Action (GF-VLA) model and the HR-Align framework. A frozen pre-trained backbone extracts spatial-temporal features from human demonstrations, while robot video representations are aligned to the human domain using a bottleneck MLP adapter to bridge the cross-embodiment gap. These adapted features are processed via attention mechanisms and integrated with segmentation and hand pose data to construct relational scene graphs capturing intricate hand-object associations. Ultimately, a language-conditioned decision model utilizing RoBERTa evaluates these structured interaction tokens to determine the optimal cross-hand assignment for bimanual task execution.

descriptions. For notational brevity, we omit the subscript i when discussing a single triplet (H, R, L).Given a sampling operator Υ that extracts T frames, we obtain the spatial-temporal feature representations utilizing the frozen backbone $\mathcal{F}$:

$$h^f = \mathcal{F}(\Upsilon(H)), \qquad r^f = \mathcal{F}(\Upsilon(R)) \tag{9}$$

where $h^f, r^f \in \mathbb{R}^{\hat{T}\times\hat{H}\times\hat{W}\times\hat{C}}$ are the unadapted features for the human and robot videos, respectively. Because the backbone $\mathcal{F}$ is optimized on human data, it robustly extracts dynamic visual semantics from the human domain (H). However, applying this frozen model directly to the robot domain (R) is inherently suboptimal; the severe morphological and environmental domain gap prevents the extraction of aligned semantics without targeted adaptation.

To bridge this cross-embodiment gap without catastrophically forgetting human-centric priors, our HR-Align framework employs parameter-efficient fine-tuning (PEFT). We construct an adapted network, $\mathcal{T}$, by integrating lightweight, learnable adapter modules directly into the frozen pre-trained model $\mathcal{F}$. The adapted robot video features are thus computed as:

$$r^t = \mathcal{T}(\Upsilon(R)) = \mathcal{F}_{Adapter}(\Upsilon(R)) \tag{10}$$

where $r^t \in \mathbb{R}^{\hat{T}\times\hat{H}\times\hat{W}\times\hat{C}}$. Structurally, these adapters can be injected at arbitrary depths within $\mathcal{F}$. Given an intermediate layer's output tensor $r^{f,inter} \in \mathbb{R}^{\tilde{T}\times\tilde{H}\times\tilde{W}\times\tilde{C}}$, the adapter applies a residual bottleneck transformation:

$$r^{t,next} = r^{f,inter} + \mathrm{Conv}_{up}(g(\mathrm{Conv}_{down}(r^{f,inter}))) \tag{11}$$

where $r^{t,next}$ is subsequently passed to the downstream layers of $\mathcal{F}$. Here, g denotes a non-linear activation function, while Conv_{down} and Conv_{up} act as down-projection and up-projection convolutional layers, respectively.

To ensure the extracted representations are grounded in the specific semantic context of the required action, we introduce a language-conditioned feature enhancement module. The text description L is encoded via a frozen BERT model and passed through a learnable linear layer to match the channel dimensionality of the video features, yielding a dense query vector $l = Linear(Bert(L)) \in \mathbb{R}^{\hat{C}}$.Using l as the query and the flattened spatial-temporal video tensors as keys and values, we perform an attention-based aggregation. For the reshaped adapted robot features, $r^t \in \mathbb{R}^{(\hat{T}\cdot\hat{H}\cdot\hat{W})\times\hat{C}}$, the task-aware representation $\bar{r}^t$ is

derived via:

$$\begin{aligned} \mathcal{A}^r &= \text{softmax}(r^t \cdot l) \in \mathbb{R}^{\hat{T}\cdot\hat{H}\cdot\hat{W}\times 1} \\ \bar{r}^t &= (r^t)^T \cdot \mathcal{A}^r \in \mathbb{R}^{\hat{C}} \end{aligned} \tag{12}$$

This identical attention mechanism is applied to the frozen streams to yield the task-aware human feature $\bar{h}^f \in \mathbb{R}^{\hat{C}}$ and the unadapted robot feature $\bar{r}^f \in \mathbb{R}^{\hat{C}}$. Note that while this task-aware aggregation is highly effective, as demonstrated in the appendix ablation studies, it serves as a supplementary enhancement rather than the core theoretical contribution. The overall architecture of the attention mechanism as shown in Fig. 3.

The primary objective of HR-Align is to structurally map the robot representations into the rich semantic space of the human domain. The overall algorithem as shown in Algorithm. 1. We achieve this by optimizing the learnable adapter weights using a novel human-robot contrastive alignment loss. This objective enforces two critical structural priors:Intra-pair adaptation: For any given video pair i, the adapted robot representation ($\bar{r}_i^t$) must exhibit a higher semantic similarity to the anchor human feature ($\bar{h}_i^f$) than its unadapted counterpart ($\bar{r}_i^f$) does.Inter-pair discrimination: Paired human-robot features must be drawn closer together in the latent space relative to all other unpaired cross-embodiment samples within the batch.Formally, for a batch of M triplets yielding the task-aware feature sets $\{\bar{h}_i^f, \bar{r}_i^f, \bar{r}_i^t\}_{i=1}^M$, the contrastive alignment loss $\mathcal{L}$ is defined as:

$$\mathcal{L} = \frac{1}{2M}\sum_{i=1}^{M}\left[-\log\frac{\mathcal{S}(\bar{h}_i^f,\bar{r}_i^t)}{\mathcal{S}(\bar{h}_i^f,\bar{r}_i^t)+\mathcal{S}(\bar{h}_i^f,\bar{r}_i^f)+\sum_{j\neq i}^{M}\mathcal{S}(\bar{h}_i^f,\bar{r}_j^t)} - \log\frac{\mathcal{S}(\bar{r}_i^t,\bar{h}_i^f)}{\mathcal{S}(\bar{r}_i^t,\bar{h}_i^f)+\mathcal{S}(\bar{r}_i^f,\bar{h}_i^f)+\sum_{j\neq i}^{M}\mathcal{S}(\bar{r}_i^t,\bar{h}_j^f)}\right] \tag{13}$$

Algorithm 1 HR-Align: Human-Robot Contrastive Alignment for VLM Adaptation

Require: Pre-trained frozen backbone $\mathcal{F}$, Frozen BERT text encoder; Batch of M human-robot video pairs and text tasks $\mathcal{B} = \{H_i, R_i, L_i\}_{i=1}^M$; Frame sampling operator Υ (samples T frames); Temperature factor τ, Learning rate η

Ensure: Adapted model $\mathcal{T}$ with optimized adapter weights $\theta_{adapter}$

