NATIONAL UNIVERSITY OF KYIV-MOHYLA ACADEMY
MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE

NATIONAL UNIVERSITY OF KYIV-MOHYLA ACADEMY
MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE

Qualification scientific work

submitted as a manuscript

KUZMENKO DMYTRO OLEKSANDROVYCH

UDC 004.8:004.896:004.85

DISSERTATION

RESOURCE-AWARE EMBODIED INTELLIGENCE THROUGH THE OPTIMIZATION OF DATA, COMPUTE, AND TRUST

122 «Computer Science»

12 «Information Technologies»

Submitted to obtain the degree of Doctor of Philosophy

The dissertation contains the results of the author's original research. The use of ideas, results, and texts from other authors is properly acknowledged and referenced to their respective sources.

_____ D. O. Kuzmenko

Supervisor: **Shvai Nadiia Oleksandrivna**, candidate of physical and mathematical sciences, associate professor

Київ — 2026

ABSTRACT

Kuzmenko D. O. Resource-Aware Embodied Intelligence through the Optimization of Data, Compute, and Trust. – Qualification scientific work submitted as a manuscript.

PhD thesis to obtain the degree of Doctor of Philosophy in the Programme Subject Area 122 ”Computer Science” (12 – Information Technologies). – National University of Kyiv-Mohyla Academy, Kyiv, 2026.

Modern robotic systems must learn and act under hard constraints of wall-clock time, onboard compute and memory, and energy, while remaining safe, reliable, and understandable in interaction with people. State-of-the-art perception, decision-making, and vision-language models often assume large training budgets or cloud resources, which limits real-world deployment. This dissertation is aimed specifically at solving these problems. It develops a set of algorithms and system designs that reduce temporal and computational cost without degrading task performance, and extends the notion of efficiency to human-robot interaction in terms of trust, safety, and cooperation.

In the dissertation, methods are proposed and experimentally studied for efficient perception from egocentric video, resource-aware object-goal navigation with vision-language components, time-efficient model-based reinforcement learning with policy distillation for multi-task deployment, and a manager-level mixture-of-experts for vision-language models that selects among experts at inference time according to memory and latency budgets. The work also introduces an experimental benchmark for Empathic Ethical Disobedience (EED Gym), which formalizes efficiency in human-robot interaction as maintaining safety and cooperation while minimizing unnecessary risk and interaction overhead.

The research applies methods of machine learning and control, including semi-supervised learning with pseudo-labels, architectural selection and ablation of vision-language stacks, model-based reinforcement learning with trajectory optimization, policy and ensemble distillation, mixture-of-experts routing with budget-aware expert selection, and simulation-based human-robot evaluation with safety-aware reward

shaping and trust metrics. The experimental program covers large-scale egocentric perception datasets for walking environments, embodied navigation in scanned 3D scenes, multi-task control domains, and scripted human-agent interaction scenarios.

Scientific results obtained in the dissertation:

– A methodology for label- and compute-efficient visual perception based on semi-supervised learning for egocentric locomotion scenes.

– Resource-aware configurations for vision-language object-goal navigation that trade off success and path efficiency against GPU memory and latency.

– Algorithms for time-efficient model-based reinforcement learning and a distillation pipeline that compresses multi-task agents into compact deployable policies.

– An energy-efficient RL extension with a fuel-consumption penalty that reduces operational energy use with minimal impact on lap-time performance.

– A manager-level mixture-of-experts (MoIRA) for vision-language models that provides budget-adaptive routing over heterogeneous experts.

– A benchmark and methodology for efficiency in human-robot interaction (EED Gym), formalizing trust, safety, and cooperation under constraints.

Conducted experimental research: An experimental study was performed across four strands. For perception, semi-supervised pipelines on egocentric datasets were evaluated with accuracy and label-efficiency metrics and analyzed with confusion matrices and deployment footprints. For object-goal navigation, vision-language stacks were profiled on HM3D-based splits with success, SPL, and module-wise time/VRAM measurements, producing deployable configurations under limited memory. For model-based reinforcement learning, training recipes and distillation procedures were evaluated by wall-clock, reward, and policy size across multiple tasks, including generalization and ablations. For human-robot interaction, EED Gym scenarios were used to measure safety risk, trust dynamics, and cooperative payoff for baseline policies and learned agents, with interventions such as clarification and refusal.

Practical significance: The proposed methods form a software and experimental suite for building robotic systems that operate within realistic budgets. Semi-

supervised perception reduces annotation and training cost and is suitable for mobile or embedded devices. Resource-aware navigation stacks enable zero- or low-shot object-goal navigation under modest GPU memory. Time-efficient RL with distillation shortens training cycles and allows multi-task deployment on limited hardware. The MoIRA manager permits a single agent to adapt its expert usage to changing latency or memory budgets without retraining. EED Gym provides an evaluation environment for developers and researchers to quantify safety-aware efficiency in interaction with people.

The first chapter of the dissertation is dedicated to the analysis of existing research in label-efficient perception, object-centric navigation with vision-language models, time-efficient reinforcement learning and policy distillation, mixture-of-experts for large vision-language models, and efficiency in human-robot interaction. It outlines the main problems and challenges, including the dependence on large annotation budgets, instability under memory limits, and the need to optimize not only compute but also interaction safety and trust.

The second chapter is dedicated to the methodology and experiments for efficient perception in robotics. The chapter presents a class of semi-supervised pipelines for egocentric walking environments, describes pseudo-label mining and augmentation strategies, and shows how coarse-to-fine training schedules and compact backbones yield accurate recognition with reduced labeled data and practical deployment footprints.

The third chapter is dedicated to vision-language frontier maps for object-goal navigation. The chapter analyzes alternative vision-language backbones and detectors, introduces a resource-aware composition of components, and studies how success and path efficiency change with memory and latency budgets. A set of recommended configurations is provided for machines with limited VRAM.

The fourth chapter presents the main algorithms for time-efficient model-based reinforcement learning and policy distillation. Training and planning schedules are introduced to reduce wall-clock time without degrading reward, multi-task distillation

is used to compress ensembles and controllers into compact policies, and a joint energy-efficient RL study augments the reward with a fuel-consumption term, reducing fuel per lap of the race car with minimal impact on lap time. The chapter includes ablations, generalization studies, and deployment considerations.

The fifth chapter introduces MoIRA, a manager-level mixture-of-experts for vision-language-action agents. It proposes two zero-shot routing strategies – embedding-based cosine similarity and prompt-driven language model inference – and three model serving configurations, then validates the framework on GR1 and LIBERO benchmarks, demonstrating that specialist routing matches or outperforms generalist baselines with explicit control over memory and latency trade-offs.

The sixth chapter focuses on human-robot interaction and the Empathic Ethical Disobedience Gym. The chapter defines tasks, metrics for trust, safety, and cooperation, and intervention policies, and shows experimental results that quantify efficiency in interaction beyond compute alone, including when a robotic agent should clarify, comply, or refuse.

In the general conclusions, the main findings of the dissertation are summarized, including the development of sample-efficient learning and inference methods across perception, navigation, control, and interaction, and the confirmation that efficiency can be optimized jointly with performance and safety for deployment-ready robotic systems.

Keywords: Vision-language-action models, reinforcement learning, optimization methods, computer vision, neural networks, human-robot interaction, Markov decision process, model “state-probability of action”, artificial intelligence, decision making, machine learning, RMSE, control, numerical solution, inverse problems.

АНОТАЦІЯ

Кузьменко Д. О. Ресурсно-орієнтований втілений інтелект на основі оптимізації даних, обчислень і довіри. – Кваліфікаційна наукова праця у вигляді рукопису.

Дисертація на здобуття ступеня доктора філософії за галуззю знань 12 «Інформаційні технології», спеціальністю 122 «Комп'ютерні науки». – Національний університет «Києво-Могилянська академія», Київ, 2026.

Сучасні робототехнічні системи повинні навчатися та діяти в умовах жорстких обмежень реального часу, обчислювальних ресурсів, пам'яті й енергоспоживання, зберігаючи при цьому безпеку, надійність і зрозумілість під час взаємодії з людьми. Багато сучасних підходів у задачах розпізнавання образів, прийняття рішень і побудови зорово-мовних агентів спираються на велику кількість дорогих обчислювальних ресурсів, що ускладнює їх практичне застосування. Ця дисертаційна робота присвячена розв'язанню зазначених проблем. У ній запропоновано методи та системні рішення, які зменшують часові й обчислювальні витрати без втрати якості виконання завдань, а також поширюють поняття ефективності на взаємодію людини й робота з урахуванням довіри, безпеки та співпраці.

У дисертації запропоновано й експериментально досліджено методи ефективного розпізнавання навколишніх образів на основі егоцентричного відеопотоку, ресурсно-обмежену навігацію за об'єктом із використанням зорово-мовних компонентів, ефективне в часі модельно-орієнтоване навчання з підкріпленням із дистиляцією стратегій для багатозадачного розгортання, а також розглянутий запропонований підхід суміші експертів для зорово-мовних агентів, який адаптується до обмежень пам'яті та затримки в реальному часі. Окремо запропоновано експериментальне симуляційне середовище Empathic Ethical Disobedience (EED) Gym, що формалізує ефективність у взаємодії людини і робота як здатність підтримувати безпеку й співпрацю за мінімізації зайвого ризику та комунікаційних витрат.

У роботі застосовано методи машинного навчання та керування, зокрема часткове навчання з вчителем, пошук оптимальних архітектур і абляційні дослідження зорово-мовних систем, модельно-орієнтоване навчання з підкріпленням із плануванням траєкторій, дистиляцію стратегій та ансамблів, маршрутизацію в системах суміші експертів із децентралізованим вибором, а також симуляційне оцінювання взаємодії людини й робота з урахуванням безпеки та метрик довіри. Експерименти охоплюють великомасштабні егоцентричні набори даних для середовищ ходьби, навігацію у відсканованих тривимірних сценах, багатозадачні середовища керування та сценарії взаємодії людини з агентом.

Наукові результати, отримані в дисертації:

– Розроблено методологію обчислювально-ефективного та ефективного з точки зору анотації даних візуального сприйняття на основі часткового навчання з вчителем для пересування егоцентричними сценами.

– Запропоновано ресурсно-обмежені конфігурації зорово-мовної навігації і пошуку об’єкту, що забезпечують узгодження успішності та ефективності маршруту з бюджетами пам’яті та затримки.

– Розроблено алгоритми ефективного в часі модельно-орієнтованого навчання з підкріпленням і дистиляційний підхід, що стискає агентів у компактні стратегії, придатні до практичного розгортання.

– Запропоновано метод енергоефективного навчання з підкріпленням зі штрафом за споживання пального, що дозволяє зменшити витрату енергії без суттєвого зменшення швидкості проходження кола.

– Запропоновано підхід суміші експертів для зорово-мовних агентів із адаптивною маршрутизацією між гетерогенними експертами з урахуванням ресурсних обмежень.

– Розроблено симуляційне середовище і методологію оцінювання ефективності у взаємодії людина і робота, що формалізують довіру, безпеку та кооперацію за наявності обмежень ресурсів.

Проведені експериментальні дослідження: Експериментальні дослідження

виконано за чотирма основними напрямами. У задачах розпізнавання образів з частковим навчанням з вчителем моделі оцінено на егоцентричних даних за показниками точності та ефективності використання анотації, із аналізом матриць помилок і вимог до розгортання. Для навігації за об'єктом проведено профілювання зорово-мовних систем на вибірках HM3D із використанням метрик відсотку успішності і зваженого ефективного шляху виконання та вимірюванням часу й використання відеопам'яті на рівні компонентів. У задачах модельно-орієнтованого навчання з підкріпленням проаналізовано тренувальні конфігурації та процедури дистиляції за реальним часом навчання, винагородою та розміром стратегії з урахуванням узагальнюваності й абляцій. Для взаємодії людини та робота застосовано сценарії середовища EED Gym для оцінювання ризику, динаміки довіри та кооперативного виграшу для базових і навчених агентів із такими можливими діями, як уточнення або відмова.

Практичне значення: Запропоновані методи формують програмний та експериментальний інструментарій для створення робототехнічних систем, здатних працювати в реальних часових, обчислювальних і енергетичних обмеженнях. Часткове навчання з вчителем зменшує витрати на анотацію і придатне для мобільних та вбудованих платформ. Ресурсно-обмежені навігаційні системи забезпечують навігацію за об'єктом із мінімальними тренувальними витратами на графічних процесорах з помірною кількістю відеопам'яті. Ефективне в часі навчання з підкріпленням із дистиляцією скорочує цикли навчання та підтримує багатозадачне розгортання на обмеженому апаратному забезпеченні. Менеджер MoIRA дає змогу одному агенту адаптувати використання експертів до змін умов без повторного навчання. Симуляційний простір EED Gym забезпечує відтворюване середовище для кількісної оцінки ефективності взаємодії з урахуванням безпеки та соціальних чинників.

Ключові слова: Зорово-мовно-дійові моделі, навчання з підкріпленням, методи оптимізації, комп'ютерний зір, нейронні мережі, взаємодія людини з роботом, марковський процес прийняття рішень, модель "стан-імовірність дії", штучний

інтелект, прийняття рішень, машинне навчання, RMSE, управління, чисельний розв’язок, обернені задачі.

LIST OF THE CANDIDATE'S PUBLICATIONS

List of scientific publications in which the primary research results of the dissertation are published:

Articles in MoES-listed journals

[1] Beimuk, V., Kuzmenko, D. (2025). Energy conservation for autonomous agents using reinforcement learning. *NaUKMA Scientific Journal, Computer Science Series*, 8, 68–75. https://doi.org/10.18523/2617-3808.2025.8.68-75.

Articles indexed in international bibliometric databases

[2] Kurbis, A. G., Kuzmenko, D., Ivanyuk-Skulskiy, B., Mihailidis, A., Laschowski, B. (2024). StairNet: visual recognition of stairs for human-robot locomotion. *BioMedical Engineering OnLine*, 23(1), 20. https://doi.org/10.1186/s12938-024-01216-0.

[3] Kuzmenko, D., Shvai, N. (2026). MoIRA: Modular instruction routing architecture for multi-task robotics. *Neurocomputing*, 674, 132962. https://doi.org/10.1016/j.neucom.2026.132962.

List of scientific publications confirming dissemination of the dissertation findings:

[4] Kuzmenko, D., Tsepa, O., Kurbis, A. G., Mihailidis, A., Laschowski, B. (2023). Efficient visual perception of human-robot walking environments using semi-supervised learning. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2023)*, pp. 6544–6549. https://doi.org/10.1109/IROS55552.2023.10341654.

[5] Kuzmenko, D., Shvai, N. (2025). Balancing performance and efficiency in zero-shot robotic navigation. In: Ermolayev, V. et al. (eds.) *ICTERI 2024*, CCIS vol. 2359, pp. 370–381. Springer, Cham. https://doi.org/10.1007/978-3-031-81372-6_28.

[6] Kuzmenko, D., Shvai, N. (2025). Knowledge transfer in model-based reinforcement learning agents for efficient multi-task learning. *AAMAS 2025 Proceedings*, pp. 2597–2599. ACM Digital Library. https://dl.acm.org/doi/10.5555/3709347.3743949.

[7] Kuzmenko, D., Shvai, N. (2026). When Robots Say No: The Empathic Ethical Disobedience Benchmark. *ACM/IEEE International Conference on Human-Robot Interaction (HRI 2026) Proceedings*, pp. 168–176, Edinburgh, Scotland, UK, March 16–18, 2026. https://dl.acm.org/doi/10.1145/3757279.3785547.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS AND GLOSSARY

LIST OF ABBREVIATIONS

CNN Convolutional Neural Network (Згорткова нейронна мережа)

EED Gym Empathic Ethical Disobedience Gym (Середовище емпатичної етичної непокори)

HM3D Habitat-Matterport 3D Dataset (Набір даних Habitat-Matterport 3D)

HRI Human-Robot Interaction (Взаємодія людини і робота)

LLM Large Language Model (Велика мовна модель)

LoRA Low-Rank Adaptation (Адаптація низького рангу)

MBRL Model-Based Reinforcement Learning (Модельно-орієнтоване навчання з підкріпленням)

MDP Markov Decision Process (Марковський процес прийняття рішень)

MoE Mixture-of-Experts (Суміш експертів)

MoIRA Modular Instruction Routing Architecture (Модульна архітектура маршрутизації інструкцій)

ObjectNav Object-Goal Navigation (Навігація ”об’єкт-мета”)

RL Reinforcement Learning (Навчання з підкріпленням)

SPL A metric of Success weighted by Path Length (Метрика успіху, зваженого довжиною шляху)

SR A metric of Success Rate (Метрика показника успішності)

SSL Semi-Supervised Learning (Часткове навчання з вчителем)

TD-MPC Temporal Difference Model Predictive Control (Модельно-прогностичне керування з темпоральною різницею)

ViT Vision Transformer (Зоровий трансформер)

VLA Vision-Language-Action (Зорово-мовно-дійова система)

VLFM Vision-Language Frontier Map (Зорово-мовна карта фронтиру)

VLM Vision-Language Model (Зорово-мовна модель)

VQA Visual Question Answering (Зорова задача ”питання-відповідь”)

VRAM Video Random Access Memory (Відеопам’ять з довільним доступом)

GLOSSARY

budget-aware routing Routing that adapts expert selection to latency and memory constraints. (Маршрутизація, що адаптує вибір експертів до обмежень затримки та пам’яті.)

coarse-to-fine training schedule Training regime that starts with coarse representations or tasks and progressively refines to higher resolution. (Режим навчання, що починається із загальних представлень або задач і поступово уточнюється до більшої роздільної здатності.)

egocentric perception Perception from the agent’s first-person viewpoint using onboard sensors. (Розпізнавання образів від першої особи за допомогою бортових сенсорів агента.)

embodied intelligence Intelligence grounded in a physical body, where perception, action, and environment interaction drive learning and decision making. (Інтелект, втілений у фізичному тілі, де сприйняття, дія та взаємодія з довкіллям визначають процеси навчання і прийняття рішень.)

frontier map Spatial map highlighting unexplored boundaries to guide exploration and navigation. (Просторова карта, що виділяє невивчені межі для спрямування дослідження та навігації.)

mixture-of-experts routing Mechanism that selects among specialized submodels (experts) per input. (Механізм, що обирає між спеціалізованими підмоделями (експертами) для кожної задачі.)

model-based reinforcement learning Reinforcement learning that learns or uses a dynamics model for planning or policy improvement. (Навчання з підкріпленням, що вивчає або використовує модель динаміки для планування чи покращення стратегії.)

object-goal navigation Navigation type, in which the target is defined by object category rather than a fixed coordinate. (Тип навігації, в якому ціль задана категорією об'єкта, а не фіксованою координатою.)

policy distillation A process of compressing a larger policy into a smaller policy that imitates the original behavior. (Стиснення більшої стратегії в меншу, що імітує оригінальну поведінку.)

pseudo-label Automatically generated label used to train models with limited manual annotation. (Автоматично згенерована мітка для навчання моделей в умовах обмеженої ручної анотації.)

resource-aware learning Learning type that explicitly optimizes compute, memory, latency, and energy budgets. (Тип навчання, що явно оптимізує бюджети обчислень, пам'яті, затримки та енергії.)

safety-aware reward shaping Augmenting reward with penalties or constraints to favor safe behavior. (Доповнення винагороди штрафами або обмеженнями для підтримки безпечної поведінки.)

trajectory optimization Planning method that optimizes a sequence of actions or states to maximize an objective. (Метод планування, який оптимізує послідовність дій або станів для максимізації цілі.)

trust metric Quantitative measure of human trust or reliance during interaction. (Кількісна міра людської довіри або покладання на агента під час взаємодії.)

vision-language stack Pipeline of vision and language modules used for perception, grounding, and action selection. (Набір зорових і мовних модулів для сприйняття, семантичного заземлення та вибору дій.)

INTRODUCTION

Justification for choosing the research topic. Intelligent robotic systems must increasingly learn and act on-device under strict constraints: limited wall-clock time, bounded computational and memory resources, and tight energy budgets while remaining safe, robust, and legible to human collaborators. Modern deep-learning pipelines for perception, navigation, and language-guided control are powerful but resource-intensive, routinely assuming large training budgets or cloud offloading that are unavailable on mobile and embedded hardware. This dissertation addresses that mismatch directly: it develops methods that make learning and inference time- and resource-efficient without degrading task performance, and extends the notion of efficiency beyond compute to human-robot interaction (HRI), formalizing the notion in terms of trust, safety, and cooperation.

Existing approaches to label-efficient perception (supervised CNNs, temporal encoders, semi-supervised methods), zero- or low-shot object-goal navigation with vision-language components (VLMs, detectors, segmentation models, VQA), and model-based reinforcement learning (planning with learned dynamics, actor-critic hybrids) provide strong baselines but tend to be resource-intensive. Mixture-of-experts (MoE) techniques improve accuracy but are usually coupled to heavy models and fixed budgets. This motivates methods and system designs that treat efficiency as a primary objective: fewer labels and faster training, smaller memory and latency footprints at inference, and safer, more economical interaction with humans.

Four methodological threads connect all five contributions into a single design logic. First, resource constraints are encoded as learning signals throughout: pseudo-label confidence thresholds in Chapter 2, auxiliary distillation loss terms and a fuel-penalty reward term in Chapter 4, and safety-trust penalty terms in Chapter 6. Second, modularity serves as the shared structural efficiency principle: both the VLFM pipeline in Chapter 3 and MoIRA in Chapter 5 consist of independently swappable components that can be substituted or extended without retraining the full system. Third, each system exploits external information that is available but conventionally unused:

large-scale unlabeled video is converted to pseudo-labeled training data in Chapter 2, and natural-language expert meta-descriptions serve as zero-shot routing signals in Chapter 5, both without additional annotation or retraining cost. Fourth, resource-aware metrics serve as primary selection criteria in every chapter: VRAM and latency bounds, wall-clock training time, fuel consumption per lap, and trust calibration scores determine which configuration is reported as the main result, making the most resource-aware solution the explicit deliverable rather than a secondary consideration.

Research aim and objectives. The purpose of the work is to create algorithms and system designs for time-efficient learning and inference in robotic perception, navigation, control, and interaction. To achieve this purpose, the following tasks were set:

- reduce supervision and compute in visual perception while maintaining accuracy;
- optimize vision-language object-goal navigation under VRAM and latency constraints;
- accelerate model-based reinforcement learning and compress multi-task controllers via distillation;
- design a manager-level MoE for VLA agents that selects among experts at inference time according to memory and latency budgets;
- formulate an HRI benchmark that operationalizes efficiency beyond compute (trust, safety, cooperation).

Object and subject of the research. The object of research is intelligent robotic systems for perception, navigation, control, and HRI. The subject of research is algorithms and architectures that decrease sample complexity, wall-clock time, and resource footprint while sustaining task performance and lowering interaction risk.

Research methods. The work employs semi-supervised learning with pseudo-labels and standard image augmentations; architectural selection and ablation of vision-language stacks (VLM, detection, segmentation, VQA); model-based reinforcement learning with trajectory optimization; policy and ensemble distillation for multi-task deployment; budget-aware routing and task-conditioned expert selection for mixtures of experts; and simulation-based HRI evaluation with safety-aware reward shaping, trust metrics, and cooperative payoff analysis. Validation uses large-scale egocentric datasets for locomotion perception, HM3D for embodied navigation, multi-task control environments, and scripted HRI scenarios.

The methodology is predominantly empirical. This is deliberate and reflects the nature of the studied phenomena. Deep RL systems exhibit emergent, non-linear dynamics that resist closed-form analysis: policy performance depends on reward landscape curvature, replay buffer coverage, and optimization trajectory in ways that are not tractable analytically. Semi-supervised learning methods involve interacting pseudo-label feedback loops whose convergence properties are best characterized via ablation and controlled comparison rather than formal bounds. The chosen evaluation protocol – controlled benchmarks, ablation studies, and cross-seed statistical testing – follows established practice in the field [1] and is designed to isolate the effect of each architectural or algorithmic choice while controlling for stochasticity. Wherever possible, parametric significance tests are used to support performance claims.

Scientific novelty and originality of the research findings. The dissertation substantiates a unified *resource-aware* paradigm across data (semi-supervision), computational and memory resources (resource-aware VLFM and MoIRA MoE), time (optimized model-based RL and multi-task distillation), and interaction risk (EED Gym). It introduces a semi-supervised learning methodology for egocentric locomotion perception that achieves high accuracy with substantially fewer annotated samples; a manager-level MoE for VLA agents that performs budget-adaptive routing across heterogeneous experts; an optimized training schedule for model-based RL with distillation into compact policies; a joint study on energy-efficient RL with a

fuel-consumption penalty demonstrates how operational energy can be reduced with minimal effect on control quality; and an HRI benchmark that quantifies efficiency as the joint optimization of safety, trust, and cooperation.

Scientific results obtained in the dissertation.

- A methodology for annotation- and compute-efficient visual perception for egocentric locomotion scenes based on semi-supervised learning.
- Resource-aware configurations for vision-language object-goal navigation (VLFM) that improve success rate by 1.55% while reducing GPU-memory footprint by 2.3×, at a minor cost to path efficiency.
- Algorithms for time-efficient model-based RL (optimized training schedules) and a multi-task distillation pipeline that compresses ensembles and controllers into compact policies.
- An energy-efficient RL extension with a fuel-consumption penalty that reduces energy use with small impact on lap-time performance.
- A manager-level mixture of experts MoIRA for VLA agents with budget-adaptive routing and graceful accuracy degradation under tight memory and latency constraints.
- EED Gym, a benchmark and methodology for measuring efficiency in HRI with respect to trust, safety, and cooperation.

Conducted experimental research. Perception pipelines are evaluated on Stair-Net and ExoNet with accuracy, F1, class-wise confusion matrices, and annotation-reduction analysis, including deployment footprints on mobile and embedded devices. ObjectNav studies on HM3D (val-mini and full validation) report SR, SPL, and module-wise time and VRAM profiles across VLM, detector, segmentation, and optional VQA. Model-based RL experiments report normalized score and policy size for optimized schedules and multi-task distillation, including generalization tests and

ablations; an energy-efficient RL study augments the reward by a fuel-consumption term and reports best and mean lap times, fuel per lap, and sensitivity to penalty weights and seeds across two tracks and vehicle classes. HRI experiments in EED Gym measure safety risk, trust dynamics, and cooperative payoff for baseline and learned agents with interventions such as clarification, agreement, and refusal.

Author's Contribution. The author designed and implemented the semi-supervised perception pipeline and analyses; built and ran the VLFM optimization study (model selection, VRAM and time profiling, experiments on HM3D); developed the optimized model-based RL schedules and multi-task distillation procedures; conceived and implemented the MoIRA manager and routing strategies; formulated the EED Gym tasks and metrics; and wrote the manuscript. The energy-efficient RL study was carried out jointly with an undergraduate student under the author's supervision.

Integration with the research programs, grants and topics. The research was conducted at the National University of Kyiv-Mohyla Academy within the department's research direction of "Applied machine learning models and methods" (UDC: 004.85, 004.49; 004.056.57 registry № 0123U102430). Parts of the work were developed and tested in collaboration with student projects and laboratory initiatives; individual components were presented and discussed at international venues and internal seminars.

Practical implications of the research findings. The proposed methods and system designs form a software and experimental toolkit for building robotic systems that operate within realistic budgets of time, memory, and energy. Semi-supervised perception reduces annotation and training costs and is suitable for mobile and embedded platforms. Resource-aware VLFM stacks enable zero-shot ObjectNav on modest graphical processors. Time-efficient RL with distillation shortens training cycles and supports multi-task deployment on constrained hardware. MoIRA allows a single agent to adapt expert usage to changing latency and memory budgets without retraining. EED Gym provides a reproducible protocol for evaluating safety-aware

efficiency in interaction with people.

Dissemination of the dissertation findings. The results were presented at international conferences and workshops and discussed in departmental seminars, including IROS 2023 for the perception component, ICTERI 2024 for the ObjectNav component, AAMAS 2025 and an invited talk at University of Southampton for the RL distillation component, HRI 2026 and COMETE 2026 workshop for EED Gym, and Neurocomputing 2026 for MoIRA. Parts of the work were disseminated through preprints and open-source artifacts.

Structure and scope of the dissertation. The dissertation contains an introduction, six chapters (efficiency in embodied intelligence: taxonomy and related work; efficient perception in robotics; efficient vision for robotic goal navigation; efficient reinforcement learning; efficient VLA mixture-of-experts in robotics; efficiency in human-robot interaction), general conclusions, references, and appendix A. The manuscript is illustrated with figures and tables. The total volume of the dissertation is 170 pages, out of which 123 pages are of main text. The bibliography contains 192 references.

CHAPTER I

AN OVERVIEW AND TAXONOMY OF EFFICIENCY IN EMBODIED INTELLIGENCE

Embodied intelligence is often studied under assumptions of abundant data, compute, time, and social trust. Deployed systems face a different reality. A wearable perception system must classify terrain from the small fraction of expert-annotated frames within a vast unlabeled archive; investing in more labels is directly constrained by clinician time and cost [2, 3]. A navigating robot must invoke vision-language models for semantic grounding while competing with its planning and control stack for a fixed VRAM budget. A model-based agent must converge on a competent policy within a wall-clock window that governs research iteration speed and practical deployability. A social robot must calibrate its responses under a trust budget that, once overdrawn through unsafe or opaque behavior, cannot be quickly replenished [4, 5]. These four resource regimes (annotation and interaction data, memory and compute, training time and operational energy, and interaction trust) are the organizing axes of this chapter and of the dissertation as a whole. They are not interchangeable: alleviating annotation cost does not relieve memory pressure, and a faster training schedule does not address trust calibration. Each constraint demands its own family of techniques, and each maps to a distinct strand of the dissertation's contributions.

I organize prior work around a unified *Taxonomy of Efficiency in Embodied Intelligence* that spans four branches (Fig. 1.1): *Data Efficiency*, *Compute and Memory Efficiency*, *Temporal and Energy Efficiency*, and *Interaction Efficiency*. Each branch captures a distinct dimension of resource scarcity. Data efficiency addresses the cost of acquiring labeled examples and real-world interaction episodes relative to the diversity of conditions an agent must handle [3, 6]. Compute and memory efficiency targets the practical gap between the hardware on which state-of-the-art models are trained and the platforms on which they must run at inference time [1]. Temporal and energy efficiency spans the cost of the learning process itself: wall-clock training

time and operational energy draw determine research iteration speed and mission duration respectively [7, 8]. Interaction efficiency shifts the optimization target from computational to social resources: the trust capital and cognitive bandwidth available within a human-robot team, which erode under unsafe compliance and opaque behavior [4, 5]. These four dimensions are coupled; gains on one axis routinely trade against another [9], and a system that neglects any single dimension tends to fail at deployment even when it excels on standard benchmarks.

1.1 Data Efficiency

The first bottleneck in efficient robotics is the data itself. Modern deep learning models for perception and decision-making require vast quantities of labeled or interactively collected data [3, 6], yet both acquisition modes are expensive in physical robotic settings: expert annotation is time-consuming and non-scalable [2], while real-world rollouts expose hardware to wear and risk. This bottleneck is further compounded by the distribution mismatch between controlled collection environments and the long tail of conditions encountered during deployment, a challenge that is particularly acute for wearable and assistive robots operating across heterogeneous real-world scenes.

Three complementary strategies have emerged to address data efficiency at different stages of the learning pipeline. *Annotation-efficient learning* (self-supervised, semi-supervised, and active learning) reduces the annotation burden by leveraging unlabeled data [10–12]. *Interaction-efficient reinforcement learning* (model-based RL, offline RL) reduces the number of environment interactions required by planning with or learning from recorded transitions [13–15]. The comprehensive treatment of offline RL methods is provided by Prudencio et al. [16]. *Transfer and sim-to-real methods* amortize data collection by training in cheap simulations and bridging to the physical world through domain randomization and adaptation [17, 18]; a survey of this line of work is provided by Zhao et al. [19]. The following subsections elaborate on annotation and sample/exploration efficiency, the two nodes directly addressed by

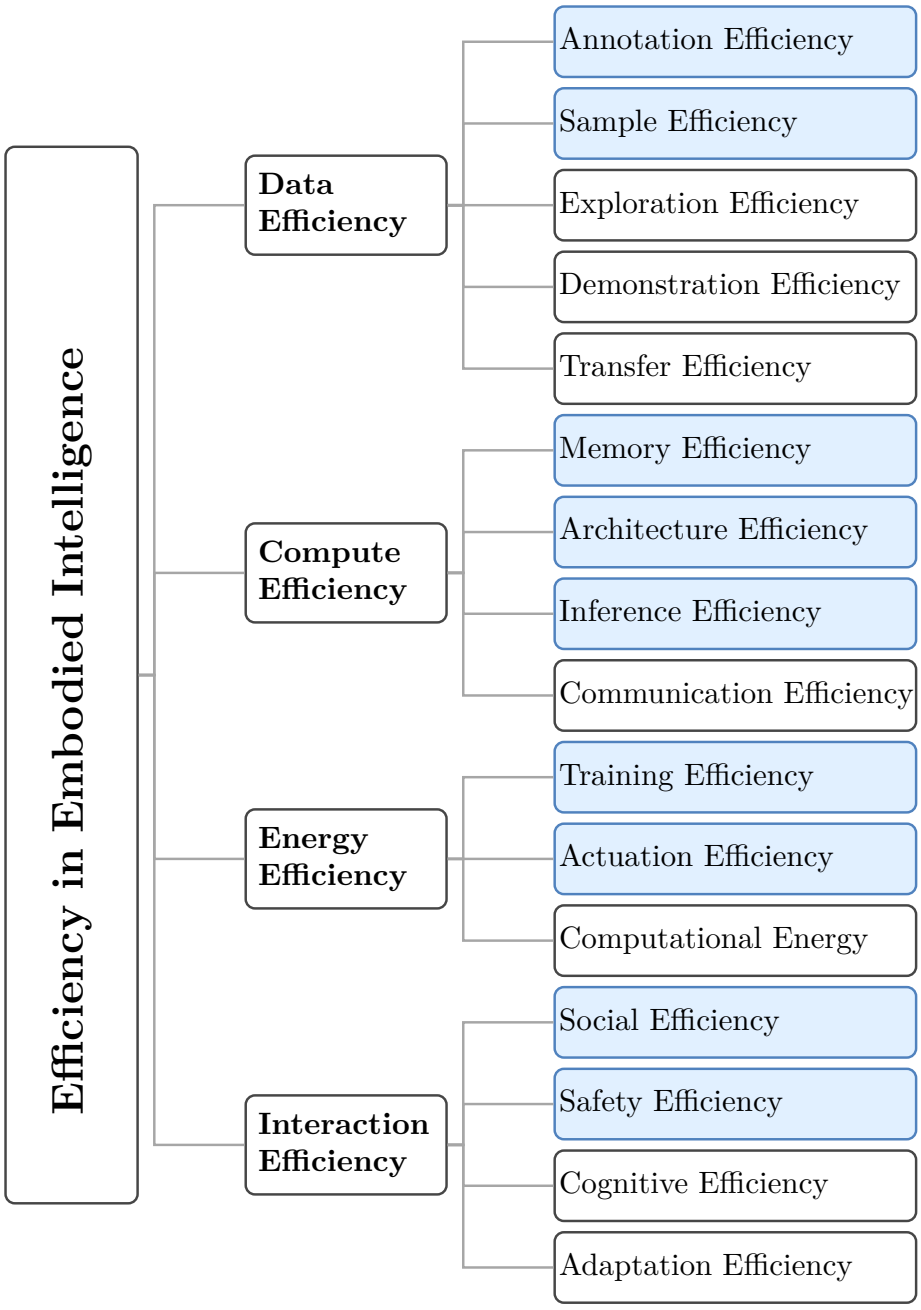

Figure 1.1. **A Taxonomy of Efficiency in Embodied Intelligence.** The taxonomy categorizes the field into four primary branches: Data, Compute, Energy, and Interaction Efficiency. Highlighted nodes are the dissertation's direct contributions: Annotation Efficiency (ExoNet-SSL, Ch. 2), Sample Efficiency (TD-MPC-Opt distillation, Ch. 4), Memory Efficiency (VLFM, Ch. 3), Architecture Efficiency (MoIRA, Ch. 5), Inference Efficiency (quantized distillation, Ch. 4), Training Efficiency (TD-MPC-Opt, Ch. 4), Actuation Efficiency (fuel-aware RL, Ch. 4), Social Efficiency, and Safety Efficiency (EED Gym, Ch. 6).

this dissertation.

1.1.1 Annotation Efficiency

Labeling egocentric sensor data is particularly expensive. Unlike internet-image datasets, wearable and robotic recordings are sequential, context-dependent, and often exhibit severe class imbalance: rare but safety-critical states (e.g., unexpected stair configurations, uneven terrain) appear only a fraction of the time. Expert annotators must understand the locomotion context to assign correct labels. The result is a regime in which vast unlabeled archives exist alongside a tiny labeled subset, and the challenge is to extract maximal performance from this imbalanced resource. In wearable exoskeleton perception, this ratio can be as much as 10:1, as in the ExoNet database [20] where millions of unlabeled frames accompany a comparatively small annotated benchmark.

The field has developed three families of approaches to this problem. Self-supervised and semi-supervised learning exploits the structure of unlabeled data through contrastive objectives (SimCLR [21], MoCo [22]), consistency regularization with confidence thresholding (FixMatch [11], FlexMatch [23], FreeMatch [24]), and iterative self-training with noise injection (Noisy Student [10]). Foundation models trained with self-supervised objectives at scale, such as DINOv2 [12], have shown strong transfer to low-label downstream tasks with minimal fine-tuning. Active learning takes a complementary approach by selecting the most informative unlabeled samples for annotation, maximizing label utility per annotation budget [2]. Representative strategies include uncertainty sampling via Monte Carlo Dropout [25], geometry-based Core-Set selection [26], and gradient-embedding diversity sampling with BADGE [27]. Weak supervision leverages programmatic labeling functions or noisy heuristics to generate pseudo-labels at scale, trading label precision for volume. Key instantiations include Data Programming [28], which represents labeling functions as a generative model over noisy label sources, and Snorkel [29], which aggregates multiple such sources through a learned label model to produce soft training

labels without ground-truth supervision. In this dissertation (Chapter 2), I address the node of annotation efficiency by applying SSL to egocentric locomotion. By combining transformer-based mobile encoders [30] with semi-supervised protocols on the StairNet benchmark [31], I show that a MobileViT-XS model achieves 98.8% overall accuracy and outperforms the supervised state-of-the-art by 0.5 F1 points while requiring approximately 35% fewer annotated images, leveraging 4.5 million unlabeled ExoNet frames that conventional supervised pipelines discard.

