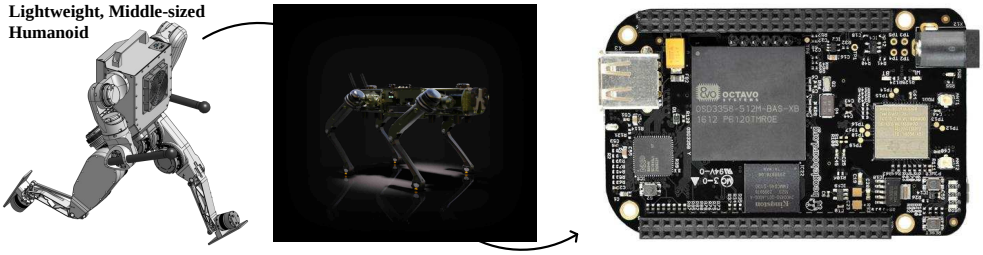
Syntrynos Robotics Research Program

Brian Edmondson
info@Syntrynos.com



(a) CAD Design (b) Training in syntrynos lab (c) microcontrollers

Figure 1: Design, training, and sim-to-real deployment of our custom-built humanoid with a learning-based controller.

Abstract: We introduce Syntrynos Robotics, a reliable and low-cost mid-scale robotics research platform designed for advanced control systems. Our lightweight, in-house-built robotics unit is specifically engineered for control algorithms with low simulation complexity, human-like motion, and high reliability against impacts. The unit's narrow sim-to-real gap enables agile and robust movement across various terrains in outdoor environments, achieved through a simple reinforcement learning controller utilizing light domain randomization. Furthermore, we demonstrate the unit traversing hundreds of meters, walking on steep unpaved trails, and performing hops with both single and double legs, showcasing its high performance in dynamic movement. Capable of omnidirectional locomotion and withstanding large perturbations with a compact design, our system is focused on scalable, real-world deployment of advanced robotics systems.

Keywords: Humanoid, Hardware Design, Reinforcement Learning

1 Introduction

There is a significant demand for mid-scale robotics systems that are designed for rapid deployment of advanced control algorithms, resilient to falls and failures, and cost-effective, while still capable of executing highly dynamic movements. Many existing bipedal and humanoid robots tend to be larger, less safe, and require multiple operators to function effectively. In contrast, working with smaller, shorter-legged robots is more practical due to their lightweight design, which eliminates the need for heavy equipment like gantry cranes. These robots can typically be handled by a single person, and falls rarely cause damage to the robot or its surroundings, making them ideal for experimental environments. Testing can also be conducted in smaller lab spaces, and rough terrain can be easily simulated for validation purposes due to their lower ground clearance. There is a clear demand for compact, reliable, and affordable robotic systems.

Legged humanoid and quadruped robots with custom high-torque density actuators are designed for rapid learning policy iteration. Mechanical design becomes more challenging for shorter-legged robots—whether humanoids or quadrupeds—due to limited space for housing components like motors, sensors, and wiring, which necessitates the use of compact, power-dense actuators. These components are often very expensive or unavailable off-the-shelf. Integrating all necessary parts into a compact volume without sacrificing performance or affordability is difficult. Additionally, mid-scale robots, including both humanoids and quadrupeds, are frequently used to push the limits of dynamic and agile tasks, requiring a high torque-to-weight ratio and increased impact reliability.

Control for mid-scale robots, particularly humanoids and quadrupeds, is more challenging due to their lower center of gravity and heightened sensitivity to disturbances, which often leads to instability. Their reduced mass and inertia make these robots more agile but also more sensitive to even small forces, which can result in significant movement. For both humanoids and quadrupeds, shorter legs result in a reduced stride length, often requiring multiple steps to counteract disturbances. Additionally, these robots require higher frequency leg movements to adjust foot or paw placement rapidly, necessitating precise coordination and control. As a result, actuation of joints must be quick and accurate to support high-frequency motions, while the control policies need to be exceptionally precise and robust to manage the short-time dynamics. Moreover, learning-based algorithms, which are predominant in the control of both humanoids and quadrupeds, face substantial sim-to-real gaps, especially in executing rapid and dynamic motions required for controlling these robots. Consequently, learning-based control for mid-scale robots presents unique challenges.