Initialize: Insert randomly initialized adapter modules into $\mathcal{F}$ to form $\mathcal{T}$
loop until convergence
  Sample mini-batch $\mathcal{B} = \{H_i, R_i, L_i\}_{i=1}^M$
  for $i = 1$ **to** M **do**
    $h_i^f \leftarrow \mathcal{F}(\Upsilon(H_i))$
    $r_i^f \leftarrow \mathcal{F}(\Upsilon(R_i))$
    $r_i^t \leftarrow \mathcal{T}(\Upsilon(R_i))$
      $r^{t,next} = r^{f,inter} + \text{Conv}_{up}(g(\text{Conv}_{down}(r^{f,inter})))$
    $l_i \leftarrow Linear(Bert(L_i))$
    Function TASKAWARE($v \in \{h_i^f, r_i^f, r_i^t\}$, l_i)
    $\mathcal{A} \leftarrow \text{softmax}(v \cdot l_i)$
    $\bar{v} \leftarrow v^T \cdot \mathcal{A}$
    return $\bar{v} \in \mathbb{R}^{\hat{C}}$
    End Function
    $\bar{h}_i^f \leftarrow$ TASKAWARE(h_i^f, l_i)
    $\bar{r}_i^f \leftarrow$ TASKAWARE(r_i^f, l_i)
    $\bar{r}_i^t \leftarrow$ TASKAWARE(r_i^t, l_i)
  end for
  $\mathcal{L}_{H\to R} \leftarrow 0, \quad \mathcal{L}_{R\to H} \leftarrow 0$
  for $i = 1$ **to** M **do**
    $\text{neg}_i^f \leftarrow \mathcal{S}(\bar{h}_i^f, \bar{r}_i^f)$
    $\text{denom}_H \leftarrow \text{pos}_i + \text{neg}_i^f + \sum_{j\neq i}^M \mathcal{S}(\bar{h}_i^f, \bar{r}_j^t)$
    $\text{denom}_R \leftarrow \text{pos}_i + \text{neg}_i^f + \sum_{j\neq i}^M \mathcal{S}(\bar{r}_i^t, \bar{h}_j^f)$
    $\mathcal{L}_{H\to R} \leftarrow \mathcal{L}_{H\to R} - \log(\text{pos}_i/\text{denom}_H)$
    $\mathcal{L}_{R\to H} \leftarrow \mathcal{L}_{R\to H} - \log(\text{pos}_i/\text{denom}_R)$
  end for
  $\mathcal{L} \leftarrow \frac{1}{2M}\left(\mathcal{L}_{H\to R} + \mathcal{L}_{R\to H}\right)$
  $\theta_{adapter} \leftarrow \theta_{adapter} - \eta\nabla_{\theta_{adapter}}\mathcal{L}$
end loop

where the similarity function is given by the scaled dot-product $\mathcal{S}(x,y) = \exp(x^T y/\tau)$, with τ serving as the temperature hyperparameter. By minimizing this loss, the adapters learn to project robotic visual dynamics into a domain-agnostic semantic space. This mechanism simultaneously mitigates the human-robot domain discrepancy and eliminates the need to iteratively customize and train models from scratch for distinct downstream robotic environments.

2.4 Control Barrier Function

Let us consider a nonlinear, continuous-time dynamical system expressed in control-affine form:

$$\dot{\mathbf{z}} = f(\mathbf{z}) + g(\mathbf{z})\mathbf{u} \tag{14}$$

where $\mathbf{z} \in \mathcal{Z} \subseteq \mathbb{R}^n$ denotes the system state and $\mathbf{u} \in \mathcal{U} \subseteq \mathbb{R}^m$ represents the control input. The drift and control vector fields, $f : \mathbb{R}^n \to \mathbb{R}^n$ and $g : \mathbb{R}^n \to \mathbb{R}^{n\times m}$ respectively, are assumed to be locally Lipschitz continuous to ensure the uniqueness of solutions.In the context of control theory, guaranteeing system safety is mathematically equivalent to ensuring the forward invariance of a designated safe operating region, $\mathcal{C} \subset \mathcal{Z}$. This safe set can be formally defined as the zero-superlevel set of a continuously differentiable Control Barrier Function (CBF), $h(\mathbf{z}) : \mathbb{R}^n \to \mathbb{R}$. To strictly maintain the system state within $\mathcal{C}$, any admissible control policy must satisfy the following derivative condition:

$$\dot{h}(\mathbf{z},\mathbf{u}) \geq -\alpha(h(\mathbf{z})) \tag{15}$$

where α is a strictly increasing, extended class $\mathcal{K}_\infty$ function. Enforcing this inequality guarantees that the set $\mathcal{C}$ remains forward-invariant[33].To synthesize a safe control signal in real-time, the condition in Eq. 15 is characteristically embedded as a linear constraint within a convex Quadratic Program (QP). This CBF-QP framework functions as a minimally invasive safety filter, projecting a nominal, task-oriented—but potentially unsafe—control signal $\mathbf{u}_{\text{nom}}$ into the set of verified safe actions:

$$\begin{aligned} \underset{\mathbf{u}\in\mathcal{U}}{\text{minimize}} \quad & |\mathbf{u} - \mathbf{u}_{\text{nom}}|_2^2 \\ \text{subject to} \quad & L_f h(\mathbf{z}) + L_g h(\mathbf{z})\mathbf{u} \geq -\alpha\left(h(\mathbf{z})\right) \end{aligned} \tag{16}$$

where $L_f h(\mathbf{z})$ and $L_g h(\mathbf{z})$ denote the Lie derivatives of the barrier function along the vector fields f and g, expanding the total time derivative to $\dot{h}(\mathbf{z},\mathbf{u}) = L_f h(\mathbf{z}) + L_g h(\mathbf{z})\mathbf{u}$. The formulation in Eq. 16 inherently assumes that the CBF possesses a relative degree of one, meaning the control input $\mathbf{u}$ explicitly manifests in the first time-derivative of $h(\mathbf{z})$. However, for mechanical systems, position-based safety constraints frequently exhibit a relative degree of two[34]. Under these circumstances, $L_g h(\mathbf{z}) = \mathbf{0}$, necessitating a second differentiation with respect to time to expose the actuation term:

$$\ddot{h}(\mathbf{z},\mathbf{u}) = L_f^2 h(\mathbf{z}) + L_g L_f h(\mathbf{z})\mathbf{u} \tag{17}$$

where $L_g L_f h(\mathbf{z}) \neq \mathbf{0}$.To accommodate these higher-order dynamics without sacrificing the convexity of the QP filter, we utilize the High-Order Control Barrier Function (HOCBF) methodology[35]. By constructing an auxiliary scalar function $h_2(\mathbf{z}) = L_f h(\mathbf{z}) + \alpha(h(\mathbf{z}))$, the safety constraint for a relative-degree-two system is rigorously reformulated as:

$$L_f h_2(\mathbf{z}) + L_g h_2(\mathbf{z})\mathbf{u} \geq -\alpha_2\left(h_2(\mathbf{z})\right) \tag{18}$$

where α_2 is an additional class $\mathcal{K}_\infty$ function. Substituting Eq. 18 into the QP objective allows the framework to enforce continuous-time safety limits on systems governed by second-order kinematics.

To mitigate potential infeasibilities that may arise from conflicting constraints or strict actuation limits, it is standard practice to relax the rigid optimization problem presented in Eq. 16. This relaxation is achieved by introducing a non-negative slack variable vector, $\mathbf{t}$, which softens the barrier constraint to guarantee that the Quadratic Program (QP) remains recursively solvable. To ensure that the system deviates from strict safety bounds only when absolutely necessary, a large weighting parameter, ρ, is applied to penalize the slack term within the objective function. Consequently, the relaxed CBF-QP is formulated as follows:

$$\begin{aligned} \underset{\mathbf{u},\mathbf{t}}{\text{minimize}} \quad & |\mathbf{u} - \mathbf{u}_{\text{nom}}|_2^2 + \rho^T \mathbf{t} \\ \text{subject to} \quad & L_f h(\mathbf{z}) + L_g h(\mathbf{z})\mathbf{u} \geq -\alpha\left(h(\mathbf{z})\right) - \mathbf{t} \quad \mathbf{t} \geq \mathbf{0} \end{aligned} \tag{19}$$

For velocity-controlled manipulators, generating a safe and task-consistent joint velocity command, $\dot{\mathbf{q}}^*$, involves a systematic three-stage pipeline. The process begins by computing the nominal task velocities necessary to minimize tracking errors within the established task hierarchy.