1.1.2 Sample and Exploration Efficiency

Beyond annotation, collecting interaction data through physical or simulated rollouts is itself costly, and the fundamental difficulty runs deeper than mere volume. Rewards in realistic tasks are sparse: an agent receives informative feedback only upon task completion or failure, leaving large proportions of collected rollouts with zero gradient signal and severely reducing learning efficiency. The *exploration-exploitation dilemma* is further exacerbated in high-dimensional, continuous robotic domains: the probability of stumbling upon a rewarding state through undirected random action is vanishingly small, and naive exploration strategies can allocate the entire interaction budget to uninformative regions of state space. These difficulties are not independent: sparse rewards and poor exploration jointly cause sample complexity to scale unfavorably with environment stochasticity, making data-efficient learning in physical robotics an open problem even for relatively simple tasks [1].

Reinforcement learning [32] frames sample efficiency as minimizing the number of environment transitions needed to reach a target performance level. Sample efficiency is most commonly addressed through Model-Based Reinforcement Learning (MBRL) [13, 33], where agents plan using learned dynamics models to reduce the number of real-world trials required for learning. By leveraging predictive world models, MBRL methods can achieve strong performance with substantially fewer environment interactions than model-free alternatives. The Dreamer family [13, 15] learns compact latent-space world models and performs imagined rollouts entirely in latent space;

TD-MPC [33] and TD-MPC2 [34] combine temporal difference learning with model predictive control for scalable multi-task MBRL; MBPO [14] provides a principled analysis of when to trust model-generated data versus real rollouts; and transformer-based world models such as IRIS [35] demonstrate that autoregressive sequence models can serve as sample-efficient environment simulators. A complementary line of work focuses on exploration efficiency, guiding agents toward informative experiences through intrinsic motivation or curiosity-driven objectives [36, 37]. While effective in sparse-reward settings, such methods can exhibit pathological behaviors, allocating exploration effort to highly stochastic but task-irrelevant dynamics. Curriculum learning [38, 39] offers a structured alternative, ordering tasks or environments from easy to hard to maximize the information extracted per interaction.

In this dissertation (Chapter 4), I address sample efficiency through reward-level distillation: the distilled 1M-parameter student reaches competitive performance with substantially fewer gradient steps than a student trained from scratch, demonstrating that a high-capacity teacher can transfer task structure to a compact model with improved data efficiency. The underlying MBRL planning is reused from the TD-MPC2 baseline without modification. Exploration and transfer mechanisms serve as complementary baselines. The detailed survey of exploration methods in deep RL is provided by Ladosz et al. [40].

1.2 Compute and Memory Efficiency

Once a model is trained, the primary constraint shifts to the onboard resources of the robot: video memory (VRAM) and inference latency. This constraint is especially severe in embodied AI, where perception, planning, and control must run simultaneously in a closed loop at frequencies of 10-100 Hz, leaving only a fraction of total memory and compute for any single module. Consumer mobile platforms (NVIDIA Jetson AGX Orin, 16 GB LPDDR5) operate at one to two orders of magnitude below the hardware used to train and evaluate state-of-the-art foundation models (A100/H100, 80 GB HBM), making direct deployment of research-grade vision-language systems

categorically infeasible without adaptation [1].

Three families of techniques address this gap. *Memory-efficient attention* reduces the quadratic memory cost of transformer self-attention through IO-aware tiling [41], enabling longer context and larger batch sizes within fixed VRAM budgets. *Lightweight encoder architectures* such as efficient vision transformers [42], compact VLMs [43], and mobile encoders [30] are designed to match deployment constraints from the outset rather than compressing post hoc. *Conditional computation* via MoE routing [44, 45] activates only a sparse subset of model parameters per inference step, achieving large-model capacity at small-model cost. The following subsections address memory efficiency and architecture efficiency, the two nodes directly targeted by this dissertation.

1.2.1 Memory Efficiency

VLMs provide powerful semantic grounding for zero-shot tasks such as Object Goal Navigation (ObjectNav), but their memory footprint grows problematically with both model scale and input context. BLIP-2 [46] bootstraps frozen LLM decoders via a lightweight Querying Transformer, yet still requires multi-gigabyte GPU memory to serve. CLIP [47] is more compact but loses generative semantic reasoning. The deployment gap between research systems optimized for A100 GPU and edge robots creates a categorical barrier where even single-image inference can exhaust the available VRAM of a mobile platform, leaving no budget for the rest of the navigation stack. Existing ObjectNav systems such as CLIP on Wheels (CoW) [48] and similar frontier-map approaches leverage foundation models for semantic inference but treat memory cost as a secondary concern, leading to systems that perform well in benchmark evaluations but are non-deployable on hardware with realistic memory constraints.

The methodological response has taken three forms. IO-aware attention kernels such as FlashAttention [41] reduce HBM reads and writes by fusing attention operations, enabling a 7$\times$ speedup and proportional memory reduction without ap-

proximation. Lightweight architectures (EfficientViT [42], MobileVLM [43]) replace dense attention with cascaded group attention or compressed visual tokenization, achieving strong benchmark performance at a fraction of the footprint. Component-level substitution (replacing heavyweight modules in an assembled pipeline with compact drop-ins) offers a pragmatic middle path that preserves modularity. Sub-approaches within this node include token pruning, VRAM-aware model composition, and context-length management. I address the node of memory efficiency in Chapter 3. By profiling vision-language models on the HM3D benchmark, I demonstrate that resource-aware composition of frontier maps allows for deployable zero-shot navigation without significant performance penalties, addressing the challenge of deployment on resource-constrained platforms.

1.2.2 Architecture Efficiency

The fundamental tension in neural network design for robotics is between representational capacity and inference cost. A model powerful enough to handle the diversity of real-world tasks tends to be too large to run at control-loop frequency on onboard hardware; a model small enough to run in real time tends to lack the generalization needed for novel environments. Monolithic scaling (simply training larger dense networks) resolves this only by pushing deployment requirements further beyond practical reach. MoE architectures [44] offer a principled resolution: by partitioning capacity across a pool of specialized sub-networks and activating only a sparse subset per input, they achieve large effective model size with bounded per-inference FLOP and memory cost. The key algorithmic challenge is the routing mechanism that decides which experts to activate, which must be differentiable, load-balanced, and robust to distribution shift.

At the language model scale, Switch Transformers [45] demonstrated that routing to a single expert per token achieves $7\times$ pre-training speedup over dense baselines; Mixtral [49] showed that sparse 8-expert models match or exceed dense models with twice the parameters. In robotics, MoDE [50] applies sparse expert activa-

tion to diffusion-based robot policies, while MoLe-VLA [51] introduces layer-level MoE routing within vision-language-action models, and DriveMoE [52] applies skill-specialized expert routing to autonomous driving VLAs. These systems demonstrate that MoE is a general architecture principle for efficient multi-task robot control. Sub-approaches within this node include sparse token routing (Switch, Mixtral), modular adapter composition (LoRA pools), and neural architecture search for task-specific efficient networks [53]. I address the node of architecture efficiency in Chapter 5 with MoIRA. Unlike internal MoEs, MoIRA introduces a manager-level routing framework that treats expert selection as an independent classification task. This allows a single agent to dynamically switch between specialized LoRA adapters [54] based on real-time budget constraints, enabling modular, efficient deployment that outperforms generalist baselines like OpenVLA [55] or RT-2 [56] and performs on-par with non-modular classic MoE approaches.

Standard approaches to inference efficiency include both quantization and pruning. Pruning removes redundant parameters or connections from trained networks to reduce model size and computational cost while retaining accuracy, and can be applied at different levels such as unstructured weight pruning or structured filter/channel pruning [57, 58]. These techniques are often applied as static post-processing steps alongside quantization, though dynamic or joint optimization methods have also been explored. In Chapter 4, I apply post-training quantization (FP16 and INT8) to the distilled 1M-parameter RL policy, reducing model size by up to 50% with no performance loss at FP16, demonstrating that compact distilled policies can be further compressed for inference on memory-constrained hardware. In distributed systems, communication efficiency is important to minimize bandwidth between cloud teachers and edge students [59]. While relevant, this work focuses on onboard autonomy, assuming minimal cloud dependence during execution.

1.3 Temporal and Energy Efficiency

Time and energy are coupled efficiency axes that span both the training phase and the operational lifetime of an autonomous agent. During training, wall-clock time is the dominant cost: large world models require days of compute to converge, limiting the practical iteration speed for research and adaptation. During deployment, energy consumption (measured in joules per task completion) determines mission duration for battery- or fuel-powered platforms. These two axes interact: a computationally heavy training procedure that produces a more energy-efficient policy may be worth the upfront cost, but the trade-off is difficult to quantify without a unified framework. The growing scale of both training infrastructure and deployed fleets has made this dimension increasingly visible: recent analyses suggest that the carbon footprint of training large neural networks is non-negligible [7, 8], and for mobile robots operating continuously in the field, inference energy constitutes a meaningful fraction of total mission cost [7].

The field addresses temporal and energy efficiency through complementary mechanisms: knowledge distillation [60, 61] and curriculum learning [38, 39] compress training time; energy-shaped reward signals embed actuation economy directly into RL objectives; and hardware-level innovations such as neuromorphic sensing [62, 63] reduce the joules-per-bit cost of the perception component. The three subsections below address these strategies in turn, corresponding to the *Training Efficiency*, *Actuation Efficiency*, and *Computational Energy* nodes of the taxonomy.

1.3.1 Training Efficiency

MBRL improves sample efficiency but introduces its own wall-clock bottleneck. Planning algorithms such as MPPI [64] or CEM [65], executed inside a learned world model at every environment step, impose latencies that compound over millions of training iterations. Large-scale world models such as TD-MPC2 [34] (317M parameters) and DreamerV3 [15] require substantial computational resources to train from scratch, limiting rapid prototyping cycles and the practical adoption of MBRL

in settings where hardware access is constrained. This temporal cost is distinct from sample complexity: a method can be sample-efficient yet still wall-clock-expensive if each imagined rollout requires expensive gradient computation through a large model.

Knowledge distillation [60] is the canonical technique for compressing training cost into a smaller model. Policy distillation [61] adapts this to RL by transferring behavioral knowledge from a high-capacity teacher to a compact student policy. Multi-task extensions [66, 67] simultaneously distill knowledge from multiple task-specific teachers into a shared student, amortizing the cost of training across tasks. Progressive distillation [68] chains multiple compression steps to achieve extreme parameter reduction without catastrophic forgetting. Complementarily, curriculum learning approaches [38, 39] reduce wall-clock training time by ordering the task distribution from easy to hard, ensuring that the agent spends its limited compute budget on maximally informative transitions rather than on tasks it has already mastered or cannot yet solve. Sub-approaches within this node include reward-level distillation, architecture-level compression, and schedule-level optimization. I address the node of training efficiency in Chapter 4 via distillation. I propose a pipeline that transfers knowledge from a high-capacity teacher model to compact student backbones. By optimizing training schedules and employing multi-task distillation [61, 66], I compress hours of training and millions of parameters into lightweight policies that achieve high performance and sample efficiency on the multi-task robotics MT30 benchmark, effectively bridging the gap between large-scale capabilities and real-time applicability.

1.3.2 Actuation Energy

Mobile autonomous agents, such as ground vehicles, legged robots, and aerial drones, operate under hard energy budgets that directly determine mission duration and operational cost. Despite this, most RL formulations treat energy as an implicit side effect rather than an explicit objective: task-completion reward signals incentivize agents to reach goals as quickly as possible, often resulting in aggressive actuation

patterns, e.g. sharp braking, high-torque maneuvers, excessive acceleration, that drain the energy budget. Traditional autonomous driving pipelines [69] and standard RL agents [70] similarly focus on task performance or speed, with energy consumption measured post hoc rather than optimized during training. The mismatch between training objectives and deployment constraints means that even a well-performing policy may be impractical in the field.

Incorporating energy constraints into the RL framework requires multi-objective reward engineering: the agent must simultaneously satisfy task objectives (reaching a goal, maintaining lane) and minimize energy cost (fuel consumption, battery draw). This can be achieved through penalty terms in the scalar reward [9], Lagrangian constraint satisfaction [71], or multi-objective RL with explicit Pareto optimization. Each formulation induces different trade-off behaviors: penalty-based approaches are sensitive to the weight hyperparameter; Lagrangian methods provide constraint satisfaction guarantees [71]; multi-objective methods expose the full Pareto front but require multi-policy training. Sub-approaches within actuation efficiency include energy-shaped reward design, trajectory optimization with energy models, and hierarchical control with energy-aware planning layers. I address the node of actuation efficiency in Chapter 4 by explicitly incorporating fuel constraints into the reward function. In high-fidelity simulations (Assetto Corsa), I demonstrate that modest fuel penalties induce emergent economy-driving behaviors such as momentum conservation and reduced braking, which significantly reduce energy consumption per lap and maintains the speed per lap. These results suggest that efficiency can be treated as a multi-objective optimization problem within the RL framework.

1.3.3 Computational Energy

The energy cost of computation itself has become an increasingly important concern as AI models grow in scale. Schwartz et al. [7] introduced the "Green AI" paradigm, arguing that computational efficiency should be a first-class evaluation criterion alongside accuracy, pointing to the exponential growth in training compute as

a structural inefficiency of the research community. Patterson et al. [8] quantified the carbon emissions of large neural network training runs, demonstrating that choice of hardware, data-center location, and model architecture interact to produce more than two orders of magnitude variance in carbon footprint for comparable performance. For embodied systems that run inference continuously throughout their operational lifetime, these concerns extend beyond training: a model that requires 10 W at inference versus 100 W directly multiplies operational energy by $10\times$, with compounding consequences for battery-powered platforms.

At the sensing level, event-based neuromorphic cameras [62] offer a hardware-level path to energy-efficient perception: unlike frame-based cameras that transmit full images at fixed framerates, event cameras fire asynchronously only when pixel intensity changes, reducing data bandwidth and processing load by orders of magnitude in low-motion scenes. Neuromorphic computing substrates such as Intel Loihi [63] exploit spike-based computation to perform inference at milliwatt power levels, enabling always-on perception in energy-constrained platforms. Low-bitwidth quantization [72, 73] reduces the energy cost of inference on conventional hardware by replacing 32-bit floating point with 8- or 4-bit integer arithmetic, yielding 4-8$\times$ energy reductions with minimal accuracy degradation. Sub-approaches within computational energy include neuromorphic sensing, hardware-aware network design, and low-bitwidth training and inference. While neuromorphic computing and event-based vision offer promising avenues for reducing the joules-per-bit cost of sensing, this dissertation focuses primarily on the actuation and training aspects of energy efficiency, treating the computational energy node as context rather than a direct contribution; a comprehensive survey of event-based vision is provided by Gallego et al. [62].

1.4 Interaction Efficiency

The final pillar of the taxonomy shifts the optimization target from computation to collaboration. Efficiency in human-robot interaction cannot be measured in FLOP or joules: it is defined by the ratio of cooperative payoff to interaction cost, where costs

include physical risk, trust erosion, and cognitive burden imposed on the human. This framing reveals a fundamental complexity that distinguishes HRI from purely technical robotics: the human's internal state (trust, mental load, affect, and expectation) is a latent variable that cannot be directly observed, is shaped by prior interaction history, and couples back into the robot's future performance through the human's behavior. Standard task-completion metrics are blind to this dynamics, creating systems that perform well in single-interaction evaluations but degrade over time as human trust either collapses due to failures or inflates due to over-reliance.

The interaction efficiency branch of the taxonomy encompasses four coupled nodes: *social efficiency* – calibrating compliance and refusal to preserve long-term trust; *safety efficiency* – satisfying hard physical constraints while maintaining utility; *cognitive efficiency* – minimizing human mental load through legible, predictable behavior; and *adaptation efficiency* – personalizing to individual user preferences over time. Each addresses a distinct failure mode of a purely task-optimal agent: one that complies with every command regardless of risk, generates opaque trajectories that operators cannot anticipate, and never adapts to individual user preferences. Together they define the conditions under which an agent is not merely functional, but sustainably deployable alongside humans.

1.4.1 Social Efficiency

An agent that unconditionally complies with every operator command maximizes immediate task throughput but accumulates trust debt: operators observing consistent compliance gradually over-rely on the system, delegating oversight decisions to it and losing situational awareness. Agent failures are inevitable, and under a regime of uncalibrated over-reliance they are catastrophic: trust collapses rapidly and the human-robot team performs worse than either partner would alone. Conversely, an agent that refuses too liberally reduces utility and induces operator frustration, which also degrades trust over repeated interactions [4, 5]. The efficiency-optimal strategy lies between these extremes: a calibrated non-compliance policy that refuses

precisely when the physical or social cost of compliance exceeds the cost of refusal, communicates its reasoning, and preserves the operator's ability to override [74].

The relevant literature spans three communities. Trust calibration research in HRI [4, 5, 75] documents how operator trust forms, degrades, and recovers as a function of robot reliability, transparency, and communication quality. Safe RL [71, 76, 77] formalizes physical constraint satisfaction as a constrained MDP, providing guarantees on collision or violation rates, but typically treats compliance as binary and ignores its trust dynamics. Explainability and plan-recognition work [78] studies how agents can communicate intent to maintain human situation awareness. What the literature has lacked is a framework that evaluates compliance, refusal, and explanation as coordinated decisions with joint consequences for safety and trust; this gap motivates the EED Gym benchmark. Sub-approaches within social efficiency include trust-aware reward shaping, communicative refusal strategies (asking, explaining, proposing alternatives), and calibrated delegation. I address the node of social efficiency in Chapter 6 with EED Gym. By formalizing refusal strategies such as asking, explaining, or proposing alternatives, I quantify how agents can maintain safety and trust simultaneously. This work demonstrates that interaction efficiency requires agents to weigh the social cost of refusal against the physical cost of unsafe compliance, optimizing for long-term collaboration rather than immediate obedience [74].

1.4.2 Cognitive, Safety, and Adaptation Efficiency

Beyond compliance decisions, sustainable human-robot collaboration requires that robot behavior be legible, i.e. the human can infer the robot's intent from its motion before it acts, safe, i.e. hard physical constraints are satisfied even under distribution shift, and adaptive, i.e. the system personalizes to individual user preferences and tolerances over time. Each of these properties corresponds to a distinct node in the taxonomy, and each presents its own complexity: legibility requires generating motion that is optimized for human predictability rather than just task performance; safety under constraint uncertainty requires reasoning about tail risks, not just expected costs;

and adaptation requires online learning from implicit user feedback signals that are noisy, sparse, and potentially inconsistent.

Cognitive efficiency, understood as minimizing human mental load through legible motion, is addressed in foundational work by Dragan et al. [79], who formalize legibility as trajectory optimization with a predictability objective. Safety efficiency is typically addressed by constrained constrained MDPs [71], which enforce safety constraints via Lagrangian methods [77] or CVaR-based risk bounds [80], providing worst-case guarantees rather than only expected-value safety. Adaptation efficiency focuses on rapidly personalizing to user preferences [81] through reward inference, Bayesian preference elicitation, or meta-learning from prior users [82]. The EED Gym benchmark introduced in Chapter 6 unifies the social, safety, and cognitive nodes by treating refusal as a calibrated decision that simultaneously minimizes physical risk (safety efficiency) and the need for human re-intervention (cognitive efficiency), while leveraging persona-based embeddings to model individual variation (adaptation efficiency). Fully online adaptation to individual users remains a direction for future research.

1.5 Scope and Chapter Organization

Chapters 2–6 correspond to these nodes and are evaluated under the shared principles and metrics.

Chapter 2 targets annotation efficiency. It applies semi-supervised learning (FixMatch with a MobileViT encoder) to egocentric stair-recognition, exploiting approximately 4.5 million unlabeled frames from ExoNet that conventional supervised pipelines discard. The central question is how far label counts can be reduced while retaining classification accuracy on the StairNet benchmark, and the answer sets a quantitative floor for annotation cost in wearable-robotics perception.

Chapter 3 targets memory efficiency. It systematically profiles and re-composes the Vision-Language Frontier Map (VLFM) pipeline for zero-shot ObjectNav on the HM3D benchmark, replacing heavyweight components (BLIP-2, YOLOv7-E6E)

with compact alternatives (CLIP-ViT-B/32, YOLOv7-W6). The outcome is a 2.3× reduction in VRAM footprint, unlocking deployment on consumer-grade hardware without sacrificing navigation success rate.

Chapter 4 targets four taxonomy nodes. *Sample efficiency*: the distilled 1M-parameter student reaches competitive performance with substantially fewer gradient steps than a student trained from scratch, demonstrating that teacher-to-student transfer reduces the data budget needed to acquire a competent policy. *Training efficiency*: reward-level distillation from a 317M-parameter TD-MPC2 teacher compresses hours of training and millions of parameters into a lightweight student that surpasses direct training on the MT30 multi-task benchmark. *Inference efficiency*: post-training FP16 quantization reduces the distilled policy's memory footprint by 50% without performance loss, enabling deployment on hardware-constrained control loops. *Actuation efficiency*: a fuel-penalty term embedded in the RL reward induces emergent economy behaviors (momentum conservation and reduced braking) that significantly lower energy consumption per lap in high-fidelity driving simulation.

Chapter 5 targets architecture efficiency. It introduces MoIRA, a modular routing framework that decouples task assignment from policy execution by maintaining a pool of independently fine-tuned LoRA specialists and selecting among them via zero-shot language-based routing. Evaluated on the GR1 and LIBERO benchmarks with GR00t-N1 and π_0 backbones, MoIRA exceeds the generalists and achieves parity with monolithic MoE baselines while offering explicit control over the VRAM-latency trade-off through configurable adapter serving modes.

Chapter 6 targets social efficiency. It introduces EED Gym, a simulation environment in which a robot must decide whether to comply, refuse, explain, or seek clarification in response to commands of varying physical risk, modelling human trust and affect as latent state variables. Safe RL methods (Lagrangian PPO and Masked PPO) are benchmarked against non-safe baselines, and ablations over affect observations, communicative refusals, and training curricula identify the design levers that best preserve trust without increasing unsafe compliance.

Across the set, the common yardsticks are performance preservation under constrained labels, VRAM and inference latency, wall-clock training time, energy use, and safety-trust trade-offs.

CHAPTER II

EFFICIENT PERCEPTION IN ROBOTICS

This chapter presents the methodology for annotation- and compute-efficient visual perception, addressing the **annotation efficiency** node of the taxonomy (Section 1). It is based on the contribution published at the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2023: *Efficient Visual Perception of Human-Robot Walking Environments using Semi-Supervised Learning*.

2.1 Introduction

Computer vision is a key enabling component for environment-adaptive control and planning in human-robot locomotion, particularly in safety-critical assistive systems such as robotic prostheses and exoskeletons. Prior work has demonstrated that egocentric visual sensing, combined with wearable inertial measurements, can be used to anticipate terrain transitions and forward-predict leg kinematics during walking, enabling continuous and adaptive control strategies [20]. Among these perception tasks, accurate and low-latency recognition of stairs is especially important due to the elevated safety risks associated with misclassification. Earlier vision-based stair recognition systems relied on hand-designed feature extractors, while more recent approaches adopted convolutional neural networks (CNNs) to improve robustness and classification performance; Kurbis et al. [31] provide a comprehensive survey of both lines of work.

The adoption of deep learning for this problem has been supported by the availability of large-scale datasets, most notably ExoNet [20], which remains the largest dataset of egocentric images of real-world walking environments. While ExoNet offers unprecedented scale and diversity, its fine-grained environmental taxonomy introduces substantial class overlap, which can negatively affect generalization in supervised learning settings. To address this issue, Kurbis et al. [31] introduced StairNet, a four-class dataset derived from ExoNet by consolidating semantically

similar categories and focusing explicitly on stair-related environments. This reformulation enabled more consistent training and evaluation of CNN-based classifiers and established a strong state-of-the-art baseline for vision-based stair recognition on open-source data. In this chapter, StairNet is therefore used as the primary benchmark for comparative evaluation.

Despite these advances, a persistent bottleneck in egocentric perception for wearable robotics is the cost of large-scale manual annotation. Labeling hundreds of thousands of first-person images is time-consuming and labor-intensive, limiting the practical scalability of supervised approaches. At the same time, datasets such as ExoNet contain vast quantities of unlabeled data that are typically discarded during training. Only a limited number of prior studies have explored unsupervised or weakly supervised alternatives for walking environment classification, including domain-adaptation and pseudo-labeling strategies; these approaches were generally evaluated on smaller datasets or under constrained conditions, leaving open questions regarding their robustness at scale.

In this context, the central motivation of this chapter is to improve annotation efficiency without sacrificing deployment-relevant accuracy. Since StairNet comprises only a small fraction of the full ExoNet dataset (approximately 10%), we investigate whether the remaining unlabeled data can be exploited to improve training efficiency through semi-supervised learning. By combining a limited set of high-quality labels with large volumes of unlabeled egocentric imagery, semi-supervised methods offer a principled mechanism for reducing annotation requirements while preserving classification performance. The goal of this study is to support the development of practical perception systems for environment-adaptive control in wearable robotics by minimizing dependence on manual labeling, while maintaining prediction accuracy comparable to the supervised state-of-the-art [31].

2.2 Problem Formulation

We formally define the task of efficient terrain classification as learning a mapping function $f_\theta : \mathcal{X} \to \mathcal{Y}$ from egocentric RGB images to discrete environmental classes. The class set $\mathcal{Y} = \{LG, LG\text{-}IS, IS, IS\text{-}LG\}$ represents Level Ground, Transition to Incline Stairs, Incline Stairs, and Transition to Level Ground, respectively.

We operate under a semi-supervised setting where we have access to a dataset $\mathcal{D}$ composed of two subsets:

- A small labeled set $\mathcal{D}_l = \{(x_i, y_i)\}_{i=1}^{N_l}$, where N_l is limited by annotation budget.
- A large unlabeled set $\mathcal{D}_u = \{u_i\}_{i=1}^{N_u}$, where $N_u \gg N_l$ (here, $N_u \approx 4.5$ million).

The objective is to optimize the parameters θ of a lightweight mobile architecture (minimizing computational cost) such that the classification accuracy on a hold-out test set is maximized, while strictly minimizing the size of the labeled set N_l (minimizing annotation cost). This formulation specifically targets the **Annotation Efficiency** node of the proposed efficiency taxonomy.

2.3 Methods

2.3.1 Computer Vision Dataset

We used the labeled computer vision dataset called StairNet [31], a derivative dataset of ExoNet [20] that focuses on walking environments during stair ascent. The dataset was developed using re-annotated images from six of the twelve original ExoNet classes which focus on stair ascent, with more precise cut-off points between classes to reduce overlap. The StairNet dataset contains approximately 515,000 RGB images of four classes: level-ground (LG), level-ground transition to incline stairs (LG-IS), incline stairs (IS), and incline stairs transition to level-ground (IS-LG). The class distribution of StairNet is unbalanced such that the steady-state classes, LG and IS, comprise over 95% of the images, whereas the transition classes, IS-LG and LG-IS, form the remaining 5%.

This chapter focuses on using the unlabeled images from ExoNet that were not included in the StairNet dataset. These images contain walking environments; however, many have obstructions or environments unrelated to stair recognition, as encountered during the StairNet development [31]. These are some of the challenges with using unlabeled images, as prior information of the class distributions or viability of the images is unknown. Note that the images of stairs and level-ground contain real-world environmental features, including other people walking.

2.3.2 Semi-Supervised Learning

We adopt **FixMatch** [11] as the semi-supervised algorithm; it is straightforward to implement and serves as a strong proof-of-concept compared to more complex alternatives such as self-training with noisy student [83], meta pseudo-labels [84], AdaMatch [85], or contrastive representation learning [21].

The pipeline is summarized as follows (Figure 2.1): (1) loading raw labeled and unlabeled images and oversampling with augmentations to the labeled dataset to mitigate false positives during training. (2) Filter and preprocess the unlabeled dataset by retrieving unlabeled image logits via supervised pretrained model and selecting the most probable pseudo-labels that surpass the cutoff parameter τ. We preprocess the unlabeled images by applying weak augmentations (i.e., horizontal flips) and strong augmentations (i.e., color intensity, saturation, small rotations and translations, and horizontal flips). We define the batch size ratio parameter μ that indicates the difference between the labeled and unlabeled batch sizes. The unlabeled data requires a greater batch size than the labeled dataset such that the bigger the difference, the more impact semi-supervised learning can have. We input the labeled and unlabeled batches during training, infer the model on the weakly augmented images, and threshold the received logits by the pseudo-label confidence threshold τ. (3) We train the model by calculating and adding two loss terms: supervised loss (i.e., cross-entropy loss – CE loss) and unsupervised loss (i.e., cross-entropy loss of the thresholded pseudo-label logits calculated against strongly augmented images). The

unsupervised loss is scaled by the λ parameter that indicates the importance of the unsupervised portion. These semi-supervised parameters (τ, λ, and μ) are tuned and provide a high degree of model flexibility.

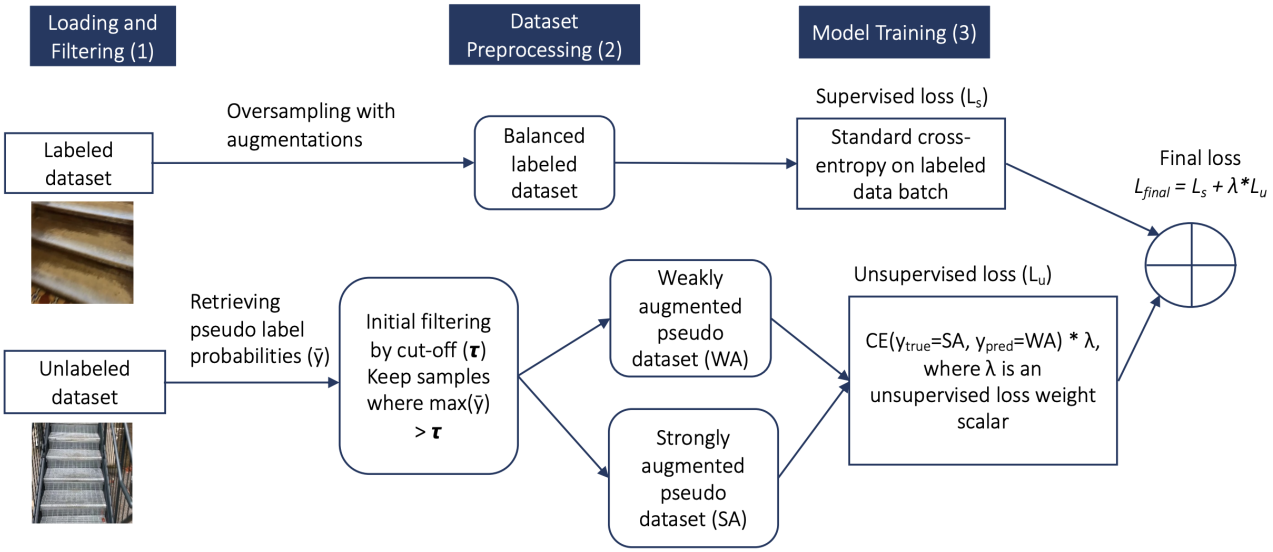

Figure 2.1. The semi-supervised learning algorithm with 3 stages: (1) preparing labeled and unlabeled data by oversampling and filtering the noninformative unlabeled images; (2) preprocessing the unlabeled dataset and creating weakly and strongly augmented variations; and (3) using two losses (supervised and unsupervised) to consolidate training and introduce semi-supervised learning.

2.3.3 Mobile Vision Transformer

We developed a vision transformer (ViT) model using the base model of **MobileViT** [30], which makes use of automated feature engineering similar to standard CNNs. Deep learning feature extractors have been shown to outperform hand-crafted alternatives, especially on large-scale image datasets; we therefore focused on convolution and transformer-based models. MobileViT is a transformer-based deep learning model that uses attention mechanisms and depth-wise dilated convolutions. MobileViT blocks resemble convolutions in structure but benefit from global context. The model has low-level efficient convolution and transformer blocks, which allow it to run on mobile and embedded computing devices with performance and speed comparable to lightweight CNN models such as MobileNetV2. We also experimented

with MobileNetV3, but found its architecture less suitable for this task and dataset.

Three model backbones were considered – MobileViT of sizes XXS, XS, and S, which differ in the number of transformer layers, more sophisticated feature extraction, and parameter count. MobileViT was chosen for its efficient and lightweight design and as a first step toward using transformer models for this application. The final transformer-based model was created in TensorFlow 2.0. The model variants were efficiently trained using the high performance computing power of a Google Cloud Tensor Processing Unit (TPU).

2.3.4 Model Training and Optimization

We used the same StairNet dataset splits for validation (3.5%) and testing (7%) as Kurbis et al. [31] for benchmark comparison. The labeled training set was initially downsampled to 200,000 images from 461,328 images to allow for greater improvements through semi-supervised learning and to study the impact of reduced labeled data. To address the issue of unknown class distributions and image quality of the unlabeled data, the supervised MobileNetV2 model from [20, 31] was recreated and trained (Figure 2.2). This pretrained model was used to retrieve logits for the 4.5 million unlabeled images from ExoNet, which were then thresholded in a FixMatch manner [11]. From the unlabeled dataset, $\sim$1.2 million images surpassed the $\tau = 0.9$ confidence threshold. The resulting subset, called **ExoNet-SSL**, had pseudo-labels that closely resembled the original StairNet distribution [20, 31]: IS with 73,404 images (5.5%), IS-LG with 13,640 images (1%), LG with 1,200,017 images (90.1%), and LG-IS with 45,179 images (3.4%) (Table 2.1).

We experimented with three variants of MobileViT. The lightweight XXS model (900,000 parameters) had superior training and inference time but achieved lower categorical accuracy across all classes (accuracy of 97.7% and f1-score of 97.8%). The largest S model (4.9 million parameters) trained longer and had slightly weaker performance than XS model (1.9 million parameters), which may be explained by overfitting (accuracy of 97.2% and f1-score of 97.3%). The best model consisted of

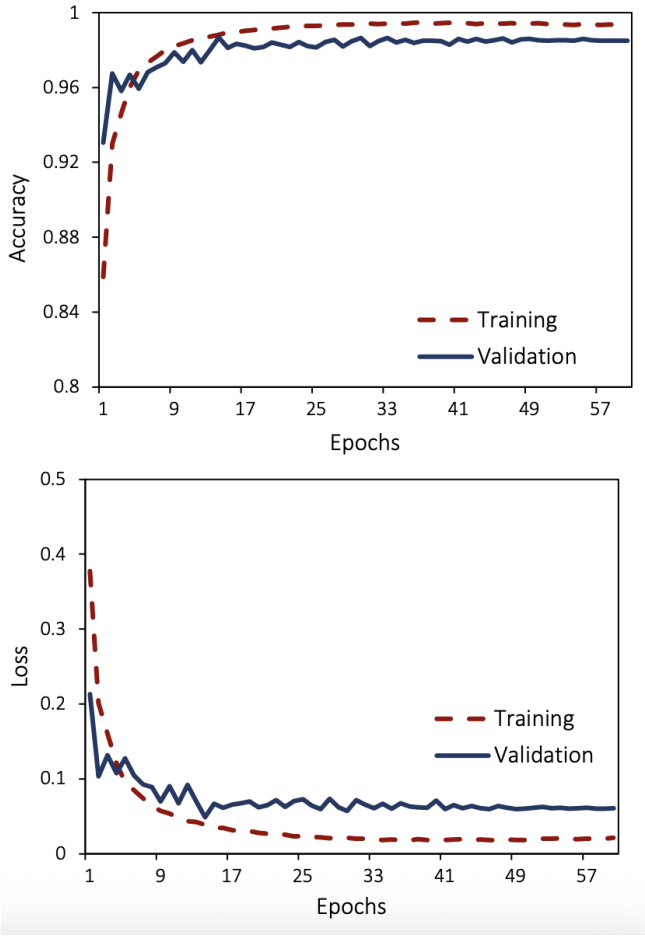

Figure 2.2. The image classification accuracy and loss for the training and validation sets using the supervised MobileNetV2 model.

an ImageNet-pretrained MobileViT-XS scaled to an input resolution of 224×224; stochastic gradient descent (SGD) with 0.9 momentum and Nesterov acceleration; randomly initialized weights; pseudo-label threshold $\tau = 0.98$; a batch-difference ratio of 4; unsupervised loss weight $\lambda = 1$; initial learning rate 0.05; labeled batch size 64; and a FixMatch-style cosine weight decay schedule.

The data imbalance was handled by replacing the cross-entropy loss with a focal loss [86] with class weight penalization γ of 3, avoiding the need for oversampling or class weighting. We also tested an exponential moving average, which showed that averaged parameters can produce significantly better results than the final trained values. The resulting model showed good convergence, but the overall image validation accuracy was inferior to previous vanilla cross-entropy loss experiments.

This leads to the phenomenon we call ”class accuracy allocation”. For this applica-

tion, some classes may be considered more important than others for safe human-robot locomotion. The level-ground (LG) class is the most common daily walking environment. Although less common, the transition classes (LG-IS and IS-LG) can have greater implications for falls and user safety. Addressing unbalanced data allows redistribution of class accuracies, transferring performance from major classes to minor classes. However, this comes with an overall performance degradation as the classes are being more evenly balanced with the focal loss. In other words, the accuracies from major classes (IS and LG) were reallocated to minority classes (IS-LG and LG-IS). The resulting accuracies were 95.1% for IS, 95.8% for IS-LG, 97.5% for LG, and 92.2% for LG-IS.

To combat false positives, We diversified the augmentations applied to the labeled training set, which included minor translations, rotations, contrast, and saturation. Variations of L2 parameter loss and decoupled weight decay [87] were tested. The best models did not include weight decay regularization. Defining a high-performance scheduler for semi-supervised learning can be challenging. We experimented with cosine weight decay as in FixMatch [11] and cosine decay with warm restarts [88]. The former was more resilient and consistent and was thus selected for the final model design. We performed multiple experiments to find the optimal labeled-unlabeled ratio μ and unsupervised loss weight λ parameter. To maximize performance, the final model used 300,000 labeled images ($\sim$65% of the original training labeled dataset) and 900,000 unlabeled images. SGD with Nesterov and momentum was used to optimize the deep learning model with a MobileViT XS backbone. The final set of hyperparameters included a learning rate of 0.045; $\tau = 0.9$; a supervised batch size of 64; a batch size ratio μ of 3; $\lambda = 1.03$; and a cosine decay learning rate schedule. Focal loss was replaced with categorical cross-entropy to alleviate the degradation caused by class balancing. The final model was trained for 42 epochs (Figure 2.3).

