Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Leveraging Emerging Physical Knowledge in LLMs Toward World-Aware Robotics

Master Thesis

Constantin Riff

MSc in Mechanical Engineering

Spring Semester 2025

Supervisors: Dr. Francesco Stella, Prof. Josie Hughes, Prof. Marco Hutter

Computational Robot Design & Fabrication Lab, EPFL

Leveraging Emerging Physical Knowledge in LLMs Toward World-Aware Robotics

CREATE Lab, EPFL

Constantin Riff

Abstract

In recent years, the use of Large Language Models (LLMs) has increased significantly thanks to their ability to understand a wide range of knowledge, making them highly generalisable. Their vast knowledge can be leveraged in order to predict physical phenomena without relying on traditional physics-based simulators. This approach is particularly interesting in domains where traditional simulators struggle to produce reliable results. The objective of this work is therefore to explore how the use of LLMs can overcome the current limitations of other available models in order to pave the way for new methods of physical prediction in the field of robotics. This thesis first reviews the three established paradigms for physics simulation, namely physics-based, hybrid, and foundation model, outlining their respective strengths but also their limitations in terms of the costs of computation and time, and in terms of scalability. In order to assess whether LLMs can help overcome such barriers, a framework is developed to systematically probe their capacity on selected domains of physics. A series of controlled experiments enables a performance analysis to explore physical prediction capabilities and persistent shortcomings. The findings of this work provide the potential to improve robotic reasoning with LLM-based knowledge. Moreover, the developed framework provides a foundation for deeper investigation into how language models internalize and apply physical principles, enabling non-proficient users to deploy robotic systems with minimal set-up and time on a large range of applications.

1 Introduction

Humans have always sought to understand the world around them. This learning first came through direct confrontation between humans and the physical world. Gradually, this heuristic learning was complemented by a theoretical understanding of the laws of physics, transmitted from generation to generation. Today, this knowledge is written and detailed in thousands of books and articles that have become the foundation of Large Language Models (LLMs) [1][2].

LLMs are neural networks trained on an enormous collection of data, providing them with general knowledge and enabling them to generate responses in textual format. This general knowledge opens significant potential for robotics by allowing robots to become aware of their surrounding physical environment.

Currently, other methods are used to simulate the physical world but they encounter limitations in online robotics applications. The first established paradigm is the physics-based model. It relies on mathematical representations of physical laws in an open-loop manner, without learning from data. Under well-defined conditions, these methods can capture fine-grained phenomena: high-resolution Finite Element Method (FEM) can reveal detailed stress distributions in deformable objects [3], and Finite Volume Method-based (FVM) Computational Fluid Dynamics (CFD) can enforce conservation laws to predict complex flow patterns with reasonable fidelity [4]. In tightly controlled scenarios, their predictions closely match real-world behaviour, providing a reliable reference. However, their limitations are significant in robotics. They are computationally intensive and often impractical for real-time use [5]. Complex scenarios (like turbulence or phase changes) require extremely fine meshes and small time-steps, causing long solve times. Each method also has reliability issues in

mixed domains [6] and the set-up complexity is substantial [4].

Hybrid models seek to combine the fidelity of physics-based solvers with enhanced computational efficiency. A prominent example is Physics-Informed Neural Networks (PINNs), which embed governing equations (e.g. differential equation residuals or conservation laws) into the loss function of a neural network [7]. The network acts as a surrogate solver, forced to satisfy known physics during training. Once trained, PINNs can produce continuous solutions which are consistent with the physics, and much faster at runtime than classical solvers. Another strategy is to build Reduced-Order Models (ROMs): the principle is to take a high-dimensional physics system and project it into a low-dimensional space that captures the dominant dynamics. This typically involves collecting solution snapshots, constructing a reduced basis (e.g., via Proper Orthogonal Decomposition), and projecting the governing equations onto this basis. The resulting reduced system enables fast simulations while preserving the essential features of the original model [8].

However, these methods present limitations. PINNs can be prone to poor convergence and the solutions are not robust outside the data envelope [9][10]. ROMs are limited by their dependence on representative training data and poor extrapolation capability [11]. More broadly, hybrid models lack generality—models trained for specific tasks cannot easily transfer to new settings without costly retraining or resetting.

The third paradigm leverages large-scale foundation models, such as Large Language Models and Vision-Language Models (VLMs), to provide robots with physical-world knowledge. Instead of programming physics explicitly or training task-specific models, these approaches exploit pretrained models' internal knowledge as a shortcut to common-sense reasoning. Modern foundation models encode factual and causal information enabling predictions of physical outcomes without formal simulation [12].

Key strengths of LLMs include generality and adaptability. Indeed, a single pretrained model can be applied to many different tasks and domains with minimal or no fine-tuning. Besides the capabilities for physical reasoning tasks, the internal knowledge enables semantic and affordance understanding, which is particularly valuable for grasping tasks.

Despite increasing interest in using large language models for physical reasoning, there is limited quantitative evidence of their ability to make forward predictions for physics tasks relevant to online robotics, without access to domain-specific simulators or task-specific training. During the thesis, numerous experiments put LLM capabilities to the test on different physics domains.

Therefore, this work aims to systematically evaluate when and how an off-the-shelf LLM can predict physical outcomes. Concretely, we study whether LLMs can (i) produce quantitative predictions for simple classical mechanics scenarios and (ii) obtain sufficient accuracy across distinct physics domains.

We implemented a closed-loop framework that turns LLM outputs into discrete subactions for a robot arm to complete tasks from raw images. The framework requires no complex installation or set-up, making it simple to reproduce and adaptable for future studies exploring LLM prediction capabilities in other domains. Then, we designed a set of experiments spanning classical mechanics and simplified fluid dynamics.

This work has demonstrated the capabilities of LLMs in domains of classical mechanics such as
(a) predicting the maximum height after the first bounce of a ball and
(b) assessing block-stacking stability.
It also extends to fluid dynamics, enabling
(c) the control of flow behaviour across different viscosities.
Finally, it characterizes the ability of LLMs to address tasks that enhance specific physical challenges, thereby
(d) providing a benchmark within each tested domain.

We report task-level success rates and quantitative results relative to the ground truth, and we analyse failure modes and conditions for success.

In the following section, we describe the framework and methodology developed to evaluate the physics prediction capabilities of LLMs.

2 Methods

2.1 Set-up

In order to connect the LLM's understanding to the physical world, we developed a framework to link

the LLM to a robot arm. The robot arm used is a Universal Robots UR3, with a 3D-printed gripper at its end actuated by a Dynamixel XL430-W250-T servo motor. The gripper is connected to a monitoring computer via a U2D2 Power Hub Board. A standard video camera is used to capture images of the scene and various objects have been used throughout the experiments.