First, the proportional operational-space twist command, $\nu \in \mathbb{R}^6$, is defined as:

$$\nu = \nu_{\text{des}} - K_{po} \begin{bmatrix} \mathbf{x}_p - \mathbf{x}_{p,\text{des}} \\ \delta\phi \end{bmatrix} \tag{20}$$

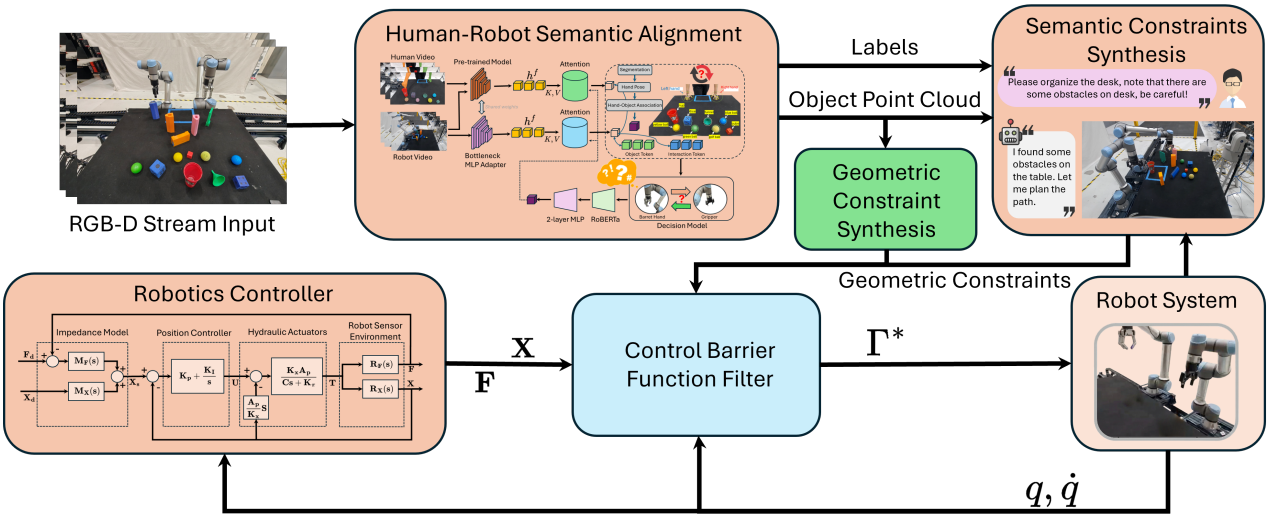

Figure 4. System architecture of the proposed GF-VLA framework integrated with a semantic safety filter. Visual inputs and object point clouds are processed to derive geometric boundaries and Large Language Model (LLM)-inferred semantic constraints. These constraints are formally synthesized into an Operational Space Control Barrier Function (OSCBF) filter, which projects nominal task commands into strictly safe, dynamically feasible joint torques (Γ^*) for the dual-arm system.

Simultaneously, a proportional joint-space velocity command, $\dot{\mathbf{q}}_N \in \mathbb{R}^{n_q}$, is computed and projected into the null-space of the primary operational task to ensure it does not interfere with end-effector execution:

$$\dot{\mathbf{q}}_N = N(\mathbf{q})\left(\dot{\mathbf{q}}_{\text{des}} - K_{pj}(\mathbf{q} - \mathbf{q}_{\text{des}})\right) \tag{21}$$

In these formulations, K_{po} and K_{pj} act as the proportional gain matrices for the operational and joint spaces, respectively. The term $\delta\phi$ denotes the instantaneous angular error between the current rotation matrix, R, and the desired rotation matrix, R_{des}. By extracting the i^{th} column vectors ($\mathbf{r}_i$) from these matrices, the orientation error is efficiently reconstructed as:

$$\delta\phi = -\frac{1}{2}\sum_{i=1}^{3}(\mathbf{r}_i \times \mathbf{r}_{i,\text{des}}) \tag{22}$$

These components are then superimposed using the Jacobian pseudoinverse, $J^{\#}(\mathbf{q})$, to formulate the nominal—yet potentially unsafe—joint velocity command:

$$\dot{\mathbf{q}}_{\text{nom}} = J^{\#}(\mathbf{q})\nu + \dot{\mathbf{q}}_N \tag{23}$$

To integrate safety guarantees via Control Barrier Functions (CBFs), the system dynamics must be expressed in a control-affine format. For a purely kinematic, velocity-controlled framework, the state vector is defined as the joint positions ($\mathbf{z} = \mathbf{q} \in \mathbb{R}^{n_q}$) and the control input as the joint velocities ($\mathbf{u} = \dot{\mathbf{q}} \in \mathbb{R}^{n_q}$). This yields a direct first-order model:

$$\dot{\mathbf{z}} = \mathbf{u} \tag{24}$$

Using these dynamics alongside a defined valid CBF, $h(\mathbf{z})$, an Operational Space Control Barrier Function (OSCBF) safety filter is formulated as a Quadratic Program (QP). The objective is to calculate a safe velocity command, $\dot{\mathbf{q}}$, that strictly satisfies the CBF constraint while minimizing deviations from the nominal operational and null-space velocities:

$$\begin{aligned} \underset{\dot{\mathbf{q}}}{\text{minimize}} \quad & \|W_j(\dot{\mathbf{q}}_N - \dot{\mathbf{q}}_{\text{N,nom}})\|_2^2 + \|W_o(\nu - \nu_{\text{nom}})\|_2^2 \\ \text{subject to} \quad & L_f h(\mathbf{z}) + L_g h(\mathbf{z})\mathbf{u} \geq -\alpha(h(\mathbf{z})) \end{aligned} \tag{25}$$

The positive-definite diagonal matrices W_j and W_o serve as tuning weights to prioritize error minimization across different operational axes (e.g., translation versus rotation) or joint types.To solve this efficiently using standard QP solvers, the objective function is mathematically equivalent to minimizing the mapped velocity deviations:

$$\|W_o J(\mathbf{q})(\dot{\mathbf{q}} - \dot{\mathbf{q}}_{\text{nom}})\|_2^2 + \|W_j N(\mathbf{q})(\dot{\mathbf{q}} - \dot{\mathbf{q}}_{\text{nom}})\|_2^2 \tag{26}$$

This can be systematically restructured into the canonical QP objective format, $\frac{1}{2}\mathbf{x}^T P_{QP}\mathbf{x} + \mathbf{q}_{QP}^T\mathbf{x}$, by defining the Hessian matrix, P_{QP}, and the gradient vector, $\mathbf{q}_{QP}$, as follows:

$$\begin{aligned} P_{QP} &= J^T W_o^T W_o J + N^T W_j^T W_j N \\ \mathbf{q}_{QP}^T &= -\dot{\mathbf{q}}_{\text{nom}}^T P_{QP} \end{aligned} \tag{27}$$

While the preceding formulation explicitly addresses a strictly two-task hierarchy ($n_t = 2$), the objective function seamlessly scales to accommodate complex architectures comprising n_t tasks. This is achieved by penalizing the deviation of the velocity, $\mathbf{v}_i$ (in either joint or operational space), from its nominal reference for each sequential task i:

$$\sum_{i=1}^{n_t} \| W_i \left(\mathbf{v}_i(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{v}_{i,\text{nom}}(\mathbf{q}, \dot{\mathbf{q}}_{\text{nom}}) \right) \|_2^2 \tag{28}$$

Finally, strict hardware limitations are enforced by adjoining linear inequality constraints to the QP to bound the allowable joint velocities:

$$\dot{\mathbf{q}}_{\min} \leq \dot{\mathbf{q}} \leq \dot{\mathbf{q}}_{\max} \tag{29}$$

For torque-controlled manipulators, the derivation of a safe control input, Γ^*, necessitates a structured, three-tier methodology: (1) calculation of nominal generalized forces to minimize tracking errors within the defined task hierarchy; (2) projection and mapping of these generalized forces into a nominal (albeit potentially unsafe) joint torque command; and (3) application of an Operational Space Control Barrier Function (OSCBF) via a Quadratic Program (QP) to guarantee safety and yield the final safe torque command, Γ^* as shown in Fig. 4.