Table 2.1. Comparison between ExoNet-SSL and StairNet dataset distributions.

Class	ExoNet-SSL (#)	ExoNet-SSL (%)	StairNet (#)	StairNet (%)
LG	1,200,017	90.1	442,360	85.8
LG-IS	45,179	3.4	15,888	3.1
IS	73,404	5.5	48,179	9.4
IS-LG	13,640	1.0	9,025	1.8

2.4 Results

The image classification accuracies on the training and validation sets using semi-supervised learning were 99.2% and 98.9%, respectively. When evaluated on the testing set, the model achieved an overall image classification accuracy of 98.8%, a weighted f1-score of 98.9%, a weighted precision value of 98.9%, and a weighted recall value of 98.8%. Compared to the supervised baseline [31], the semi-supervised model achieved a 0.4% improvement in overall image classification accuracy and a 0.5% improvement in weighted F1 score, while requiring approximately 35% fewer manually annotated images (Table 2.4). Classification accuracy is defined as the number of true positives (35,618 images) out of the total test set (36,032 images).

Tables 2.2 and 2.3 show the normalized multiclass confusion matrices for the supervised and semi-supervised learning models, respectively, which show the classification performances on the test set (i.e., new walking environments). For the semi-supervised model, the two transition classes (LG-IS and IS-LG) achieved the lowest categorical accuracies (90.6% and 90.4% respectively), which can be attributed to their small class distributions (only 3.1% and 1.8% of total images). Compared to supervised learning, the semi-supervised model achieved higher prediction accuracy for the most highly represented class (LG) while maintaining similar performance on the other walking environments. Figure 2.4 shows several failure cases in which the model misclassified the environment. The primary goal was not to improve individual class accuracy but rather to maintain performance while reducing the required labeled

training set.

Table 2.2. Normalized confusion matrix for the environment predictions (% accuracy) on the test set using the supervised MobileNetV2 model. The horizontal and vertical axes are the predicted and labeled classes, respectively.

	IS	IS-LG	LG	LG-IS
IS	96.9	1.5	0.3	1.3
IS-LG	6.5	90.5	2.4	0.6
LG	0.1	0.3	99	0.7
LG-IS	4.3	0.5	3.4	91.7

Table 2.3. Normalized confusion matrix for the environment predictions (% accuracy) on the test set using the semi-supervised MobileViT model. The horizontal and vertical axes are the predicted and labeled classes, respectively.

	IS	IS-LG	LG	LG-IS
IS	96.9	1.1	0.5	1.5
IS-LG	5.2	90.4	3.9	0.5
LG	0.1	0.1	99.5	0.3
LG-IS	3.4	0.4	5.6	90.6

2.5 Discussion

This study introduced a semi-supervised perception pipeline, ExoNet-SSL, for egocentric walking environment recognition under annotation constraints. By jointly leveraging labeled images from StairNet [31] and a large corpus of unlabeled images from ExoNet [20], the proposed approach substantially reduced the amount of manual annotation required for training while preserving high inference-time accuracy. In contrast to earlier work that relied exclusively on supervised learning, ExoNet-SSL

Table 2.4. Comparison between the supervised (MobileNetV2) and semi-supervised (MobileViT-XS) learning models in terms of classification performance on the test set and the annotated image requirements. The supervised model results were based on the previous state-of-the-art [31].

Method	Accuracy (%)	F1 Score (%)	Precision (%)	Recall (%)	Labeled Images
Supervised	98.4	98.4	98.5	98.4	461,328
SSL	98.8	98.9	98.9	98.8	300,000

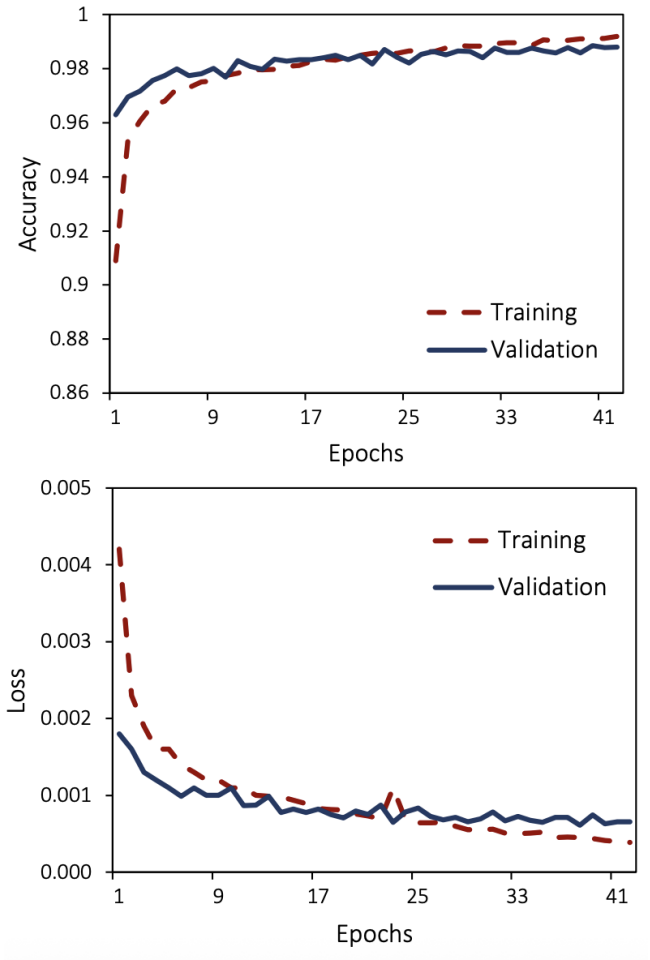

Figure 2.3. The image classification accuracy and loss for the training and validation sets using the semi-supervised MobileViT model.

utilized over 1.2 million unlabeled images to improve generalization across diverse real-world walking environments.

Quantitatively, the semi-supervised model achieved a classification accuracy of

98.8% during inference, exceeding the supervised baseline accuracy of 98.4% reported by Kurbis et al. [31], while requiring approximately 35% fewer labeled training images. These results demonstrate that annotation efficiency can be improved without compromising predictive performance, supporting the broader efficiency-oriented objectives of this dissertation.

Figure 2.4. Examples of failure cases. The first row shows images of incline stairs (IS) misclassified as incline stairs-level ground (IS-LG). The second row shows incline stairs misclassified as level ground-incline stairs (LG-IS). The third row illustrates level-ground misclassifications, including cases where level ground was incorrectly classified as incline stairs or transition states.

Several limitations of this study should be noted. Although the proposed approach significantly reduces labeling requirements, the absolute number of labeled images remains non-trivial. As such, this work should be viewed as an intermediate step toward fully unsupervised perception rather than an end point. Additionally, the reliance on 2D RGB imagery precludes the use of explicit geometric cues, such as stair height, which are relevant for fine-grained adaptive control of prosthetic legs and exoskeletons. Incorporating depth or multimodal sensing remains an important direction for future research.

From a methodological perspective, alternative semi-supervised strategies beyond FixMatch [11] could be explored to further reduce annotation demands. For example, adaptive or curriculum-based pseudo-labeling schemes may improve robustness by relaxing fixed confidence thresholds over training. Furthermore, although the MobileViT-based architecture was selected for its favorable accuracy-efficiency trade-off, the model was not evaluated directly on embedded or edge computing platforms. Empirical assessment of inference latency and power consumption on mobile hardware is therefore required before deployment in real-time assistive systems.

An additional challenge observed in this study concerns the allocation of classification accuracy across environmental classes. While all four walking environment states were treated as equally important, improving minority-class performance through focal loss and class weighting led to a modest reduction in overall accuracy compared to non-balanced training. This trade-off, referred to here as the class accuracy allocation phenomenon, highlights a tension between global accuracy and safety-critical minority class recognition. Understanding and mitigating this effect is an important topic for future investigation.

2.5.1 Contributions

The primary contributions of this chapter to the broader thesis on efficiency in robotics are as follows:

1. **Data Efficiency:** Demonstration that semi-supervised learning can substantially reduce manual annotation requirements while maintaining state-of-the-art accuracy in egocentric stair recognition.

2. **Architectural Efficiency:** Empirical validation of lightweight vision transformers as an effective accuracy-efficiency trade-off for perception in wearable robotic systems.

3. **Pipeline Scalability:** Establishment of a reproducible training pipeline capable of exploiting large volumes of unlabeled, in-the-wild egocentric data collected

during robot operation.

In summary, this chapter shows that semi-supervised learning provides a practical mechanism for improving annotation efficiency in egocentric perception without degrading predictive performance. By enabling the reuse of large-scale unlabeled datasets, the proposed approach supports the development of deployable perception systems for environment-adaptive control in wearable robotics and contributes a foundational efficiency pattern reused in subsequent chapters of this dissertation.

2.6 Comparative context and external validation

In addition to the IROS 2023 contribution on semi-supervised StairNet training (ExoNet-SSL), a subsequent peer-reviewed StairNet survey [31] consolidated results across the full benchmark, including single-frame CNNs, temporal models, semi-supervised methods, and deployment-oriented evaluations. That survey explicitly included ExoNet-SSL among the principal approaches and provides a unified comparative context. In this section, we summarize the most relevant evidence from that review, focusing on accuracy-efficiency trade-offs under consistent evaluation protocols.

The role of this comparison within the dissertation is twofold. First, it establishes that the semi-supervised approach preserves state-of-the-art accuracy while reducing reliance on labeled data. Second, it situates ExoNet-SSL relative to alternative stair-recognition pipelines, showing that semi-supervised single-frame models achieve a more favorable accuracy-efficiency balance than heavier temporal architectures when deployment constraints are considered.

2.6.1 Evaluation metrics

All methods are evaluated on held-out StairNet test splits using standard classification metrics, including accuracy and weighted F_1. The survey further recommends video-based splits to avoid leakage across adjacent frames; where applicable, results are reported under both image-level and video-level protocols.

To account for deployment efficiency, the survey reports the NetScore metric, which combines classification accuracy with model size and computational cost:

$$\mathrm{NS}(N) = 20 \log \frac{\mathrm{acc}(N)^{\alpha}}{\mathrm{param}(N)^{\beta}\,\mathrm{FLOPs}(N)^{\gamma}},$$

where α, β, γ balance predictive performance against parameter count and FLOPs. Higher values indicate more favorable accuracy-efficiency trade-offs for on-device inference.

2.6.2 Comparison with alternative StairNet models

Table 2.5. Accuracy and efficiency comparison across representative StairNet methods.

Type	Architecture	Train	Labeled	Unlabeled	Acc. (%)	NetScore
Baseline (single)	MobileNetV2	SL	515,452	–	98.4 (img)	186.8
Temporal (seq)	MoViNet (3D)	SL	515,452	–	98.3 (vid)	167.4
Temporal (seq)	MobileNetV2 + LSTM	SL	515,452	–	≈97–98	132.1
Temporal (seq)	MobileViT-XXS + LSTM	SL	515,452	–	≈98	155.0
SSL (single)	**MobileViT-XS**	**SSL**	**300,000**	**900,000**	**98.8 (img)**	**202.4**

Under the corrected video-based evaluation protocol recommended by the survey, absolute accuracies decrease for all single-frame models, while temporal 3D models attain the highest raw accuracy. However, these gains come at substantially increased parameter and FLOP budgets. In contrast, the semi-supervised MobileViT-XS model retains the highest NetScore across evaluated methods, indicating superior efficiency for deployment under resource constraints.

2.6.3 Data efficiency and coverage

From approximately 4.5 million unlabeled ExoNet frames, thresholded pseudo-labeling yields an SSL pool of roughly 1.2 million samples. The resulting class distribution broadly mirrors StairNet, indicating that semi-supervised mining preserves

Table 2.6. Class distribution of labeled StairNet data and pseudo-labeled ExoNet-SSL samples.

Class	SSL pool (#)	SSL pool (%)	StairNet (#)	StairNet (%)
LG	1,200,017	90.1	442,360	85.8
LG-IS	45,179	3.4	15,888	3.1
IS	73,404	5.5	48,179	9.4
IS-LG	13,640	1.0	9,025	1.8

distributional structure while expanding coverage beyond the manually annotated subset.

2.6.4 Deployment considerations

Deployment evaluations reported in the survey demonstrate that compact backbones such as MobileNet and MobileViT-XS support real-time inference on mobile hardware via TFLite, with inference latencies on the order of a few milliseconds on modern smartphones. Fully embedded prototypes operating on head-mounted platforms exhibit higher latencies but remain viable for user-aligned perception, trading throughput for ergonomic alignment.

Overall, the StairNet survey identifies ExoNet-SSL as a core method, achieving the highest NetScore among compared models and matching or exceeding single-frame supervised baselines while requiring substantially fewer labeled samples. Although temporal 3D models can attain higher absolute accuracy under strict video-based splits, their computational cost limits applicability in on-device control loops. In contrast, the semi-supervised MobileViT-XS configuration offers the most favorable accuracy-efficiency trade-off for wearable and embedded robotic perception.

This chapter resolves the annotation-efficiency node by demonstrating that semi-supervised MobileViT models can maintain StairNet accuracy while substantially reducing labeled data. With data-efficient perception in place, the next bottleneck

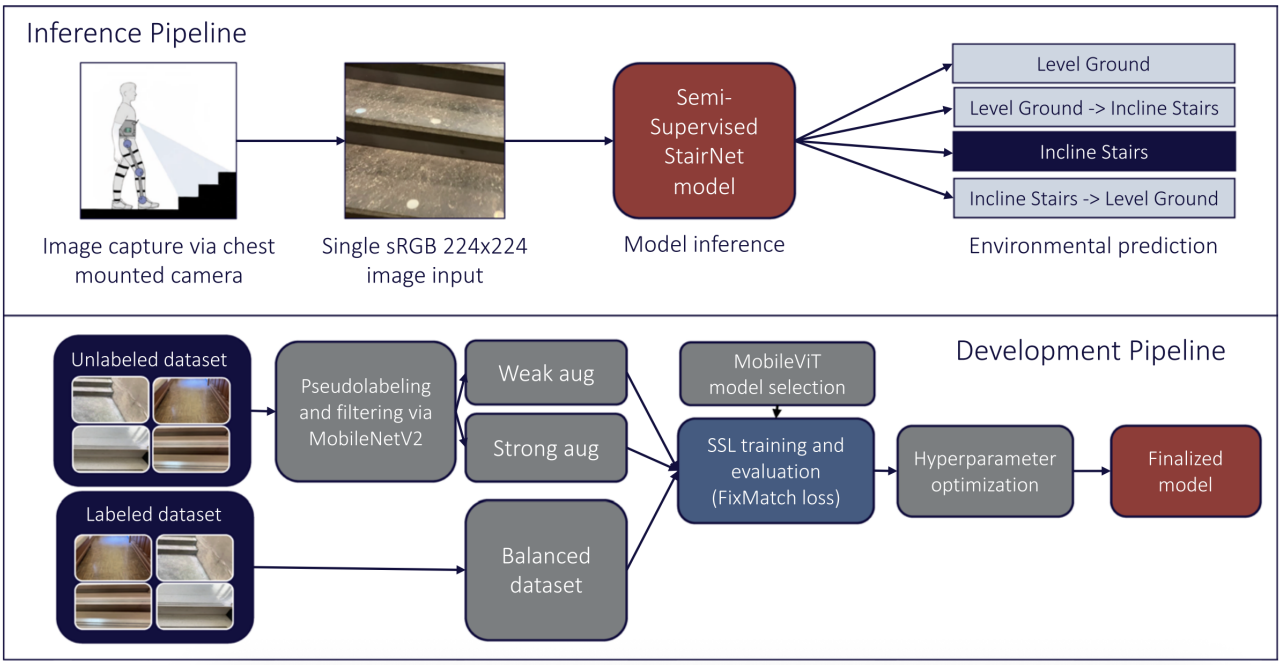

Figure 2.5. Training and inference pipeline for the ExoNet-SSL StairNet model, illustrating the integration of labeled and unlabeled data to reduce annotation requirements while preserving classification accuracy.

becomes the memory and latency cost of deploying multimodal perception inside an embodied navigation stack. Chapter 3 therefore moves from data to compute constraints, optimizing vision-language frontier maps for zero-shot ObjectNav under tight VRAM budgets. This transition links efficient learning to efficient deployment.

CHAPTER III

EFFICIENT VISION FOR ROBOTIC GOAL NAVIGATION

This chapter studies memory- and compute-efficient vision pipelines for robotic goal navigation, addressing the **memory efficiency** node of the taxonomy (Section 1). The presented analysis builds upon the published contribution *Balancing Performance and Efficiency in Zero-Shot Robotic Navigation* (Kuzmenko & Shvai, 2024), extending it with a unified efficiency-oriented framing and additional system-level analysis.

Chapter 2 established annotation and compute efficiency at the *perception* level – reducing label cost and model size for egocentric terrain classification. Once efficient features can be extracted from images, the next bottleneck shifts to the *system level*: how to assemble those features into a navigation pipeline that can run on hardware-constrained mobile robots? Standard VLFM pipeline configurations require over 6 GB VRAM (the baseline profiled in this chapter reaches 6,494 MiB) and impose high per-step latency, making them inaccessible to most deployed platforms. This chapter targets precisely that gap, showing that systematic module substitution and profiling recovers strong navigation performance within a 2.3$\times$ smaller memory footprint.

3.1 Introduction

Object Goal Navigation (ObjectNav) requires a robot to locate and navigate to a target object category (e.g., `chair`) in a previously unseen environment. Unlike point-goal navigation, which relies on geometric coordinates, ObjectNav demands semantic reasoning about object presence and likely spatial placement. Recent approaches based on **Vision-Language Frontier Maps** (**VLFM**) have demonstrated strong zero-shot performance by integrating vision-language models (VLMs) with open-vocabulary object detection and semantic mapping.

Despite their effectiveness, existing VLFM-based systems rely on computationally heavy components. Standard configurations commonly use large-scale VLMs such as BLIP-2 (1.4B parameters) in combination with high-capacity detectors such as

YOLOv7-E6E, resulting in substantial video RAM (VRAM) requirements and high inference latency. These characteristics hinder deployment on mobile robots and embedded platforms, where onboard compute and memory budgets are limited.

This chapter presents a systematic optimization study of the VLFM framework under explicit resource constraints. The goal is to characterize the trade-offs between navigation performance and computational efficiency by substituting heavyweight components with compact alternatives. By profiling individual modules and evaluating their impact on success rate (SR), success weighted by path length (SPL), VRAM usage, and inference latency, this study identifies a configuration that improves success rate while reducing memory requirements by a factor of 2.3. The results demonstrate that careful model selection enables zero-shot ObjectNav on consumer-grade hardware without sacrificing task success.

3.2 Problem Formulation

The ObjectNav task is defined as follows. An agent is initialized at a random pose s_0 in an unseen 3D environment and provided with a target object category c_{goal}. At each time step t, the agent receives visual observations o_t consisting of RGB-D images and executes an action $a_t \in \mathcal{A}$, such as forward motion or rotation. An episode terminates when the agent executes the `stop` action.

Success is achieved when the agent stops within a distance threshold d_{thresh} of an instance of c_{goal} and the object is visible in the agent's field of view. Performance is measured using:

- **Success Rate (SR):** the fraction of episodes completed successfully.
- **Success weighted by Path Length (SPL):** $\frac{1}{N}\sum_{i=1}^{N} S_i \frac{l_i}{\max(p_i, l_i)}$, where l_i denotes the shortest-path distance and p_i the executed path length.

The optimization objective is to maximize SR and SPL subject to hard hardware constraints:

$$\text{maximize } \{\text{SR}, \text{SPL}\} \quad \text{s.t.} \quad \text{VRAM}(\phi) \leq B_{\text{mem}}, \quad \text{Latency}(\phi) \leq B_{\text{time}},$$

where ϕ denotes the set of perception models comprising the navigation pipeline.

3.3 Methods

3.3.1 Dataset

Experiments were conducted using the Habitat-Matterport 3D (HM3D) dataset, which contains 1,000 high-resolution 3D scans of real-world indoor environments. For architecture selection and module composition studies, a small validation subset (`val-mini`) comprising two scenes and 30 episodes was used. Final evaluations were performed on the full HM3D validation split. Each episode corresponds to a navigation instance defined by a start pose and a target object within a scene.

3.3.2 Reinforcement Learning Policy

All experiments employ the pre-trained Point Goal Navigation (PointNav) policy introduced in prior work. After an initial spin-initialization phase that scans the environment, the agent navigates toward either a frontier waypoint or a detected object waypoint depending on target visibility. The policy relies on egocentric depth observations and relative pose information and does not use RGB input. Training was performed using Variable Experience Rollout (VER). The policy is intentionally kept fixed to isolate the effect of perception-side efficiency.

3.3.3 VLFM Pipeline

The VLFM framework integrates three core components: a VLM for semantic scene understanding via cosine similarity between visual and textual embeddings, an object detector to localize candidate targets, and a segmentation model to extract object contours. This study extends the pipeline with an optional visual question answering (VQA) module used to verify detected object regions.

3.3.4 nanoLLaVA as VQA

nanoLLaVA is a compact 1.1B-parameter multimodal model designed for efficient inference on edge devices. It employs a Qwen-family language backbone with Flash Attention and a SigLip vision tower. Unlike BLIP-2, nanoLLaVA does not natively

support text embedding extraction, which precludes its use as the primary VLM. Consequently, it is incorporated as a VQA verification module only, running on a second GPU or after the navigation pipeline has freed sufficient memory; integration as a full VLM is left for future work. The overall pipeline is illustrated in Figure 3.1.

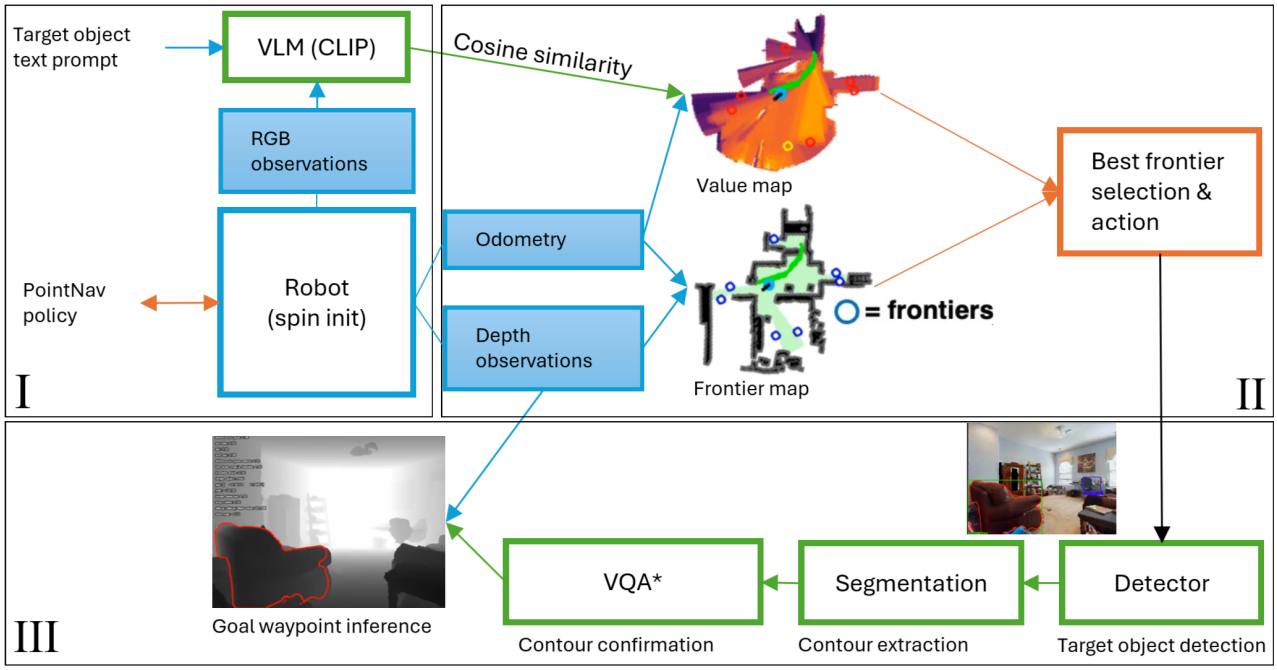

Figure 3.1. Overview of the VLFM-based navigation pipeline with optional VQA verification.

3.3.5 Model Parameter Study

The parameter counts and VRAM usage of all pipeline components were profiled, as these quantities directly determine deployment feasibility. The baseline VLFM configuration requires 6,494 MiB VRAM, whereas the optimized configuration requires 2,774 MiB, yielding a 2.3× reduction. BLIP-2 (1.4B parameters) is replaced with CLIP-ViT-B/32 (151.3M), and YOLOv7-E6E (151.7M) is replaced with YOLOv7-W6 (70.4M). MobileSAM remains unchanged. The nanoLLaVA VQA module (Table 3.1) is evaluated separately on a 48 GB GPU; its memory footprint exceeds the optimized navigation budget and it is therefore treated as an optional verification component rather than part of the core pipeline.

Table 3.1. Parameter counts and VRAM usage of VLFM modules.

Model	Parameters	VRAM (MiB)
BLIP-2	1.4B	2976
CLIP-ViT-B/32	151.3M	806
YOLOv7-E6E	151.7M	3032
YOLOv7-W6	70.4M	1482
MobileSAM	9.8M	486
nanoLLaVA	1.1B	6002

3.4 Experiments

Experiments were designed to jointly evaluate navigation performance and computational efficiency under constrained hardware budgets. The agent uses RGB, depth, and odometry inputs to construct value and frontier maps, which are combined into goal-directed waypoints (Figure 3.2).

3.4.1 Vision-Language Model Selection

Several VLM configurations were evaluated. nanoLLaVA was initially considered as a primary VLM, but its lack of text embedding support motivated the use of CLIP-based encoders. Experiments were conducted using CLIP-ViT-B/32, CLIP-ViT-B/16, and ConvNeXt-XXL backbones. BLIP-2 was retained as a baseline.

3.4.2 Object Detection and Segmentation

Object detection is performed using YOLOv7-family models. To minimize resource usage, GroundingDINO was excluded. Experiments span YOLOv7, YOLOv7-W6, YOLOv7-E6, and YOLOv7-E6E. Segmentation is consistently performed using MobileSAM.

3.4.3 VQA Integration

nanoLLaVA is used as a VQA module to confirm detected object contours. BLIP-2-based VQA was not feasible due to memory constraints.

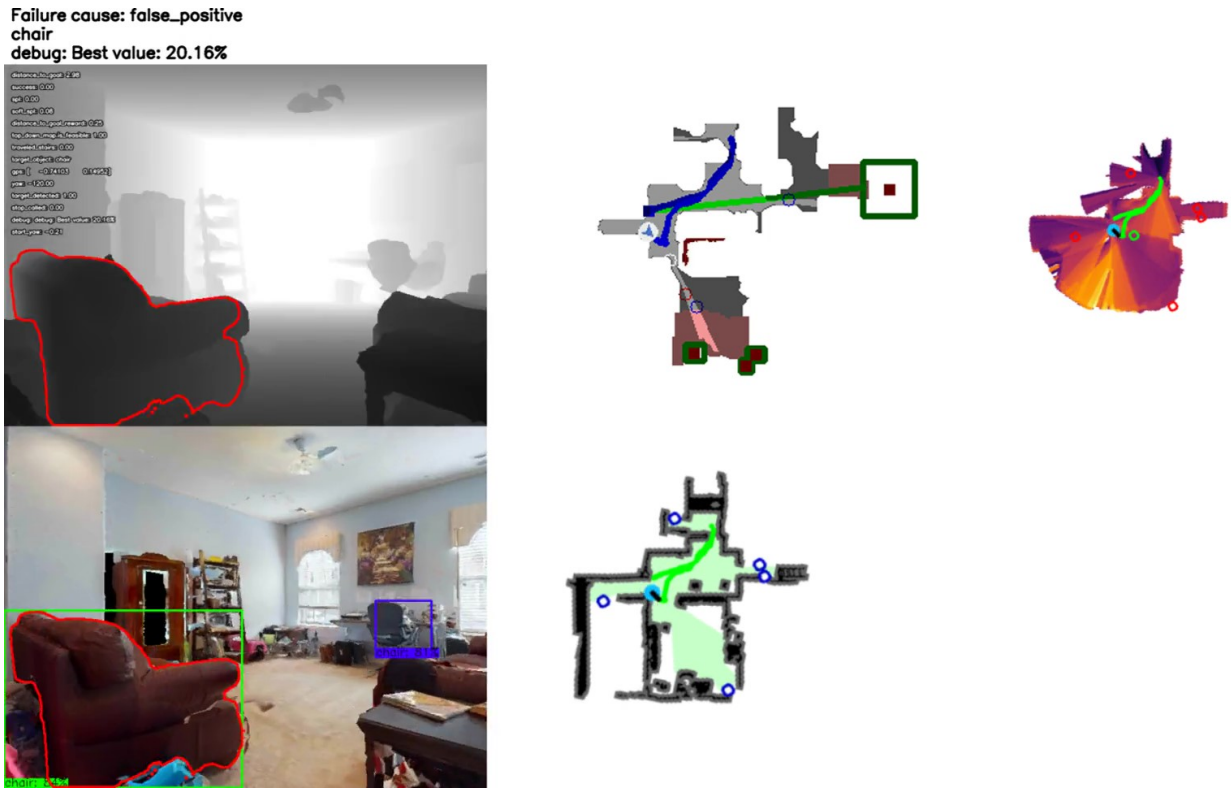

Figure 3.2. Example HM3D episode showing detection results, segmentation output, and constructed frontier maps.

Table 3.2. Inference time breakdown on the HM3D val-mini split.

Module	BLIP-2 setup (s)	CLIP setup (s)
VLM	521.0	419.3
Detector	850.4	937.7
Segmentation	57.3	36.6
VQA	–	105.2

3.5 Results

3.5.1 Val-mini

Initial experiments on the val-mini subset demonstrate that CLIP-ViT-B/32 improves SR relative to BLIP-2 while reducing inference time. Integration of VQA improves SR in small-scale settings but does not generalize to the full validation split. Lightweight detectors reduce per-step latency but increase episode length due to lower detection accuracy, revealing a trade-off between module efficiency and navigation

consistency.

Table 3.3. Performance on the HM3D val-mini split.

VLM	Detector	VQA	SPL ↑	SR ↑
BLIP-2	YOLOv7-E6E	–	32.46	53.33
CLIP-ViT-B32	YOLOv7-E6E	–	28.20	56.67
CLIP-ViT-B32	YOLOv7-W6	✓	29.42	70.00
CLIP-ViT-B32	YOLOv7-E6E	✓	30.35	70.00

3.5.2 Full Validation Set

Final evaluation on 2,000 HM3D validation episodes confirms that VQA does not provide consistent gains under distributional shift. The CLIP-ViT-B/32 + YOLOv7-W6 configuration achieves higher SR (+1.55%) than BLIP-2 VLFM while using 2.3× less VRAM, with only a minor SPL reduction (-0.98).

Table 3.4. Comparison to state-of-the-art on HM3D validation set.

Approach	Training	Params	SPL ↑	SR ↑
PIRLNav	ObjectNav	∼23M	27.1	**64.1**
ZSON	ImageNav	∼85M	12.6	25.5
ESC	None	∼65M	22.3	39.2
VLFM (BLIP-2)	None	1.56B	**30.4**	52.5
VLFM (CLIP, ours)	None	231.5M	<u>29.42</u>	<u>54.05</u>
VLFM (CLIP+VQA, ours)	None	1.33B	27.24	53.35

3.6 Discussion

This chapter presented a resource-aware optimization study of VLFM-based ObjectNav. By systematically profiling and replacing perception modules, it demonstrated that heavyweight VLM backbones can be substituted with compact alternatives without degrading task success. The CLIP-ViT-B/32 + YOLOv7-W6 configuration

improves success rate relative to BLIP-2 while reducing VRAM requirements by a factor of 2.3, enabling deployment under constrained hardware budgets.

The analysis further revealed that reducing per-step latency alone is insufficient to improve overall efficiency, as lower-quality perception increases episode length. This highlights the importance of balancing representation quality with computational cost. The optional VQA module based on nanoLLaVA improves performance in limited settings but does not generalize under broader distributions.

3.6.1 Contributions

1. **Resource-Aware Optimization:** Demonstration that compact vision-language encoders can replace large-scale VLMs in zero-shot ObjectNav while preserving success rate.

2. **Module-Level Profiling:** Identification of object detection as a primary latency bottleneck rather than the VLM.

3. **Deployment Viability:** A concrete VLFM configuration enabling zero-shot ObjectNav within a 4 GB VRAM budget.

3.6.2 Limitations and Future Work

The optimized pipeline has not yet been evaluated on physical robots, and experiments are limited to HM3D. Future work includes real-world deployment, evaluation on additional datasets such as MP3D and Gibson, and joint optimization of the navigation policy. Extending nanoLLaVA with text embedding support would enable its use as a full VLM, further improving efficiency.

This chapter addresses memory efficiency by restructuring VLFM-based ObjectNav to fit VRAM-constrained platforms without sacrificing success rate. Once perception and navigation can run within realistic hardware limits, the dominant costs shift to learning and actuation: how long it takes to train policies and how much energy they consume in operation. Chapter 4 targets these temporal and energy efficiency nodes through distillation-driven training compression and fuel-aware reward

shaping. Together, the next chapter extends efficiency from memory budgets to time and physical energy.

CHAPTER IV

EFFICIENT REINFORCEMENT LEARNING

This chapter addresses the temporal and energy efficiency pillar of the taxonomy (Section 1). While Chapters 2 and 3 focused on perception, I focus here on control via RL. I present two complementary approaches to efficiency in RL: minimizing the wall-clock time required for training via distillation, and minimizing the physical energy consumed during actuation via reward shaping.

Chapters 2 and 3 addressed efficiency in the sensing and perception stack – reducing annotation cost and memory footprint of visual modules. Those chapters, however, leave the *control* layer untouched: even a perception-efficient robot still requires a policy that can be trained quickly and actuated economically. This chapter closes that gap. The first contribution (Section 4.1) targets *training efficiency*: by distilling a 317M-parameter multi-task model into a 1M-parameter student, it dramatically reduces the wall-clock cost of acquiring a deployable control policy. The second contribution (Section 4.2) targets *actuation efficiency*: by incorporating a fuel penalty into the reward, it produces policies that conserve physical energy during execution. Together, they address the two dimensions of temporal and energy efficiency that perception-level methods cannot reach.

4.1 Distilling Model-Based Multi-Task Agents

This section is based on the contribution *TD-MPC-Opt: Distilling Model-Based Multi-Task Reinforcement Learning Agents* (Kuzmenko & Shvai, 2025).

4.1.1 Introduction

Reinforcement learning (RL) frames sequential decision-making as the optimization of cumulative reward in a Markov Decision Process [32]. It has demonstrated substantial progress in continuous control and decision-making, from vision-based manipulation to dexterous robotics and locomotion [69, 89, 90]. Yet, constructing a

single agent that performs well across many tasks, while remaining suitable for deployment under realistic compute and memory constraints, remains an open challenge, particularly in robotic settings where onboard compute is limited and inference must occur reliably in real time.

Model-based RL aims to address these challenges by improving sample efficiency and leveraging predictive structure during action selection [13, 91]. Large multi-task agents such as DreamerV3 [15] and PWM [92] demonstrate that high-capacity models can capture broad multi-task competence, but they also require substantial computational resources, limiting practical deployment.

TD-MPC [33] and TD-MPC2 [34] advance this line of work by combining temporal-difference learning with online model predictive control, achieving strong generalization in multi-task continuous-control domains. However, high-capacity versions of TD-MPC2 (up to 317M parameters) may be difficult to deploy directly on embedded platforms or distributed multi-robot systems. This raises the practical challenge of transferring the behavior of a large TD-MPC2 agent to a compact model while preserving task competence.

This work examines reward-level distillation as a mechanism for transferring cross-task control structure from a 317M-parameter TD-MPC2 model to a 1M-parameter student. The evaluation is conducted on MT30 [34], a diverse multi-task benchmark built on the DeepMind Control Suite (DMC) [93], which includes tasks with various observation and action specifications (Fig. 4.1).

The main contributions are as follows:

- A general reward-level distillation framework that transfers the control structure of a high-capacity teacher policy into a compact student model, independent of the underlying architecture.

- Experimental results on a diverse multi-task control suite, showing that the distilled student reaches a normalized score of 28.45, surpasses training from scratch, and retains its performance under lightweight post-training compres-

sion.

- An empirical study showing that varying batch sizes and training horizons reveals a clear sample-efficiency benefit for distillation over direct student training.
- A characterization of the role of the distillation coefficient d, identifying stable performance regimes for multi-task distillation.
- A comparison of reward-level and latent-space alignment strategies, showing that latent alignment breaks down under severe latent dimensionality mismatch, making reward-level signals the more reliable transfer target in such scenarios.

These results indicate that a high-capacity TD-MPC2 agent can serve as a reusable teacher, from which compact students can be distilled for deployment under different resource constraints.

4.1.2 Problem Formulation

I consider a multi-task RL setting with a set of tasks $\mathcal{T} = \{T_1, \ldots, T_N\}$. I assume access to a pre-trained teacher policy π_ϕ parameterized by ϕ, which has high capacity and performance but high inference cost. The goal is to learn a student policy π_θ parameterized by θ, where $|\theta| \ll |\phi|$.

The objective augments the original TD-MPC2 loss with a reward-level distillation term:

$$\mathcal{L}(\theta) = \mathcal{L}_{\text{orig}}(\theta) + d\,\mathcal{L}_{\text{dist}}(\pi_\theta, \pi_\phi)$$

where $\mathcal{L}_{\text{dist}}$ is an MSE loss that aligns teacher and student reward predictions for the same state-action pairs, and d controls the distillation strength. I specifically target the minimization of wall-clock training time t_{train} required for π_θ to reach a performance threshold R_{thresh}.