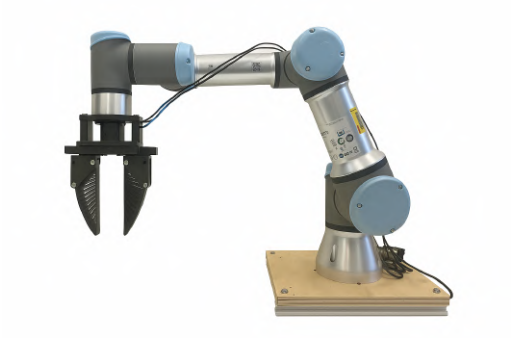


Figure 1: Experimental set-up: Universal Robots UR3 robotic arm equipped with a gripper used to perform tasks

In the experiments presented, the LLM employed is **GPT-5**, accessed via its API. It includes the parameter `reasoning.effort` which can be specified as *minimal*, *low*, *medium*, or *high*. We observed no notable differences between the *low*, *medium*, and *high* settings, whereas the *minimal* setting consistently led to worse results. Hence, for all the experiments, this parameter was set to *low*.

2.2 Framework

During the course of this thesis, the framework was extended with new functionalities, leading to improved predictive performance of the LLM. The main components are presented in the following sections.

2.2.1 Core Framework

We evaluate the physical capabilities of LLMs by controlling a UR3 manipulator through discrete, low-level actions derived from camera observations. The system comprises three main modules, `Task_Router`, `Task_Detector`, and `Executor`, that interact with the LLM through an API.

Perception and task identification:
At the start of each trial, the `Task_Router` acquires an RGB image of the workspace and forwards it to the `Task_Detector`. The `Task_Detector` first queries the LLM with this image and a set of reference scene descriptions predefined in the prompt. The LLM performs a matching step:

If the current scene matches one of the predefined cases, the LLM returns the corresponding task label. If no match is found, the LLM returns `Unknown`.

Prompt selection and generation:
When a match is returned, the `Task_Detector` selects a prewritten execution prompt corresponding to that task and passes it to the `Executor`. Each prewritten prompt specifies (i) the task goal and (ii) the discrete set of subactions available to the LLM at each iteration. These subactions are elementary moves that the robot can undertake in order to perform the task (e.g., `Up_1 cm`, `Down_1 cm`, `Left_1 cm`, `Right_1 cm`, `Tilt_3°`, `OpenGrip`, `CloseGrip`...).

If the LLM replies `Unknown`, the `Task_Detector` issues a second query consisting of the current image and a generator prompt. The generator prompt asks the LLM to synthesize a task-specific execution prompt tailored to the observed scene. The generated prompt must explicitly state the task goal and list the same fixed library of subactions in order to achieve the task. The `Task_Detector` then forwards this generated prompt to the `Executor`.

Closed-loop execution:
When receiving either a prewritten or a generated prompt, the `Executor` enters a while-loop that repeatedly:

- sends the current image and the execution prompt to the LLM;
- receives the corresponding subaction response from the LLM;
- converts the chosen subaction into low-level commands via the `ur_rtde` library and executes them on the UR3;
- captures a new image and continues.

The loop terminates when the LLM returns `Success`, at which point the `Executor` closes logs and subprocesses and commands the robot to return to its home configuration.

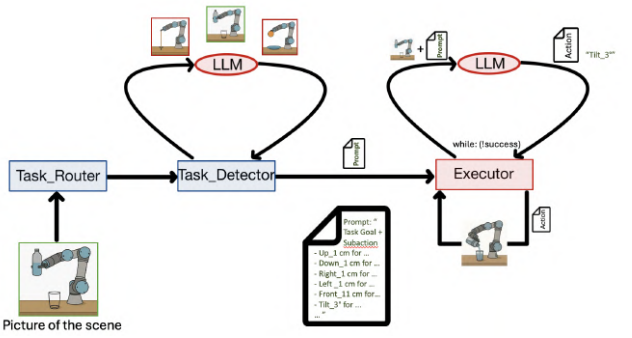


Figure 2: Framework in the matched-task case: The `Task_Detector` matches the scene to a predefined task and sends the corresponding prewritten prompt to the `Executor`. The `Executor` then iteratively queries the LLM for low-level actions, executes them on the UR3 robot, and stops when the LLM signals `Success`.

The choice of two paths, using either a prewritten or a generated prompt, enables rigorous experimental control and targeted generalisation. First, the prewritten path makes it possible to hard-code the instruction that enters the executor loop; repeated runs of the same experiment therefore use an identical prompt and action set, maximizing repeatability. Moreover, the prewritten path provides the option to hard-code the task goal and its decomposition when needed, meaning that we are not forced to rely on the LLM's own interpretation of what the task is and how it should be achieved. Second, the prompt-generation module confers scalability: it enables the automatic creation of task-specific execution prompts directly from the observed scene, allowing the framework to generalise to a wide variety of manipulation tasks without a tedious, task-by-task prompt authoring.

2.2.2 Bringing Memory

The LLM used with the API is stateless: each API call is conditionally independent of the past so the model has no built-in access to previous observations or actions. In manipulation, this can degrade logical continuity (e.g., oscillatory or contradictory commands). To mitigate this limitation, we therefore introduce two complementary memory mechanisms.

First, the **Short-Term Memory** (STM) module is introduced. Specifically, instead of providing only the current frame and the execution prompt, we also append the last n frames along with the corresponding subactions previously selected by the LLM. This sliding-window context provides the model with temporal coherence and a notion of progression within the task. For all the experiments, the short-term memory sliding window size was fixed to n = 5. Therefore, the input at each iteration becomes

$$\mathcal{C}_t = \{(I_t, \text{prompt}), (I_{t-1}, a_{t-1}), \ldots, (I_{t-n}, a_{t-n})\}.$$

where:

t	current timestep
I_t	frame at timestep t
prompt	execution prompt provided to the LLM
a_t	subaction decided by the LLM at timestep t
n	size of the sliding window in short-term memory
$\mathcal{C}_t$	input query to the LLM API at timestep t

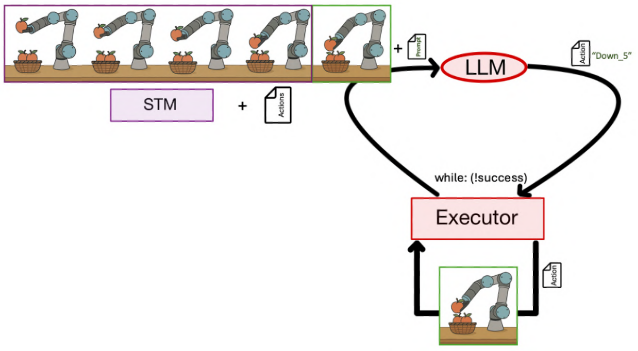


Figure 3: Short-term memory (STM) provides the LLM with the last n frame–action pairs in addition to the current frame, ensuring temporal continuity during execution.