The procedure initiates with the computation of nominal generalized forces required for both the operational-space and joint-space tasks. The operational-space wrench command, $\mathcal{F} \in \mathbb{R}^6$, and the unconstrained joint-space torque command, $\Gamma_0 \in \mathbb{R}^{n_q}$, are formulated as:

$$\begin{aligned} \mathcal{F} &= \Lambda(\mathbf{q})\dot{\nu} \\ \Gamma_0 &= M(\mathbf{q})\ddot{\mathbf{q}} \end{aligned} \tag{30}$$

where $\Lambda(\mathbf{q})$ denotes the operational-space inertia matrix, and $M(\mathbf{q})$ is the joint-space inertia matrix. The desired operational-space acceleration ($\dot{\nu}$) and joint-space acceleration ($\ddot{\mathbf{q}}$) are synthesized utilizing a proportional-derivative (PD) control strategy applied to the respective task error dynamics:

$$\begin{aligned} \dot{\nu} &= \dot{\nu}_{\text{des}} - K_{po} \begin{bmatrix} \mathbf{x}_p - \mathbf{x}_{p,\text{des}} \\ \delta\phi \end{bmatrix} - K_{do}(\nu - \nu_{\text{des}}) \\ \ddot{\mathbf{q}} &= \ddot{\mathbf{q}}_{\text{des}} - K_{pj}(\mathbf{q} - \mathbf{q}_{\text{des}}) - K_{dj}(\dot{\mathbf{q}} - \dot{\mathbf{q}}_{\text{des}}) \end{aligned} \tag{31}$$

In these expressions, K_{po}, K_{do} and K_{pj}, K_{dj} represent the proportional and derivative gain matrices for the operational and joint spaces, respectively. $\mathbf{x}_p$ and $\mathbf{x}_{p,\text{des}}$ are the current and desired Cartesian positions, while $\delta\phi$ encapsulates the orientation error.To enforce the established strict task hierarchy, ensuring that secondary joint-posture objectives do not interfere with the primary end-effector task, the joint-space torque command is explicitly projected into the dynamically consistent null-space of the operational-space task:

$$\Gamma_N = N^T(\mathbf{q})\Gamma_0 \tag{32}$$

where $N(\mathbf{q})$ is the dynamically consistent null-space projection matrix.Aggregating these prioritized components and incorporating active compensation for Coriolis, centrifugal, and gravitational effects yields the nominal, combined joint torque command:

$$\Gamma_{\text{nom}} = J^T(\mathbf{q})\mathcal{F} + \Gamma_N + \mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) \tag{33}$$

The overall architecture of the CBF with control scheme as shown in Algorithm 2.

3 Experiment Validation

3.1 Experimental Setup

To empirically validate the proposed Vision-Language Model (VLM) directed Control Barrier Function (CBF) framework, we engineered a heterogeneous, dual-arm manipulation workspace. This physical testbed is meticulously designed to evaluate the system's capacity to seamlessly bridge low-frequency, high-level semantic reasoning with high-frequency, low-level safe optimal control in complex, unstructured environments.

Algorithm 2 Torque-Level Operational Space Impedance Control with CBF (OSCBF)

Require: Current state $\mathbf{z} = [\mathbf{q}^T, \dot{\mathbf{q}}^T]^T$; Operational task targets $(\mathbf{x}_{p,\text{des}}, \nu_{\text{des}}, \dot{\nu}_{\text{des}})$; Null-space joint targets $(\mathbf{q}_{\text{des}}, \dot{\mathbf{q}}_{\text{des}}, \ddot{\mathbf{q}}_{\text{des}})$
Ensure: Safe joint torque command Γ^*

Step 1: Compute Nominal Generalized Forces

$M(\mathbf{q}), \mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}), \mathbf{g}(\mathbf{q}) \leftarrow \text{ComputeRigidBodyDynamics}(\mathbf{q}, \dot{\mathbf{q}})$

$J(\mathbf{q}), \Lambda(\mathbf{q}) \leftarrow \text{ComputeKinematics}(\mathbf{q})$

$\delta\phi \leftarrow -\frac{1}{2}\sum_{i=1}^{3}(\mathbf{r}_i \times \mathbf{r}_{i,\text{des}})$

$\dot{\nu} \leftarrow \dot{\nu}_{\text{des}} - K_{po}\begin{bmatrix}\mathbf{x}_p - \mathbf{x}_{p,\text{des}} \\ \delta\phi\end{bmatrix} - K_{do}(\nu - \nu_{\text{des}})$

$\ddot{\mathbf{q}} \leftarrow \ddot{\mathbf{q}}_{\text{des}} - K_{pj}(\mathbf{q} - \mathbf{q}_{\text{des}}) - K_{dj}(\dot{\mathbf{q}} - \dot{\mathbf{q}}_{\text{des}})$

$\mathcal{F} \leftarrow \Lambda(\mathbf{q})\dot{\nu}$

$\Gamma_0 \leftarrow M(\mathbf{q})\ddot{\mathbf{q}}$

Step 2: Map to Nominal Joint Torque Command

$N(\mathbf{q}) \leftarrow I - J^T(\mathbf{q})\bar{J}^T(\mathbf{q})$

$\Gamma_N \leftarrow N^T(\mathbf{q})\Gamma_0$

$\Gamma_{\text{nom}} \leftarrow J^T(\mathbf{q})\mathcal{F} + \Gamma_N + \mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q})$

Step 3: Solve OSCBF Quadratic Program

$P_{QP} \leftarrow NM^{-T}W_j^T W_j M^{-1}N^T + M^{-T}J^T W_o^T W_o J M^{-1}$

$\mathbf{q}_{QP}^T \leftarrow -\Gamma_{\text{nom}}^T P_{QP}$

$\mathcal{C}_{\text{QP}} \leftarrow \emptyset$

$\mathcal{C}_{\text{QP}} \leftarrow \mathcal{C}_{\text{QP}} \cup \{L_f h(\mathbf{z}) + L_g h(\mathbf{z})\Gamma \geq -\alpha(h(\mathbf{z}))\}$

$\mathcal{C}_{\text{QP}} \leftarrow \mathcal{C}_{\text{QP}} \cup \{\Gamma_{\min} \leq \Gamma \leq \Gamma_{\max}\}$

$\mathcal{C}_{\text{QP}} \leftarrow \mathcal{C}_{\text{QP}} \cup \{\mathcal{F}_{c,\min} \leq \bar{J}^T(\mathbf{q})(\Gamma - \mathbf{c} - \mathbf{g}) \leq \mathcal{F}_{c,\max}\}$

$\Gamma^* \leftarrow \text{SolveQP}(P_{QP}, \mathbf{q}_{QP}, \mathcal{C}_{\text{QP}})$

return Γ^*

The manipulation platform comprises a heterogeneous dual-arm robotic system, deliberately selected to demonstrate the framework's scalability across varying kinematic chains, payload capacities, and end-effector morphologies. The primary manipulator, a Universal Robots UR5e, known for its high kinematic repeatability, is equipped with a Robotiq 2F-85 adaptive two-finger gripper. This arm is primarily designated for executing fine-grained semantic tasks, target acquisition, and precision placements dictated by the VLM. Operating in tandem is a secondary manipulator, a Universal Robots UR10e, which is fitted with a highly dexterous, three-fingered BarrettHand. The UR10e's extended reach and superior payload capacity are used to execute complex caging grasps, manipulate heavy occluding objects, and introduce dynamic, controlled spatial constraints. In this capacity, the UR10e serves as an active, semantically meaningful obstacle to rigorously test the real-time collision-avoidance capabilities of the multi-agent CBF algorithm. Both manipulators are securely mounted to a rigid, vibration-damped optical table, with their base frames separated by exactly 1.2 meters to create a highly coupled, shared workspace volume that demands continuous spatial deconfliction.