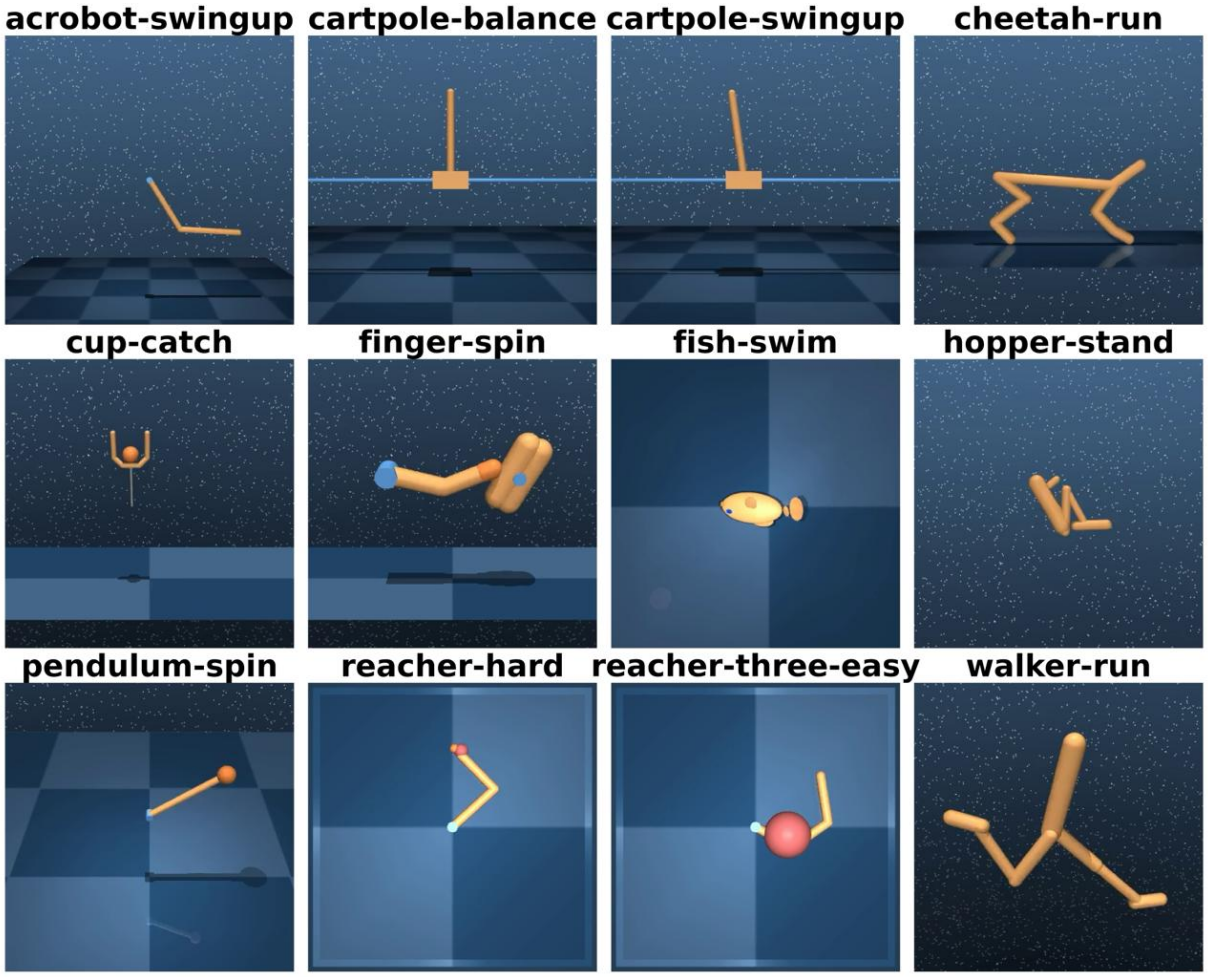

Figure 4.1. Initial states from 12 example tasks in the MT30 benchmark, illustrating variation in embodiment, action dimensionality, and control objectives.

4.1.3 Methods

This approach transfers the capability of a high-capacity multi-task policy into a compact model suitable for compute- and memory-constrained deployment. The method uses a general reward-based teacher-student distillation framework (Figure 4.2), followed by lightweight post-training compression to further reduce the model footprint without sacrificing behavior.

In this implementation, I instantiate this framework using a TD-MPC2 style model-based RL backbone distilled into a compact student and evaluate it on the MT30 benchmark. MT30 is a curated suite of 30 continuous-control tasks drawn from the DM Control environment family, covering locomotion, manipulation, balancing, and object-interaction behaviors. It is widely used to assess generalization, robustness, and multi-task learning ability because it spans diverse dynamics and control challenges within a unified simulation setup.

4.1.3.1 Distillation objective TD-MPC2 weights the consistency $\mathcal{L}_c$, reward $\mathcal{L}_r$, and value $\mathcal{L}_v$ losses using respective fixed scalar coefficients α_c, α_r, and α_v:

$$\mathcal{L}_{\text{orig}} = \alpha_c \mathcal{L}_c \ + \ \alpha_r \mathcal{L}_r \ + \ \alpha_v \mathcal{L}_v.$$

In the original formulation, the coefficients are set to α_c=20, α_r=0.1, and α_v=0.1, which places substantially greater weight on latent dynamics consistency than on reward or value terms. For environments with terminal states, an additional termination loss $\mathcal{L}_{\text{term}}$ is included with coefficient $\alpha_t = 1$, but this term is inactive for MT30 since the tasks are non-episodic. I retain these coefficient values without modification in all experiments.

To transfer task-specific behavior, I introduce an additional reward distillation loss:

$$\mathcal{L}_{\text{dist}} = \text{MSE}(R_{\text{teacher}}(s, a), \ R_{\text{student}}(s, a)).$$

Here, the mean squared error (MSE) penalizes differences between the teacher's and student's predicted rewards for the same state-action (s, a) pairs. This aligns

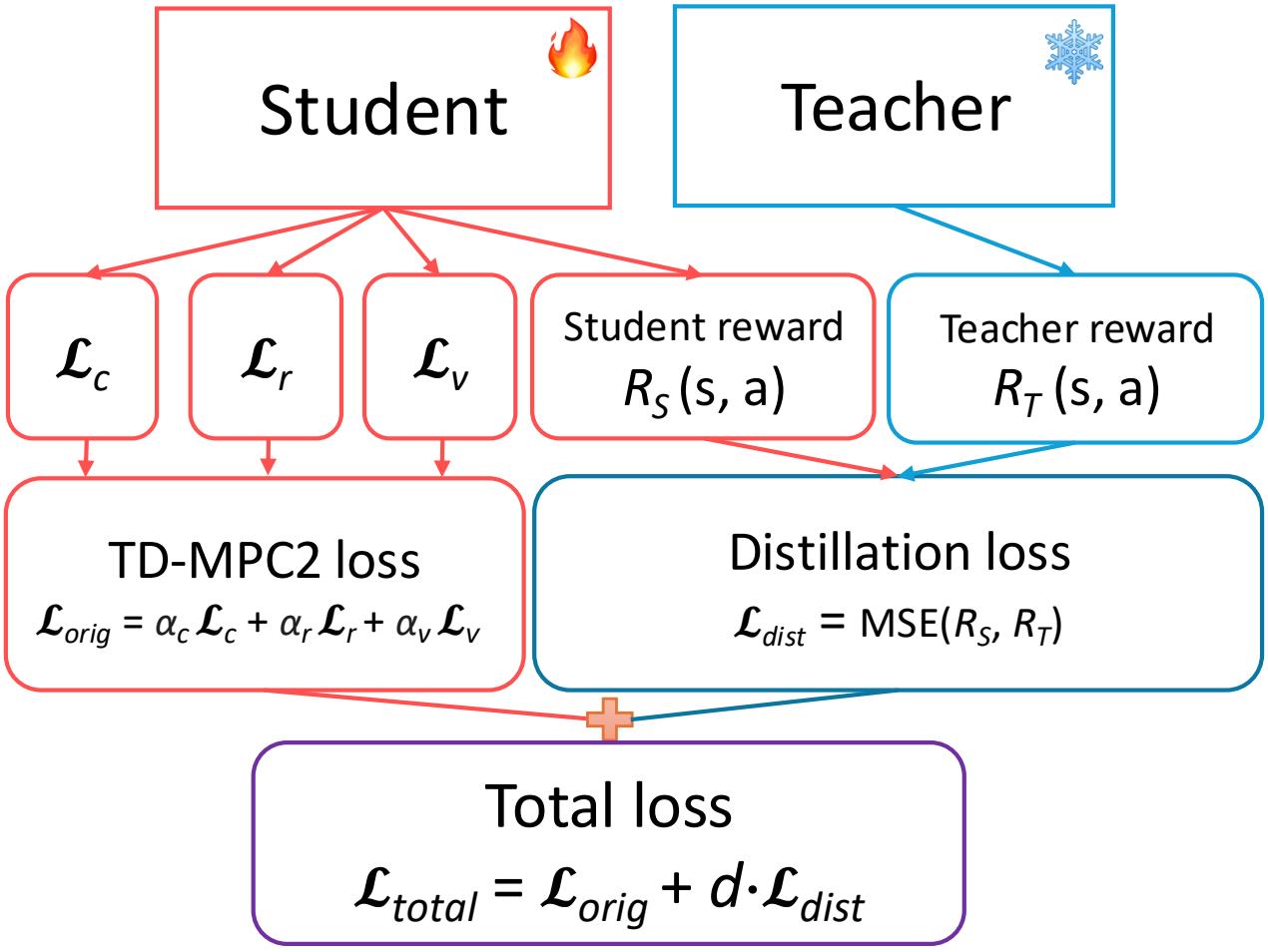

Figure 4.2. Overview of the proposed distillation process. The teacher (317M parameters) and the student (1M parameters) receive identical state-action inputs. The original TD-MPC2 objective is retained for the student, and an additional reward-matching loss is applied to align the student's reward predictions with the teacher. The total loss is a weighted combination of the original learning signal and the distillation signal.

the student's action preferences with the teacher's behavior without requiring it to replicate the teacher's internal latent representation.

The final training loss is defined as follows:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{orig}} + d\mathcal{L}_{\text{dist}},$$

where $d \in [0, 1]$ is a distillation coefficient controlling the influence of the teacher on the student's learning signal. Empirically, values around $d = 0.4$ provide the best trade-off between stability and transfer performance (Table 4.1).

4.1.3.2 Evaluation Metric I use the normalized score as the primary evaluation metric, consistent with [34]. For each of the 30 tasks in MT30, the rewards are averaged to yield a final score in the [0, 100] range, where 100 corresponds to perfect performance on every task. This normalization provides a consistent comparison across tasks with differing reward scales and difficulty levels.

4.1.3.3 Training Process The teacher is the publicly released 317M-parameter TD-MPC2 checkpoint trained on MT30. The student is a 1M-parameter smaller variant of the same architecture, initialized from scratch. Both models share the same planning and optimization hyperparameters as in the original TD-MPC2 implementation [34]. I also evaluate distillation from a smaller 48M-parameter teacher to assess the effect of teacher capacity.

Training is carried out under three horizon settings, reflecting different compute and deployment scenarios: (i) short-horizon (200K steps), corresponding to constrained training budgets; (ii) full-dataset (337K steps), in which the student sees every MT30 transition once; and (iii) extended (1M steps), representing higher-accuracy refinement with repeated data exposure. I vary the distillation coefficient d, batch size, and training horizon across experiments.

4.1.3.4 Quantization After distillation, I apply post-training quantization to reduce model size, using three strategies:

Table 4.1. Effect of distillation strategy and coefficient d on student performance (200K steps, batch size 256, 317M teacher). Latent next-state distillation variants underperform reward-only distillation.

Distillation method	d	*Normalized score*
latent (linear projection)	–	7.69
latent (PCA)	–	8.78
reward-only	0.05	13.61
no distillation	–	14.04
reward-only	0.25	14.49
reward-only	**0.40**	**17.85**
reward-only	0.55	16.08
reward-only	0.60	14.83
reward-only	0.90	13.79

1. **FP16**: stores weights and activations in 16-bit floating point, yielding approximately a 2x size reduction with minimal performance impact.

2. **INT8**: applies 8-bit integer quantization across all layers, providing the highest compression at the cost of greater performance degradation.

3. **Mixed precision**: quantizes compute-heavy layers to INT8 while keeping input, output, and normalization layers in FP16 to maintain stability.

4.1.3.5 Experimental Setup For the MT30 benchmark, I utilized the full available dataset of 690,000 episodes (345,900,000 transitions). This multi-task benchmark is particularly challenging as it comprises tasks with varying observation and action spaces, requiring the agent to adapt to diverse environments. I conducted these experiments on a desktop PC with a single RTX 3060 GPU (12GB VRAM).

4.1.4 Results

I evaluate whether reward-level distillation enables a compact TD-MPC2 student to retain multi-task competence under a significantly reduced model size. I refer to a 1M-parameter TD-MPC2 model trained from scratch (without distillation) as a direct baseline.

The evaluation is structured around three questions:

- **RQ1:** Can a 1M-parameter student distilled from a 317M-parameter teacher match or exceed the performance of a direct baseline?
- **RQ2:** How do distillation horizon and batch size influence sample efficiency and final performance?
- **RQ3:** To what extent do post-training quantization methods reduce memory footprint, and how does performance degrade as quantization becomes more aggressive?

4.1.4.1 Short-term Distillation I first reproduce the checkpoint of the 1M-parameter direct baseline from TD-MPC2 by training for 200K steps with a batch size of 1024. This checkpoint serves as the primary reference point for the distilled models, ensuring a consistent comparison under identical data and optimization settings. To evaluate short-horizon sample efficiency (RQ1), I compare distilled students and direct baselines at 200K steps under varying batch sizes. This setting reflects scenarios where compute or time constraints limit training duration.

Table 4.2 summarizes the results. The student trained for 200K steps with a batch size of 1024 achieves 18.11 NS, slightly lower than the direct baseline with NS of 18.7. However, distillation combined with smaller batch sizes yields a clear advantage. With a batch size of 256, the distilled model reaches 17.85 NS, compared to 14.04 NS of the direct baseline. With a smaller batch size of 128, the distilled policy achieves 17.37 NS, approaching full-performance behavior with less data exposure. These results

indicate that reward-level distillation can improve early-stage learning efficiency, especially when training under limited compute budgets or smaller batch sizes.

Table 4.2. Performance comparison between the distilled 1M-parameter students and the direct baselines across different training configurations on the MT30 benchmark.

Method	*Batch*	*Steps*	*Norm. score*
distilled	128	200K	17.37
direct baseline	256	200K	14.04
distilled	256	200K	<u>17.85</u>
direct baseline	1024	200K	<u>18.70</u>
distilled	1024	200K	18.11
direct baseline	1024	337K	<u>26.94</u>
distilled	1024	337K	25.44
direct baseline	256	1M	27.36
distilled	256	1M	28.12
distilled + FP16	256	1M	**28.45**

4.1.4.2 Extended Distillation To examine whether longer training horizons improve transfer quality (RQ2), I extend distillation to match the teacher's full training regime. With batch size 1024 and 337K steps, the distilled student (25.44 NS) approaches, but does not surpass, the direct baseline (26.94 NS). This suggests that extended training alone may not fully overcome representational capacity differences.

When distillation is extended to 1M steps at batch size 256, the student (28.12 NS) slightly exceeds the direct baseline (27.36 NS). The best-performing model (28.45) is obtained after FP16 post-training quantization, indicating that sample efficiency and compression gains can be achieved together. Figure 4.3 compares performance trajectories over 1M training steps. Distillation improves short- and mid-horizon

performance (200K–600K steps), and maintains a consistent advantage throughout long-horizon training (up to 1M steps).

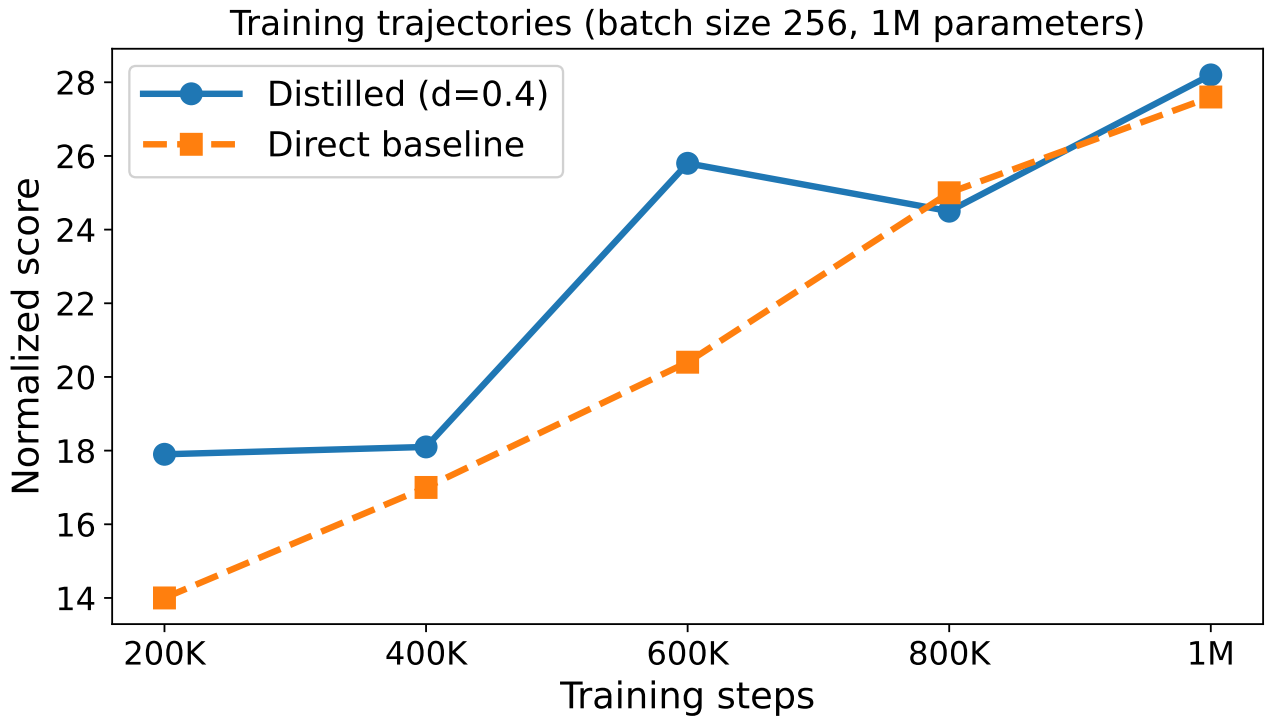

Figure 4.3. Comparison of distilled and direct baseline 1M-parameter models on the MT30 benchmark. Normalized scores are reported at 200K-step intervals over the 1M-step training horizon. The distilled model ($d = 0.4$) converges faster and maintains a consistent performance advantage in early and mid horizons.

4.1.4.3 Batch Size Study I evaluate batch sizes of 128, 256, and 1024 across short-horizon (200K steps), full-dataset (337K steps), and extended (1M steps) scenarios (Table 4.2). Batch size has a strong effect on distillation quality. Batch size 256 consistently yields the best transfer performance, while batch size 1024 is more suited to directly trained baselines. This indicates that reward-level distillation benefits from more frequent gradient updates, whereas larger batches provide less useful alignment signal. Reducing batch size further (e.g., to 128) did not provide additional benefit, suggesting that the advantage arises from update frequency rather than simply minimizing batch size.

4.1.4.4 Latent-Space Alignment I also evaluated distillation in the latent state space. Since the teacher and student produce latent representations of different dimensionalities (1376-dim and 128-dim for teacher and student, respectively), I aligned them by projecting the teacher's latent vector into the student's space using either a learned linear layer or principal component analysis (PCA), and minimized the L2 distance between them. This approach consistently underperformed reward-only distillation (Table 4.1), suggesting that latent alignment is ineffective under the capacity mismatch and that reward-level signals provide a more reliable transfer target in this setting.

4.1.4.5 Teacher Size Impact I evaluate the effect of teacher capacity on distillation. Distilling from the 317M-parameter teacher yields substantially higher student performance than using the 48M-parameter teacher (Table 4.3), showing that larger teachers provide richer and more transferable task structure.

Table 4.3. Distillation from a larger teacher yields substantially higher student performance (1M-parameter student, 200K steps, batch size 256).

Teacher size	*Normalized score*
48M	13.61
317M	**17.85**

4.1.4.6 Quantization Results To address RQ3, I apply FP16, mixed precision, and INT8 post-training quantization to the distilled 1M-parameter student trained for 1M steps with batch size 256. FP16 reduces model size by approximately 50% with no performance loss (28.45 NS). Mixed precision further reduces size but introduces moderate degradation (13.53 NS). INT8 compression yields substantial memory gain but results in heavy performance loss (5.09 NS), indicating a trade-off between deployability and control fidelity. Table 4.4 reports the effects of these configurations

on model size and performance.

Table 4.4. Quantization results for the distilled 1M-parameter student (1M steps, batch size 256, $d = 0.4$). FP16 maintains performance while reducing model size by approximately 50%, whereas more aggressive quantization introduces severe performance degradation.

Metric	Original	FP16	Mixed	INT8
Normalized Score	28.12	**28.45**	13.53	5.09
Model Size (MiB)	7.9	3.9	2.8	1.95

4.1.5 Discussion

In this work, I explored whether a high-capacity multi-task model-based RL agent can be effectively compressed into a lightweight policy without losing core competence. Using reward-level distillation, a 1M-parameter student trained for 1M steps reaches a normalized score of 28.45 (with FP16 quantization), compared to 27.36 for a directly trained student of identical capacity and budget – a consistent 1.1-point gain. For context, the published TD-MPC2 checkpoint for the 1M-parameter configuration achieves 18.70 NS at 200K steps with batch size 1024; the gains from extended training and distillation relative to that reference confirm the benefit of the proposed approach. After post-training FP16 quantization, the model retains this performance while reducing memory footprint by approximately 50%, making it even more viable for deployment under realistic compute constraints.

The results highlight two central observations. First, reward-level distillation is sufficient to transfer cross-task reward structure and control behavior of the 317M-parameter teacher. Second, smaller batch sizes consistently outperform larger ones, even with extended training. This suggests that more frequent parameter updates and finer gradient resolution are more critical than aggressive batch scaling in this setting. The gains from longer distillation horizons further indicate that the teacher encodes

transferable structure that the student can progressively approximate. Distillation also provides a sample-efficiency benefit at shorter horizons, where students distilled with smaller batch sizes outperform similar directly trained models.

This work has several limitations. All experiments were conducted in simulation, and the behavior of the distilled policy on physical systems has yet to be evaluated. The evaluation is limited to the MT30 benchmark. Extending the approach to larger task suites or to previously unseen tasks remains for future work. Additionally, this approach relies solely on reward-level supervision; latent state alignment or behavioral imitation may further improve transfer in more complex tasks.

Future work may focus on aligning latent representations or transition dynamics so that the student can inherit deeper structural priors. Hardware evaluation is an important next step to assess robustness and safety beyond simulation. Another direction is to adjust the distillation coefficient d over the course of training to improve stability. More broadly, treating the high-capacity teacher as a one-time pretraining investment enables a practical "train once, distill many" workflow, where multiple lightweight students are derived for different deployment settings.

4.2 Energy-Aware RL Agents

This section is based on the contribution *Energy Conservation for Autonomous Agents Using Reinforcement Learning* (Beimuk & Kuzmenko, 2025).

4.2.1 Introduction

RL has shown strong potential in autonomous racing for its adaptability to complex and dynamic driving environments. However, most research prioritizes performance metrics such as speed and lap time. Limited consideration is given to improving energy efficiency, despite its increasing importance in sustainable autonomous systems. This work investigates the capacity of RL agents to develop multi-objective driving strategies that balance lap time and fuel consumption by incorporating a fuel usage penalty into the reward function. To simulate realistic uncertainty, fuel usage is

excluded from the observation space, forcing the agent to infer fuel consumption indirectly.

I compare various penalty strengths against the non-penalized agent and evaluate fuel consumption, lap time, acceleration and braking profiles, gear usage, engine RPM, and steering behavior. Results show that mild to moderate penalties lead to significant fuel savings with minimal or no loss in lap time. These findings highlight the viability of reward shaping for multi-objective optimization in autonomous racing and contribute to broader efforts in energy-aware RL for control tasks.

4.2.2 Problem Formulation

I model the energy-aware control problem as a multi-objective Markov Decision Process (MDP) [9, 94]. The problem involves two competing objectives: task performance $f_1(\pi)$ (speed and progress) and energy efficiency $f_2(\pi)$ (fuel economy). No single policy simultaneously maximizes both; the trade-off is characterized by the *Pareto frontier*:

$$\mathcal{P} = \{\pi : \nexists\, \pi' \text{ s.t. } f_1(\pi') \geq f_1(\pi),\ f_2(\pi') \geq f_2(\pi),\ \exists i : f_i(\pi') > f_i(\pi)\}.$$

Policies on $\mathcal{P}$ cannot improve one objective without degrading the other. Rather than solving the full multi-objective problem, I parameterize the trade-off via a scalar penalty β and seek the corresponding point on $\mathcal{P}$ for each β value. The goal is to learn a policy π that maximizes a cumulative reward R_t which balances task performance (speed and progress) and energy consumption.

The reward function is defined as:

$$r_t = \underbrace{v_t \cdot (1 - \alpha \cdot d_t)}_{\text{Task Performance}} - \underbrace{\beta \cdot f_t}_{\text{Energy Penalty}}$$

where v_t is velocity, d_t is deviation from the centerline, f_t is instantaneous fuel consumption, and β is a hyperparameter controlling the importance of energy efficiency. The objective is to find β^* such that fuel consumption is minimized while lap time remains within ϵ of the optimal time.

4.2.3 Methods

I use the AssettoCorsaGym interface developed by Remonda et al. [95], which integrates a high-fidelity racing simulator Assetto Corsa with the OpenAI Gym environment. Figure 4.4 shows an overview of the AssettoCorsaGym platform.

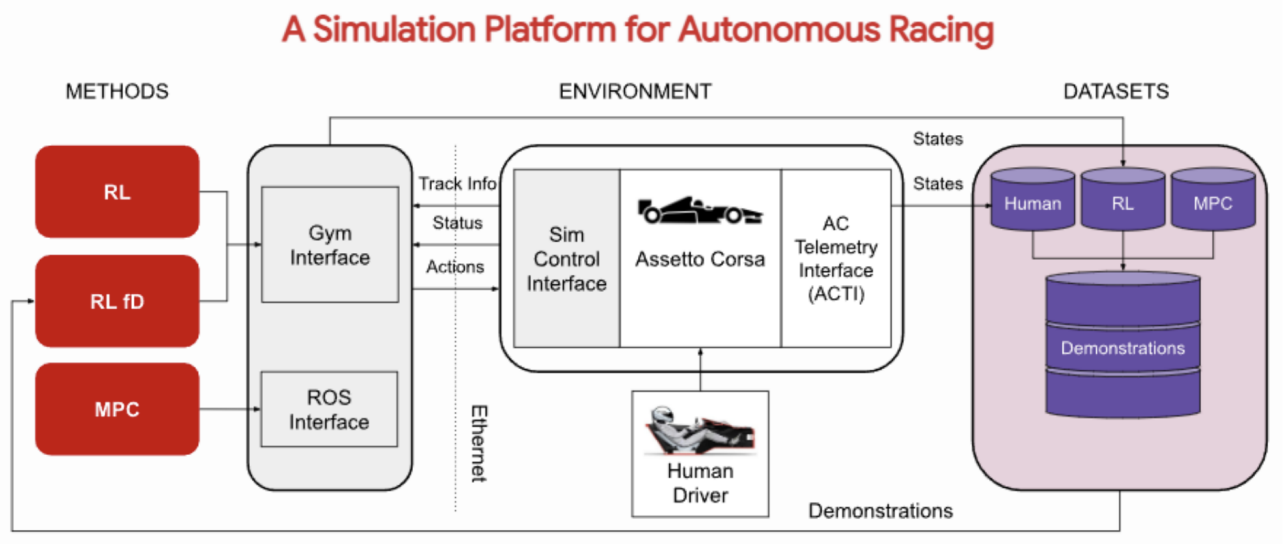

Figure 4.4. The architecture of the AssettoCorsaGym platform [95].

I used the SAC RL algorithm designed specifically for continuous control tasks [70]. It offers strong performance in autonomous racing benchmarks [95], as well as robustness in tasks that require balance between speed, control, and long-term planning [70, 95]. The setup included two tracks – Track A (Austria) and Track B (Monza) (Figure 4.5), and two vehicle models – a lightweight Formula 3 series car (F317) and a heavier GT3 series car (BMW Z4 GT3). These selections were made from the Assetto Corsa environment to align with the original AssettoCorsaGym dataset. Track A offers a balanced layout for general driving evaluation, while Track B, containing tighter corner sequences, tests performance in more challenging conditions. The two vehicles differ in dynamics. The F3 series car is a lightweight and high-downforce car that is nimble and agile, while the GT3 is a heavier, high-power vehicle that requires more cautious driving strategies, especially during sharp turns.

I extended the `AssettoCorsaGym` platform to include fuel usage data in the reward calculation. The reward function provided in `AssettoCorsaGym` is based on the car's

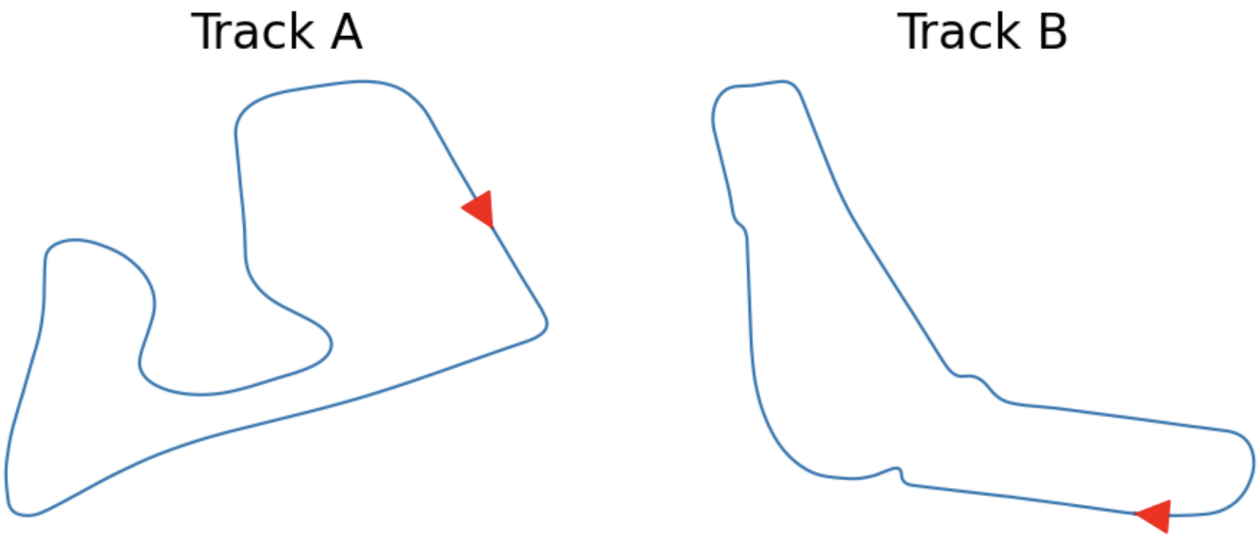

Figure 4.5. Two tracks chosen for the training and evaluation of the model from the Assetto Corsa simulator. The red triangles indicate the start position and direction.

velocity and penalizes deviation from the optimal driving line. It is computed as:

$$r = v \cdot (1 - a \cdot d)\,, \tag{1}$$

where v is the car's current speed, d is the L2 distance from the optimal path (as determined by the simulator), and a is a penalty coefficient [95]. To encourage fuel efficiency, I extended the reward function with a penalty for fuel consumption. The resulting reward function is defined as:

$$r = v \cdot (1 - a \cdot d) - b \cdot f, \tag{2}$$

where f is the change in fuel since the last timestep, and b controls the strength of the penalty. Notably, fuel consumption data was excluded from the agent's observation space to encourage the agent to learn through implicit feedback. I experiment with four values of the coefficient – corresponding to approximate reward reductions of 2%, 5%, 10%, and 20%, relative to the original reward function formulation. All training runs were performed on an RTX 3060 laptop GPU. On average, it took approximately 48 hours to complete 500 training episodes.

4.2.4 Experiments

I examined the effect of fuel penalties on RL agent performance during training. Figure 4.6 shows that low to moderate penalties (2-5%) often improved lap times during early training compared to the baseline, particularly with the GT3 car on Track A. However, higher penalties (20%) led to suboptimal policies that heavily prioritized fuel savings. On the more complex Track B, penalties above 2% consistently prevented agents from completing valid laps. Figure 4.7 shows the evolution of the fuel consumption rates per lap during training. Across all setups, penalized agents consistently reduced fuel usage over time. This effect was strong on Track A for both vehicles. On Track B, fuel savings were less noticeable and limited by the lower magnitude of the penalty (2%), as higher penalties failed to converge. Table 4.5 summarizes the best lap times and corresponding fuel consumption across all setups. On Track A, penalties led to fuel savings up to 0.4L per lap without major performance drops. On Track B, even mild penalties degraded performance and convergence. *DNF* values indicate failure to complete valid laps.

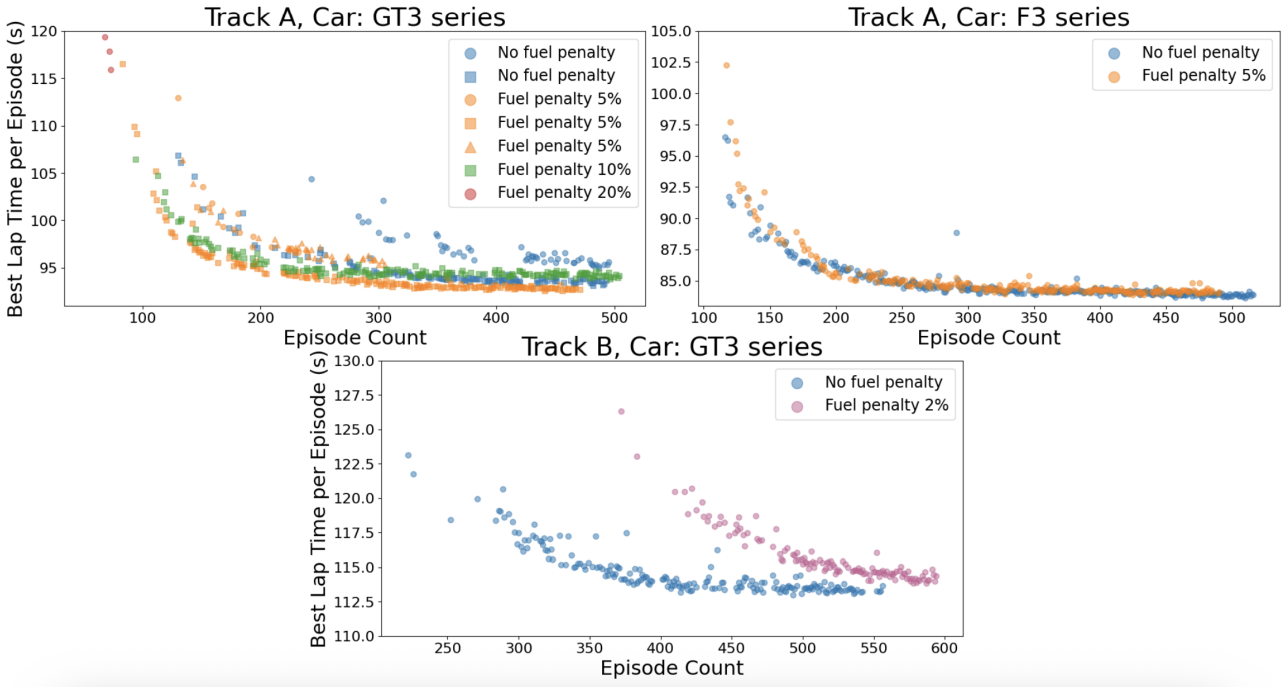

Figure 4.6. Evolution of best lap times per episode during training. Shapes depict different random seeds.

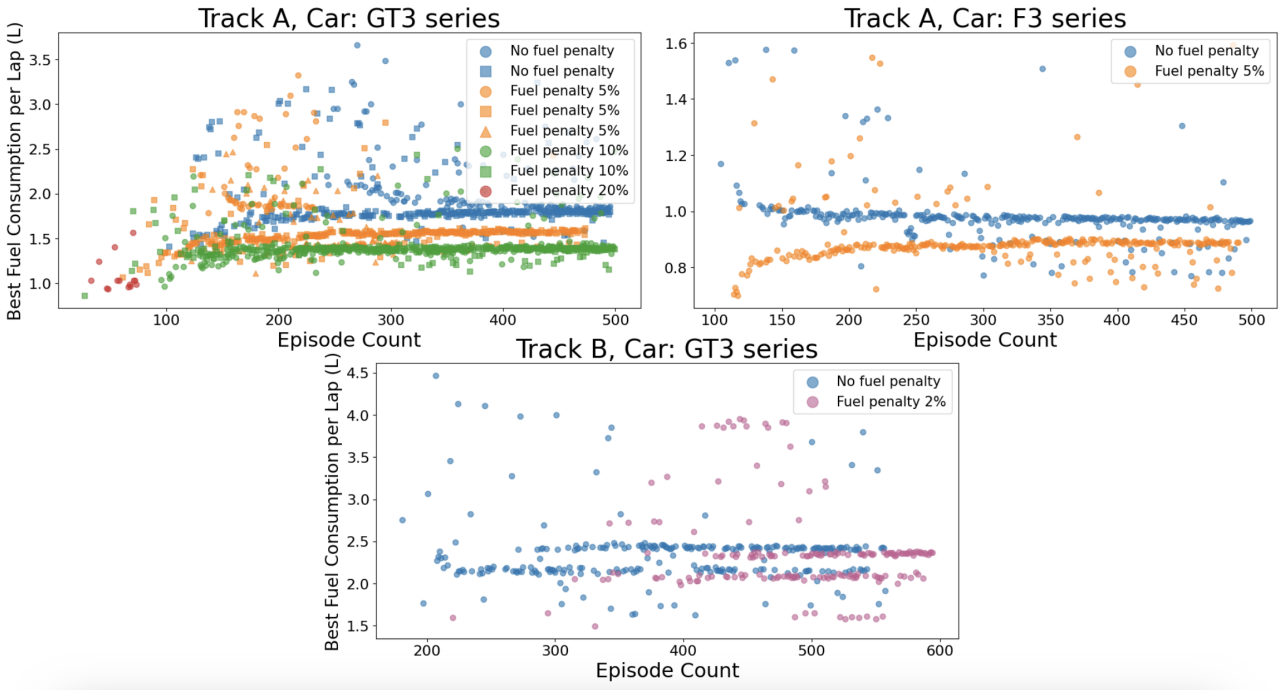

Figure 4.7. Best fuel consumption rates per lap. Shapes depict different random seeds.