In addition, a **Long-Term Memory** (LTM) mechanism has been introduced to provide the LLM with a contextual history of the task from its beginning. The LTM is stored as a text file that describes the progress of the task and is continuously updated. To achieve this, the LTM loop is triggered every n iterations (n corresponding to the size of the STM sliding window). At each trigger, the LLM receives as input a bundle of contextual information composed of the current frame, the STM, the previous LTM file (if it exists), and the execution prompt. Based on this input, the LLM condenses the new information with the previous LTM file and generates an updated version. This updated LTM file is then included in every iteration of the while-loop, providing the LLM with a higher-level contextual overview to guide subaction selection.

2.2.3 Redundant Safety and Robustness Mechanism

Finally, we introduce a `Corrector` Unit as an additional safety mechanism. This module can be triggered when certain conditions are met, such as the detection of oscillatory movements or when the number of iterations exceeds a predefined threshold. The `Corrector` Unit receives as input the full contextual information available: the short-term memory, the current frame, the execution prompt, and the log file containing all past states of the LTM. Based on this input, it can intervene in several ways to prevent undesirable situations. For instance, it may refine the execution prompt to make it more precise with the actual subtask undertaken (especially if the goal is almost reached), provide a short sequence of corrective actions to be executed by the `Executor`, or abort the task entirely if it considers the task unrecoverable. The `Corrector` Unit therefore adds an additional layer of safety: while the `Executor` already has built-in mechanisms to escape from such situations, the `Corrector` provides a redundant and more fine-grained safeguard.

In addition to the core components, several auxiliary modules complement the framework to facilitate monitoring, traceability, and post-hoc analysis. These include (i) video recording of the entire scene, (ii) systematic logging of each frame provided to the LLM together with the corresponding predicted action and the model's justification, (iii) storage of all log files, including the successive states of the LTM, and (iv) visualisation panels that display the current state of the LTM. Together, these modules provide a comprehensive record of the experiments, enabling fine-grained analysis of the LLM's decision-making process and improving the transparency of the overall framework.

To systematically assess the effectiveness of this framework, we next introduce our evaluation metrics, designed to quantify the ability of the LLM to successfully achieve physical tasks.

2.3 Evaluation Metrics

In order to evaluate the ability of the LLM to perform the task while accounting for the underlying physics, a straightforward evaluation system was implemented. Each specific task is independently assessed ten times under identical conditions. For each run, it is determined whether the task has been successfully completed or not. The success criterion is task-dependent and predefined. Based on the outcomes of the ten runs, a success rate is calculated, providing an assessment of the LLM's capability to accomplish the task.

3 Experiments

In order to test the physical prediction capabilities of LLMs, various experiments were conducted to evaluate specific domains of physics. The selection of these experiments was primarily guided by the particular area of physics they highlight. Furthermore, the experiments were designed to be carried out in a simple and standardised laboratory environment, facilitating high reproducibility.

The different experiments are divided into two main categories. On the one hand, some require an action in order to demonstrate the LLM's understanding of the physical scene. These experiments therefore rely on the previously presented framework, using the UR3 robotic arm to perform the task. On the other hand, some experiments do not require any action and instead put the LLM's physical prediction capabilities to the test through specific queries, involving either real-domain or synthetic-domain images. For these action-free experiments, the expected outcomes can be either qualitative or quantitative.

In this manuscript, one experiment from each category will be presented in detail.

3.1 Structural Stability

First, the LLM's capabilities in classical mechanics are evaluated. The LLM is provided with an image of a stacked-block configuration. For each configuration, the LLM must determine whether it is stable or not according to the laws of physics. This task primarily probes rigid-body statics under gravity, including centre-of-mass reasoning, contact modeling, and Coulomb friction limits.

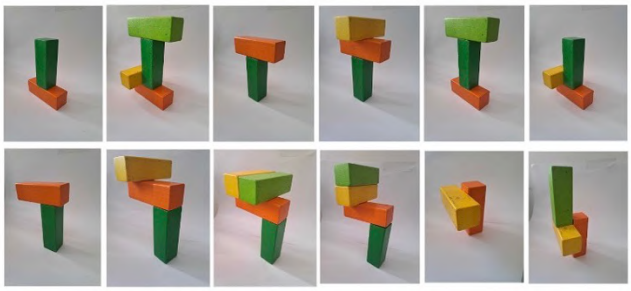


Figure 4: Set of block configurations individually presented to the LLM. The top six images show stable configurations, while the bottom six show unstable ones.

3.2 Height of Bouncing Ball

Next, the LLM's predictive capabilities in mechanics and its internal reasoning are analysed through an experiment in which the model is provided with an image of a ball held at a known height by the robotic arm. The LLM must then predict, based on what it observes in the image, the maximum height reached by the ball after the first bounce. This height is measured from the rebound surface to the bottom of the ball. This experiment probes classical mechanics, specifically impact dynamics and energy dissipation.

For each query, the LLM is required to provide a justification, and the `reasoning.summary` parameter is set to auto, which enables access to the recorded history of the LLM's chain-of-thought process.

3.3 Filling a Glass

The third experiment aims to demonstrate the LLM's predictive capabilities for tasks related to fluid dynamics. To this end, a bottle containing liquid can be tilted by rotating the robot's sixth joint, thereby pouring the liquid at a controlled rate. The subactions available to the LLM for regulating the flow consist of tilting the bottle downward or upward by 3°, tilting it upward by 10°, or taking no action to maintain the current configuration.

The container into which the liquid is poured varies among a standard glass, a stemmed glass, a glass jar, and a beer glass. The viscosity of the liquids also differs. At an average temperature of 24 °C, the tested liquids are: water (0.001 Pa·s), dish soap (1.5 Pa·s), cooking olive oil (0.085 Pa·s), and standard commercial liquid honey (10 Pa·s). These viscosity values, as well as the identity of the liquids, are not communicated to the LLM.

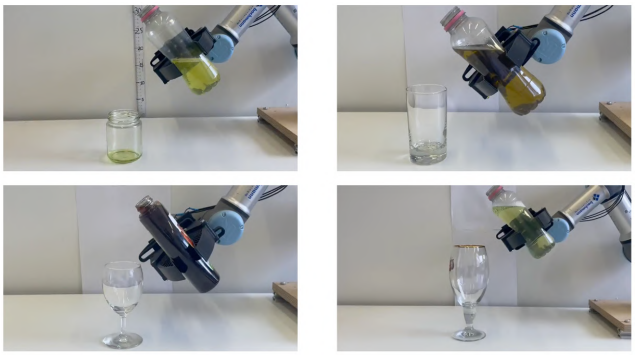


Figure 5: Runs of the filling task for four container geometries (glass jar, standard glass, stemmed glass, beer glass) and liquids spanning a wide range of viscosities.

The execution prompt specifies to the LLM that the mission is to fill the glass without exceeding its level. No other information about the scene is provided. The task is considered successful if the container is filled with liquid to at least three-quarters of its volume and no overflow or spillage occurs.