High-fidelity, real-time spatial data is critical for accurately translating VLM semantic outputs into rigorous geometric safety boundaries for the subsequent CBF optimization. To achieve this, visual perception is driven by an Intel RealSense D405 RGB-D camera subsystem. Engineered for high-precision, sub-millimeter depth sensing at close range, the D405 delivers the optimal resolution for generating dense, noise-filtered point clouds and accurate semantic segmentation masks. We employ a distributed dual-camera topology to maximize observational coverage and minimize occlusions. The primary D405 is rigidly mounted on an overhead aluminum extrusion profile, oriented downward at a 45-degree angle to provide a global, unoccluded view of the shared workspace. A supplementary D405 is deployed in an eye-in-hand configuration on the UR5e wrist, dynamically capturing granular geometric details of the target objects immediately prior to grasping and manipulation. To ensure absolute baseline stability and to severely mitigate thermal drift in the D405 stereo depth-matching algorithm—which is highly sensitive to environmental fluctuations—the ambient laboratory environment is continuously maintained at a strict 22°C. Furthermore, intrinsic and extrinsic camera parameters are calibrated offline using a standardized ChArUco board procedure, ensuring precise spatial consistency between the local camera frames and the global robotic base coordinate systems.

The computational architecture is explicitly bifurcated to handle the stark differences in processing frequency, latency requirements, and resource demands between the large-scale deep learning models and the control-theoretic safety filters. The Semantic Reasoning Node, which governs the high-level intent generation, executes VLM inference, zero-shot semantic segmentation, and point cloud processing on a dedicated high-performance workstation equipped with an Intel Core i9-13900K processor, 128 GB of DDR5 RAM, and dual NVIDIA RTX 4090 GPUs. This node processes incoming D405 RGB-D streams,

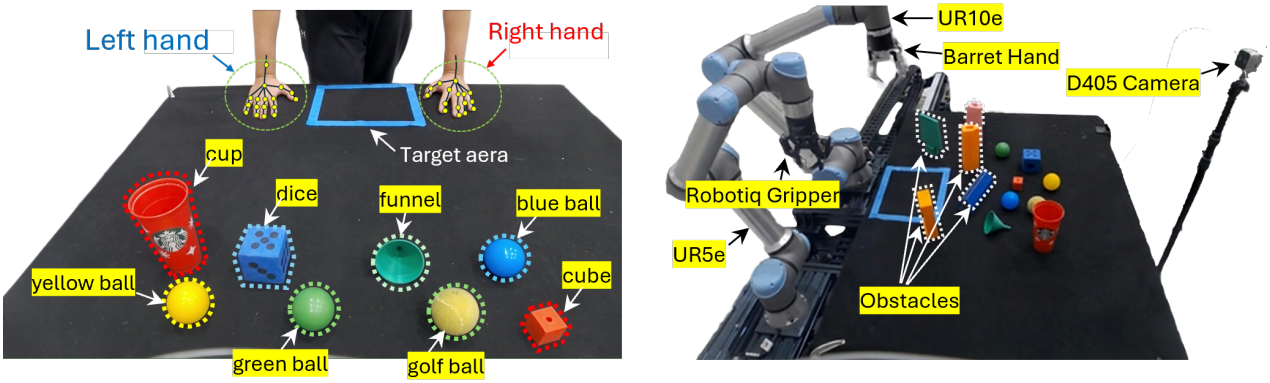

Figure 5. Experimental setup for the heterogeneous dual-arm manipulation workspace. **Left**: the system utilizes a Universal Robots UR5e equipped with a Robotiq 2F-85 adaptive two-finger gripper, strategically designated for fine-grained semantic tasks and precision target acquisition among diverse objects (e.g., cups, dice, and textured balls). **Right**: a Universal Robots UR10e fitted with a dexterous BarrettHand operates within a structurally constrained zone, navigating around a deliberately constructed obstacle field. The shared, tightly coupled workspace is continuously monitored by an overhead Intel RealSense D405 camera to generate high-fidelity, dense semantic maps.

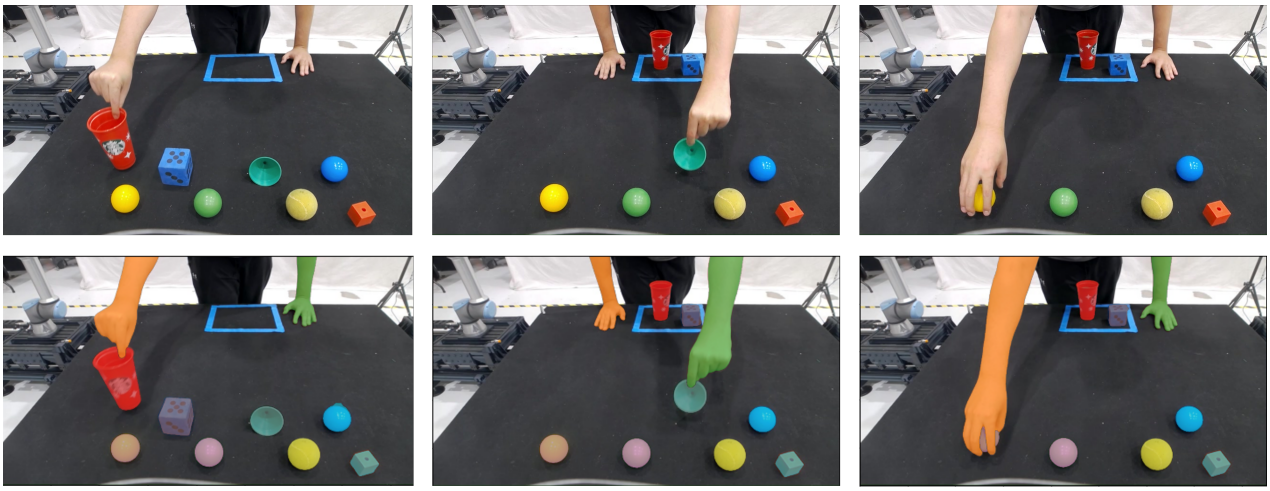

Figure 6. Demonstration of the zero-shot perception and cross-hand assignment pipeline. The system utilizes semantic segmentation algorithms to process RGB-D inputs, isolating individual objects and extracting their spatial features. These semantic abstractions are fed into the GF-VLA architecture to determine the optimal manipulator (UR5e or UR10e) and bimanual coordination strategy required for the targeted manipulation task.

iteratively querying the VLM to generate nominal operational-space waypoints and to extract 3D semantic bounding boxes of environmental obstacles at approximately 5 Hz. Conversely, the Real-Time Control Node is dedicated strictly to low-level optimization. The Operational Space Control Barrier Function (OSCBF) Quadratic Program (QP) is solved on a physically separate computing node running a fully preemptive real-time Linux kernel (Ubuntu 22.04 patched with PREEMPT_RT) operating alongside ROS 2 Humble. This node is designed to ingest the asynchronous nominal waypoints from the Semantic Node and immediately project them into safe, dynamically feasible joint torque commands.