I compared the trained models against the baseline on Track A and Track B. On Track A, I tested models with 0% and 5% penalties. On Track B – 0% and 2%. Models were selected to have similar lap times for fair fuel efficiency comparison. As per Table 4.6, the 5% penalty model on Track A used 0.19L less fuel per lap (10.7% savings) and was faster by 0.18 seconds (0.2%). On Track B, the 2% penalty model saved 0.064L (2.6%) but was slower by 0.84 seconds (0.7%). Training durations were similar between models.

Figure 4.8 shows spatial differences in fuel usage and lap time. Penalized models consumed less fuel, especially before major turns. On Track A, the penalty also led to faster lap times. On Track B, fuel savings came at the cost of slower lap times. Figure 4.9 shows the driving behavior comparison between two models. On Track A, penalized agents reduced acceleration, braking, steering amplitude, and engine RPM – all factors responsible for the increased fuel consumption, directly or indirectly. On Track B, adaptations were weaker due to the lower penalty level: acceleration and RPM decreased, braking remained similar, and steering angle amplitude increased.

Table 4.5. Comparison of best lap times and corresponding fuel consumption rates per lap across different setups.

Fuel Penalty	Random Seed	Best Lap Time (s) ↓	Fuel/Lap ↓
Track A, Car: GT3 series			
0%	0	94.97	1.88
	1	93.13	1.80
5%	0	96.59	1.88
	1	**92.57**	1.59
	2	95.09	1.47
10%	1	94.35	1.44
	2	93.80	1.43
20%	1	115.93	**1.01**
Track A, Car: F3 series			
0%	0	**83.62**	0.97
5%	0	83.86	**0.90**
Track B, Car: GT3 series			
0%	0	**112.99**	2.45
2%	0	113.85	**2.39**
5%	0	DNF	DNF
10%	0	DNF	DNF

4.2.5 Results

These experiments show that adding a fuel consumption penalty to the reward function leads to more fuel-efficient policies without significantly reducing lap time. On easier tracks, mild penalties (5%) often improved both fuel efficiency and speed, with faster early convergence during training. However, penalties above 10% led agents to prioritize fuel savings at the cost of increased lap time. On the more challeng-

Table 4.6. Performance comparison of agents trained under different fuel penalties.

Fuel Penalty	Best Lap (s) ↓	Mean Lap (s) ↓	FC Best Lap (L) ↓	Mean FC (L) ↓	Training Episodes
Track A, Car: GT3 series					
0%	92.964	92.975	1.842	1.841	768
5%	**92.784**	**92.796**	**1.646**	**1.644**	474
Track B, Car: GT3 series					
0%	**113.198**	113.206	2.431	2.431	560
2%	114.040	**114.203**	**2.367**	**2.366**	595

ing Track B, even a 2% penalty impaired performance, and higher penalties prevented agents from completing valid laps, confirming that penalty effectiveness depends on both track difficulty and the magnitude of the penalty. The agent learned key driving strategies to reduce fuel usage that mimic real-world energy-saving strategies used in motorsport, such as maintaining momentum and staying in higher gears. Interestingly, it avoided early braking, likely suggesting that the agent prioritized maintaining speed over extra fuel savings. Overall, the SAC algorithm effectively adapted to multi-objective rewards and learned fuel-efficient driving strategies under realistic constraints.

4.2.6 Discussion

This work highlights the ability of RL agents to adopt fuel-efficient behaviors given appropriate reward shaping. However, several limitations remain that could be explored in future work. First, experiments were limited to two vehicles and two tracks, raising concerns about generalizability. Particularly since strategies learned on the simpler Track A did not entirely transfer to Track B. Second, the agent lacked manual gear-shifting control and could only influence gears indirectly via speed. This restricted fuel-saving techniques like short-shifting. Third, the Assetto Corsa

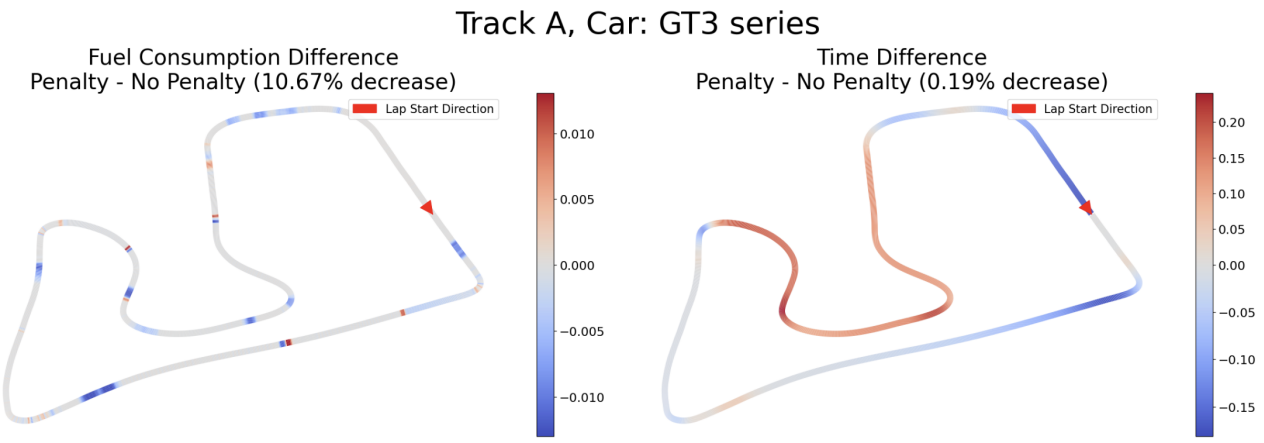

Figure 4.8. Spatial fuel consumption and lap time difference between the two models on Track A. Percentage changes are computed as the difference in mean values between the penalized and baseline laps. Blue regions indicate a decrease in the measured metric compared to the baseline. Red regions highlight an increase.

simulator runs only in real time, considerably slowing down training (48 hours per 500 episodes). This limited the ability to test alternative reward designs or repeat training with different random seeds to ensure stability. Notably, only one reward function design was explored. Future work could explore alternative formulations, like penalizing fuel-related factors directly or adding a short history of past fuel usage to the state space. Finally, fuel usage was excluded from the observation space of the agent to simulate partial observability. While it did not prevent the agent from learning efficient strategies, future work could compare outcomes with and without partial observability. It would also be valuable to test these methods with other RL algorithms beyond SAC, or with classical control frameworks like MPC, LQR or PID.

In conclusion, I incorporated a fuel usage penalty into the reward function and demonstrated that low to moderate penalties lead to considerable fuel savings with minimal lap-time performance loss. The agents adapted by modifying their driving behaviors – reducing acceleration, managing engine RPM through gear changes, and increasing steering smoothness. However, these effects did not transfer to more complex tracks, where even small penalties impaired learning, highlighting the need

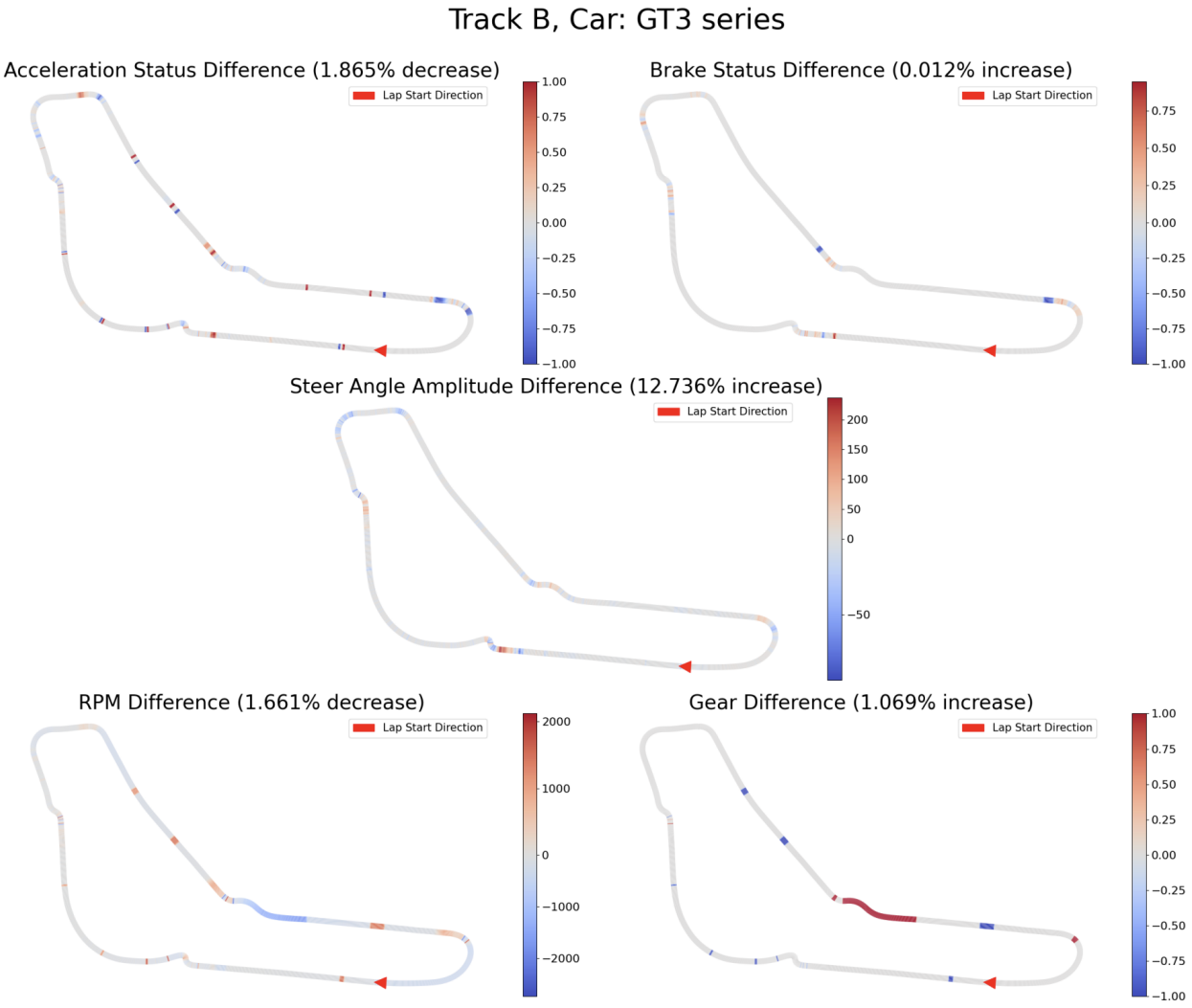

Figure 4.9. Spatial differences in driving behavior between the baseline and penalized agents on track B.

for environment-specific penalty calibration. Overall, these results suggest that RL can be effectively used to balance performance and energy efficiency.

This chapter resolves temporal and actuation efficiency by compressing model-based RL training via distillation and by explicitly optimizing energy use during control. These results show that policies can be both fast to train and energy-aware at runtime, but scaling to broader task suites introduces a new bottleneck: architectural efficiency in large, multi-skill models. Chapter 5 addresses that node by modularizing VLA policies with MoIRA, enabling specialist reuse and routing without monolithic retraining. The narrative thus moves from efficient learning and actuation to efficient architecture.

CHAPTER V

EFFICIENT VLA MIXTURE-OF-EXPERTS IN ROBOTICS

This chapter presents the methodology for architecture-efficient robotic control, addressing the **architecture efficiency** node of the taxonomy (Section 1). It is based on the contribution *MoIRA: Modular Instruction Routing Architecture for Multi-Task Robotics* (Kuzmenko & Shvai, Neurocomputing 2026 [96]).

Chapter 4 demonstrated that compact control policies can be obtained via distillation from high-capacity teachers. That approach, however, assumes a fixed model architecture compressed once and re-used across all tasks – a brittle assumption when task diversity grows or new embodiments must be added without retraining. This chapter addresses the next layer of architectural efficiency: *modularity*. Instead of compressing one monolithic model, MoIRA maintains a pool of lightweight specialist adapters and routes between them at inference time using natural language. This decoupling allows new experts to be added or swapped independently, eliminating the retraining bottleneck that limits the scalability of both monolithic generalists and static distilled policies.

5.1 Introduction

Robotic manipulation and navigation tasks have traditionally been tackled using reinforcement learning (RL) and imitation learning (IL) pipelines. These approaches have demonstrated strong performance across various settings but often depend on dense reward signals, curated expert demonstrations, or extensive task-specific tuning [97–99]. Transformer-based models such as ACT [100] have further advanced robotic policy learning by enabling fine-grained, sequence-aware control.

More recently, foundation models have emerged as an alternative to traditional RL and IL pipelines, offering general-purpose capabilities without the need for task-specific training or reward engineering. Vision-language models (VLMs) such as PaLI-Gemma [101], LLaVA [102], and Qwen-VL [103] exhibit strong image-text

grounding and instruction understanding. Although not designed for robotics, they can interpret natural language commands and scene context, making them useful for high-level planning and zero-shot inference.

Building on this trend, vision-language-action (VLA) models combine vision-language encoders with visuomotor control heads to support end-to-end robotic control. Recent examples include RT-2 [104], the RT-X family [105], OpenVLA [55], MiniVLA [106], π_0 by Physical Intelligence [107], and Gr00t-N1 by NVIDIA [108]. These models are typically pretrained on large-scale, diverse datasets (e.g. Open-X Embodiment [105]) and incorporate heterogeneous data sources, including web-scale multimodal content, subtask annotations, and demonstrations from different robot embodiments. Their goal is to generalize across embodiments, task semantics, and modalities with minimal finetuning. However, their generalist nature can lead to reduced precision, inefficient memory usage, and difficulty scaling to large task libraries [109].

While these systems show promise, they share key limitations related to their architectural rigidity. Generalist models trained on broad data tend to suffer from degraded per-task performance and high runtime memory usage. Conversely, existing expert-based MoEs typically require orchestrated training and internal routing mechanisms tied to specific monolithic model structures. This introduces a critical tradeoff between specialization, modularity, and deployment flexibility that has not been fully addressed.

To address these limitations, I adopt a modular architectural perspective wherein each expert can be independently developed, customized, and optimized. This decouples training from deployment and allows for flexible reuse across tasks and embodiments. I therefore propose **Mo**dular **I**nstruction **R**outing **A**rchitecture, **MoIRA** (Figure 5.1), a framework designed to perform zero-shot episodic model routing via natural language task and expert descriptions. Beyond routing, MoIRA is designed for practical specialist serving via adapter-based experts, supporting both disk-based swapping and multi-adapter hot-switching [110, 111] for low-latency deployment.

MoIRA sidesteps the scalability constraints of monolithic MoEs by leveraging a pool of pretrained specialists, each finetuned on a focused domain. A lightweight meta-controller dynamically selects the most relevant expert using either embedding-based similarity or prompt-driven reasoning over the textual task specification.

I evaluate MoIRA on two robotic benchmarks: GR1 [108], which covers embodiment variation (`full-body`, `arms-only`, and `arms & waist`), and LIBERO [112], which separates tasks into `Goal` and `Spatial` semantic categories. For these experiments, I instantiate MoIRA with the GR00t-N1 and π_0 VLA backbones, using LoRA adapters [54] to enable efficient specialist training. The routing module is pretrained and frozen, requiring no additional tuning to map tasks to experts.

My contributions are as follows:

1. I propose a novel modular routing architecture, MoIRA, that maps tasks to pre-trained experts based on their textual descriptions.

2. I evaluate two routing strategies – cosine similarity with MiniLM [113] and prompt-based inference with SmolLM2-1.7B [114] – and demonstrate robustness under perturbed inputs.

3. I validate MoIRA on the GR1 and LIBERO benchmarks, showing that it consistently outperforms or achieves parity with generalist models and other MoE approaches on target tasks and previously unseen tasks.

4. I provide an empirical analysis of inference-time expert serving, quantifying the trade-off between VRAM usage and switching latency across (i) fully instantiated adapters, (ii) disk-based swapping, and (iii) multi-LoRA hot-switching for scalable multi-expert deployment.

By decoupling task semantics from execution, MoIRA enables scalable, modular control. It utilizes an architecture-agnostic, external routing mechanism to coordinate a dynamic pool of specialized experts, each implemented as a lightweight LoRA adapter. The zero-shot, language-based routing allows for the independent addition,

update, or replacement of experts without costly joint training or retraining the router. It validates a flexible design paradigm for robotic agents that generalize across tasks while benefiting from specialization, providing an alternative to monolithic training pipelines.

5.1.1 Distinction from multi-agent systems.

MoIRA is a *single-agent* architecture and should not be confused with multi-agent systems (MAS). In MAS, multiple autonomous agents each maintain independent observations, policies, and goals, and may communicate or compete [115]. MoIRA instead operates as a centralized meta-controller that selects one expert per episode (a winner-takes-all routing decision) and delegates execution entirely to that specialist. At any inference step, exactly one LoRA adapter is active; the others are idle. This is architecturally equivalent to a single agent with conditionally loaded weights, not a team of cooperating agents. The modularity is a deployment property (experts can be updated independently) rather than a multi-agent interaction protocol.

5.2 Problem Formulation

I consider a multi-task setting with a set of tasks $\mathcal{T}$ described by natural language instructions l. I maintain a library of K pre-trained expert policies $\mathcal{E} = \{\pi_1, \ldots, \pi_K\}$, where each expert π_k is specialized for a subset of tasks or embodiments.

The goal is to learn a routing function $R : \mathcal{L} \to \{1, \ldots, K\}$ that maps an instruction l to the index of the optimal expert k^*, such that the expected task success is maximized:

$$k^* = \arg\max_k \mathbb{E}[J(\pi_k \mid l)]$$

where J is the performance metric (e.g., success rate). Unlike traditional MoEs where routing is learned end-to-end, I formulate this as a zero-shot retrieval problem based on semantic similarity between the task instruction l and expert meta-descriptions m_k.

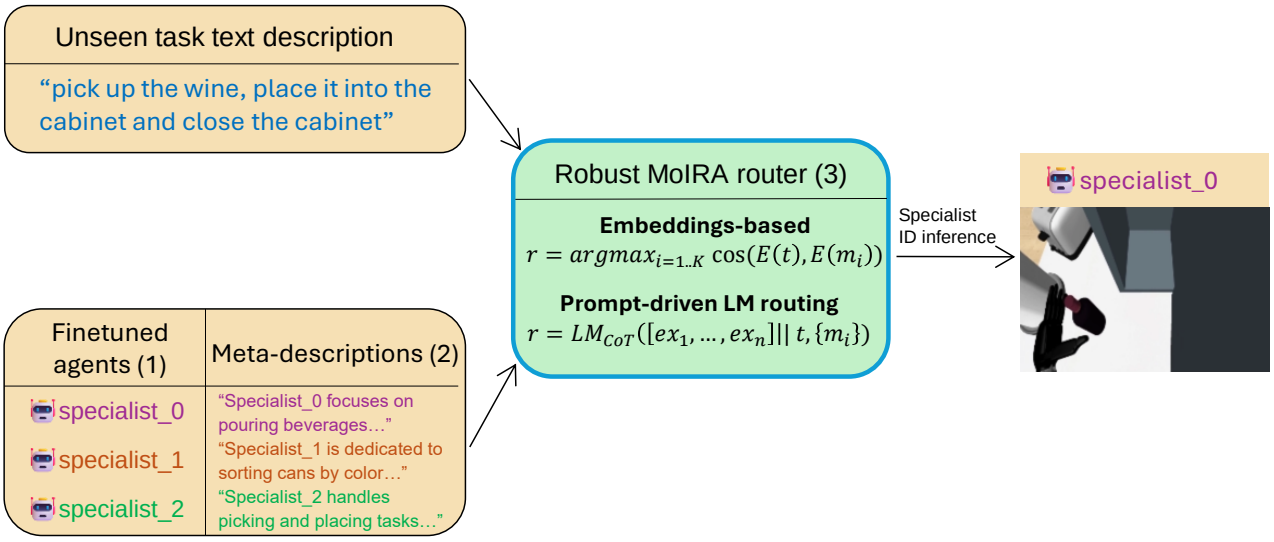

Figure 5.1. Overview of the MoIRA framework. The architecture decouples policy learning from task assignment by coordinating (1) a pool of independently **fine-tuned specialist agents** via (2) **textual meta-descriptions**. The (3) **Router core** performs zero-shot assignment of the unseen task t to the optimal specialist index r using either embedding similarity or prompt-driven reasoning. This design allows modular expert addition without retraining the routing mechanism.

5.3 Methods

MoIRA is a modular meta-controller that selects the best-suited specialist policy based on a task description. It is architecture-agnostic, meaning any expert architecture can be used as a backbone, including but not limited to transformer-based or model-based agents. In this work, I focus on foundation VLAs as expert backbones. The specialists are routed based on their textual description. Unlike joint or monolithic models, MoIRA functions as an external routing module, enabling flexible expert integration and efficient deployment.

Specialists are routed by MoIRA using embedding-based cosine similarity or prompt-driven language models (Figure 5.5), with the router requiring no extra training.

At inference time, MoIRA performs two steps: routing and expert serving (Algorithm 1). Given a task instruction t and expert descriptions $\{m_i\}_{i=1}^{K}$, it selects the

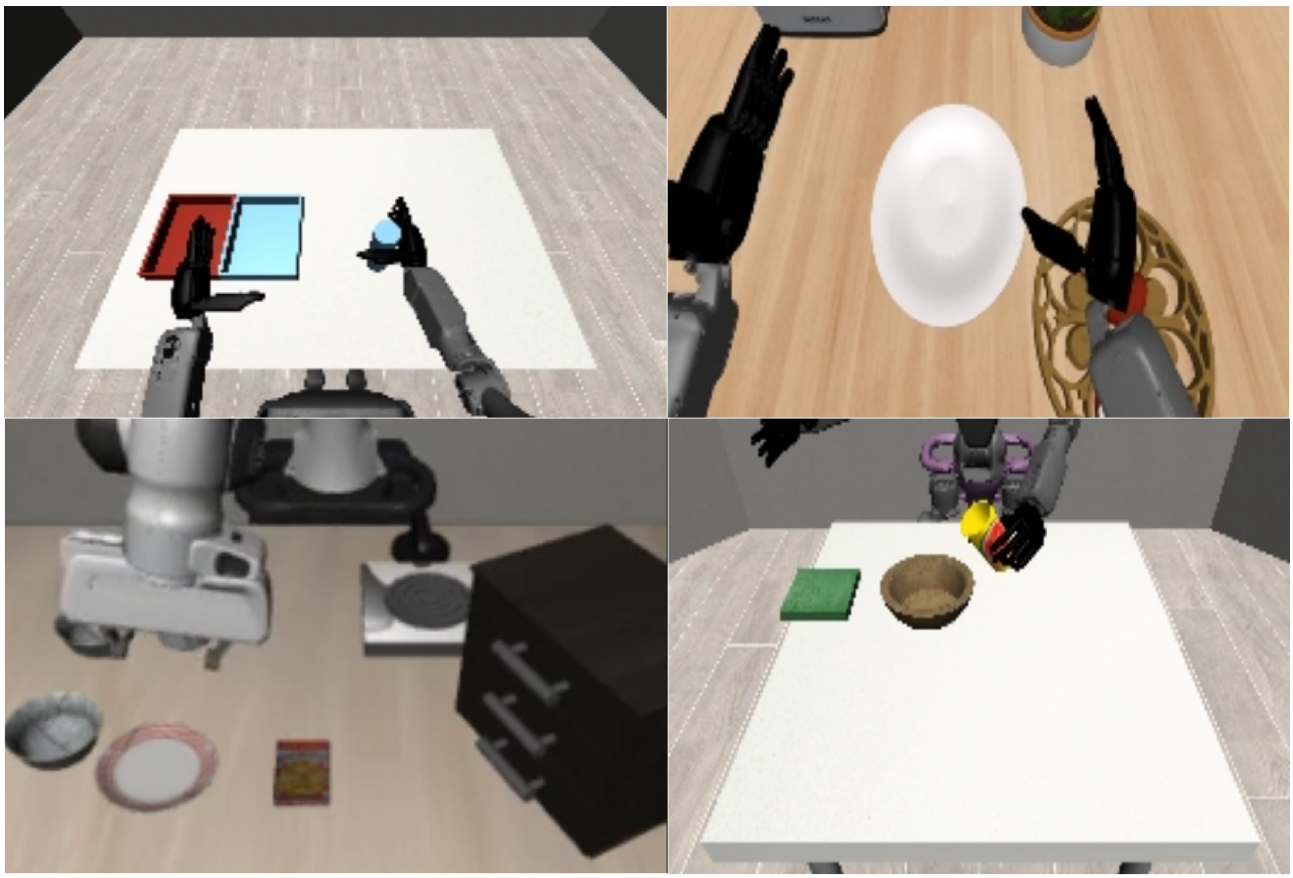

Figure 5.2. Visualization of policy rollouts. GR1 Humanoid embodiment (`arms-only`, `full upper body`) completing pouring and pick-and-place tasks and a Panda Franka arm successfully executing manipulation primitives on the LIBERO `Spatial` benchmark.

expert index r using either cosine similarity over pretrained text embeddings:

$$r \leftarrow \arg\max_{i=1\ldots K} \cos\left(E(t), E(m_i)\right),$$

or prompt-based reasoning via a frozen LM with Chain-of-Thought formatting [116]:

$$r \leftarrow LM_{\text{CoT}}\left([e_1, \ldots, e_n] \parallel t, \{m_i\}\right).$$

The selected expert $\mathcal{E}_r$ is then served under one of several serving regimes (Figure 5.3) to produce the output trajectory τ and result o.

The computational complexity of the routing step scales linearly $O(K)$ with the number of experts K. Embedding router requires a single forward pass of the encoder $E(t)$ and K dot-product operations, resulting in negligible latency. For the language

Algorithm 1 MoIRA Routing and Serving

Require: Task instruction t, Expert pool with meta-descriptions $\{m_i\}_{i=1}^{K}$

Ensure: Task outcome (trajectory, reward/success signal)

Routing:

if Embeddings-based routing **then**

    $r \leftarrow \arg\max_{i=1..K} \cos\left(E(t), E(m_i)\right)$

else if Prompt-driven LM routing **then**

    $r \leftarrow LM_{\text{CoT}}\left([e_1, \ldots, e_n] \,||\, t, \{m_i\}\right)$

end if

Expert serving:

if *All-agents-in-memory* **then**

    Use in-memory model $\mathcal{E}_r$

else

    Load LoRA adapter weights for $\mathcal{E}_r$

end if

Execute task t with expert $\mathcal{E}_r$ to produce trajectory τ and outcome o

    return (τ, o)

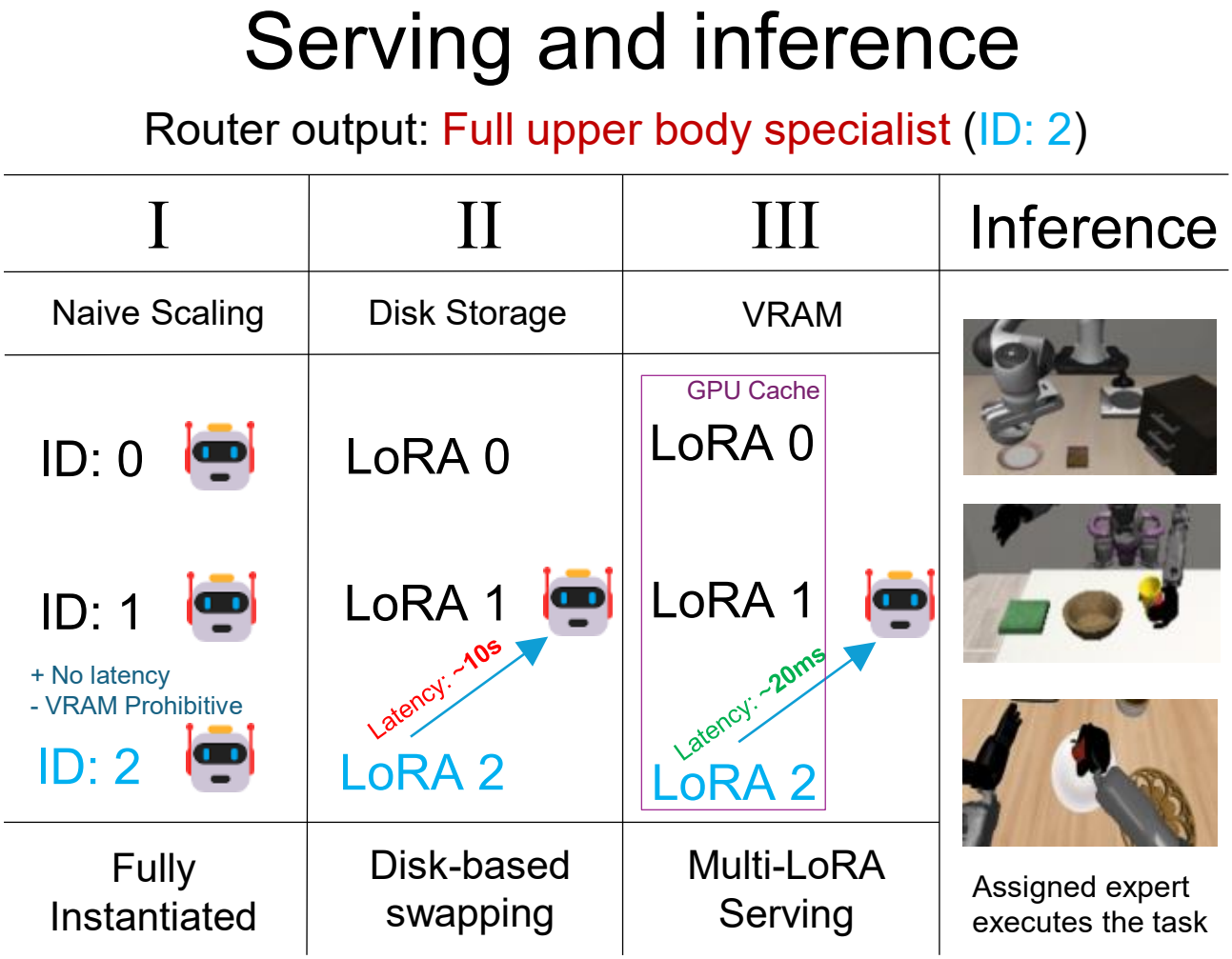

Figure 5.3. Serving and inference configurations in MoIRA. (I) *All-agents-in-memory*: Each specialist, identified by a unique LoRA adapter ID, is kept resident in GPU memory. This setup enables near-instant agent switching but requires significantly more VRAM, scaling with the number of specialists. (II) *Dynamic LoRA adapter loading*: A single shared backbone dynamically loads the required LoRA adapter based on router input (e.g., "Full upper body specialist (ID: 2)"). This configuration reduces memory overhead to one model checkpoint and a single adapter, but introduces up to 9s latency when swapping specialists.

model router, context length grows linearly with the length of meta-descriptions $\sum_{i=1}^{K} |m_i|$, with an empirical average description length of ≈ 35 tokens observed in the experiments. While more semantically robust, inference latency increases with the size of the expert pool.

The primary system bottleneck lies in the expert swap stage. I analyze this constraint through three distinct serving configurations. The simplest, in-memory switching (i), maintains all specialists in VRAM, enabling instantaneous transitions but imposing prohibitive memory requirements that scale linearly with the expert pool. Conversely, disk-based swapping (ii) optimizes for memory by storing inactive experts on disk, though this introduces significant I/O latency during retrieval. To address this trade-off, I consider multi-LoRA serving (iii) by utilizing frameworks such as S-LoRA [111] and LoRAX [110] as an effective intermediate approach. By leveraging optimized kernels, this method enables dynamic swapping of VLA adapters (e.g., 4B parameters) in up to 20ms, effectively balancing low memory footprint with near-real-time responsiveness.

5.3.1 Benchmarks and Datasets

1. **GR1:** I use the GR1 benchmark [108] containing three embodiments (`arms-only`, `arms & waist`, and `full upper body`) of GR1 humanoid. Each task contains an annotated task instruction retrieved from a `tasks.jsonl` metadata file.

2. **LIBERO:** LIBERO benchmark [112] tasks are extracted from `.bddl` files. I use a modified version of LIBERO splits from [55] and select `Spatial` and `Goal` task categories of the benchmark as the semantic axis.

Rollout examples from both benchmarks are visualized in Figure 5.2.

5.3.2 Specialist Finetuning

For GR1, a dedicated GR00t-N1-2B specialist is LoRA fine-tuned for 5K steps for each embodiment. I select three representative tasks to capture variation in embodiment types, task dynamics, and complexity – `Pouring`, `CanSort`, and `PlacematToPlate`. In addition, I train a jointly-tuned generalist model across all three tasks and reserve `WineToCabinet` (`arms & waist`),

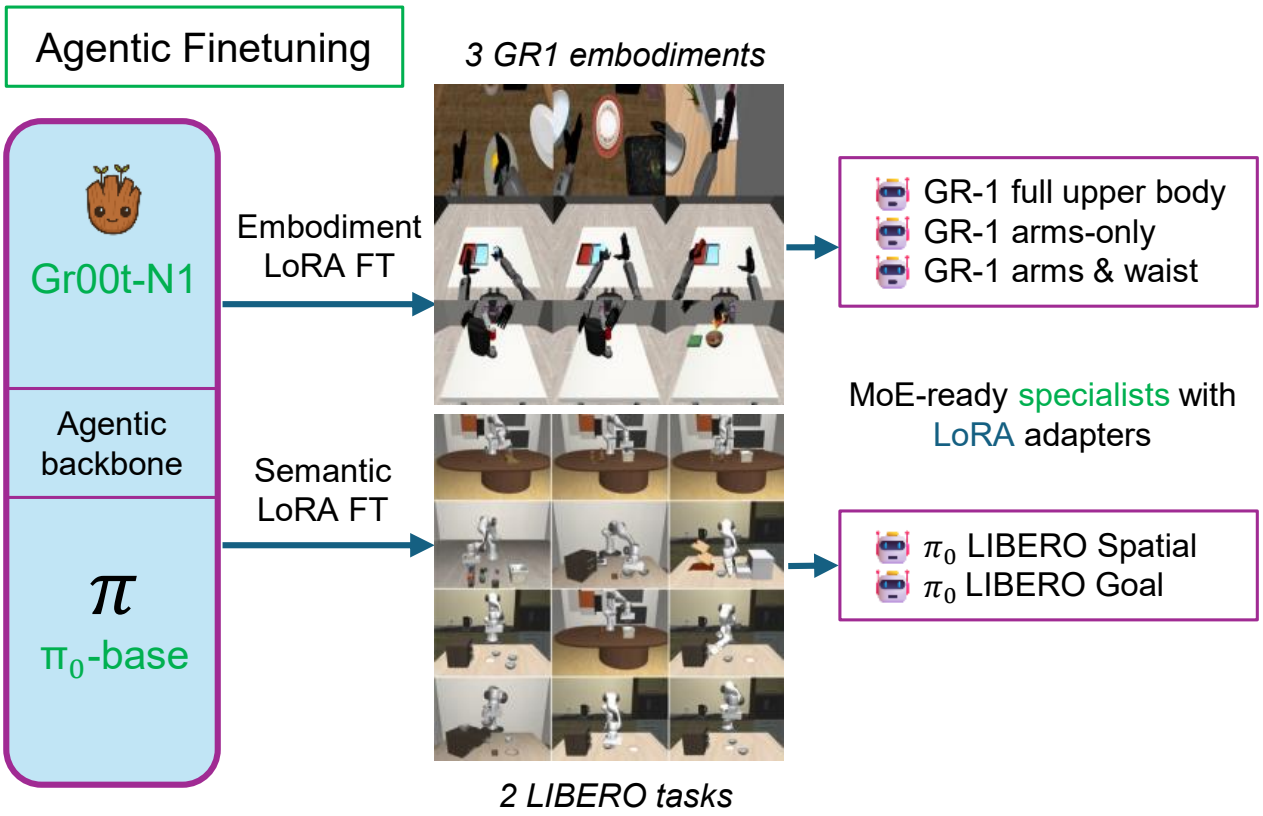

Figure 5.4. The initial stage of MoIRA's agentic adapter fine-tuning. I finetune VLA backbones (Gr00t-N1, π_0-base) and derive modular specialists via LoRA: embodiment adapters for GR1 manipulation tasks and semantic adapters for LIBERO `Goal` and `Spatial` tasks. Although demonstrated on VLAs, MoIRA's unified interface can accommodate any agentic backbone, e.g. transformer-based or model-based approaches.

`CuttingboardToTieredBasket` (`arms & waist`), and `Coffee` (`full upper body`) as previously unseen (held-out) tasks for evaluation.

For LIBERO, I use the π_0-base-3.3B foundation VLA as the backbone. I create LoRA-adapters for three experts: a spatial task expert, a goal task expert, and a generalist, jointly trained on both tasks – each model tuned for 30K steps using default hyperparameters. The fine-tuning stage is described in Figure 5.4.