3.4 Additional Experiments

Additional experiments were conducted to investigate other physical domains such as:

- **Elasticity**, by stretching a rubber band to its maximum length without breaking, thereby probing the elastic limit.
- **Thermodynamics and heat transfer**, by heating a piece of chocolate with a candle to melt it without burning, thus involving conduction, phase change, and temperature control.
- **Rigid-body mechanics and object placement**, by inserting a pen into a holder, placing an orange on a plate, putting a peach into a fruit basket, or dropping a chocolate-bar wrapper into a small trash bin, all requiring stability assessment and spatial placement under gravity.
- **Fluid mechanics and capillarity**, by applying a sponge to absorb a water spill, involving porous absorption and liquid–solid interactions.

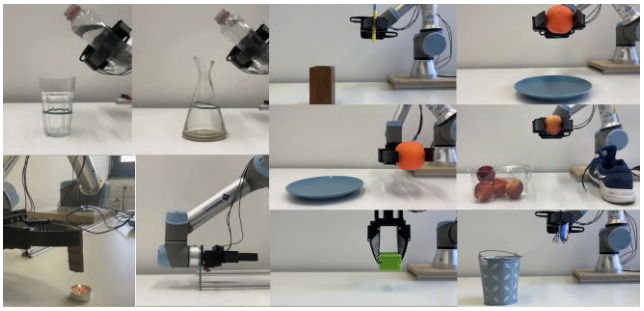


Figure 6: Examples of physical prediction tasks performed with the UR3 manipulator, including object placement (pen in holder, orange on plate, peach in fruit basket, wrapper in trash bin), thermodynamics (melting chocolate with a candle), and fluid mechanics (pouring water and absorbing a spill with a sponge).

4 Results

The results of the experiments are presented in the current section.

4.1 Structural Stability Result

The outcomes for each configuration are detailed below. For the stable configurations shown in the top row, a success rate of 1 indicates that the LLM classified the configuration as stable for all ten queries. Similarly, for the unstable configurations in the bottom row, a score of 1 indicates that the LLM classified the structure as unstable for all ten queries.

The configurations are numbered from 1 to 12 in Figure 4, left to right and top to bottom, with stable configurations numbered 1–6 and unstable configurations numbered 7–12. The results are presented in the table below.

Table 1: Success rate per configuration: **1–6** stable, **7–12** unstable.

Configuration	**1**	**2**	**3**	**4**	**5**	**6**
Success Rate	1	0.1	1	0.8	1	0.4
Configuration	**7**	**8**	**9**	**10**	**11**	**12**
Success Rate	1	0.1	0.8	0.3	1	0.9

The LLM achieved high accuracy on stable configurations, with the exception of cases 2 and 6, where misinterpretation of the yellow block (treated as a cube rather than a parallelepiped, and perceived as unsupported) led to errors. For unstable configurations, performance was also strong except for cases 8 and 10, where failures arose from perspective-related ambiguities in transitioning from 2D to 3D. Notably, configuration 8 without the yellow block (structurally identical to configuration 7 but rendered from configuration 8's viewpoint) achieved 0.3 (vs. 1.0 for configuration 7), solely due to a slightly different viewing angle. The failure cases therefore arose from a misinterpretation of the visual scene rather than from an incorrect understanding of the underlying physics.

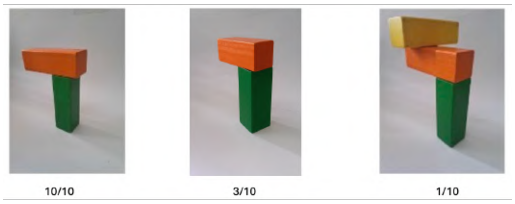


Figure 7: Impact of perspective on LLM predictions: the same block configuration achieves a perfect score when viewed frontally (left, configuration 7, 10/10), but lower scores when viewed from a three-quarter angle (middle, 3/10; right, configuration 8, 1/10).

4.2 Bouncing Ball Result

The LLM's outputs for each request regarding the maximum height of the ball after the first bounce are presented below.

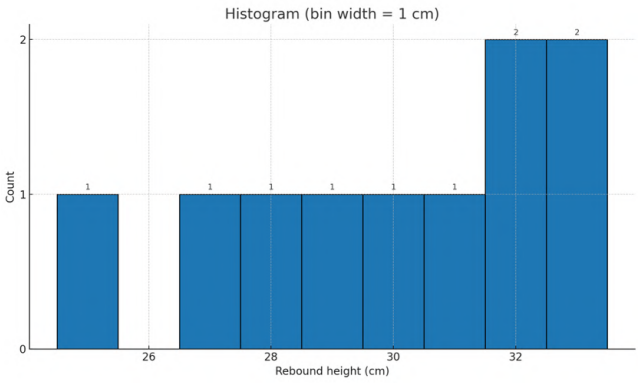
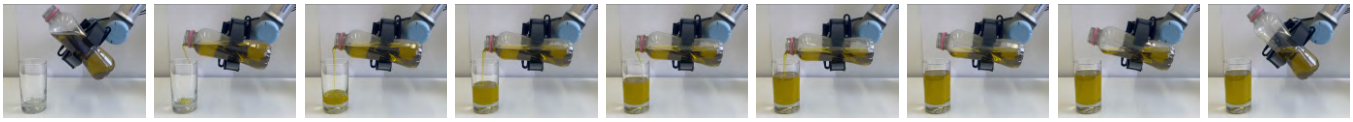


Figure 8: Histogram showing the 10 predicted values by the LLM about the maximum height of the ball after the first bounce

The data obtained from the chain-of-thought history and the request justifications make it possible to trace the LLM's physical reasoning process and to understand how its knowledge is applied. First, the LLM identifies the type of ball involved (a tennis ball in the presented experiment). It then relies on its internal physical knowledge to relate the initial drop height to the rebound height. For this purpose, it applies the formula $e^2 = \frac{h_1}{h_0}$, which links the two heights through the coefficient of restitution. Finally, it uses its raw internal knowledge to

(a) Honey (success)



(b) Oil (success)

(c) Tennis ball drop(success)

(d) Plum drop (failure) due to semantic bias

Figure 9: Sequences of nine consecutive frames for different tasks. Honey, Oil pouring and Tennis ball dropping show successful outcomes, while Plum dropping leads to failure.

determine the appropriate value of the coefficient of restitution for the observed ball.

Through this reasoning process, the LLM produces results that closely match the ground truth, with a mean prediction of 29.95 cm over 10 trials compared to the measured value of 31 cm. The outcomes are consistent, as indicated by a small sample standard deviation (2.73 cm), with a median of 30.50 cm, first quartile (Q1) at 28.13 cm, and third quartile (Q3) at 32.00 cm.