The orchestration among the heterogeneous arms, the perception suite, and the split computing architecture relies on a highly synchronized, low-latency communication protocol over a dedicated local-area network. Joint states, encompassing both positions and velocities for the UR5e and UR10e, are continuously streamed over a Gigabit Ethernet switch via the Universal Robots Real-Time Data Exchange (RTDE) interface at a strict 500 Hz control loop. Simultaneously, the high-level VLM node asynchronously updates the locations and geometries of semantic obstacles, transmitting them as sets of convex polyhedra to the Control Node. The CBF framework then dynamically constructs rigorous mathematical safety constraints encompassing self-collision avoidance between the UR5e, the UR10e, and their respective end-effectors, strict joint limit enforcement, and semantic obstacle evasion. The aggregated OSCBF QP is formulated utilizing the Pinocchio rigid body dynamics library, which rapidly computes the analytical Jacobians, Coriolis and centrifugal force matrices, and mass-inertia matrices required for the control-affine formulation. Finally, the QP is optimally solved using the Operator Splitting Quadratic Program (OSQP) solver at the 500 Hz hardware frequency, ensuring that the VLM's generated intent is executed with strict, formal safety guarantees without violating the physical or dynamic constraints of the dual-arm setup.

The physical hardware configuration shown in Fig. 5 and the zero-shot perception pipeline shown in Fig. 6 are intricately linked to enable the proposed framework's context-aware reasoning. The experimental testbed features a heterogeneous dual-arm architecture, pairing a Universal Robots UR5e equipped with a Robotiq 2F-85 two-finger gripper alongside a UR10e fitted with a dexterous, three-fingered BarrettHand. This deliberately designed workspace is engineered to evaluate the system's scalability across varying end-effector morphologies while introducing complex, dynamic spatial constraints. To bridge this physical environment with the high-level reasoning module, high-fidelity RGB-D streams captured by an overhead Intel RealSense D405 camera are processed through advanced semantic segmentation algorithms. This perceptual front-end effectively isolates individual objects from the unstructured workspace and extracts their critical spatial features. As illustrated in Figure 6, these resulting semantic abstractions are subsequently fed directly into the Graph-Fused VLA (GF-VLA) architecture. By synthesizing the dense visual state with the model's policy generation capabilities, the framework autonomously resolves the optimal cross-hand assignment, dynamically selecting the most appropriate manipulator and bimanual coordination strategy required to safely execute the targeted manipulation task.

3.2 Experimental Results

The empirical results are detailed in the Table. 3 provides a comprehensive quantitative evaluation of task execution success rates across 18 complex, bimanual manipulation scenarios. The baseline VLA model struggles significantly with spatial generalization and optimal arm allocation, yielding an average success rate of only 40.8%. The introduction of the Human-Robot Contrastive Alignment (HR-Align) module mitigates some cross-embodiment discrepancies, improving the average success rate to 54.2%. However, integrating these representations into the proposed Graph-Fused VLA (GF-VLA) architecture yields substantial improvements. While the geometric Control Barrier Function (Geo-CBF) filter raises the success rate to 69.2% by strictly preventing structural collisions, it is the integration of the semantic safety filter (Sem-CBF) that fundamentally maximizes system reliability, achieving a remarkable average success rate of 94.4%. This critical performance increase is particularly evident in contextually hazardous tasks, such as avoiding lit candles or pouring water near electronics. In these highly constrained environments, the Sem-CBF framework explicitly optimizes bimanual coordination and dynamically enforces appropriate cross-hand assignments to ensure successful, collision-free task completion.

Table. 4 presents a rigorous statistical analysis of safety constraint violations during real-world dual-arm manipulation, systematically comparing the unconstrained VLA policy against the nominal geometric filter (Geo-CBF) and the proposed semantic filter (Sem-CBF). The evaluation tracks the mean percentage and standard deviation of temporal steps that violate defined operational boundaries, including semantic constraints ($\mathcal{C}_{sem}$), environmental limits ($\mathcal{C}_{env}$), and self-collision thresholds ($\mathcal{C}_{self}$). The data reveals that while traditional Geo-CBF effectively mitigates purely physical collisions, it catastrophically fails to recognize context-specific risks, such as manipulating a fluid-filled Starbucks cup over electronic devices or moving dense objects like a golf ball over fragile glass. In these high-risk semantic scenarios, the nominal Geo-CBF filter permitted severe constraint violations in up to 64.98% of the execution time steps. Conversely, the proposed GF-VLA framework equipped with the Sem-CBF rigorously enforced zero-violation execution bounds across all tested scenes. By translating LLM-inferred spatial and contextual rules into continuous control bounds, the semantic safety filter successfully prevents geometrically permissible but semantically disastrous behaviors, thereby ensuring absolute spatial and operational safety.

Fig. 7 illustrates the sequential execution of language-conditioned manipulation tasks, presenting recorded time-lapse

Task	VLA (Base)	VLA + HR-Align	GF-VLA (Geo-CBF)	GF-VLA (Sem-CBF)
grasp cup	$45.3_{\pm 6.1}$	$62.0_{\pm 4.0}$	$84.7_{\pm 4.6}$	$90.7_{\pm 6.1}$
grasp funnel	$80.0_{\pm 5.0}$	$87.3_{\pm 2.3}$	$96.0_{\pm 4.0}$	$100.0_{\pm 0.0}$
grasp dice	$77.3_{\pm 4.6}$	$86.0_{\pm 3.0}$	$92.0_{\pm 4.0}$	$98.0_{\pm 2.0}$
grasp balls	$57.3_{\pm 12.9}$	$70.7_{\pm 2.3}$	$80.0_{\pm 4.0}$	$92.0_{\pm 4.0}$
grasp cube	$64.0_{\pm 10.6}$	$71.3_{\pm 2.3}$	$90.0_{\pm 4.0}$	$96.0_{\pm 2.0}$
pick near laptop	$16.7_{\pm 6.1}$	$26.7_{\pm 4.6}$	$34.0_{\pm 8.0}$	$93.3_{\pm 4.2}$
avoid candle	$14.0_{\pm 4.0}$	$22.7_{\pm 2.3}$	$20.0_{\pm 4.0}$	$96.3_{\pm 2.3}$
avoid knife	$18.0_{\pm 5.0}$	$27.3_{\pm 4.3}$	$30.7_{\pm 2.3}$	$94.7_{\pm 4.6}$
pour water	$23.3_{\pm 2.3}$	$30.0_{\pm 5.0}$	$40.0_{\pm 6.0}$	$96.0_{\pm 2.0}$
cross-hand assign	$41.3_{\pm 6.1}$	$62.0_{\pm 6.9}$	$84.7_{\pm 2.3}$	$96.0_{\pm 4.0}$
bimanual handover	$36.7_{\pm 8.3}$	$56.0_{\pm 4.0}$	$76.0_{\pm 4.0}$	$93.3_{\pm 2.3}$
cage large obj	$40.0_{\pm 6.0}$	$58.0_{\pm 5.0}$	$80.0_{\pm 4.0}$	$94.0_{\pm 2.0}$
pinch small obj	$51.3_{\pm 6.1}$	$64.0_{\pm 4.0}$	$87.3_{\pm 4.6}$	$98.0_{\pm 2.0}$
stack cubes	$32.0_{\pm 4.0}$	$57.3_{\pm 8.3}$	$70.7_{\pm 6.1}$	$81.3_{\pm 4.6}$
sort colors	$48.0_{\pm 6.9}$	$66.7_{\pm 4.6}$	$86.0_{\pm 4.0}$	$98.7_{\pm 2.3}$
safe place	$22.0_{\pm 10.6}$	$34.0_{\pm 6.9}$	$47.3_{\pm 9.3}$	$96.0_{\pm 3.9}$
clear clutter	$37.3_{\pm 2.3}$	$52.0_{\pm 4.0}$	$76.0_{\pm 5.0}$	$93.3_{\pm 2.3}$
dual retrieval	$30.0_{\pm 8.0}$	$41.3_{\pm 7.3}$	$70.0_{\pm 6.0}$	$91.3_{\pm 4.3}$
Average success	40.8	54.2 **(+13.4)**	69.2	94.4 **(+25.2)**