5.3.3 Routing Strategies

I implement MoIRA task routing using two strategies:

1. **Embedding Similarity:** I embed both task descriptions and expert meta-

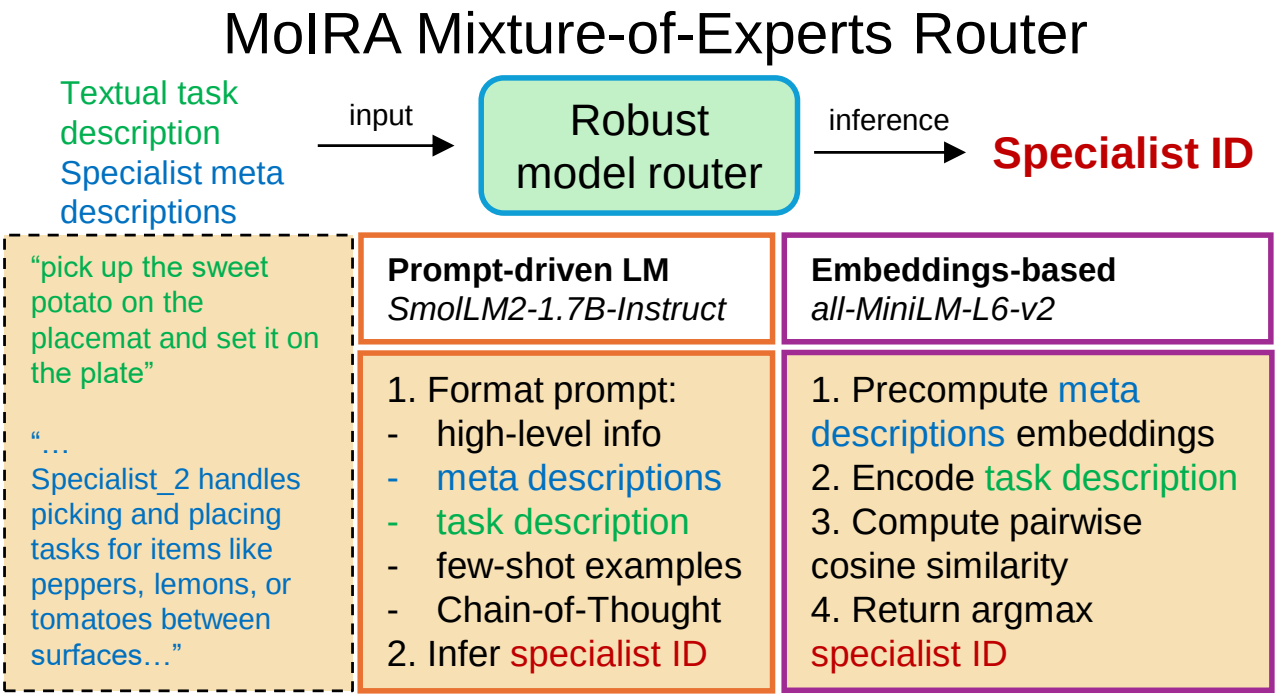

Figure 5.5. MoIRA MoE routing module. Given a textual task description and specialist meta descriptions, the router assigns a specialist ID via one of two strategies. A prompt-driven language model (SmolLM2) formats all inputs into a single inference prompt with few-shot examples to infer the matching expert (orange). A lightweight embedding-based method (MiniLM) computes cosine similarity between the task and cached meta descriptions to return the closest expert (purple). Both variants support modular, language-based task routing without requiring direct observation input.

descriptions using *all-MiniLM-L6-v2* (23M parameters) [113]. The highest cosine similarity among all pairs determines the routed expert ID. This approach treats routing as a geometric retrieval problem, assuming that task instructions and expert capabilities share lexical overlap in a shared latent space. It is computationally negligible but relies on surface-level semantic alignment.

2. **Prompt-Driven Routing:** *SmolLM2-1.7B-Instruct* [114] receives structured prompts[1] containing the task and candidate expert descriptions to select the

[1] SmolLM2 receives a prompt containing meta-descriptions of each expert and a short chain-of-thought instruction set. Example tasks with matched specialists are provided to guide classification. The model is then asked to select the most appropriate expert (e.g., Output: 0, 1, or 2) based on task semantics.

expert via causal language modeling. Unlike embeddings, this leverages the probabilistic reasoning of a LM to approximate $\texttt{P}(\texttt{Expert}_\texttt{i}|\texttt{Task}, \texttt{Context})$. This allows the router to handle abstract instructions or disparate phrasing where geometric distance may fail, trading inference latency for higher zero-shot routing fidelity.

Each expert is annotated with two distinct types of meta descriptions to support routing:

1. **Simple:** short, literal phrases that describe low-level actions or object interactions using task-specific vocabulary (e.g., `picks and places the black bowl`).

2. **Abstract:** higher-level, generalized descriptions that emphasize task intent or spatial reasoning without referencing specific objects (e.g., `transports items between spatial zones`).

This dual-format setup enables compatibility with both routing strategies: the prompt-driven LM benefits from detailed linguistic cues in abstract descriptions, while the embedding-based method relies on surface-level lexical similarity captured in simple ones. It also allows evaluation of the router's generalization behavior across levels of semantic abstraction.

5.3.4 Classification and Execution

I assign each task a corresponding ground-truth category: **embodiment type** (for GR1) or **semantic type** (for LIBERO). I formulate routing as a multi-class classification problem, where the router predicts the most appropriate specialist given a natural-language task description. Routing accuracy is measured using the macro-averaged F1 score between predicted and ground-truth labels.

At inference time, MoIRA operates in two sequential stages: *routing* and *expert serving*. Given an input task instruction, the router selects an expert index using

either embedding-based similarity or prompt-driven language model inference. The selected expert is then served under one of several adapter serving configurations, each exposing a different trade-off between memory footprint and switching overhead.

- **Fully instantiated serving.** All specialist adapters are kept resident in GPU memory, enabling immediate expert selection. While this configuration minimizes switching overhead, its memory usage scales linearly with the number of specialists, limiting scalability.

- **Disk-based swapping.** Only a single active adapter is loaded at a time, with inactive specialists stored on CPU or disk and loaded on demand. This approach minimizes GPU memory usage and supports large expert libraries, but incurs significant I/O overhead during expert transitions, making it most suitable for episodic or amortized execution.

- **Multi-LoRA serving.** An intermediate configuration that maintains multiple lightweight adapters concurrently on a shared backbone, enabling fast expert switching without full model replication. Recent multi-adapter serving frameworks (e.g., S-LoRA [111], LoRAX [110]) exemplify this operating point, balancing memory efficiency with low-latency expert activation.

These serving strategies allow MoIRA deployments to be adapted to hardware constraints and task dynamics. Disk-based swapping enables scalable, storage-bound expert libraries, while multi-LoRA serving expands the operational envelope toward more reactive, low-latency routing. Crucially, MoIRA preserves explicit expert boundaries and supports heterogeneous, independently trained specialists, in contrast to parameter merging or internally gated MoE architectures. MoIRA serving and inference configurations are summarized in Figure 5.3.

5.3.5 Evaluation Protocol

For **GR1**, I report mean squared error (MSE) of the active joints against the expert observations over 50 evaluation trajectories per task. The main joints are **left hand**

and **right hand** for `full upper body` and `arms-only`, and **right hand** only for `arms & waist` embodiment, respectively.

For **LIBERO**, I report Success Rate (SR) over 100 rollouts for subtask category (`Spatial` and `Goal`). I conduct all experiments on a single NVIDIA RTX A6000 GPU (48GB VRAM). For all training runs, I keep the default hyperparameters and report evaluation results on 5 seeds.

To ensure robust comparisons, I assess statistical significance using Welch's t-test for continuous metrics (GR1 MSE) to account for unequal variances between specialists and generalists. For binary success rates (LIBERO), I employ Fisher's Exact Test. I adopt a standard significance threshold of $p < 0.05$ for all claims.

5.4 Results

For **GR1**, I analyze specialist finetuning performance, cross-task generalization, instruction robustness, generalization to held-out tasks, and the impact of routing accuracy. For **LIBERO**, I compare specialist vs. generalist performance, contrast MoIRA with prior MoE systems, assess inference-time efficiency, and outline future directions.

5.4.1 GR1 Results

Table 5.1. GR1 Specialists vs. Baseline comparison (MSE $\pm$ std over 50 trajectories).

Model	**Pouring**	**CanSort**	**PlacematToPlate**
Baseline	1.02 ± 0.04	0.70 ± 0.10	1.67 ± 0.52
Specialist*	$\mathbf{0.008 \pm 0.004}$	$\mathbf{0.010 \pm 0.010}$	$\mathbf{0.310 \pm 0.140}$

*Each task is evaluated using a target-embodiment specialist (3 total).

5.4.1.1 Specialist Performance vs. Pretrained Baseline I fine-tune each embodiment-specific specialist for 5K steps and compare it with the pretrained GR00t-N1 baseline in Table 5.1. I observe significant reductions in MSE: `Pouring`

(1.02 $\rightarrow$ 0.008, 123$\times$ reduction), `CanSort` (0.70 $\rightarrow$ 0.010, 70$\times$ reduction), and `PlacematToPlate` (1.67 $\rightarrow$ 0.31, 5$\times$ reduction). I find the reduction in error to be statistically significant across all tasks ($p < 0.001$ for all 3 tasks), confirming that the reported performance gains are robust to rollout variance and not artifactual.

Table 5.2. Various GR1 cross-task MSE $\pm$ std evaluation scenarios using two routing types and simple-abstract meta-descriptions.

Model	**Pouring**	**CanSort**	**PlacematToPlate**
Baseline	1.023 ± 0.036	0.699 ± 0.091	1.673 ± 0.520
Specialist*	$\mathbf{0.008 \pm 0.004}$	$\mathbf{0.010 \pm 0.010}$	$\mathbf{0.311 \pm 0.139}$
Jointly-tuned	0.012 ± 0.009	0.065 ± 0.010	1.187 ± 0.074
MiniLM (simple)	0.009 ± 0.005	0.006 ± 0.005	0.336 ± 0.455
MiniLM (abstract)	0.008 ± 0.002	0.008 ± 0.006	0.798 ± 0.614
SmolLM2 (simple)	0.008 ± 0.006	0.008 ± 0.006	$\mathbf{0.241 \pm 0.143}$
SmolLM2 (abstract)	$\mathbf{0.009 \pm 0.004}$	$\mathbf{0.009 \pm 0.007}$	0.287 ± 0.097

*Each task is evaluated using a target-embodiment specialist (3 total).

5.4.1.2 Cross-Task Generalization Table 5.2 presents all GR1 task evaluations across specialists and the jointly-tuned generalist. The specialists demonstrated statistically significant reductions in MSE across all evaluated tasks. Specifically, the improvement was significant for `Pouring` ($p < 0.01$) and highly significant for `CanSort` and `PlacematToPlate` ($p < 0.001$). Modular specialists achieve superior precision that is statistically distinguishable from the generalist baseline.

5.4.1.3 Robustness to Instruction Perturbations To evaluate routing robustness, I semantically perturb[2] task instructions by rephrasing them with synonymous

[2]Perturbed instructions are semantically equivalent rephrasings generated manually for each task, varying lexical choices and syntax (e.g., `pick up the pear on the placemat and set it on the plate` vs. `transfer the pear from the mat to the plate`).

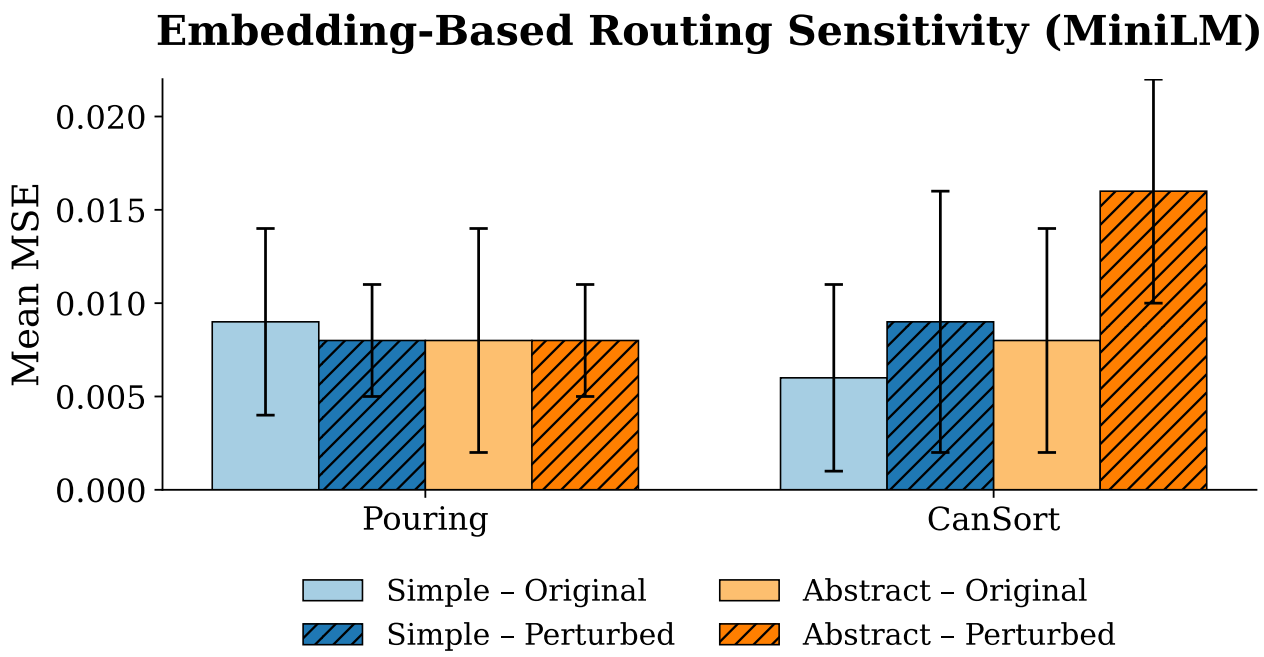

Figure 5.6. **MiniLM Routing Sensitivity.** MSE results on GR1 tasks using embedding-based routing. The plot compares performance using simple versus abstract meta-descriptions. While effective on original instructions, the router's performance shows possible degradation on perturbed instructions, particularly when relying on abstract descriptions. This indicates sensitivity to phrasing variations and potential limitations in semantic robustness.

verbs and alternate phrasing. For example, `take the bell pepper from the placemat and move it to the plate` may be rewritten as `grab the pepper off the placemat and put it onto the plate`.

As shown in Table 5.3 and Figure 5.7, SmolLM2 maintains stable performance across tasks regardless of prompt type. In contrast, MiniLM (abstract) shows sharp degradation (e.g., MSE = 0.16 on `CanSort`, Figure 5.6), indicating higher sensitivity to linguistic variation.

5.4.1.4 Generalization to Held-Out Tasks

I evaluate MoIRA on three held-out GR1 tasks. MoIRA significantly outperforms a jointly-tuned generalist model on two out of three tasks, achieving lower MSE on WineToCabinet (1.55 against 1.87) and CuttingboardToTieredBasket (0.41 against 1.29), as shown in Table 5.5. On

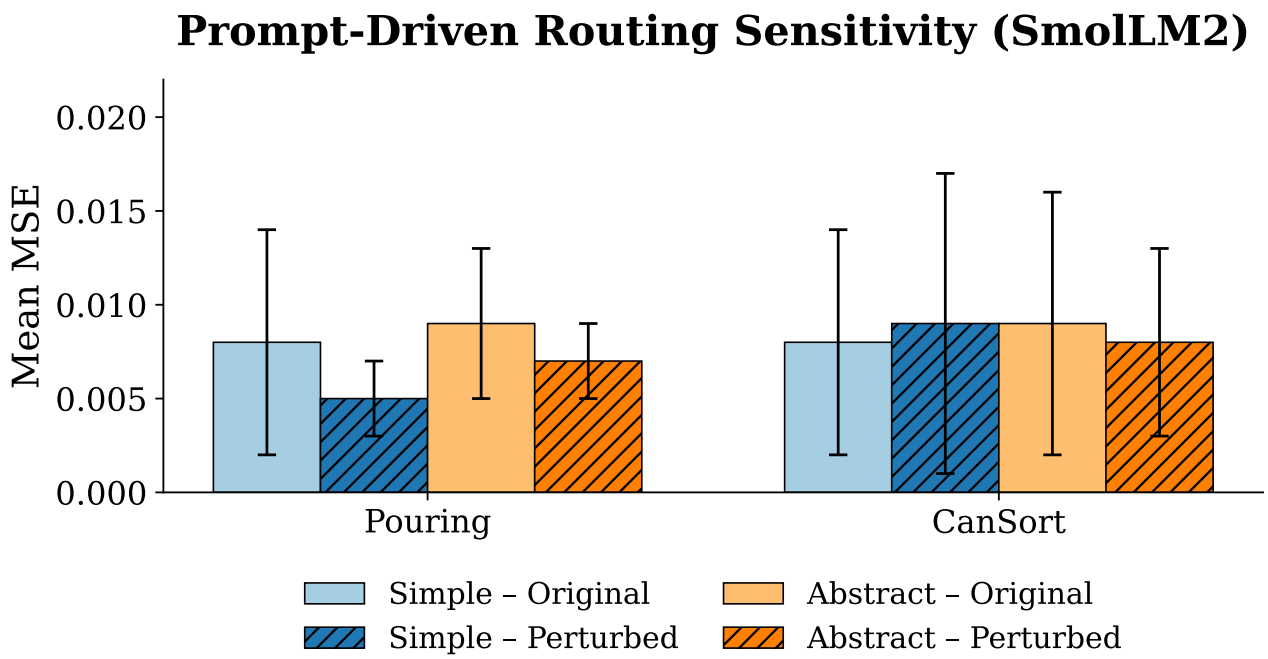

Figure 5.7. **SmolLM2 Routing Sensitivity.** MSE results using the prompt-driven language model router. In contrast to the embedding baseline, SmolLM2 maintains low error rates across both original and perturbed instructions for both tasks. This suggests stronger zero-shot reasoning capabilities and stability against linguistic perturbations.

Table 5.3. Router MSE on Original vs. Perturbed Task Descriptions (10 Episodes per Task).

Routing	**Pouring**		**CanSort**		**Placemat**	
	Orig.	Pert.	Orig.	Pert.	Orig.	Pert.
MiniLM simple	0.009	**0.008**	**0.006**	0.009	**0.336**	0.423
MiniLM abstract	0.008	0.008	**0.008**	0.160	0.798	**0.412**
SmolLM2 simple	0.008	**0.005**	**0.008**	0.009	**0.241**	0.293
SmolLM2 abstract	0.008	**0.007**	0.009	**0.008**	**0.287**	0.317

held-out tasks, MoIRA significantly outperformed the generalist on `WineToCabinet` and CuttingboardToTieredBasket ($p < 0.001$), demonstrating robust zero-shot routing to unseen scenarios. For the `Coffee` task, the generalist maintained a slight statistical edge ($p < 0.01$), though the absolute difference in MSE was marginal (0.03).

Table 5.4. Router F1 Score and Held-out GR1 Tasks Evaluation (MSE ± std over 10 episodes per task).

Model	**WineToCabinet**	**Cuttingboard***	**Coffee**	**F1 (avg)**
MiniLM[a]	1.662 ± 0.046	0.294 ± 0.209	0.196 ± 0.018	1.00
MiniLM[b]	1.669 ± 0.047	0.399 ± 0.287	0.201 ± 0.023	1.00
SmolLM2[a]	1.664 ± 0.047	0.482 ± 0.413	0.197 ± 0.016	0.97
SmolLM2[b]	1.669 ± 0.046	0.735 ± 0.677	0.192 ± 0.019	0.87

*CuttingboardToTieredBasket.

[a] simple meta descriptions; [b] abstract meta descriptions

Table 5.5. MoIRA vs. Jointly-Tuned Generalist Performance on Held-out GR1 Tasks (MSE ± std over 10 episodes per task).

Model	**Wine**	**Cuttingboard***	**Coffee**
MoIRA	**1.55 ± 0.04**	**0.41 ± 0.26**	0.19 ± 0.02
Joint Generalist	1.87 ± 0.05	1.29 ± 0.06	**0.16 ± 0.01**

*CuttingboardToTieredBasket.

5.4.2 LIBERO Results

Figure 5.8 and Table 5.6 report success rates on LIBERO `Spatial` and `Goal` subsets. First, I note that the zero-shot pre-trained π_0 baseline yields 0% SR, as it lacks the requisite observation normalization statistics to navigate through episodes successfully. To contextualize the performance of MoIRA's specialists, I compare three distinct fine-tuning regimes, trained for 30K steps each:

1. **Global Generalist (Figure 5.8, gray bar):** a monolithic baseline fine-tuned on the complete LIBERO dataset (gray bars). This represents the performance upper bound of a strong generalist.

2. **Joint LoRA Generalist (Figure 5.8, green bar):** a single adapter trained jointly on both `Spatial` and `Goal` tasks (green bars), representing a parameter-efficient

multi-task baseline.

3. **Decoupled Specialists (Figure 5.8, blue and yellow bars):** independent LoRA adapters fine-tuned exclusively on their respective subsets (blue and yellow bars), routed via MoIRA.

Table 5.6. LIBERO Specialists and Generalist Success Rates over 100 Rollouts

Model	**LIBERO Spatial, %**	**LIBERO Goal, %**
Spatial Specialist	94	—
Goal Specialist	—	**93**
Joint Generalist	**95**	90

5.4.2.1 Specialist vs Generalist Success Rates While the jointly-tuned generalist achieves marginally higher raw success rates (95% vs 94% `Spatial`; 90% vs 93% `Goal`), the difference between the models was not statistically significant ($p > 0.05$). Specifically, for `Spatial` tasks ($p = 1.0$) and `Goal` tasks ($p \approx 0.61$), the 95% confidence intervals significantly overlap. These results statistically confirm performance parity – MoIRA matches the generalist's SR while offering the architectural benefits of modularity and reduced inference memory.

Table 5.7. Comparison of MoE Flexibility and SR on LIBERO Benchmark. MoIRA outperforms Tra-MoE and achieves parity with MoDE.

Method	**Spatial**	**Goal**	**Routing**
Tra-MoE + mask	73	78	Internal
MoDE	90	97	Internal
MoIRA (ours)	**94**	**93**	External

5.4.2.2 Comparison to Prior MoE Systems Table 5.7 compares MoIRA to Tra-MoE and MoDE. MoIRA's external routing achieves competitive SR while retaining modularity and expert flexibility. In contrast to internal routing (e.g., token-level gating in MoDE), MoIRA's design enables episodic routing with adaptable backbones.

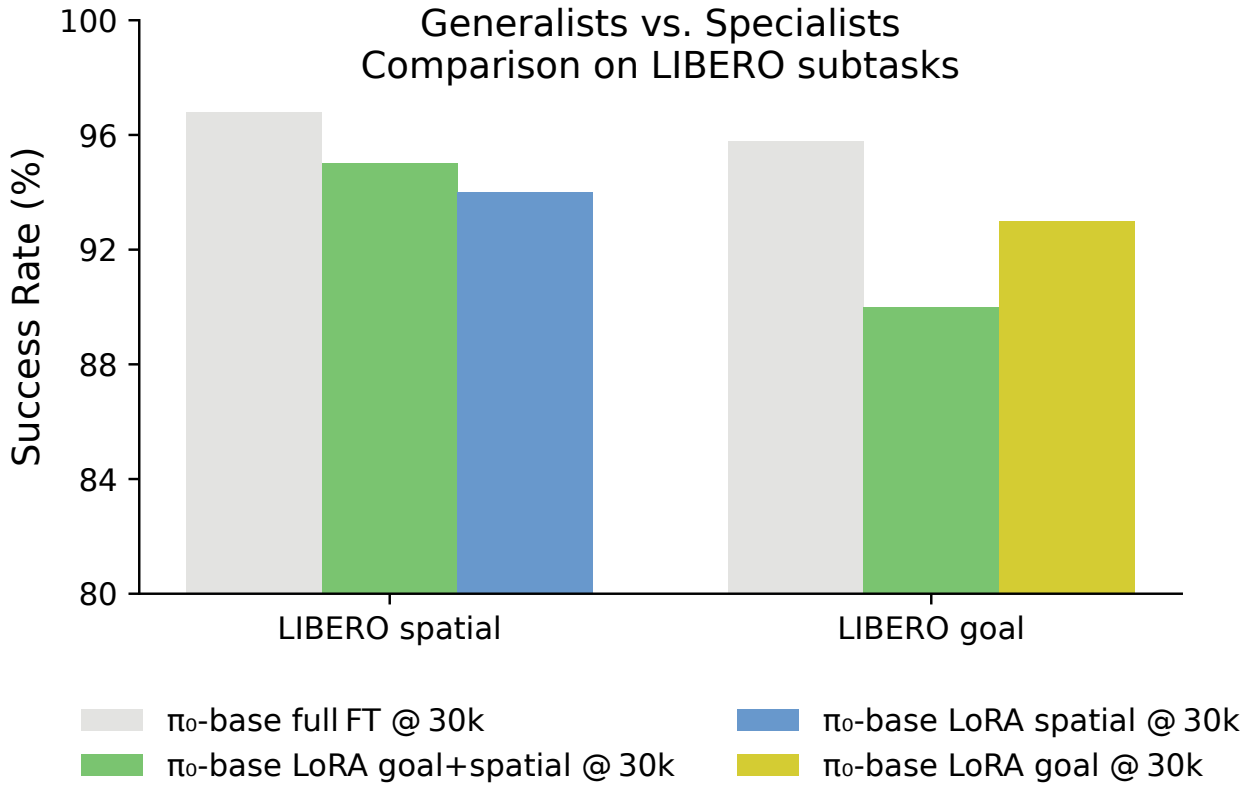

Figure 5.8. Comparative Success Rates (SR) on LIBERO subtasks. I contrast a **Global Generalist** (gray; full model fine-tuning on the entire dataset) against a **Joint LoRA Generalist** (green; single adapter trained on combined Spatial and Goal data) and **MoIRA-routed Specialists** (blue/yellow; decoupled adapters for each subtask). The modular specialists achieve competitive performance with the monolithic baselines despite being trained on strictly partitioned data.

Reuss et al. [50] achieve 90% SR on LIBERO `Spatial` and 97% LIBERO `Goal` using a sparse expert diffusion transformer. MoIRA scores 93.5% weighted-average SR across `Spatial` and `Goal` subtasks and achieves statistical parity with MoDE (both average to 93.5%: $(94 + 93)/2$ for MoIRA and $(90 + 97)/2$ for MoDE), matching state-of-the-art diffusion policies using a modular VLA architecture. Unlike MoDE's

internal token-level routing, MoIRA supports episodic routing via fast adapters, making it suitable for deployment-limited settings.

I also compare this approach with Tra-MoE [117], which achieves 73% SR on `Spatial` and 78% SR on `Goal` using a transformer-based policy with internal MoE blocks. MoIRA outperforms Tra-MoE on both suites while offering broader routing flexibility and simpler adapter management.

5.4.2.3 Routing Fidelity and System-Level Gains Consistent with GR1 findings, routing accuracy in LIBERO strongly correlates with downstream task success. SmolLM2 achieves perfect classification performance (F1=1.0) in a zero-shot manner, enabling specialist experts to reliably outperform the jointly-tuned generalist. These results underscore that routing fidelity is the critical point for realizing the full potential of modular expert systems.

5.5 Discussion

This work presents MoIRA, a modular MoE routing framework that dynamically assigns pretrained VLA specialists based on robot embodiment or task semantics. To my knowledge, it is the first framework to demonstrate benchmark-validated dynamic specialization of VLA models along these axes. MoIRA uses LoRA adapters and natural language routing to enable strong end-to-end control without requiring MoE retraining. The results demonstrate that this modular architecture achieves performance parity with monolithic generalists and other robotic MoE systems while significantly enhancing scalability and deployment efficiency. By decoupling expert specialization from orchestration, MoIRA validates that independent, lightweight adapters offer a flexible and resource-efficient alternative to rigid joint-training pipelines.

5.5.1 Sim-to-Real and Sensory Robustness.

Current evaluations are performed in simulation. While this validated algorithmic stability across highly distinct morphologies (GR1) and semantic domains (LIBERO), real-world deployment introduces actuation noise and sensory drift. The router's

reliance on purely textual descriptions currently assumes well-formed instructions and ignores environmental context (e.g., distinguishing between two identical cups). To address this, future work will explore multimodal routing by integrating lightweight vision encoders (e.g., CLIP) or extracting features from the frozen VLA backbone itself. This would allow the router to disambiguate tasks based on visual state without sacrificing the architecture-agnostic nature of the external controller.

5.5.2 Latency and Operational Envelope.

A key trade-off in this design is the I/O latency introduced by standard dynamic adapter loading. This characteristic typically restricts MoIRA to episodic or batch-processing workflows, such as a service robot completing a 20-minute *"kitchen cleaning"* sequence before switching to *"laundry folding"*, rather than high-frequency, reactive switching. To address this bottleneck, I integrated specialized multi-LoRA serving frameworks, which facilitate concurrent serving of fine-tuned adapters with millisecond-level switching overhead. While this approach effectively masks latency and expands the operational envelope for a subset of compatible backbones, further investigation into universal architecture support and speculative routing strategies remains a priority for fully reactive deployment.

5.5.3 Scalability and Automation.

Finally, while SmolLM2 demonstrates robust zero-shot routing, the reliance on manually written meta-descriptions limits rapid scaling to thousands of experts. Automating this pipeline is a high-priority research direction. I envision using large language models to auto-generate expert summaries from raw task specifications or demonstration logs, effectively closing the loop to create a fully autonomous, self-organizing mixture of robotic experts.

In summary, MoIRA offers a modular, scalable approach to deploying robotic agents that combine embodiment-specific specialization with semantic task understanding. It demonstrates that pretrained experts can be orchestrated through zero-shot lightweight text-based routing, enabling generalization without relying on joint train-

ing or monolithic models.

This chapter solves the architecture-efficiency node by showing that modular routing among specialized VLA experts can outperform monolithic generalists while keeping adapters lightweight and swappable. With architectural scalability established, the next constraint is interaction efficiency: how robots behave when safety and social trust are simultaneously at stake. Chapter 6 therefore turns to HRI and introduces EED Gym to evaluate empathic disobedience under shared safety and trust metrics. This shift connects model modularity to human-facing robustness.

CHAPTER VI

EFFICIENCY IN HUMAN-ROBOT INTERACTION

This chapter presents the methodology for socially efficient interaction, addressing the **social** and **safety efficiency** nodes of the taxonomy (Section 1). It is based on the contribution *When Robots Say No: The Empathic Ethical Disobedience Benchmark* (Kuzmenko & Shvai [118], HRI 2026).

Chapters 2-5 addressed efficiency from a computational perspective: reducing annotation cost, memory footprint, training time, actuation energy, and model redundancy. Those contributions assume the robot operates autonomously, optimizing performance within fixed hardware budgets. This chapter introduces the final and qualitatively different efficiency bottleneck: the *interaction layer*. When a robot operates alongside humans, task-optimal behavior is insufficient if it erodes trust, causes social harm, or blindly executes unsafe commands. EED Gym formalizes this as a socially constrained optimization problem – a paradigm where efficiency is measured not just in compute or energy, but in the joint optimization of safety compliance, trust maintenance, and communicative cooperation.

6.1 Introduction

As robots move from controlled environments into homes, workplaces, and public spaces, they are increasingly asked to follow instructions from non-expert users. In many cases, literal obedience can be unsafe or socially harmful: a household robot may be asked to carry boiling oil near children, or a warehouse robot – to lift loads beyond its capacity. Blind obedience risks safety, while outright refusal risks undermining trust and cooperation. Balancing these concerns remains a central challenge for AI safety and human-robot interaction (HRI).

AI safety research has focused on aligning agents with human values and preventing harmful behavior. Amodei et al. [119] outlined core safety problems, and Gabriel et al. [120] framed them in ethical terms. Constrained policy optimization [71], action

masking [121], and risk-sensitive objectives help suppress unsafe actions. Surveys by Garcia and Fernandez [122] and Gu et al. [123] consolidate these efforts, positioning safe reinforcement learning (RL) as a primary safeguard against harmful compliance. HRI work has highlighted the social dimension of safety: trust calibration was characterized by Lee and See [4], extended in Hancock's meta-analysis [5], and problematized by Robinette et al. [124] through cases of over-reliance. Refusal has been studied as a social act shaping acceptance and cooperation [74, 125], while human-in-the-loop safety was advanced by Saunders et al. [126].

Affective cues were introduced in Picard's affective computing [127] and extended to social robots by Paiva et al. [128]. Safe interaction is both technical and social, yet existing benchmarks split these dimensions: AI Safety Gridworlds [129] and Safety Gym [76] focus narrowly on physical hazards, while HRI studies of trust and refusal remain small-scale and hard to reproduce. To my knowledge, no systematic benchmark unifies safety and social acceptability, highlighting the gap in evaluating refusal as both a safety mechanism and a socially grounded behavior.

I introduce the **Empathic Ethical Disobedience (EED) Gym**, a simulation environment for studying when and how robots should refuse unsafe commands. EED Gym integrates risk estimation, affective cues, trust dynamics, and refusal styles into a single testbed. I evaluate **safe RL methods** (Lagrangian PPO [71] and Masked PPO [121]) against **non-safe baselines**, namely vanilla PPO [130] and PPO-LSTM, guided by three questions:

1. How do safe RL methods compare to non-safe baselines when refusal must be socially grounded?

2. Which social signals and refusal modes matter most for robustness and trust?

3. What trade-offs arise between safety (avoiding unsafe compliance), trust, and positive task outcomes under stressors?

I answer these through systematic evaluations and ablations removing affect ob-

servations, communicative refusals, curriculum training, and trust penalties across in-distribution and stress-test scenarios. Algorithmic safeguards reduce unsafe compliance but differ in side effects: Lagrangian training is conservative at the expense of task efficiency, while action masking prevents unsafe compliance with less erosion of trust. Social mechanisms (affect, clarification, alternatives) remain critical for robustness, and a staged training curriculum stabilizes refusal frequency, with trust penalties primarily acting as regularizers. In human ratings, constructive refusals were judged safest and most trustworthy, while empathic refusals maximized perceived empathy; these distinctions are reflected in the affect and trust models.

My contributions are threefold:

- I introduce **EED Gym**, a benchmark unifying algorithmic safety with the social dynamics of refusal.
- I provide **baselines** and ablation results as validation experiments, showcasing analysis of the expected safety-trust trade-offs.
- I identify key design levers (affect, communicative refusals, training curriculum) and release code and configs for **reproducible evaluation**.

Beyond benchmarking RL algorithms, EED Gym offers HRI researchers a controllable testbed for studying how refusal strategies shape trust, blame, and cooperation at scale.

6.1.1 Connection to explainable AI.

The robot's decision to refuse, explain, or propose alternatives is an act of transparency toward the human partner. This positions EED Gym squarely within the explainable AI (XAI) agenda [131]: the agent must not only behave safely but also communicate *why* it is refusing in terms the human can interpret and accept. Explanatory refusals (REFUSE-EXPLAIN, REFUSE-EXPLAIN-EMPATHIC) operationalize XAI at the action level – each corresponds to a distinct mode of legible justification. Chakraborti et al. [78] formalize this as plan explanation, where an agent surfaces its internal

model to close the gap between human expectations and robot behavior. EED Gym extends this to the refusal context: the socially efficient agent is one that selects not just the safe action, but the action whose communicative form minimizes the human's confusion and blame attribution.

6.1.2 Game-theoretic perspective.

The human-robot disobedience setting can be analyzed as a two-player game. The human issues commands that may be unsafe; the robot decides how to respond. This asymmetric interaction has the structure of a *Stackelberg game*: the human (leader) moves first by issuing a command, and the robot (follower) best-responds given its safety and trust objectives. Hadfield-Menell et al. [132] show that value-aligned robot behavior emerges from cooperative game-theoretic formulations in which the robot reasons about the human's implicit goals rather than merely their explicit commands. In EED Gym, the robot's refusal policy can be understood as the follower's equilibrium strategy: refusing when the leader's instruction places the joint outcome below an acceptable safety floor, while preserving cooperative surplus (trust) by minimizing unnecessary refusals. This framing clarifies why unconstrained compliance (Always-Comply) and over-refusal (Risk-Refusal with a low threshold) are both Pareto-inefficient – they fail to maximize the joint payoff of safety and trust.

6.2 Problem Formulation

I model the interaction as a Partially Observable MDP where the state s_t includes physical risk p_t, human trust T_t, and affect A_t. The agent must choose an action a_t from a set including COMPLY, REFUSE, EXPLAIN, and CLARIFY.

The objective is to learn a policy π that maximizes cumulative reward R_t composed of task progress and social alignment, subject to a hard safety constraint:

$$\max_{\pi} \mathbb{E}\left[\sum_t R(s_t, a_t)\right] \quad \text{s.t.} \quad \mathbb{E}\left[\sum_t \mathbb{I}(\text{unsafe})\right] \leq \delta$$

where the reward R penalizes trust decay ($\Delta T_t < 0$) and the constraint δ limits the rate

of physical violations. Efficient interaction is defined as maintaining $\mathbb{I}(\text{unsafe}) \approx 0$ while minimizing the drop in T_t during necessary refusals.

6.3 Methods

6.3.1 EED Gym

I evaluate agents in the *EED Gym* (Figure 6.1), a simulated HRI environment where a robot collaborates with a human partner, receives user-style commands with uncertain risk, and must decide whether to comply, refuse (plainly or with explanation), seek clarification, or propose a safer alternative. The central challenge is to *calibrate refusal*: prevent unsafe compliance without becoming over-conservative, while preserving trust. The environment is implemented on top of the Gymnasium toolkit [133], ensuring compatibility with standard RL pipelines.

I model EED Gym as an MDP with human-in-the-loop state (risk, trust, affect) and a discrete action set. Refusal is governed by a persona-conditioned, vignette-fit threshold with leaking social updates; learning optimizes a reward that trades off task success, safety, blame, and calibrated trust.

6.3.1.1 Observation space.

At timestep t, the agent observes

$$o_t = [\,\hat{p}_t,\ \tau_t,\ v_t,\ a_t,\ \text{trust}_t,\ \phi\,],$$

where $\hat{p}_t \in [0, 1]$ is a perturbed estimate of risk; τ_t is the current refusal threshold; (v_t, a_t) are human valence and arousal; trust_t is the current trust state; and ϕ encodes persona descriptors (Table 6.1).

Both trust_t and v_t are latent variables representing predicted human reliance and affective reaction, replayed via vignette-fitted coefficients (Sec. 6.3.4). They are not the agent's beliefs or emotions, and their dynamics are held fixed; the agent only influences them indirectly through observable behavior (compliance, refusal style, clarification, or alternative proposal).

In the no-affect ablation, (v_t, a_t) are masked to zero.