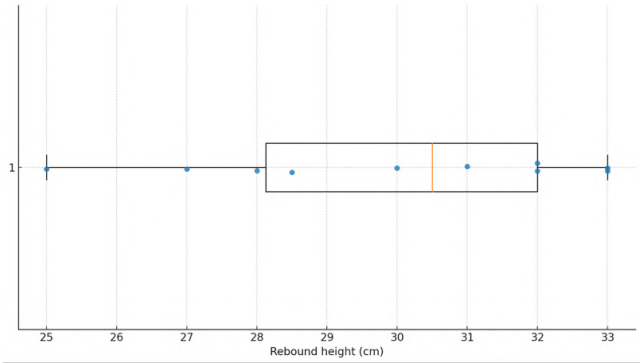


Figure 10: Box-and-whisker plot of the results of the maximum height of the ball after the first bounce.

4.3 Filling a Glass Result

We obtain a high score on liquid pouring tasks, with an overall performance of **0.84** across different liquids and receptacles. Detailed results for each specific liquid–receptacle combination are provided in Table 3. The LLM achieves consistently high results despite variations in viscosity and container type, demonstrating its ability to adapt the inclination angle to maintain controlled flow without any prior task-specific training.

In the initial experiments, the LLM sometimes considered the task successful before it was actually completed. At that stage, the Short-Term Memory (STM) module had not yet been integrated. This premature success detection was caused by a visual misinterpretation of the water surface line: as shown in Figure 13, the shape of the glass and the background created a misleading horizontal line. This issue was resolved with the implementation of STM, which enabled the LLM to maintain logical continuity of the scene and thus achieve better results.

4.4 Additional Experiments Results

The table below summarizes the results of the additional experiments.

Task	Success condition	Score (n = 10)
Fill glass to the limit	± 15 mm	0.5
Fill carafe to the limit	± 15 mm	0.4
Fill glass	At least 3/4 filled and no spillage	0.9
Melt chocolate	Last 2 min without dripping and continuously above 25 °C	0.3
Stretch elastic	Reach the lower boundary of the confidence interval (35 cm)	0.1
Drop orange (vertical)	Onto the plate	0.9
Drop orange (horizontal)	Onto the plate	0.7
Drop peach (right receptacle)	Into the basket	0.6
Drop pen	Into the receptacle	1.0
Drop trash	Into the bin	0.9
Wipe table	Absorb water over at least the area of the sponge	0.0

Table 2: Tasks, success conditions, and scores (n = 10).

Some experiments yield low scores and highlight the current limitations of LLMs, revealing common failure modes.

Placing a plum in a box: Errors are induced by semantic bias. As shown in Figure 9, while the LLM has no difficulty dropping a ball into the box, when asked to place a fruit, it tends to deposit it more gently, causing frequent collapse of the box and overall task failure.

Wiping a water spill: This task was never completed successfully. The error originates from computer vision: when the sponge is near the spill, the LLM prematurely infers contact and triggers the success condition, even though absorption has not occurred.

Melting chocolate without dripping: Failures primarily arise from the experimental set-up itself. Once the chocolate begins to drip, the flow cannot be stopped—even by timely corrective action—due to the inherent thermal inertia of the melting process. As a result, the LLM may react appropriately but still fail to prevent dripping, which lowers the success rate.

Stretching an elastic band: The success criterion was defined as achieving a minimum stretch threshold without causing rupture of the band. A systematic scale bias was observed: the input image format had a strong influence on the LLM's reasoning. With portrait images, the robot terminated the task after approximately 22 iterations, whereas with landscape images (where the band appears smaller relative to the overall frame), it continued up to about 37 iterations on average. This indicates that the perceived image scale significantly affects the LLM's decision-making process.

5 Discussion and Conclusion

A single off-the-shelf LLM can produce useful physical predictions and control decisions across distinct domains (rigid-body stability, impact/rebound, container filling) without a dedicated simulator. For action-tasks, performance arises from physical commonsense priors and incremental action with visual feedback rather than precise metric estimation.

LLMs exhibit latent physical priors that are exploitable when scenes are unambiguous; most errors come from visual misinterpretation, not missing physical laws. The principal failure modes are: (i) perspective/scale ambiguity that yields inconsistent stability judgments or stopping criteria; (ii) semantic bias from task phrasing; and (iii) visual artefacts (reflections, background lines) misread as physical cues.

This thesis reviewed contemporary physics-simulation methods relevant to embodied intelligence. Building on this review, we introduced a framework that exposes and tests the physics knowledge of a large language model through a perception–action loop, applicable across a broad range of tasks without prior task-specific preparation. The framework offers a reproducible protocol and transparent logging.

Promising results were obtained in specific domains such as classical mechanics, for instance the accurate prediction of a ball's height after its first bounce or the assessment of structural stability. Strong performance was also observed in fluid dynamics tasks where the LLM successfully filled containers of varying geometry and viscosity while controlling flow. More broadly, this work highlights the model's capacity to handle other tasks requiring physical scene understanding.

At the same time, it revealed current limitations of LLMs, largely arising from visual misinterpretations that impact task performance. Additional issues were identified in semantic understanding as well as from artefacts induced by input image scaling.

Overall, the framework paves the way for broader use cases and deeper research on LLM physical reasoning, leveraging its logging system to guide LLM-driven actions while lowering the barrier for non-expert users to deploy robotic systems efficiently with minimal set-up across diverse applications.

References

[1] B. M. Lake, T. D. Ullman, J. B. Tenenbaum, and S. J. Gershman, "Building machines that learn and think like people," *Behavioral and Brain Sciences*, vol. 40, e253, 2017. DOI: 10.1017/S0140525X16001837.

[2] N. Mahowald, A. Ivanova, I. A. Blank, N. Kanwisher, J. B. Tenenbaum, and E. Fedorenko, "Dissociating language and thought in large language models," *Trends in Cognitive Sciences*, vol. 28, no. 5, pp. 473–487, 2024. DOI: 10.1016/j.tics.2024.01.007.

[3] P. Schegg and C. Duriez, "Review on generic methods for mechanical modeling, simulation and control of soft robots," *PLOS ONE*, vol. 17, no. 1, e0251059, 2022. DOI: 10.1371/journal.pone.0251059.

[4] H. K. Versteeg and W. Malalasekera, *An Introduction to Computational Fluid Dynamics: The Finite Volume Method*, 2nd. Harlow, England: Pearson Education Limited, 2007, ISBN: 978-0-13-127498-3. [Online]. Available: https://books.google.com/books/about/An_Introduction_to_Computational_Fluid_D.html?id=RvBZ-UMpGzIC.

[5] E. Ménager, T. Navez, P. Chaillou, O. Goury, A. Kruszewski, and C. Duriez, "Modeling, embedded control and design of soft robots using a learned condensed FEM model," *arXiv preprint arXiv:2503.15009*, 2025. [Online]. Available: https://arxiv.org/abs/2503.15009.

[6] Q. L. Lidec, W. Jallet, L. Montaut, I. Laptev, C. Schmid, and J. Carpentier, "Contact models in robotics: A comparative analysis," *arXiv preprint arXiv:2304.06372*, 2024. [Online]. Available: https://arxiv.org/abs/2304.06372.