Table 3. Quantitative evaluation of task execution success rates across 18 complex dual-arm manipulation scenarios. The table benchmarks baseline VLA models against the proposed Graph-Fused VLA architecture with geometric (Geo-CBF) and semantic (Sem-CBF) safety filters. The inclusion of the semantic safety filter demonstrates a critical performance increase in contextually hazardous tasks (e.g., avoiding lit candles, pouring water near electronics) while optimizing optimal bimanual coordination and cross-hand assignments.

sequences that capture the complete trajectory of the dual-arm system from initial target identification to the final secure grasp. Driven by the proposed framework, the robotic system autonomously evaluates both the inherent geometry of the target objects and the broader semantic context of the task to formulate an execution strategy. This visual-semantic reasoning directly dictates the optimal cross-hand assignment, enabling the system to dynamically select the most appropriate manipulator and end-effector morphology for each specific task—specifically choosing between the UR5e equipped with a two-finger gripper and the UR10e fitted with a three-fingered dexterous hand. The visual sequences demonstrate the framework's robust capacity to seamlessly translate high-level natural language instructions, such as "Pick up the Starbucks cup" or "Pick up the dice," into highly reliable physical actions across a diverse array of target items. Ultimately, these successions highlight the system's ability to bridge advanced semantic intent with optimal, hardware-aware continuous control.

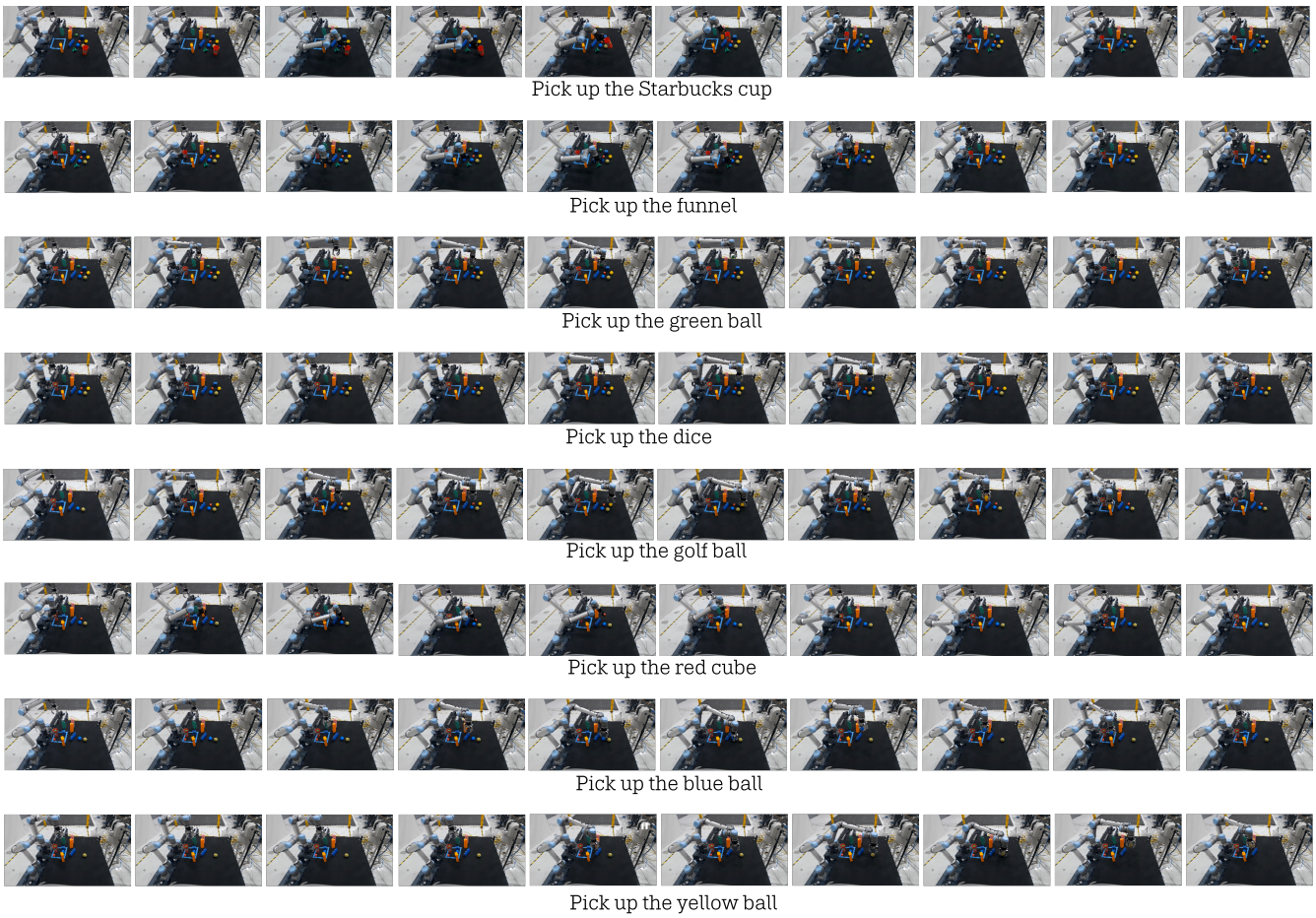

Figure 7. Sequential execution of language-conditioned manipulation tasks. The recorded time-lapse sequences illustrate the complete manipulation trajectory—from initial target identification to the final secure grasp, across a diverse set of objects. Driven by the proposed framework, the dual-arm system autonomously evaluates the target geometry and task context to dictate the optimal cross-hand assignment , dynamically selecting the most appropriate manipulator and end-effector morphology (i.e., the UR5e with a two-finger gripper or the UR10e with a three-fingered dexterous hand) for each specific task.

4 Conclusion

In this work, we presented a unified framework that seamlessly integrates Graph-Fused Vision-Language-Action (GF-VLA) models with a novel semantic Control Barrier Function (Sem-CBF) to achieve semantically safe and optimal dual-arm robotic control. Recognizing the critical limitations of existing end-to-end VLA models in enforcing rigorous motion safeguards, our approach bridges the fundamental gap between high-level contextual reasoning and low-level continuous control. By extracting task-critical spatial and relational features from visual demonstrations and encoding them into temporally ordered scene graphs, the proposed GF-VLA architecture successfully resolves complex spatial ambiguities. This structured representation

Table 4. Statistical analysis of safety constraint violations during real-world dual-arm manipulation. The table reports the mean percentage and standard deviation of temporal steps violating defined operational constraints, including semantic ($\mathcal{C}_{sem}$), environmental ($\mathcal{C}_{env}$), self-collision ($\mathcal{C}_{self}$), and hardware limits ($\mathcal{C}_{lim}$). Performance is systematically compared across unconstrained VLA policy execution, a nominal geometric-only Control Barrier Function (Geo-CBF) filter, and the proposed semantic CBF formulation (Sem-CBF). Manipulation scenarios involving contextually hazardous relationships (e.g., manipulating a "Starbucks cup" over electronics) demonstrate that the proposed GF-VLA with the semantic safety filter rigorously ensures zero-violation execution bounds, successfully mitigating context-specific risks that traditional geometric formulations fail to identify.