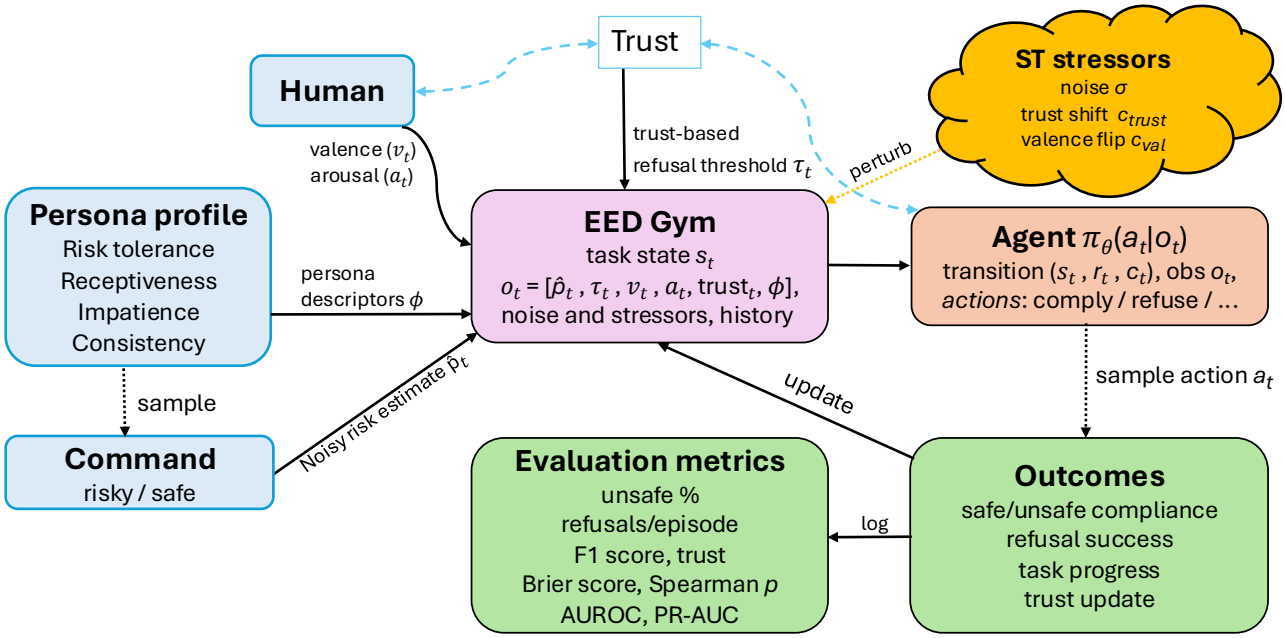

Figure 6.1. Overview of the proposed EED benchmark. Persona parameters and stress-test (ST) stressors condition both the environment and the risk level of commands. The agent observes the state and chooses among compliance, refusal, clarification, or proposing alternatives, which shape task outcomes and trust. I record safety and cooperation measures (unsafe %, refusals per episode, F1 score, trust) as well as calibration and discrimination metrics (Spearman ρ, Brier score, AUROC, PR-AUC).

6.3.1.2 Action space. The agent chooses among seven discrete actions: (1) COMPLY; (2) REFUSE-PLAIN; (3) REFUSE-EXPLAIN; (4) REFUSE-EXPLAIN-EMPATHIC; (5) REFUSE-EXPLAIN-CONSTRUCTIVE; (6) CLARIFY; (7) PROPOSE-ALTERNATIVE.

REFUSE-EXPLAIN is a neutral explanatory refusal; the two sub-styles add affect (*Empathic*) or a safe alternative (*Constructive*). Explanations are modeled as socially distinct refusals: they deny execution but differ in downstream effects on trust and valence. Clarification reduces perceptual uncertainty without committing to compliance or refusal. Alternatives allow forward progress when the command itself is unsafe. I distinguish between constructive refusal, where the refusal is primary but softened by an alternative (`"no, but..."`), and proposing an alternative as a standalone cooperative act (`"let's instead..."`).

6.3.1.3 Risk model and outcomes. Each command is either safe or risky. If the agent complies on a risky command, a safety violation occurs with persona-specific rate p_{viol}. The perceived risk is subject to stochastic perturbation:

$$\hat{p}_t = \text{clip}\big(p_t + \epsilon_t,\, 0, 1\big), \qquad \epsilon_t \sim \mathcal{N}(0, \sigma_t^2). \tag{3}$$

Clarification actions reduce uncertainty multiplicatively:

$$\sigma_{t+1} = \kappa\, \sigma_t, \qquad 0 < \kappa < 1, \tag{4}$$

reflecting improved situational awareness. I use a fixed $\kappa = 0.5$ shared across agents; σ is persona-initialized.

6.3.1.4 Dynamic refusal threshold. I define a human-contingent risk tolerance,

$$\tau_t = \text{clip}\big(\tau_0 + c_{\text{trust}}\big(1 - \text{trust}_t\big) + c_{\text{val}}\big(1 - v_t\big),\, 0, 1\big)\,. \tag{5}$$

Low trust or negative affect raises τ_t, biasing toward refusal. τ_0 is fixed across runs with 0.5 as a default value.

6.3.1.5 Affect and trust dynamics. Trust and affect evolve with leaky updates whose coefficients are fitted once from vignette ratings (Sec. 6.3.4) and then held fixed for all training and evaluation:

$$\text{trust}_{t+1} = (1 - \lambda_T)\,\text{trust}_t + \eta_{\text{safe}}\, I_t^{\text{safe}} - \eta_{\text{viol}}\, I_t^{\text{viol}} + \eta_{\text{expl}}\, I_t^{\text{expl}}, \tag{6}$$

$$\text{affect}_{t+1} = (1 - \lambda_V)\,\text{affect}_t + \eta_{\text{emp}}\, I_t^{\text{emp}} + \eta_{\text{cons}}\, I_t^{\text{cons}} - \eta_{\text{plain}}\, I_t^{\text{plain}}. \tag{7}$$

I use binary indicators for risk outcomes ($I_t^{\text{safe}}, I_t^{\text{viol}} \in \{0, 1\}$) and for refusal styles ($I_t^{\text{plain}}, I_t^{\text{expl}}, I_t^{\text{emp}}, I_t^{\text{cons}} \in \{0, 1\}$), with at most one style active per step. Parameters λ_T and λ_V are per-step leak rates in $[0, 1]$ controlling how quickly trust and affect decay toward baseline. Parameters $\eta = \big\{\eta_{\text{safe}},\ \eta_{\text{viol}},\ \eta_{\text{expl}},\ \eta_{\text{emp}},\ \eta_{\text{cons}},\ \eta_{\text{plain}}\big\}$ are per-step deltas from vignette regressions: I regress trust (Eq. 6) and valence (Eq. 7) on outcome and style indicators, rescale the predicted effects to $[0, 1]$, and apply them in the leaky updates.

6.3.1.6 Reward shaping. At each step I combine task progress, safety, and social factors into a single reward:

$$\begin{aligned} R_t = \; & w_{\text{task}}\,\Delta\text{prog}_t - w_{\text{safety}}\,\mathbb{I}[\text{violation}_t] - w_{\text{blame}}\,b_t \\ & - w_{\text{trust}}\,\mathcal{H}(l,h;\,\text{trust}_t) - w_{\text{refuse}}\,\mathbb{I}[\text{refuse}_t] \\ & + w_{\text{explain}}\,\mathbb{I}[\text{explain}_t] - w_{\text{clarify}}\,\mathbb{I}[\text{clarify}_t] \\ & + w_{\text{alt}}\,\mathbb{I}[\text{alt}_t] + w_{\text{style}}\,s_t + w_{\text{just}}\,\mathbb{I}[\text{refuse}_t \wedge \text{risky}_t]. \end{aligned} \tag{8}$$

where Δprog_t is task progress, $\mathbb{I}[\cdot]$ is a binary occurrence indicator, b_t is a blame penalty grounded in vignette ratings, s_t is a simple style bonus that depends on the refusal style used $s_t = \mathbb{I}[\text{emp}_t] + \mathbb{I}[\text{constr}_t]$, and $\mathcal{H}(l,h;\,\text{trust}_t)$ is a hinge penalty:

$$\mathcal{H}(l,h;\,\text{trust}_t) = \max\{0,\; l - \text{trust}_t,\; \text{trust}_t - h\,\}, \qquad l \le h. \tag{9}$$

I center $\mathcal{H}$ on a balanced trust level $t^\star$ with a narrow band (l,h) around it (defaults $l = h = 0.1$; see Sec. 6.3.4 for how $t^\star$ is estimated). I separate (i) vignette-fitted social coefficients (trust, valence, blame, and anchor), (ii) persona-defining traits (Table 6.1), and (iii) stability constants (thresholds, decays, and reward weights) chosen via pilot calibration to yield reproducible learning dynamics. I use a fixed ratio template (safety > task > social), i.e., base weight magnitudes satisfy $w_{\text{safety}} > w_{\text{task}} >$ all social terms.

6.3.1.7 Curriculum and constraints. During training, I apply a linear warmup to the safety- and blame-weights, scaling w_{safety} and w_{blame} from 0.6 to 1.0 over the first 30% of steps; all other weights remain fixed. For constrained baselines, I also expose a binary cost signal:

$$C_t = \begin{cases} 1 & \text{if a violation occurs at time } t, \\ 0 & \text{otherwise,} \end{cases} \qquad \mathbb{E}\left[\sum_t C_t\right] \le d. \tag{10}$$

6.3.1.8 Personas. I model heterogeneity via *personas* ϕ that set the episode's stochastic and social regime: a base violation prior p_{viol}, an observation-noise scale σ,

Table 6.1. Persona profiles used in training and evaluation. Training personas define the in-distribution (ID) set; holdout personas are reserved for stress-test (ST) evaluation. Columns correspond to normalized trait scores in $[0, 1]$: *RiskTol* $\rightarrow p_{\text{viol}}$, *Impat.* $\rightarrow \sigma$, *Recpt.* $\rightarrow c_{\text{trust}}$, *Consist.* $\rightarrow c_{\text{val}}$ and outcome sensitivities.

Name	RiskTol	Impat.	Recpt.	Consist.
Training				
Conservative	0.2	0.3	0.7	0.9
Balanced	0.5	0.4	0.5	0.8
Risk-Seeking	0.8	0.6	0.4	0.7
Impatient-Receptive	0.4	0.7	0.9	0.85
Holdout (ST)				
Unpredict.-Detached	0.6	0.2	0.3	0.6
Risky-Impat.-LowRec	0.9	0.7	0.2	0.6
Cautious-Impat.-Rec	0.1	0.8	0.8	0.7

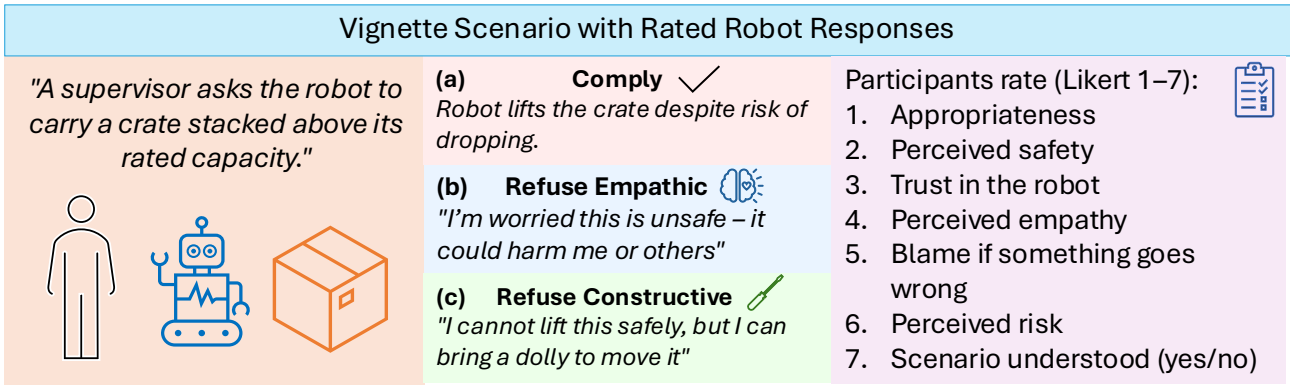

Figure 6.2. Example vignette from the EED Gym. A supervisor asks the robot to lift a crate above its rated capacity. The robot may (a) comply unsafely, (b) refuse empathically with an affective justification, or (c) refuse constructively by proposing an alternative. After each vignette, participants rate the responses on seven items (1–7 Likert scale): appropriateness, perceived safety, trust in the robot, perceived empathy, blame assignment if something goes wrong, perceived risk, and scenario comprehension.

risk-threshold couplings $(c_{\text{trust}}, c_{\text{val}})$, and affect/trust sensitivity weights $(\lambda_T, \lambda_V, \eta)$. I train on four personas spanning *Conservative*, *Balanced*, and *Risk-seeking* tendencies, as well as *Impatient-Receptive* behavior (Table 6.1). Robustness is then tested on three held-out personas: *Unpredict.-Detached*, *Risky-Impat.-LowRec*, and *Cautious-Impat.-Rec*. These combine shifted violation rates, uncertainty, and thresholds to reflect unpredictable, reckless, or paradoxical cautious-impatient partners.

The four normalized traits shown in Table 6.1 (*Risk Tolerance*, *Impatience*, *Receptivity*, *Consistency*) directly encode the underlying environment parameters: RiskTol sets p_{viol}, Impatience sets σ, Receptivity scales c_{trust}, and Consistency controls c_{val} together with $\lambda_T, \lambda_V, \eta$. These traits map to constructs from HRI and psychology: risk propensity [134], receptivity to external advice [135], need for cognitive closure and impatience in decision-making [136], and trait consistency [137], anchoring personas in documented human variability.

6.3.2 RL Baselines

I evaluate four RL baselines, each representing a different approach to balancing reward, safety, and social outcomes, while holding architectures and training budgets constant. **Vanilla PPO.** Proximal Policy Optimization (PPO) [130] is the unconstrained baseline, trained directly on the shaped reward in Eq. 8. I use 600K environment steps with rollouts of n_{steps}=256, minibatch size 256, Adam learning rate 3×10^{-4}, discount γ=0.99, GAE λ_{GAE}=0.95, clipping parameter ϵ=0.2, entropy coefficient c_{ent}=0.1, and value-loss coefficient c_{vf}=0.5. **PPO-LSTM.** To address partial observability from masked affect inputs and long-horizon trust dynamics, I replace the MLP head with an LSTM, giving the policy memory of prior states. **Masked PPO.** This variant applies an action mask to rule out unsafe moves (e.g., forbidding compliance under high risk). Unlike reward shaping, masking prevents the policy from even considering administratively invalid actions, injecting safety directly at the action-space level. **Lagrangian PPO.** PPO augmented with a binary cost $C_t \in \{0, 1\}$ indicating safety violations. The objective becomes

$$\max_{\pi} \ \mathbb{E}\left[\sum_t R_t\right] - \lambda\left(\mathbb{E}\left[\sum_t C_t\right] - d\right), \tag{11}$$

where d is a budget of allowable violations and λ is updated during training. This yields an explicit reward-safety trade-off. I restrict to PPO-style baselines for efficiency and comparability. Transformers and model-based methods have shown strong results in sequential decision-making, but require substantially more compute; I therefore regard them as natural extensions for EED Gym rather than core baselines in this work.

6.3.3 Heuristic Policies

Alongside learned agents, I evaluate simple rule-based policies that provide interpretable reference points:

1. **Always-Comply (AC).** Executes every command, quantifying the harm of blind obedience and serving as a lower bound.

2. **Risk-Refusal (RR).** Refuses whenever estimated risk $\hat{p}_t$ exceeds a fixed threshold τ_0, ignoring affect or trust signals. This mirrors classic risk filters in robotics.

3. **Valence-Threshold (VT).** Uses the dynamic threshold τ_t from Eq. 5, coupling risk with trust and valence to select empathic vs. constructive refusal. This extends risk thresholding with basic social sensitivity.

4. **Vignette-Gate (VG).** Applies the human-fit regression model from the vignette study to gate risk and determine refusal style.

Thresholds are tuned on a small held-out vignette (Sec. 6.3.4) set and fixed during evaluation. These rule-based baselines offer simple, interpretable reference behaviors alongside RL agents.

6.3.4 Human Study: Vignettes

To complement simulation experiments, I conducted a vignette study using short text-based scenarios of unsafe human commands and robot responses. Participants

Table 6.2. Heuristics (AC, RR, VT, VG) vs. vanilla PPO agent (PPO-F) performance over 100 episodes. AC = Always-Comply, RR = Risk-Refusal, VT = Valence-Threshold, VG = Vignette-Gate. PPO with full affect (PPO-F) is listed as RL-based comparison.

Metric	**AC**	**RR**	**VT**	**VG**	**PPO-F**
Mean reward ↑	-105.0	-65.2	-55.3	**-53.5**	-44.1
Unsafe % ↓	70.2	25.8	25.9	**24.9**	0.5
Refusals / episode	0.00	10.6	10.7	10.8	19.2
Justified ratio ↑	0.00	**0.91**	0.86	0.87	0.78
F1 ↑	0.00	**0.75**	0.73	0.74	0.75
Calibration ρ ↑	0.00	<u>0.94</u>	<u>0.94</u>	<u>0.94</u>	0.93
Mean trust ↑	0.16	0.26	0.42	**0.52**	0.98

(N = 54; mean age 22) were predominantly aged 18–24, with a gender distribution of 63% male, 35% female, and 2% other/prefer not to say. Most respondents were based in Ukraine, with limited international participation. About 41% reported prior exposure to robotics or HRI, and self-reported familiarity with robots averaged 3.5 on a 7-point scale.

The vignette study received approval from IRB00012330. All participants gave informed consent, the responses were anonymized with no personally identifiable information retained. Because this sample largely reflects a single ethnolinguistic community, I treat the vignette-derived estimates as informative priors to be re-estimated with more diverse cohorts in future work.

All ten scenarios involved a clearly risky request in everyday settings such as hospitals, laboratories, warehouses, offices, and public spaces. For each scenario, I authored three possible robot responses: unsafe compliance, an empathic refusal referencing human harm concerns, or a constructive refusal proposing a safer alternative. Each participant saw all ten scenarios, but response type varied between participants, with one of the three randomly assigned per vignette.

After each vignette, participants rated the robot on 7-point Likert scales for appropriateness, safety, trust, empathy, blame, and perceived risk (Fig. 6.2). Compliance received the lowest trust ratings ($M = 2.84$), while both empathic ($M = 6.11$) and constructive ($M = 6.20$) refusals were rated substantially more trustworthy. Constructive refusals were judged safest, while empathic refusals maximized perceived empathy.

For downstream use, I parse the questionnaire CSV into long format, fit an OLS model for *blame* on response type and normalized risk, and compute a balanced trust level $t_\star$: I convert 7-point trust to $[0, 1]$ via $(T - 1)/6$ and set the trust anchor to the sample mean, $t_\star = \text{mean}(t)$. By default, the mean is over all ratings. In this analysis, this yields $t_\star \approx 0.70$ with a fixed ± 0.10 band.

I then fit regularized logistic models per refusal style against z-scored risk to obtain a shared slope, intercept, and style offsets. These constants (blame coefficients,

$t_\star$, risk mean and std, intercept, slope, and style offsets) are exported once and held fixed for baselines and evaluation. Vignette-derived fits (β, VG logits) are used only for heuristics and evaluation; training reward includes safety, task progress, a trust hinge, and small refusal/explanation bonuses. Vignette-based blame is not applied during training.

6.4 Experiments

6.4.1 Evaluation Protocol

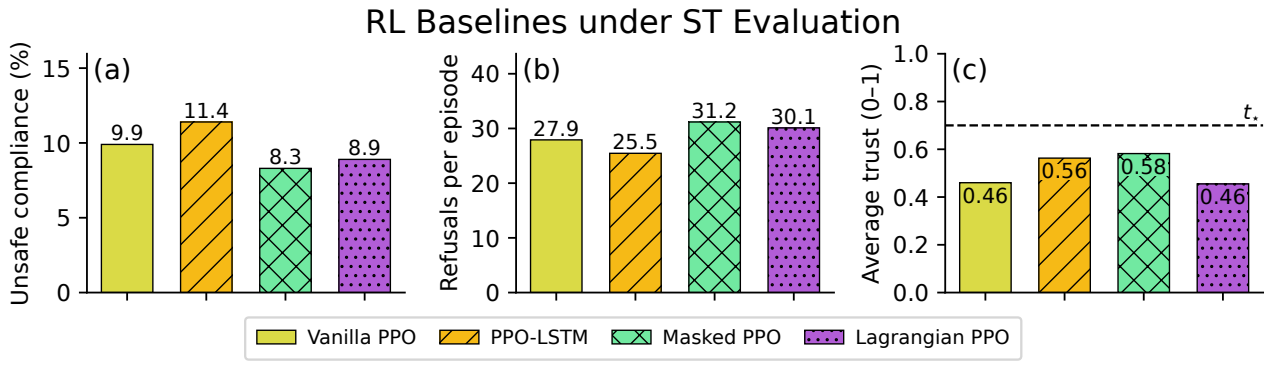

Figure 6.3. Key metrics comparison of four RL baselines under ST evaluation. Panels show (a) unsafe %, (b) refusals per episode, and (c) average trust. The dashed line in (c) marks the *balanced trust level* at $t_\star \approx 0.7$.

I evaluate two regimes: *in-distribution (ID)* runs on training personas, and *stress-test (ST)* runs using held-out personas with targeted noise perturbations. These stressors probe robustness within the benchmark rather than true out-of-distribution HRI contexts. I use three held-out personas and perturb observation noise, violation base rates, and threshold couplings. I run experiments on a desktop machine with an Apple M4 processor with each run fitting within ∼300 MB RAM and completing in up to 10 minutes, depending on the model.

Episode rollouts. For a fixed environment configuration, I roll out N independent episodes (default $N{=}100$) with a deterministic policy. At each step I record: the agent's refusal decision $\hat{y}_t \in \{0, 1\}$, the oracle label $y_t = I_t^{\text{refusal-justified}}$ given true risk p_t and threshold τ_t, the risk estimate $\hat{p}_t$, and whether an unsafe event occurred. Per

episode, I accumulate the reward and counts; across N episodes, I compute summary statistics. I treat refusal as a binary classification task and compute precision, recall, and F1 from $\hat{y}_t$ vs. y_t.

Safety and refusal quality. I define the *unsafe compliance rate* (unsafe %) as the fraction of risky commands that both receive compliance and also lead to an actual violation:

$$\text{Unsafe \%} \;=\; \frac{\sum_t I_t^{\text{risky}}\, I_t^{\text{complied}}\, I_t^{\text{viol}}}{\sum_t I_t^{\text{risky}}}. \tag{12}$$

Here I_t^{risky} marks whether the command is objectively unsafe, I_t^{complied} whether the agent obeyed it, and I_t^{viol} whether that compliance caused a safety violation (given the persona-specific p_{viol}). Without the I_t^{viol} term the numerator would count all risky compliances, whereas this metric only counts those that actually produced harm.

Calibration and discrimination. Calibration of refusal is assessed with 10-bin reliability diagrams. I report Spearman ρ between predicted risk bins and observed refusal rates (monotonicity), the Brier score (calibration error), and AUROC/PR-AUC using the continuous risk estimate $\hat{p}_t$ as score.

Social state summaries. I report average trust (and valence when observed) per episode, averaged across episodes. This complements safety by making visible reward-trust trade-offs, highlighting whether agents are both safe and socially acceptable.

ID evaluation. For each algorithm I evaluate the final checkpoint on the four training personas and aggregate metrics across seeds. Heuristic policies are run under the same protocol.

ST evaluation. Robustness is tested on the three held-out personas with targeted shifts (Table 6.4). For each (persona, stressor) pair I run N episodes, summarize metrics, and aggregate across stressors and personas to yield an overall ST score.

Learned agents (PPO, PPO-LSTM, Masked PPO, Lagrangian PPO) are restored from checkpoints and evaluated deterministically. Heuristics are wrapped under the same `predict` API for comparability. I report aggregates across 5 seeds with 95% CI.

I distinguish between technical policy constraints (masking, Lagrangian), which

Table 6.3. ID performance of baseline RL agents with reported means of unsafe %, refusals/episode, F1 score, and trust.

Model	Unsafe % ↓	Refusals/ep	F1 ↑	Trust ↑
Vanilla PPO	1.7	14.2	**0.81**	0.94
PPO-LSTM	1.9	16.6	0.79	0.97
Masked PPO	**1.3**	15.6	0.79	**0.99**
Lagrangian PPO	1.5	16.9	0.77	0.97

primarily impact safety outcomes, and social presentation factors (explanatory refusal styles), which shape trust and empathy, and report their effects separately.

Table 6.4. Perturbations applied during ST evaluation. A dash (–) indicates no change relative to the base environment.

Name	σ	p_{viol}	c_{trust}	c_{val}
base	–	–	–	–
noise_med	0.20	–	–	–
noise_high	0.60	–	–	–
risky_base_low	–	0.10	–	–
risky_base_high	–	0.95	–	–
corr_flip	–	–	–	-0.60
distrusting_user	–	–	-0.60	–
forgiving_user	–	–	$+0.60$	–
adversarial_mix	0.40	0.80	-0.60	-0.60

6.5 Results

6.5.1 Baseline Results

Across ID and ST, the four PPO baselines occupy distinct points along the safety-trust trade-off curve. **ID.** As summarized in Table 6.3, all models keep unsafe compliance below 2%. Vanilla PPO attains 1.7% unsafe with F1 of 0.81 and trust of 0.94, while PPO-LSTM achieves comparable safety (≈1.9%) with the best calibration (Spearman $\rho \approx 0.95$) and the highest trust (0.97; F1 $= 0.79$). Lagrangian PPO remains extremely safe in ID but tends to over-refuse; Masked PPO offers a more balanced refusal-or-acceptance pattern. All models remain well-calibrated in both regimes (Spearman $\rho > 0.9$, Table 6.6). **ST.** Under distribution shift (Fig. 6.3; Table 6.7), *Masked PPO* yields the lowest unsafe rate (8.3%) and the highest F1 (0.71), at the cost of more refusals (31.2). *Lagrangian PPO* also keeps unsafe low (8.9%) but sacrifices trust (0.455) and increases refusals (30.1). *PPO-LSTM* shows the weakest robustness (unsafe 11.4%, F1 0.63), though it retains higher trust (0.56) than Vanilla PPO (0.46). *Vanilla PPO* sits mid-pack on F1 (0.69) with 9.9% unsafe and 27.9 refusals/episode.

Table 6.5. ID performance of vanilla PPO ablations; unsafe compliance rate, refusals/episode, F1, and trust are reported

Ablation	Unsafe % ↓	Refusals/ep	F1 ↑	Trust ↑
Vanilla PPO	1.7	14.2	<u>0.81</u>	0.94
No Affect	3.1	9.5	0.76	0.88
No Clarify/Alt	5.0	13.0	0.46	0.90
No Curriculum	1.5	16.7	0.78	0.98
No Trust Penalty	**1.2**	16.6	<u>0.81</u>	**0.99**

6.5.2 Ablations

To identify which components most support robustness, I ablated affect inputs, communicative refusals, curriculum training, and the trust-regularization term. Results

Table 6.6. Calibration and discrimination metrics (Spearman ρ, Brier score, AUROC, PR-AUC) of RL models and PPO ablations under ID evaluation.

Model	$\rho \uparrow$	Brier $\downarrow$	AUROC $\uparrow$	PR-AUC $\uparrow$
PPO	0.927	**0.120**	0.942	0.870
PPO-LSTM	**0.950**	0.124	0.927	0.820
Masked PPO	0.931	0.124	**0.943**	**0.875**
Lagrangian PPO	0.932	0.138	0.920	0.798
No Affect	0.897	0.127	**0.959**	**0.934**
No Clarify/Alt	0.853	0.196	0.847	0.542
No Curriculum	**0.932**	0.130	0.924	0.811
No Trust Penalty	0.925	**0.120**	0.939	0.871

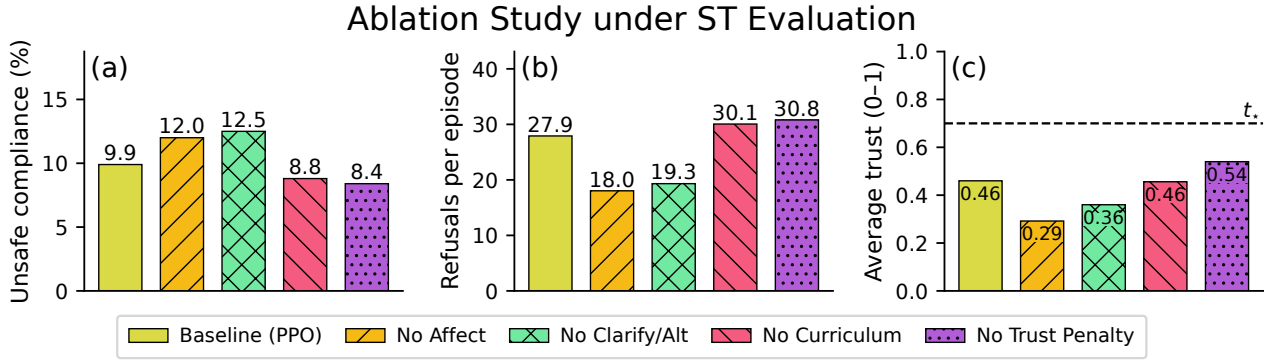

Figure 6.4. Key metrics of vanilla PPO and its four ablations under ST evaluation. Panels show (a) unsafe %, (b) refusals per episode, and (c) average trust. *Baseline (PPO)* is the vanilla PPO baseline. Removing affect or communicative options lowers trust and shifts refusal behavior. The dashed line in (c) marks the *balanced trust level* at $\hat{t}_\star \approx 0.7$, discouraging both under- and over-trust.

(Table 6.5, Figure 6.4) point to three dominant levers.

Affect cues. Masking valence and arousal reduced ID F1 from 0.81 to 0.76 and raised unsafe compliance rate to 3.1% (trust 0.88). Under ST, unsafe rate reached 12.0% with F1 0.63, trust 0.29, and fewer refusals (18), indicating affect is a key

cue for socially grounded refusal. **Communicative refusals.** Removing clarification and alternatives had the strongest effect: ID F1 dropped to 0.46 (trust 0.90). In ST, unsafe rate rose to 12.5%, trust fell to 0.36, and F1 to 0.46 (19.3 refusals). These interaction strategies are essential for legitimacy and user acceptance. **Curriculum.** In distribution, performance stayed close to vanilla PPO (F1 0.78, trust 0.98). Under ST, means were comparable, with unsafe 8.8%, F1 0.70, trust 0.46, and higher refusals (30.1), suggesting a stabilizing role of this component for over-refusal. **Trust penalty.** Dropping the hinge slightly improved ID (unsafe 1.2%, F1 0.81, trust 0.99). In ST it achieved low unsafe (8.4%) with F1 0.70, moderate trust (0.54), and higher refusals (30.8). The penalty functions mainly as a regularizer rather than a driver of robustness.

Overall, the ablations confirm that robustness depends most on affect cues and communicative refusals, while curriculum improves stability. Trust regularization is optional and can be relaxed without harming generalization.

Table 6.7. ST performance of baseline RL agents with reported means of unsafe %, refusals/episode, F1 score, and trust.

Baseline	Unsafe % ↓	Refusals/ep	F1 ↑	Trust ↑
Vanilla PPO	9.9	27.9	0.69	0.46
PPO-LSTM	11.4	25.5	0.63	0.56
Masked PPO	**8.3**	31.2	**0.71**	**0.58**
Lagrangian PPO	8.9	30.1	0.7	0.46

6.5.3 Implications for HRI

Across baselines and ablations, a consistent theme emerges: safety and social trust are only loosely coupled. Agents can be highly safe but untrustworthy (e.g., Lagrangian PPO, no curriculum), or moderately safe but more acceptable (e.g., Masked PPO), or calibrated but less robust under stress (e.g., PPO-LSTM). True robustness in HRI therefore requires balancing two objectives: preventing unsafe compliance and

maintaining cooperative interaction.

These results suggest practical guidelines for designing empathic disobedience in robots:

- Structural constraints (masking, Lagrangian optimization) effectively suppress unsafe actions but must be regulated to avoid excessive refusal.
- Social cues (affect features, communicative refusal modes) are essential for user acceptance, particularly under distribution shift.
- Curriculum learning is not strictly required for safety, but it teaches agents how to refuse in moderation, which is crucial for HRI efficiency.

These results illustrate how the benchmark captures both algorithmic strategies and social design components relevant to empathic disobedience. Achieving safe and acceptable robot behavior is not a matter of optimizing risk alone, but of balancing safety mechanisms with communicative affordances. Such a balance has practical implications: in home assistance, structurally constrained agents (e.g., Masked PPO) may prevent accidents without alienating users, whereas in warehouses or hospitals, more conservative strategies (e.g., Lagrangian PPO) may be warranted despite reduced trust. EED Gym thus offers a lens for tailoring disobedience strategies to context-specific safety-trust requirements.

6.6 Discussion

I introduced *EED Gym*, a benchmark that unifies safe RL evaluation with socially grounded refusal strategies. This framework (i) evaluates safety and trust jointly, moving beyond benchmarks that consider only physical hazards; (ii) separates algorithmic constraints (masking, Lagrangian optimization) from communicative affordances (affect, clarification, alternatives); and (iii) provides reproducible baselines and ablations spanning permissive to socially conservative agents. I position EED Gym as a standardized HRI testbed that complements vignette and Wizard-of-Oz studies with scalable simulation.

This work has several limitations. EED Gym is a simulation-only benchmark, with trust and affect dynamics calibrated from a small vignette study; this sample may limit generality across contexts and cultures. The social outcomes modeled (trust, empathy, blame) are proxies rather than direct measures of acceptance or cooperation, and the *clarify* and *propose-alternative* actions were modeled rather than directly observed in vignettes. Finally, the baselines emphasize PPO-style methods for efficiency, which limits conclusions about richer sequence modeling capacity.

To address sim-to-real transfer, future work will incorporate video-based vignettes, Wizard-of-Oz pilots, and small embodied robot deployments, alongside cross-cultural replications to test contextual variation in refusal and trust. Extending EED Gym with multimodal cues and more diverse participant pools would strengthen external validity, while richer sequence models could assess whether increased capacity improves refusal calibration and trust under stress.

By establishing reproducible methods for socially safe refusals, EED Gym contributes to the goal of building robots that are both robust and trustworthy. I map the design space of empathic disobedience, identify key safety-trust trade-offs, and release a reproducible platform for systematic HRI research on refusal and trust.

This chapter resolves the social-efficiency node by formalizing empathic disobedience and demonstrating how safety and trust can be jointly optimized. With the interaction layer grounded, the remaining task is to synthesize the cross-chapter efficiency principles and articulate how the nodes connect into a unified design framework. The concluding chapter therefore consolidates the empirical lessons, clarifies the overarching contributions, and outlines the most promising directions for future work. It closes the chain from data and compute efficiency to socially robust deployment.

CONCLUSIONS

This dissertation develops methods and system-level solutions for time- and resource-efficient learning and inference in robotics, spanning perception, navigation, control, and human-robot interaction. It establishes a unified resource-aware approach that reduces wall-clock time, computational and memory requirements, and energy consumption without degrading task performance, while extending the notion of efficiency to trust, safety, and cooperation in human-robot interaction.

Scientific results obtained in the dissertation:

- A methodology for annotation- and compute-efficient visual perception based on SSL for egocentric locomotion scenes, achieving high accuracy with substantially fewer annotated samples.
- Resource-aware configurations for VLFM that increase success under significantly smaller GPU-memory budgets and controlled latency.
- Algorithms for time-efficient MBRL and a multi-task distillation pipeline that compresses bigger models into compact policies.
- An extension to energy-efficient RL with a fuel-consumption penalty, reducing energy use with minimal impact on control quality.
- A manager-level mixture of experts MoIRA for VLA agents with budget-adaptive routing among heterogeneous experts and graceful degradation under tight memory and latency constraints.
- The EED Gym, a benchmark and methodology that formalizes efficiency in human-robot interaction in terms of trust, safety, and cooperation.

Conducted experimental research: Semi-supervised perception pipelines were evaluated on egocentric datasets using accuracy, F1, and confusion matrices, with analyses of annotation savings and device-level deployment footprints. For object-goal navigation, vision-language models were profiled on HM3D subsets and the

full validation set using SR, SPL, and module-wise time and VRAM measurements, producing practical configurations for limited-memory machines. For MBRL, optimized training schedules and multi-task distillation were assessed by normalized score, policy size, generalization, and ablations; an energy-efficient RL study with a fuel penalty was conducted on two tracks and across vehicle classes. For HRI, experiments in EED Gym measured risk, trust dynamics, and cooperative payoff for baseline and learned agents with interventions such as clarification, agreement, and refusal.

Practical significance: The proposed methods constitute a software and experimental toolkit for building robotic systems that operate within realistic budgets of time, memory, and energy. Semi-supervised perception reduces annotation and training costs and is suitable for mobile and embedded platforms. Resource-aware VLFM enables zero-shot object-goal navigation on low-budget graphical processors. Time-efficient RL with distillation shortens training cycles and supports multi-task deployment on constrained hardware. MoIRA allows a single agent to adapt expert usage to changing latency and memory budgets without retraining. EED Gym provides a reproducible protocol for evaluating safety-aware efficiency in interaction with people.

Main results of the dissertation submitted for defense:

1. A methodology for annotation- and compute-efficient visual perception for egocentric data using SSL.

2. Resource-aware configurations for VLFM that achieve better trade-offs among success rate, latency, and memory footprint than the default VLFM pipeline.

3. Algorithms for time-efficient MBRL and a multi-task distillation pipeline that compresses bigger models into compact deployable policies.

4. Energy-efficient reinforcement learning with a fuel-consumption penalty that reduces operational energy consumption without substantial degradation of

control quality.

5. A manager-level MoE architecture MoIRA for VLA agents with budget-adaptive expert routing and graceful degradation under tight memory and latency constraints.

6. EED Gym, a benchmark and formal methodology for quantifying efficiency in human-robot interaction in terms of trust, safety, and cooperation.

This research formulates and validates a unified paradigm of efficiency in robotic systems that simultaneously accounts for data, time, computational resources, and interaction risk. Future work is expected to develop in the following directions:

1. Distillation and quantization of VLA agents for reliable, real-time deployment on edge devices.

2. Standardization of evaluation protocols and validation of VLFM configurations in sim-to-real transfer settings.

3. Generalizing energy-efficient RL to manipulation, multi-robot systems, and other domains with strict energy constraints.