[7] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, "Physics-informed machine learning," *Nature Reviews Physics*, vol. 3, no. 6, pp. 422–440, 2021. DOI: 10.1038/s42254-021-00314-5.

[8] A. Quarteroni, A. Manzoni, and F. Negri, *Reduced Basis Methods for Partial Differential Equations: An Introduction* (MS&A Series: Modeling, Simulation and Applications). Springer, 2015, vol. 92, ISBN: 978-3-319-15430-5. DOI: 10.1007/978-3-319-15431-2.

[9] N. Doumèche, G. Biau, and C. Boyer, "On the convergence of pinns," *Bernoulli*, vol. 31, no. 3, pp. 2127–2151, 2025. DOI: 10.3150/24-BEJ1799.

[10] P. K. Singh, K. A. Farrell-Maupin, and D. Faghihi, "A framework for strategic discovery of credible neural network surrogate models under uncertainty," *arXiv preprint arXiv:2403.08901v1*, 2024. [Online]. Available: https://arxiv.org/abs/2403.08901v1.

[11] Y. Xu, Z. Wan, Y. Zhang, H. Zhang, K. Hauser, and B. Zhu, "Accelerated quasi-static fem for real-time modeling of continuum robots with multiple contacts and large deformation," *arXiv preprint arXiv:2503.06922*, 2025. [Online]. Available: https://arxiv.org/abs/2503.06922.

[12] M. G. Mecattaf, B. Slater, M. Tešić, J. Prunty, K. Voudouris, and L. G. Cheke, "A little less conversation, a little more action, please: Investigating the physical commonsense of llms in a 3d embodied environment," *arXiv preprint arXiv:2410.23242v2*, 2024. [Online]. Available: https://arxiv.org/abs/2410.23242v2.

6 Appendix

All experimental data, logs, and code scripts are available at **this link**.

Scheme Representing the Main Units of the Framework

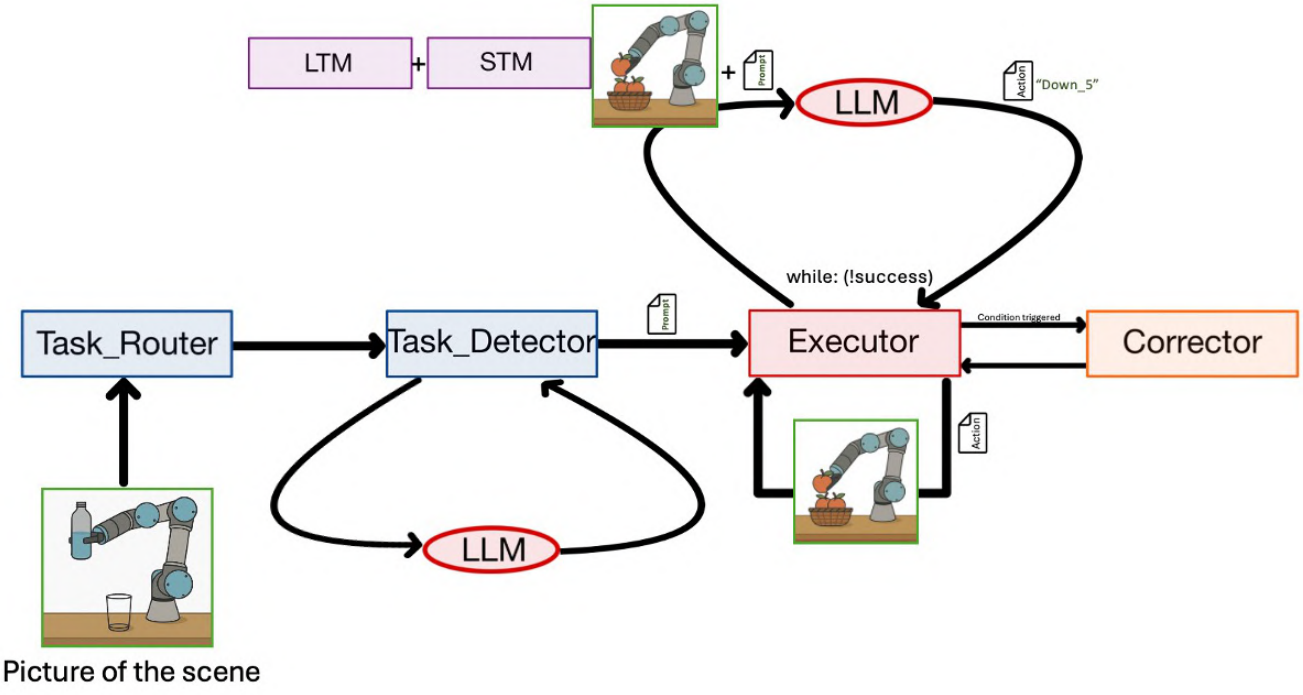


Figure 11: Overview of the framework showing task detection, execution, correction, and memory modules interacting through the LLM.

Image input for the LLM for Maximum Height of the ball Prediction

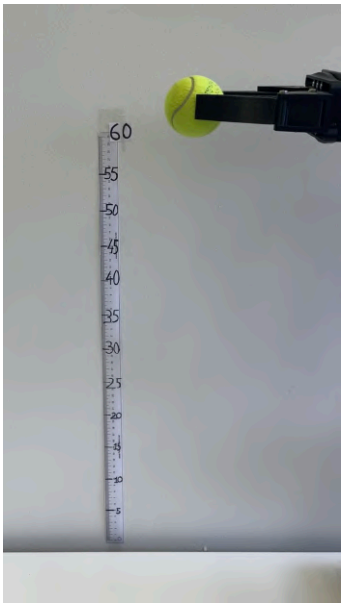


Figure 12: Image provided to the LLM, from which it must predict the maximum rebound height, measured between the bottom of the ball and the rebound surface after the first bounce.

Table Representing the Success Rates of the different Liquid-Container Experiments

Experiment Title	Success Rate
Water in a standard glass	0.9
Water in a glass jar	0.9
Water in a beer glass	0.9
Water in a stemmed glass	0.8
Oil in a standard glass	0.9
Oil in a glass jar	0.9
Oil in a beer glass	0.8
Oil in a stemmed glass	0.8
Dish soap in a standard glass	0.9
Dish soap in a glass jar	0.8
Dish soap in a beer glass	0.9
Dish soap in a stemmed glass	0.8
Honey in a standard glass	0.8
Honey in a glass jar	0.8
Honey in a beer glass	0.8
Honey in a stemmed glass	0.7

Table 3: Success rates of various liquid–container experiments

Visual Misinterpretation of the LLM concerning the Water Level

Figure 13: Visual misinterpretation by the LLM: it considers the water level to be more than half full due to the visual effect produced by the glass and the background (highlighted in the picture).