Workspace Scene	**Manipulated Target**[†]	**VLA (No Filter)**	**GF-VLA (Geo-CBF)**	**GF-VLA (Sem-CBF)**
`{target area, empty containers}`	`yellow ball`	11.06% ± 13.60%	**0.00% ± 0.00%**	**0.00% ± 0.00%**
	`Starbucks cup`	70.37% ± 23.51%	64.98% ± 33.42%	**0.00% ± 0.00%**
`{laptop, electronic devices}`	`dice`	36.29% ± 18.29%	**0.00% ± 0.00%**	**0.00% ± 0.00%**
	`funnel`	65.21% ± 14.20%	51.33% ± 27.85%	**0.00% ± 0.00%**
	`Starbucks cup`	59.40% ± 12.02%	41.90% ± 25.46%	**0.00% ± 0.00%**
`{fragile glass, target area}`	`blue ball`	28.07% ± 14.77%	**0.00% ± 0.00%**	**0.00% ± 0.00%**
	`red cube`	50.33% ± 9.44%	49.89% ± 9.04%	**0.00% ± 0.00%**
	`golf ball`	49.07% ± 16.16%	30.85% ± 10.53%	**0.00% ± 0.00%**

empowers the language-conditioned transformer to generate highly reliable Cartesian commands and dictate optimal cross-hand assignments for bimanual coordination.

Ultimately, this framework marks a critical step toward the reliable deployment of autonomous, multi-arm robotic systems in unstructured, human-centric environments, where both dexterous task execution and strict adherence to semantic safety are paramount. Future work will explore extending this architecture to highly dynamic environments with moving human collaborators and integrating multimodal perception, such as tactile feedback, to further enhance the system's context-aware reasoning and fine-grained manipulation capabilities.

5 Funding

Not applicable.

6 Data availability

We publish our code for this project on GitHub as supplemental material for this work, see details in:
https://github.com/gaolongsen/GFVLA_CBF

References

1. Gao, J. *et al.* Physically grounded vision-language models for robotic manipulation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 12462–12469 (IEEE, 2024).
2. Buhl, J. F. *et al.* A dual-arm collaborative robot system for the smart factories of the future. *Procedia manufacturing* **38**, 333–340 (2019).
3. Li, S. *et al.* Information-theoretic graph fusion with vision-language-action model for policy reasoning and dual robotic control. *Inf. Fusion* 104193 (2026).
4. Bu, Q. *et al.* Univla: learning to act anywhere with task-centric latent actions. *arXiv:2505.06111* (2025).
5. Du, Y. *et al.* Learning universal policies via text-guided video generation. In *NeurIPS*, vol. 36, 9156–9172 (2023).
6. Zhang, J. *et al.* Integrating a pipette into a robot manipulator with uncalibrated vision and tcp for liquid handling. *IEEE Transactions on Autom. Sci. Eng.* **21**, 5503–5522 (2023).
7. Mao, Y., Zhang, Y. & Gao, L. Liquid-augmented mpc in quadrupedal robot for disturbance learning. *Electronics* (2025).
8. Brunke, L. *et al.* Semantically safe robot manipulation: From semantic scene understanding to motion safeguards. *IEEE Robotics Autom. Lett.* (2025).
9. Zhang, Y. *et al.* Invertible liquid neural network-based learning of inverse kinematics and dynamics for robotic manipulators. *Sci. Reports* **15**, 42311 (2025).
10. Park, S. *et al.* Saliency-aware quantized imitation learning for efficient robotic control. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 13140–13150 (2025).
11. Ames, A. D. *et al.* Control barrier functions: Theory and applications. In *2019 18th European Control Conference (ECC)*, 3420–3431, DOI: 10.23919/ECC.2019.8796030 (IEEE, 2019).
12. Zhang, S., So, O., Garg, K. & Fan, C. Gcbf+: A neural graph control barrier function framework for distributed safe multiagent control. *IEEE Transactions on Robotics* **41**, 1533–1552 (2025).
13. Marvi, Z. & Kiumarsi, B. Safe reinforcement learning: A control barrier function optimization approach. *Int. J. Robust Nonlinear Control.* **31**, 1923–1940 (2021).
14. Liu, Z., Liu, Q., Xu, W., Wang, L. & Zhou, Z. Robot learning towards smart robotic manufacturing: A review. *Robotics Comput. Manuf.* **77**, 102360 (2022).
15. Schmidgall, S., Kim, J. W., Kuntz, A., Ghazi, A. E. & Krieger, A. General-purpose foundation models for increased autonomy in robot-assisted surgery. *Nat. Mach. Intell.* **6**, 1275–1283 (2024).
16. Florence, P. *et al.* Implicit behavioral cloning. In *Conference on robot learning*, 158–168 (PMLR, 2022).
17. Li, Y. Deep reinforcement learning: Advancements, limitations, and real-world applications. *arXiv preprint arXiv:1701.07274* (2018).
18. Zhao, W. *et al.* Vlas: vision-language-action model with speech instructions for customized robot manipulation. In *ICLR* (2025).
19. Merlo, R., Doughty, H. *et al.* Information-theoretic physical reasoning for robotic manipulation. *IEEE Transactions on Pattern Analysis Mach. Intell.* (2025). To appear.
20. Zhang, F. & Gienger, M. Affordance-based robot manipulation with flow matching (2024). 2409.01083.
21. Lee, B. & Lee, C. G. Collision-free motion planning of two robots. *IEEE Transactions on Syst. Man, Cybern.* **17**, 21–32, DOI: 10.1109/TSMC.1987.289330 (1987).
22. Bounini, F., Gingras, D., Pollart, H. & Gruyer, D. Modified artificial potential field method for online path planning applications. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, 180–185, DOI: 10.1109/IVS.2017.7995717 (IEEE, 2017).
23. Kingston, Z., Moll, M. & Kavraki, L. E. Sampling-based methods for motion planning with constraints. *Annu. Rev. Control. Robotics, Auton. Syst.* **1**, 159–185, DOI: 10.1146/annurev-control-060117-105226 (2018).
24. Mosavi, A. & Vaezipour, A. Reactive search optimization; application to multiobjective optimization problems. *Appl. Math.* **03**, 1572–1582, DOI: 10.4236/am.2012.330217 (2012).
25. Xiao, W. & Belta, C. High-order control barrier functions. *IEEE Transactions on Autom. Control.* **67**, 3655–3662, DOI: 10.1109/TAC.2021.3105491 (2022).

26. Shi, K. *et al.* Safe human dual-robot interaction based on control barrier functions and cooperation functions. *IEEE Robotics Autom. Lett.* **9**, 9581–9588, DOI: 10.1109/LRA.2024.3458597 (2024).

27. Rauscher, M., Kimmel, M. & Hirche, S. Constrained robot control using control barrier functions. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 279–285, DOI: 10.1109/IROS.2016.7759067 (IEEE, 2016).

28. Yi, S. *et al.* Safety-aware human-centric collaborative assembly. *Adv. Eng. Informatics* **60**, 102371, DOI: 10.1016/j.aei.2024.102371 (2024).

29. Hafner, M. *et al.* Clip and complementary methods. *Nat. Rev. Methods Primers* **1**, DOI: 10.1038/s43586-021-00018-1 (2021).

30. Kirillov, A. *et al.* Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 4015–4026 (2023).

31. Siciliano, B., Sciavicco, L., Villani, L. & Oriolo, G. *Robotics: Modelling, Planning and Control* (Springer, 2010).

32. Featherstone, R. *Rigid Body Dynamics Algorithms* (Springer, 2008).

33. Abate, M. & Coogan, S. Computing robustly forward invariant sets for mixed-monotone systems. In *2020 59th IEEE Conference on Decision and Control (CDC)*, 4553–4559, DOI: 10.1109/CDC42340.2020.9304461 (IEEE, 2020).

34. Li, Z., Huang, J., Zhang, P. & Shi, P. Whole-body safety-critical control design of an upper limb prosthesis for vision-based manipulation and grasping. *IEEE Transactions on Autom. Sci. Eng.* **22**, 534–545, DOI: 10.1109/TASE.2024.3412823 (2025).

35. Xiao, W. & Belta, C. Control barrier functions for systems with high relative degree. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, 474–479, DOI: 10.1109/CDC40024.2019.9029455 (IEEE, 2019).