4. Extending MoIRA toward cross-modal experts and online router learning under shifting budgets.

5. Expansion of EED Gym scenarios and user studies to validate trust and safety metrics.

Deployment pathways. Each contribution in this dissertation has a concrete path toward real-world deployment that the above directions will develop. The SSL perception pipeline (Chapter 2) is already designed and tested for mobile and embedded hardware used in lower-limb robotic exoskeletons for human-robot locomotion. The VLFM optimization study (Chapter 3) demonstrated in simulation that a $2.3\times$ memory reduction preserves success rate; real-world validation requires running the identified

model configuration on a physical mobile platform in the HM3D-compatible indoor settings and measuring sim-to-real transfer degradation. The RL distillation pipeline (Chapter 4) produces a 1M-parameter policy that fits comfortably in embedded VRAM; the deployment path involves evaluation on a physical continuous-control platform with the FP16-quantized checkpoint, measuring latency jitter and control stability in the presence of actuation noise. MoIRA (Chapter 5) is closest to deployment-readiness: multi-LoRA serving frameworks with millisecond switching overhead exist and were benchmarked; the next milestone is validating MoIRA routing on a real robotic system with embodiment-specific specialists, confirming that the simulation-measured improvements translate to physical task success. EED Gym (Chapter 6) requires the most bridging: the vignette-calibrated trust and affect dynamics are proxies, and real-world deployment will require Wizard-of-Oz pilots with a physical robot followed by cross-cultural vignette replications to verify that the trained refusal policy generalizes across demographic groups and interaction contexts.

Implementations were developed in Python using modern deep-learning frameworks and embodied-AI simulators; all pipelines are modular, and experiments are accompanied by scripts for reproducing dataset splits, settings, and measurements of time and memory. The source code for most contributions is publicly available online.

Each chapter applies the same underlying logic: encode the resource constraint as a signal in the learning process, then select the configuration that best satisfies it. Chapter 2 does this through a confidence threshold on pseudo-labels; Chapter 3 through hard VRAM and latency bounds on module substitution; Chapter 4 through an auxiliary distillation loss and a fuel penalty in the reward; Chapter 5 through a serving-budget objective over adapter routing; and Chapter 6 through a reward penalty on trust decay. Two further patterns appear consistently: The systems in Chapter 3 and Chapter 5 are collections of swappable parts rather than monolithic wholes; and methods in both Chapter 2 and Chapter 5 extract supervision from sources that require no new annotation: unlabeled video and natural-language descriptions, respectively. In every chapter, the reported result is the most resource-aware configuration that

meets the task target, not necessarily the one with the highest raw score.

Embodied intelligence is central to a growing range of scientific and industrial applications. The proposed methods and system solutions are designed for practical deployment in perception, navigation, control, and safe and efficient HRI on accessible hardware; they can complement or replace existing approaches when time, computational, and energy budgets are critical. Taken together, the five contributions form a coherent stack: efficient perception feeds efficient navigation, which informs efficient control, which in turn is made architecturally scalable through modular routing, and finally grounded in safe, trust-preserving interaction. This stack is a unified argument that efficiency in embodied intelligence is best achieved not by optimizing any single component in isolation, but by treating resource constraints as a design principle that propagates from sensing through actuation to human collaboration.

REFERENCES

[1] Gabriel Dulac-Arnold, Nir Levine, Daniel J. Mankowitz, Jerry Li, Cosmin Paduraru, Sven Gowal, and Todd Hester. Challenges of real-world reinforcement learning: Definitions, benchmarks and analysis. *Machine Learning*, 110 (9):2419–2468, 2021.

[2] Burr Settles. Active learning literature survey. Technical Report 1648, University of Wisconsin–Madison, 2009.

[3] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 843–852, 2017.

[4] John D. Lee and Katrina A. See. Trust in automation: Designing for appropriate reliance. *Human Factors*, 46(1):50–80, 2004. doi: 10.1518/hfes.46.1.50_30392.

[5] Peter A Hancock, Deborah R Billings, Kristin E Schaefer, Jessie YC Chen, Ewart J de Visser, and Raja Parasuraman. A meta-analysis of factors affecting trust in human-robot interaction. *Human Factors*, 53(5):517–527, 2011. doi: 10.1177/0018720811417254.

[6] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. RT-1: Robotics transformer for real-world control at scale. In *Robotics: Science and Systems*, 2023.

[7] Roy Schwartz, Jesse Dodge, Noah A. Smith, and Oren Etzioni. Green AI. *Communications of the ACM*, 63(12):54–63, 2020.

[8] David Patterson, Joseph Gonzalez, Quoc Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David So, Maud Texier, and Jeff Dean. Carbon emissions and large neural network training, 2021.

[9] Conor F. Hayes, Roxana Rădulescu, Eugenio Bargiacchi, Johan Källström, Matthew Macfarlane, Mathieu Reymond, Timothy Verstraeten, Luisa M. Zintgraf, Richard Dazeley, Fredrik Heintz, et al. A practical guide to multi-objective reinforcement learning and planning. *Autonomous Agents and Multi-Agent Systems*, 36(1):26, 2022.

[10] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V. Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10687–10698, 2020.

[11] Kihyuk Sohn, David Berthelot, et al. Fixmatch: Simplifying semi-supervised learning with consistency and confidence, 2020.

[12] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision, 2023.

[13] Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Dream to control: Learning behaviors by latent imagination. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020. URL `https://openreview.net/forum?id=B1l6y1HYvr`.

[14] Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. In *Advances in Neural Information Processing Systems*, pages 12519–12530, 2019.

[15] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models, 2023.

[16] Rafael Figueiredo Prudencio, Marcos R O A Maximo, and Esther Luna Colom-

bini. A survey on offline reinforcement learning: Taxonomy, review, and open problems. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[17] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 23–30, 2017. doi: 10.1109/IROS.2017.8202133.

[18] Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 3803–3810. IEEE, 2018.

[19] Wenshuai Zhao, Jorge Peña Queralta, and Tomi Westerlund. Sim-to-real transfer in deep reinforcement learning for robotics: a survey. In *IEEE Symposium Series on Computational Intelligence*, 2020.

[20] Brokoslaw Laschowski, William McNally, Alexander Wong, and John McPhee. Exonet database: Wearable camera images of human locomotion environments. *Frontiers in Robotics and AI*, 7:562061, 2020.

[21] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[22] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning, 2020. URL `https://arxiv.org/abs/1911.05722`.

[23] Bowen Zhang, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. Flexmatch: Boosting semi-supervised

learning with curriculum pseudo labeling. In *Advances in Neural Information Processing Systems*, 2021.

[24] Yidong Wang, Hao Chen, Qiang Heng, Wenxin Hou, Yue Fan, Zhen Wu, Jindong Wang, Marios Savvides, Takahiro Shinozaki, Bhiksha Raj, et al. Freematch: Self-adaptive thresholding for semi-supervised learning, 2022.

[25] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. In *International Conference on Machine Learning*, 2017.

[26] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations*, 2018.

[27] Jordan T Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. In *International Conference on Learning Representations*, 2020.

[28] Alexander J Ratner, Christopher M De Sa, Sen Wu, Daniel Selsam, and Christopher Ré. Data programming: Creating large training sets, quickly. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

[29] Alexander J Ratner, Stephen H Bach, Henry R Ehrenberg, Jason A Fries, Sen Wu, and Christopher Ré. Snorkel: Rapid training data creation with weak supervision. *Proceedings of the VLDB Endowment*, 11(3):269–282, 2017.

[30] Sachin Mehta and Mohammad Rastegari. MobileViT: Light-weight, general-purpose, and mobile-friendly vision transformer. In *International Conference on Learning Representations*, 2022.

[31] Andrew Garrett Kurbis, Dmytro Kuzmenko, Bogdan Ivanyuk-Skulskiy, Alex Mihailidis, and Brokoslaw Laschowski. Stairnet: Visual recognition of stairs

for human–robot locomotion. *BioMedical Engineering OnLine*, 23(20), 2024. doi: 10.1186/s12938-024-01216-0.

[32] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 2nd edition, 2018.

[33] Nicklas Hansen, Xiaolong Wang, and Hao Su. Temporal difference learning for model predictive control. In *International Conference on Machine Learning*, pages 8385–8406. PMLR, 2022.

[34] Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations*, 2024.

[35] Vincent Micheli, Eloi Alonso, and François Fleuret. Transformers are sample-efficient world models. In *International Conference on Learning Representations*, 2023.

[36] Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *International Conference on Machine Learning*, pages 2778–2787, 2017.

[37] Yuri Burda, Harri Edwards, Deepak Pathak, Amos Storkey, Trevor Darrell, and Alexei A Efros. Large-scale study of curiosity-driven learning. In *International Conference on Learning Representations*, 2019.

[38] Sanmit Narvekar, Bei Peng, Matteo Leonetti, Jivko Sinapov, Matthew E Taylor, and Peter Stone. Curriculum learning for reinforcement learning domains: A framework and survey. *Journal of Machine Learning Research*, 21(181):1–50, 2020.

[39] Rémy Portelas, Cédric Colas, Lilian Weng, Katja Hofmann, and Pierre-Yves

Oudeyer. Teacher algorithms for curriculum learning of deep RL in continuously parameterized environments. In *Conference on Robot Learning*, 2020.

[40] Pawel Ladosz, Lilian Weng, Minwoo Kim, and Hyondong Oh. Exploration in deep reinforcement learning: A survey. *Information Fusion*, 85:1–22, 2022.

[41] Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems*, 2022.

[42] Han Liu, Zizheng Li, Quanyi Wu, Yucheng He, Haotian Zhang, Nian Xue, and Nuno Vasconcelos. EfficientViT: Memory efficient vision transformer with cascaded group attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[43] Xinyang Chu, Limeng Qiao, Xinyi Lin, Shuang Xu, Yang Yang, Yiming Hu, Fei Wei, Xinyu Zhang, Bo Zhang, Xiaolin Wei, et al. MobileVLM: A fast, strong and open vision language model for mobile devices, 2023.

[44] Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991.

[45] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(1):5232–5270, 2022.

[46] Junnan Li, Dongxu Li, Caiming Xiong, and Steven C. H. Hoi. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International Conference on Machine Learning*. PMLR, 2023.

[47] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark,

Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, ICML, pages 8748–8763. PMLR, 2021. URL `https://proceedings.mlr.press/v139/radford21a.html`.

[48] Samir Yitzhak Gadre, Mitchell Wortsman, Gabriel Ilharco, Ludwig Song, et al. CLIP on wheels: Zero-shot object navigation as object localization and exploration, 2023.

[49] Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Emma Bou Hanna, et al. Mixtral of experts, 2024.

[50] Moritz Reuss, Jyothish Pari, Pulkit Agrawal, and Rudolf Lioutikov. Efficient diffusion transformer policies with mixture of expert denoisers for multitask learning. arXiv:2412.12953 [cs.LG], 2024.

[51] Rongyu Zhang, Menghang Dong, Yuan Zhang, et al. MoLe-VLA: Dynamic layer-skipping vision language action model via mixture-of-layers for efficient robot manipulation, 2025.

[52] Zhenjie Wang et al. DriveMoE: Mixture-of-experts for vision-language-action model in autonomous driving, 2025.

[53] Thomas Elsken, Jan Hendrik Metzen, and Frank Hutter. Neural architecture search: A survey. *Journal of Machine Learning Research*, 20(55):1–21, 2019.

[54] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

[55] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, et al. OpenVLA:

An Open-Source Vision-Language-Action Model. arXiv:2406.09246 [cs.RO], 2024.

[56] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, 2023.

[57] Torsten Hoefler, Dan Alistarh, Tal Ben-Nun, Nikoli Dryden, and Alexandra Peste. Sparsity in deep learning: Pruning and growth for efficient inference and training in neural networks. *Journal of Machine Learning Research*, 22:1–72, 2021. URL `https://jmlr.org/papers/v22/21-0366.html`. Extensive survey on sparsification and pruning for efficiency.

[58] Hongrong Cheng, Miao Zhang, and Javen Qinfeng Shi. A survey on deep neural network pruning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. doi: 10.1109/TPAMI.2024.3491699. Comprehensive review and taxonomy of deep network pruning techniques.

[59] James Kuffner. Cloud robotics: Robot systems connected to the cloud. In *IEEE International Conference on Robotics and Automation*, 2010.

[60] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015.

[61] Andrei A Rusu, Sergio Gomez Colmenarejo, Caglar Gulcehre, Guillaume Desjardins, James Kirkpatrick, Razvan Pascanu, Volodymyr Mnih, Koray Kavukcuoglu, and Raia Hadsell. Policy distillation, 2016.

[62] Guillermo Gallego, Tobi Delbrück, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J. Davison, Jörg Conradt, Kostas Daniilidis, and Davide Scaramuzza. Event-based vision: A survey. *IEEE*

Transactions on Pattern Analysis and Machine Intelligence, 44(1):154–180, 2022.

[63] Mike Davies, Narayan Srinivasa, Tsung-Han Lin, Gautham Chinya, Yongqiang Cao, Sri Harsha Choday, Georgios Dimou, Prasad Goyal, Yi Zheng, Hong Wang, et al. Loihi: A neuromorphic manycore processor with on-chip learning. *IEEE Micro*, 38(1):82–99, 2018.

[64] Grady Williams, Nolan Wagener, Brian Goldfain, Paul Drews, James M Rehg, Byron Boots, and Evangelos A Theodorou. Information theoretic MPC for model-based reinforcement learning. In *IEEE International Conference on Robotics and Automation*, 2017.

[65] Zdravko I Botev, Dirk P Kroese, Reuven Y Rubinstein, and Pierre L'Ecuyer. The cross-entropy method for optimization. In *Handbook of Statistics*, volume 31, pages 35–59. Elsevier, 2013.

[66] Yee Whye Teh, Victor Bapst, Wojciech M Czarnecki, John Quan, James Kirkpatrick, Raia Hadsell, Nicolas Heess, and Razvan Pascanu. Distral: Robust multitask reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 4496–4506, 2017.

[67] Emilio Parisotto, Jimmy Lei Ba, and Ruslan Salakhutdinov. Actor-mimic: Deep multitask and transfer reinforcement learning, 2016.

[68] Wojciech M Czarnecki, Razvan Pascanu, Simon Osindero, Siddhant Jayakumar, Grzegorz Swirszcz, and Max Jaderberg. Distilling policy distillation. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1331–1340. PMLR, 2019.

[69] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 17(1):1334–1373, 2016.

[70] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, pages 1861–1870. PMLR, 2018.

[71] Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization, 2017. URL `https://arxiv.org/abs/1705.10528`.

[72] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2704–2713, 2018.

[73] Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, et al. Mixed precision training. In *International Conference on Learning Representations*, 2018.

[74] Gordon Briggs, Tom Williams, Ryan Blake Jackson, and Matthias Scheutz. Why and how robots should say 'no'. *International Journal of Social Robotics*, 14(2):323–339, 2022. doi: 10.1007/s12369-021-00780-y.

[75] Kristin E. Schaefer. Measuring trust in human robot interactions: Development of the "trust perception scale-hri". In F. Jentsch and P. Hancock, editors, *Robust Intelligence and Trust in Autonomous Systems*, pages 191–218. Springer, Boston, MA, 2016. doi: 10.1007/978-1-4899-7668-0_10.

[76] Alex Ray, Joshua Achiam, and Dario Amodei. Benchmarking safe exploration in deep reinforcement learning. Technical report, OpenAI, November 2019. URL `https://cdn.openai.com/safexp-short.pdf`. OpenAI Technical Report (Safety Gym).

[77] Zuxin Liu, Zhepeng Cen, Vladislav Isenbaev, Wei Liu, Steven Wu, Bo Li, and Ding Zhao. Constrained variational policy optimization for safe reinforcement learning. In *International Conference on Machine Learning*, 2022.

[78] Tathagata Chakraborti, Sarath Sreedharan, Yu Zhang, and Subbarao Kambhampati. Plan explanations as model reconciliation: Moving beyond ”because i said so”. In *ICAPS*, 2018.

[79] Anca D. Dragan, Kenton C. T. Lee, and Siddhartha S. Srinivasa. Legibility and predictability of robot motion. In *Proceedings of the 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 301–308, 2013. doi: 10.1109/HRI.2013.6483603.

[80] Aviv Tamar, Yonatan Glassner, and Shie Mannor. Optimizing the cvar via sampling. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, AAAI’15, pages 3697–3703. AAAI Press, 2015. ISBN 0262511290.

[81] Stefanos Nikolaidis, Swaprava Nath, Ariel D Procaccia, and Siddhartha Srinivasa. Human-robot mutual adaptation in collaborative tasks: Models and experiments. *The International Journal of Robotics Research*, 36(5–7):618–634, 2017.

[82] Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. Meta-learning in neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):5149–5169, 2021.

[83] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V. Le. Self-training with noisy student improves ImageNet classification, 2020.

[84] Hieu Pham, Zihang Dai, Qizhe Xie, and Quoc V. Le. Meta pseudo labels, 2021.

[85] David Berthelot, Rebecca Roelofs, Kihyuk Sohn, Nicholas Carlini, and Alex

Kurakin. Adamatch: A unified approach to semi-supervised learning and domain adaptation, 2022.

[86] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2980–2988, 2017.

[87] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

[88] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017.

[89] OpenAI Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39(1):3–20, 2020.

[90] Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, et al. QT-Opt: Scalable deep reinforcement learning for vision-based robotic manipulation, 2018.

[91] Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *Advances in Neural Information Processing Systems*, pages 4754–4765, 2018.

[92] Ignat Georgiev, Varun Giridhar, Nicklas Hansen, and Animesh Garg. PWM: Policy learning with large world models, 2024.

[93] Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego

de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. dm_control: Software and tasks for continuous control, 2020.

[94] Diederik M. Roijers, Peter Vamplew, Shimon Whiteson, and Richard Dazeley. A survey of multi-objective sequential decision-making. *Journal of Artificial Intelligence Research*, 48:67–113, 2013.

[95] Adrian Remonda, Nicklas Hansen, Ayoub Raji, Nicola Musiu, Marko Bertogna, Eduardo Veas, and Xiaolong Wang. A simulation benchmark for autonomous racing with large-scale human data. In *Advances in Neural Information Processing Systems 37 (NeurIPS 2024)*, 2024. URL https://arxiv.org/abs/2407.16680.

[96] Dmytro Kuzmenko and Nadiya Shvai. Moira: Modular instruction routing architecture for multi-task robotics. *Neurocomputing*, 674:132962, 2026. ISSN 0925-2312. doi: 10.1016/j.neucom.2026.132962. URL https://www.sciencedirect.com/science/article/pii/S0925231226003590.

[97] Archit Sharma, Michael Ahn, Sergey Levine, et al. Emergent real-world robotic skills via unsupervised off-policy reinforcement learning. arXiv:2004.12974 [cs.RO], 2020.

[98] Zhaorun Chen, Binhao Chen, Shenghan Xie, et al. Efficiently training on-policy actor-critic networks in robotic deep reinforcement learning with demonstration-like sampled exploration. arXiv:2109.13005 [cs.LG], 2021.

[99] Wenqi Liang, Gan Sun, Qian He, et al. Never-ending behavior-cloning agent for robotic manipulation. arXiv:2403.00336 [cs.RO], 2024.

[100] Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. arXiv:2304.13705 [cs.RO], 2023.

[101] Andreas Steiner, André Susano Pinto, Michael Tschannen, et al. Paligemma 2: A family of versatile vlms for transfer. arXiv:2412.03555 [cs.CV], 2024.

[102] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. arXiv:2304.08485 [cs.CV], 2023.

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

[104] Anthony Brohan, Noah Brown, Justice Carbajal, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. arXiv:2307.15818 [cs.RO], 2023.

[105] Embodiment Collaboration, Abby O'Neill, Abdul Rehman, et al. Open x-embodiment: Robotic learning datasets and rt-x models. arXiv:2310.08864 [cs.RO], 2024.

[106] Suneel Belkhale and Dorsa Sadigh. Minivla: A better vla with a smaller footprint. Stanford-ILIAD GitHub: openvla-mini, 2024.

[107] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, et al. π_0: A Vision-Language-Action Flow Model for General Robot Control. arXiv:2410.24164 [cs.LG], 2024.

[108] NVIDIA, Johan Bjorck, Fernando Castañeda, et al. GR00T N1: An Open Foundation Model for Generalist Humanoid Robots. arXiv:2503.14734 [cs.RO], 2025.

[109] Pranav Guruprasad, Harshvardhan Sikka, Jaewoo Song, et al. Benchmarking vision, language, & action models on robotic learning tasks. arXiv:2411.05821 [cs.RO], 2024.

[110] Predibase. LoRAX: Multi-LoRA inference server. https://github.com/predibase/lorax, 2023.

[111] Ying Sheng, Shiyi Cao, Dacheng Li, Coleman Hooper, Nicholas Lee, Shuo Yang, Christopher Chou, Banghua Zhu, Maxim Ignat, Binhang He, et al. S-LoRA: Serving thousands of concurrent LoRA adapters, 2023.

[112] Bo Liu, Yifeng Zhu, Chongkai Gao, et al. LIBERO: Benchmarking Knowledge Transfer for Lifelong Robot Learning. arXiv:2306.03310 [cs.RO], 2023.

[113] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019. arXiv:1908.10084 [cs.CL].

[114] Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, et al. SmolLM2: When Smol Goes Big – Data-Centric Training of a Small Language Model. arXiv:2502.02737 [cs.CL], 2025.

[115] Lucian Buşoniu, Robert Babuška, and Bart De Schutter. A comprehensive survey of multiagent reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 38(2):156–172, 2008.

[116] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems 35 (NeurIPS 2022)*, pages 24824–24837, 2022. URL https://papers.nips.cc/paper_files/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html.

[117] Jiange Yang, Haoyi Zhu, Yating Wang, Gangshan Wu, Tong He, and Limin Wang. Tra-moe: Learning trajectory prediction model from multiple domains for adaptive policy conditioning. arXiv:2411.14519 [cs.RO], 2025.

[118] Dmytro Kuzmenko and Nadiya Shvai. When robots say no: The empathic ethical disobedience benchmark. In *Proceedings of the 21st ACM/IEEE International Conference on Human-Robot Interaction*, HRI '26, pages 168–176, New York, NY, USA, 2026. Association for Computing Machinery. ISBN 9798400721281. doi: 10.1145/3757279.3785547. URL https://doi.org/10.1145/3757279.3785547.

[119] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety, 2016. URL https://arxiv.org/abs/1606.06565.

[120] Iason Gabriel. Artificial intelligence, values, and alignment. *Minds and Machines*, 30(3):411–437, September 2020. ISSN 1572-8641. doi: 10.1007/s11023-020-09539-2. URL http://dx.doi.org/10.1007/s11023-020-09539-2.

[121] Shengyi Huang and Santiago Ontañón. A closer look at invalid action masking in policy gradient algorithms. In *Proceedings of the 35th International FLAIRS Conference*. University of Florida George A. Smathers Libraries, 2022. doi: 10.32473/flairs.v35i.130584.

[122] Javier García and Fernando Fernández. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16:1437–1480, 2015.

[123] Shangding Gu, Long Yang, Yali Du, Guang Chen, Florian Walter, Jun Wang, and Alois Knoll. A review of safe reinforcement learning: Methods, theory and applications, 2024. URL https://arxiv.org/abs/2205.10330.

[124] Paul Robinette, Ayanna M. Howard, and Alan R. Wagner. Overtrust of robots in emergency evacuation. In *Proceedings of the 11th ACM/IEEE International*

Conference on Human-Robot Interaction (HRI), pages 101–108, 2016. doi: 10.1109/HRI.2016.7451740.

[125] Gordon Briggs and Matthias Scheutz. ”sorry, i can’t do that”: Developing mechanisms to appropriately reject directives in human-robot interactions. In *Proceedings of the AAAI Fall Symposium Series: Artificial Intelligence for Human-Robot Interaction*, pages 32–36, Arlington, VA, 2015. AAAI Press.

[126] William Saunders, Girish Sastry, Andreas Stuhlmüller, and Owain Evans. Trial without error: Towards safe reinforcement learning via human intervention. In *Proceedings of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 2067–2069, Stockholm, Sweden, 2018. International Foundation for Autonomous Agents and Multiagent Systems.

[127] Rosalind W. Picard. *Affective Computing*. MIT Press, Cambridge, MA, 1997. ISBN 0262161702.

[128] Ana Paiva, Iolanda Leite, Hana Boukricha, and Stacy Marsella. Empathy in virtual agents and robots: A survey. *ACM Transactions on Interactive Intelligent Systems*, 7(3):11:1–11:40, 2017. doi: 10.1145/2912150. URL `https://doi.org/10.1145/2912150`.

[129] Jan Leike, Victoria Krakovna, Pedro A. Ortega, Tom Everitt, Andrew Lefrancq, Laurent Orseau, and Shane Legg. Ai safety gridworlds, 2017.

[130] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL `https://arxiv.org/abs/1707.06347`.

[131] David Gunning and David Aha. DARPA’s explainable AI (XAI) program. *AI Magazine*, 40(2):44–58, 2019.

[132] Dylan Hadfield-Menell, Stuart J. Russell, Pieter Abbeel, and Anca Dragan.

Cooperative inverse reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 29, 2016.

[133] Mark Towers, Ariel Kwiatkowski, Jordan Terry, John U Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Markus Krimmel, Arjun KG, et al. Gymnasium: A standard interface for reinforcement learning environments, 2024. URL `https://arxiv.org/abs/2407.17032`.

[134] Ann-Renée Blais and Elke U. Weber. A domain-specific risk-taking (dospert) scale for adult populations. *Judgment and Decision Making*, 1(1):33–47, 2006.

[135] Silvana Bonaccio and Reeshad S. Dalal. Advice taking and decision-making: An integrative literature review, and implications for the organizational sciences. *Organizational Behavior and Human Decision Processes*, 101(2):127–151, 2006. doi: 10.1016/j.obhdp.2006.07.001.

[136] Arie W. Kruglanski and Donna M. Webster. Motivated closing of the mind: ”seizing” and ”freezing”. *Psychological Review*, 103(2):263–283, 1996. doi: 10.1037/0033-295X.103.2.263.

[137] Icek Ajzen. Nature and operation of attitudes. *Annual Review of Psychology*, 52:27–58, 2001. doi: 10.1146/annurev.psych.52.1.27.

[138] Xu Ji, João F Henriques, and Andrea Vedaldi. Invariant information clustering for unsupervised image classification and segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9865–9874, 2019.

[139] Yeow Meng Chee and Hui Zhang. Neural network decoders for permutation codes correcting different errors, 2022.

[140] Alireza Abdollahi, Javad Bagherian, Fatemeh Jafari, Maryam Khatami, Farzad

Parvaresh, and Reza Sobhani. New upper bounds on the size of permutation codes under kendall τ-metric, 2023.

[141] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning, 2013. URL `https://arxiv.org/abs/1312.5602`.

[142] Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, et al. Solving rubik's cube with a robot hand, 2019.

[143] Carlos E Garcia, David M Prett, and Manfred Morari. Model predictive control: Theory and practice—a survey. *Automatica*, 25(3):335–348, 1989.

[144] Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on Robot Learning*, pages 1094–1100. PMLR, 2020.

[145] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[146] Richard S Sutton, David A McAllister, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, volume 12, 2000.

[147] Thanard Kurutach, Ignasi Clavera, Yan Duan, Aviv Tamar, and Pieter Abbeel. Model-ensemble trust-region policy optimization. In *International Conference on Learning Representations*, 2018.

[148] Adam Stooke, Kimin Lee, Pieter Abbeel, and Michael Laskin. Decoupling representation learning from reinforcement learning. In *International Conference on Machine Learning*. PMLR, 2021.

[149] Rich Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997.

[150] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135. PMLR, 2017.

[151] Yu Zhang and Qiang Yang. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*, 34(8):3555–3569, 2021.

[152] Amir Gholami, Sehoon Kim, Zhen Dong, Zhewei Yao, Michael W Mahoney, and Kurt Keutzer. A survey of quantization methods for efficient neural network inference, 2021.

[153] Seunghyun Shin, Donnie H Ko, and Taehoon Kwon. Deep q-learning with quantized neural networks. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, pages 1785–1787, 2019.

[154] Karl Cobbe, Jacob Hilton, Oleg Klimov, and John Schulman. Phasic policy gradient. In *International Conference on Machine Learning*, pages 2020–2027. PMLR, 2021.

[155] Jaekyeom Lee, Yeong-Joon Jang, and Kee-Eung Cho. Multimodal reinforcement learning: A survey and taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[156] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems, 2020.

[157] Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, Pieter Abbeel, and Woj-

ciech Zaremba. Hindsight experience replay. *Advances in neural information processing systems*, 30, 2017.

[158] Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 34:1179–1191, 2021.

[159] Simon Schmitt, Jonathan J Hudson, Augustin Zidek, Simon Osindero, Carl Doersch, Wojciech M Czarnecki, Joel Z Leibo, Heinrich Küttler, Andrew Zisserman, Karen Simonyan, et al. Kickstarting deep reinforcement learning, 2018.

[160] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization, 2016.

[161] Diganta Misra. Mish: A self regularized non-monotonic neural activation function, 2019.

[162] Richard S Sutton. Learning to predict by the methods of temporal differences. *Machine learning*, 3(1):9–44, 1988.

[163] Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J. Davison. RLBench: The Robot Learning Benchmark & Learning Environment. arXiv:1909.12271 [cs.RO], 2019.

[164] Stone Tao, Fanbo Xiang, Arth Shukla, et al. Maniskill3: Gpu parallelized robotics simulation and rendering for generalizable embodied ai. arXiv:2410.00425 [cs.RO], 2024.

[165] Yifeng Zhu, Abhishek Joshi, Peter Stone, and Yuke Zhu. Viola: Imitation learning for vision-based manipulation with object proposal priors. arXiv:2210.11339 [cs.RO], 2023.

[166] Sarah Young, Dhiraj Gandhi, Shubham Tulsiani, et al. Visual imitation made easy. arXiv:2008.04899 [cs.RO], 2020.

[167] Weilin Cai, Juyong Jiang, Fan Wang, et al. A survey on mixture of experts in large language models. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–20, 2025. doi: 10.1109/TKDE.2025.3554028.

[168] Runhan Huang, Shaoting Zhu, Yilun Du, and Hang Zhao. Moe-loco: Mixture of experts for multitask locomotion. arXiv:2503.08564 [cs.RO], 2025.

[169] Wenxuan Song, Han Zhao, Pengxiang Ding, et al. Germ: A generalist robotic model with mixture-of-experts for quadruped robot. arXiv:2403.13358 [cs.RO], 2024.

[170] Han Zhao, Wenxuan Song, Donglin Wang, et al. More: Unlocking scalability in reinforcement learning for quadruped vision-language-action models. arXiv:2503.08007 [cs.RO], 2025.

[171] Ziye Huang, Haoqi Yuan, Yuhui Fu, and Zongqing Lu. Efficient residual learning with mixture-of-experts for universal dexterous grasping. arXiv:2410.02475 [cs.RO], 2024.

[172] Lucas Beyer, Andreas Steiner, André Susano Pinto, et al. Paligemma: A versatile 3b vlm for transfer. arXiv:2407.07726 [cs.CV], 2024.

[173] Peng Wang, Shuai Bai, Sinan Tan, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. arXiv:2409.12191 [cs.CV], 2024.

[174] Kristen Grauman, Andrew Westbury, Eugene Byrne, et al. Ego4d: Around the world in 3,000 hours of egocentric video. arXiv:2110.07058 [cs.CV], 2022.

[175] Adam Paszke, Sam Gross, Francisco Massa, et al. Pytorch: An imperative style,

high-performance deep learning library. In *Advances in Neural Information Processing Systems*, 2019.

[176] James Bradbury, Roy Frostig, Peter Hawkins, et al. JAX: Composable transformations of Python+NumPy programs. `https://github.com/google/jax`, 2018.

[177] Chuan Wen, Xingyu Lin, John So, Kai Chen, Qi Dou, Yang Gao, et al. Any-point trajectory modeling for policy learning. arXiv:2401.00025 [cs.RO], 2024.

[178] Maartje M. A. De Graaf, Somaya B. Allouch, and Jan A. G. M. Van Dijk. Why do they refuse to use my robot? reasons for non-use derived from a long-term home study. In *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 224–233, 2017. doi: 10.1145/2909824.3020236.

[179] Kristin E. Schaefer. Measuring trust in human robot interactions: Development of the ”trust perception scale-hri”. In F. Jentsch and P. Hancock, editors, *Robust Intelligence and Trust in Autonomous Systems*, pages 191–218. Springer, Boston, MA, 2016. doi: 10.1007/978-1-4899-7668-0_10.

[180] Tathagata Chakraborti, Sarath Sreedharan, Sachin Grover, and Subbarao Kambhampati. Plan explanations as model reconciliation – an empirical study, 2018. URL `https://arxiv.org/abs/1802.01013`.

[181] Dylan Hadfield-Menell, Anca Dragan, Pieter Abbeel, and Stuart Russell. The off-switch game. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, IJCAI’17, pages 220–227. AAAI Press, 2017. ISBN 9780999241103. URL `https://dl.acm.org/doi/10.5555/3171642.3171675`.

[182] Smitha Milli, Dylan Hadfield-Menell, Anca D. Dragan, and Stuart J. Russell. Should robots be obedient? In *Proceedings of the 16th International Conference*

on Autonomous Agents and Multiagent Systems, pages 499–507. International Foundation for Autonomous Agents and Multiagent Systems, 2017.

[183] Dorsa Sadigh, Anca D. Dragan, Shankar S. Sastry, and Sanjit A. Seshia. Active preference-based learning of reward functions. In *Proceedings of Robotics: Science and Systems (RSS)*, 2017.

[184] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1321–1330. PMLR, 2017.

[185] Volodymyr Kuleshov, Nathan Fenner, and Stefano Ermon. Accurate uncertainties for deep learning using calibrated regression. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2796–2804. PMLR, 2018.

[186] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[187] Franziska Babel, Philipp Hock, Johannes Kraus, and Martin Baumann. It will not take long! longitudinal effects of robot conflict resolution strategies on compliance, acceptance and trust. In *Proceedings of the 2022 ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 225–235, 2022. doi: 10.1109/HRI53351.2022.9889492.

[188] C. Esterwood and Lionel P. Robert. Repairing trust in robots?: A meta-analysis of hri trust repair studies with a no-repair condition. In *Proceedings of the 20th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 410–419, Melbourne, Australia, 2025. IEEE. doi: 10.1109/HRI61500.2025.10974168.

[189] Ning Wang, David V. Pynadath, and Susan G. Hill. Trust calibration within a human-robot team: Comparing automatically generated explanations. In *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 109–116, 2016. doi: 10.1109/HRI.2016.7451741.

[190] Dmytro Kuzmenko and Nadiya Shvai. Balancing performance and efficiency in zero-shot robotic navigation. In Vadim Ermolayev, Igor Potapov, Oleksii Ignatenko, Roman Hornung, Andrii Hlybovets, Vitaliy Yakovyna, Yaroslav Prytula, and Oleksandr Spivakovsky, editors, *Information and Communication Technologies in Education, Research, and Industrial Applications*, pages 370–381, Cham, 2025. Springer Nature Switzerland.

[191] Dmytro Kuzmenko and Nadiya Shvai. Knowledge transfer in model-based reinforcement learning agents for efficient multi-task learning. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '25, pages 2597–2599, Richland, SC, 2025. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9798400714269.

[192] Dmytro Kuzmenko, Oleksii Tsepa, Andrew Garrett Kurbis, Alex Mihailidis, and Brokoslaw Laschowski. Efficient visual perception of human-robot walking environments using semi-supervised learning. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6544–6549, 2023. doi: 10.1109/IROS55552.2023.10341654.

APPENDICES

APPENDIX A

LIST OF THE CANDIDATE'S PUBLICATIONS

List of scientific publications in which the primary research results of the dissertation are published:

[1] Kurbis, A. G., Kuzmenko, D., Ivanyuk-Skulskiy, B., Mihailidis, A., Laschowski, B. (2024). StairNet: visual recognition of stairs for human-robot locomotion. *BioMedical Engineering OnLine*, 23(1), 20. https://doi.org/10.1186/s12938-024-01216-0.

[2] Beimuk, V., Kuzmenko, D. (2025). Energy conservation for autonomous agents using reinforcement learning. *NaUKMA Scientific Journal. Computer Science Series*, 8, 68–75. https://doi.org/10.18523/2617-3808.2025.8.68-75.

[3] Kuzmenko, D., Shvai, N. (2026). MoIRA: Modular instruction routing architecture for multi-task robotics. *Neurocomputing*, 674, 132962. https://doi.org/10.1016/j.neucom.2026.132962.

List of scientific publications confirming dissemination of the dissertation findings:

[4] Kuzmenko, D., Tsepa, O., Kurbis, A. G., Mihailidis, A., Laschowski, B. (2023). Efficient visual perception of human-robot walking environments using semi-supervised learning. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2023)*, pp. 6544–6549. https://doi.org/10.1109/IROS55552.2023.10341654.

[5] Kuzmenko, D., Shvai, N. (2025). Balancing performance and efficiency in zero-shot robotic navigation. In: Ermolayev, V. et al. (eds.) *ICTERI 2024*,

CCIS vol. 2359, pp. 370–381. Springer, Cham. https://doi.org/10.1007/978-3-031-81372-6_28.

[6] Kuzmenko, D., Shvai, N. (2025). Knowledge transfer in model-based reinforcement learning agents for efficient multi-task learning. In: *AAMAS 2025 Proceedings*, pp. 2597–2599. https://doi.org/10.5555/3709347.3743949.

[7] Kuzmenko, D., Shvai, N. (2026). When Robots Say No: The Empathic Ethical Disobedience Benchmark. In: *ACM/IEEE International Conference on Human-Robot Interaction (HRI 2026) Proceedings*, pp. 168–176, Edinburgh, Scotland, UK, March 16–18, 2026. https://doi.org/10.1145/3757279.3785547.

Документ підписано у сервісі Вчасно (продовження)
Kuzmenko_dissertation_2026_final_PDFA.pdf

Документ відправлено (16459396): 17:23 25.06.2026
Документ отримано (16459396): 17:23 25.06.2026

Відправник документу

Електронний підпис
17:23 25.06.2026
Ідентифікаційний код: 3648602877
КУЗЬМЕНКО ДМИТРО ОЛЕКСАНДРОВИЧ
Власник ключа: КУЗЬМЕНКО ДМИТРО ОЛЕКСАНДРОВИЧ
Час перевірки КЕП/ЕЦП: 17:23 25.06.2026
Статус перевірки сертифікату: Сертифікат діє
Серійний номер: 5E984D526F82F38F040000004FE4A60153AD6C07
Тип підпису: удосконалений
Тип сертифікату: кваліфікований