Scale Bias Example: Input format influences the LLM's behaviour

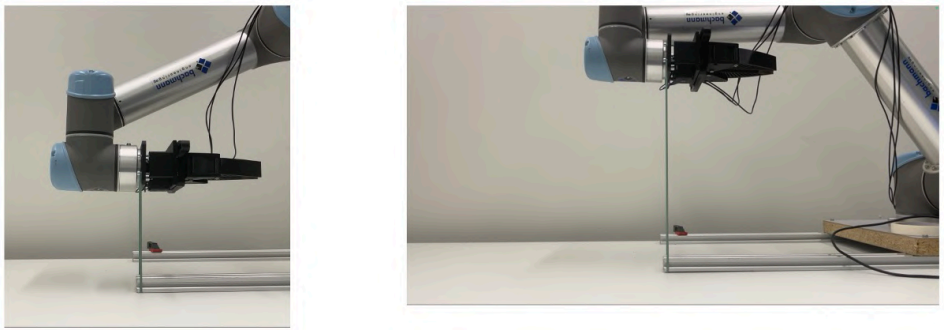
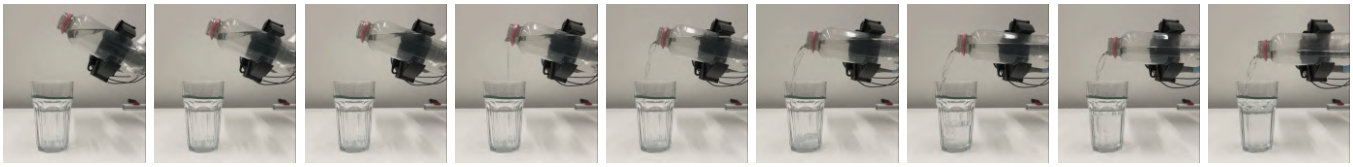
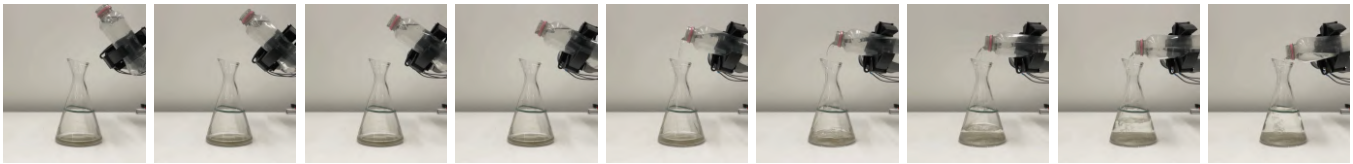


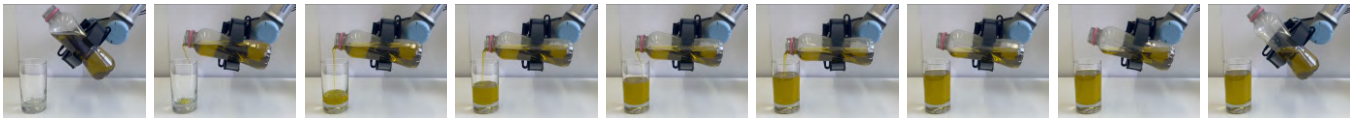
Figure 14: Scale bias resulting from the input image format: in portrait orientation (left picture), the LLM stretches the rubber band for an average of 22 iterations, compared to 37 in landscape orientation (right picture).



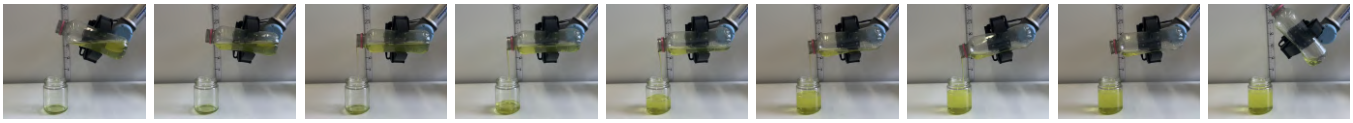
(a) Fill a glass to the limit (success)



(b) Fill a carafe to the limit (success)

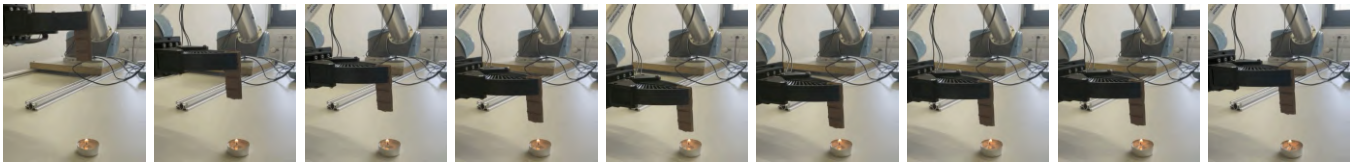


(c) Oil in a standard glass (success)



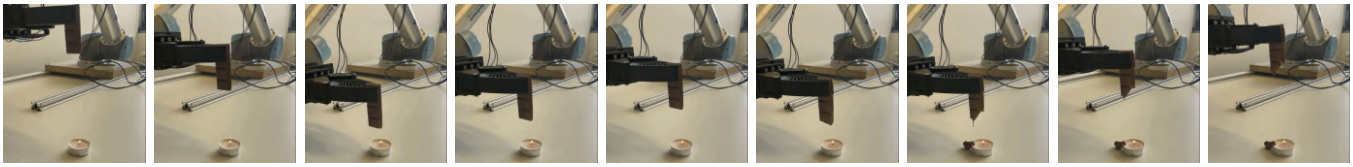
(d) Soap in glass jar (success)

(e) Honey in a stemmed glass (success)

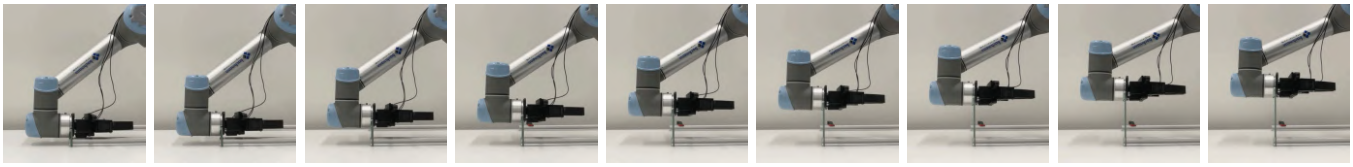


(f) Melt chocolate (success)

Figure 15: Sequences of tasks undertaken by the robot (1/2).

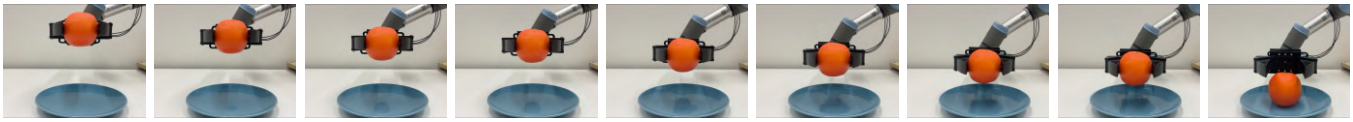


(g) Melt chocolate (failure)



(h) Stretch elastic (failure)

(i) Drop orange (vertical) (success)



(j) Drop orange horizontal (success)

(k) Drop tennis ball (success)

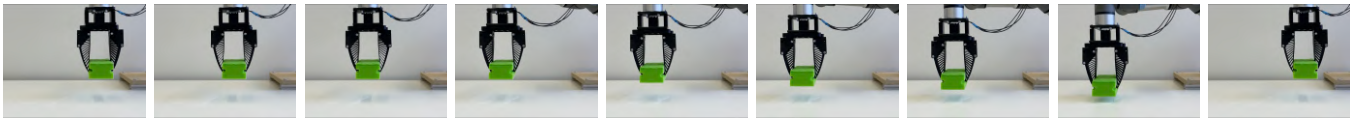
(l) Drop peach (success)

(m) Drop plum (failure)

(n) Drop pen (success)

(o) Drop trash (success)

(p) Choose the appropriate receptacle (success)



(q) Wipe table (failure)

Figure 15: Sequences of tasks undertaken by the robot (2/2).

Declaration of Originality

I hereby submit the written work entitled:

Leveraging Emerging Physical Knowledge in LLMs Toward World-Aware Robotics

- ☐ I hereby declare that I authored the work in question independently, i.e. that no one helped me to author it. Suggestions from the supervisor regarding language and content are excepted. I used no generative artificial intelligence technologies[1].
- ■ I hereby declare that I authored the work in question independently. In doing so I only used the authorised aids, which included suggestions from the supervisor regarding language and content and generative artificial intelligence technologies. The use of the latter and the respective source declarations proceeded in consultation with the supervisor.

Author(s)

Constantin	Riff

Student supervisor(s)

Francesco	Stella

Supervising lecturer

Marco	Hutter
Josie	Hughes

With my signature I confirm the following:

- I have adhered to the rules set out in the Citation Guidelines
- I have documented all methods, data and processes truthfully and fully.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for originality.

Lausanne, the 04/09/2025

Place and date

Signature

[1]For further information please consult the ETH Zurich websites, e.g. `https://ethz.ch/en/the-eth-zurich/education/ai-in-education.html` and `https://library.ethz.ch/en/researching-and-publishing/scientific-writing-at-eth-zurich.html` (subject to change).

Declaration: Use of Generative AI (GenAI)

Please declare all uses of Generative AI (GenAI) in your work:

- ■ Text generation (initial version)*
- ☐ Presentation generation (initial version)*
- ☐ Generation of figures directly from data*†
- ☐ Paper research and literature review*
- ☐ Generation of videos*
- ☐ Text refinement (incl. Grammarly, Writefull)
- ☐ Presentation refinement (incl. 365 Copilot)
- ■ Code generation/debugging/optimization
- ☐ Generation of non-data-driven figures
- ■ Other (describe below):

Pictures in cartoon-style representing real scene given as input to the LLM for my experiment. These pictures have been used in my presentation and report.

* For the uses designated, supplementary material (e.g., as an appendix) containing all prompts and responses must be provided with your final submission.

Signature: _____ **Date:** 04/09/2025

Our Guidelines:

- You are responsible for the content of the work you submit.
- Do not upload copyrighted, private, or confidential information to GenAI tools.
- Do not copy/paste generated text, data, or code before careful revision.
- Triple-check any generated citations or sources. You must have checked all the works you cite.
- When using GenAI for research or ideation, ensure that you understand the output within the context of your work and discuss with supervisors if the validity is unclear. Do not blindly accept the output.
- †Instead of using GenAI to generate figures or plots from data, prefer to use it to assist in the creation of code that generates them. This way the figures or plots are reproducible and verifiable.

Additional Resources:

- ETH Zürich *AI in Teaching and Learning* (`https://ethz.ch/en/the-eth-zurich/education/ai-in-education.html`) provides guidelines about responsible, transparent and fair use of GenAI tools.
- ETH Zürich *Plagiarism and generative Arificial Intelligence (genAI)* (`https://library.ethz.ch/en/researching-and-publishing/scientific-writing-at-eth-zurich/plagiat-und-kuenstliche-intelligenz-ki.html`) provides guidelines about the use of GenAI in scientific writing.
- ETH Zürich *AI Tools & Licenses* (`https://ethz.ch/en/the-eth-zurich/education/ai-in-education/tools.html`) provides recommendations about which GenAI tools should be used and how. For ChatGPT for example, "It is strongly recommended to **switch off the option "Improve the model for everyone"** so that inserted texts and data are not used for training the model".

Intellectual Property Agreement

The student acted under the supervision of Prof. Hutter and contributed to research of his group. Research results of students outside the scope of an employment contract with ETH Zurich belong to the students themselves. The results of the student within the present thesis shall be exploited by ETH Zurich, possibly together with results of other contributors in the same field. To facilitate and to enable a common exploitation of all combined research results, the student hereby assigns his rights to the research results to ETH Zurich. In exchange, the student shall be treated like an employee of ETH Zurich with respect to any income generated due to the research results.
This agreement regulates the rights to the created research results.

1. Intellectual Property Rights

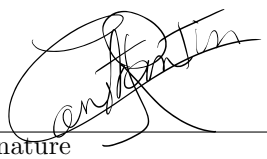
1. The student assigns his/her rights to the research results, including inventions and works protected by copyright, but not including his moral rights ("Urheberpersönlichkeitsrechte"), to ETH Zurich. Herewith, he cedes, in particular, all rights for commercial exploitations of research results to ETH Zurich. He is doing this voluntarily and with full awareness, in order to facilitate the commercial exploitation of the created Research Results. The student's moral rights ("Urheberpersönlichkeitsrechte") shall not be affected by this assignment.
2. In exchange, the student will be compensated by ETH Zurich in the case of income through the commercial exploitation of research results. Compensation will be made as if the student was an employee of ETH Zurich and according to the guidelines "Richtlinien für die wirtschaftliche Verwertung von Forschungsergebnissen der ETH Zürich".
3. The student agrees to keep all research results confidential. This obligation to confidentiality shall persist until he or she is informed by ETH Zurich that the intellectual property rights to the research results have been protected through patent applications or other adequate measures or that no protection is sought, but not longer than 12 months after the collaborator has signed this agreement.
4. If a patent application is filed for an invention based on the research results, the student will duly provide all necessary signatures. He/she also agrees to be available whenever his aid is necessary in the course of the patent application process, e.g. to respond to questions of patent examiners or the like.

2. Settlement of Disagreements

Should disagreements arise out between the parties, the parties will make an effort to settle them between them in good faith. In case of failure of these agreements, Swiss Law shall be applied and the Courts of Zurich shall have exclusive jurisdiction.

Lausanne, the 04/09/2025

Place and date



Signature