To address these issues, we propose the custom development of mid-scale humanoid and quadruped platforms with a special focus on facilitating learning-based control. To achieve accurate, robust, and agile control, we leverage learning-based algorithms and focus on narrowing the sim-to-real gap through optimized hardware design. These algorithms allow us to use more cost-effective, noisier sensors, reducing overall costs. To further optimize simulation performance and achieve high-performance actuation, we employ custom modular actuators with integrated transmission, hollow shafts, and EtherCAT for communication.

Our contributions are summarized as follows: (i) We present reliable, low-cost, mid-scale humanoid and quadruped research platforms that focus on narrowing the sim-to-real gap with design considerations tailored for learning-based control. (ii) We demonstrate that our design choices enable the use of minimally composed control policies to perform dynamic and robust locomotion across complex terrains, notably the challenging task of walking on steep, narrow, and unpaved trails. (iii) The codebase for policy training using the recent Isaac Lab release will be open-sourced to support future research in humanoid and quadruped robotics.

2 Related Work

Robot Design: We categorize robots—humanoid, quadruped, and aerial—into three primary sizes: (a) full-scale, comparable to the size of an average adult, (b) mid-scale, similar to the size of a child, and (c) miniature, which refers to smaller, non-human-sized robots. Full-scale humanoid or biped research platforms generally have a larger weight and utilize high gear ratio Harmonic Drive actuators. These platforms are primarily capable of walking and performing simple arm manipulations. Some platforms also incorporate Cycloidic Drive Actuators for high-load joints, along with spring and linkage designs. This setup simplifies the design of reduced-order, step-to-step model-based controllers. However, for recent learning-based algorithms, these designs, while optimized for model-based control, inadvertently affect training and deployment.

In contrast, more lightweight platforms featuring Quasi-Direct-Drive (QDD) actuators and primarily dummy arms have recently been developed, enabling them to perform more dynamic tasks. Besides full-scale humanoids, mid-scale or miniature humanoid research platforms have grown in popularity over recent years. These platforms also opt for QDD actuators, designed for improved dynamic performance, although most lack fully articulated legs.

Beyond humanoid platforms, quadruped robots have gained traction due to their ability to navigate complex terrains, leveraging robust leg designs and QDD actuators for agile movement. These

Table 1: Comparison of existing electric humanoid locomotion research platforms.

Robot	Size[a]	Avg. Leg Len.(m) [b]	Leg Dof	Weight (kg)	Price (USD)	Actuator[c] Type	Max HFE Tor.(Nm)	Max KFE Tor.(Nm)	Transmission Complexity	T/F Sensor
TORO [1]	F	∼0.4	6	76.4	-	H	100	130	++	Joint
LOLA [2]	F	∼0.44	6	68.2	-	H	370	390	+++	Feet
WALK-MAN [3]	F	∼0.38	6	132	-	H	270-400	270-400	++	Feet
Unitree H1 [5]	F	∼0.4	5	47	90K	P	270	360	+	✗
Digit [4]	F	∼0.5	6	50	250K	C, H	200	230	+++	✗
ARTEMIS [16]	F	∼0.38	5	37	-	P	250	250	+	Feet
Cassie [15]	F	∼0.5	5	35	250K	C, H	195	195	+++	✗
MIT [18]	M	∼0.28	5	24	-	P	72	144	+	✗
Unitree G1 [19]	M	∼0.3	6	35	16K	P	88	139	+	✗
HECTOR [17]	M	∼0.22	5	16	-	P	33.5	51.9	+	✗
iCub [44]	M	∼0.2	6	24	300K	H	40	40	++++	Feet
BRUCE [10]	S	∼0.17	5	3.3	6.5K	P	10.5	10.5	+	✗
NAO [45]	S	∼0.15	6	4.5	14K	S	1.61	1.61	+	Feet
DARwIn-OP [46]	S	∼0.09	6	2.8	-	S	2.35	2.35	+	Feet
Surena-Min [47]	S	∼0.085	6	3.3	-	S	3.1	7.3	+	✗
Ours	M	**∼0.2**	**6**	**16[d]**	**10K[e]**	**P**	**62.6**	**81.1**	**+**	✗

[a] F, M, and S represent Full, Middle, and Small, respectively.
[b] Average length of thigh and calf.
[c] H, P, C, and S represent Harmonic Drive, Planetary, Cycloidal Drive, and Servo Motor with a high reduction ratio, respectively.
[d] Without arms. The estimated weight of two 4 DoF arms is 6kg, the total weight will be 22kg.
[e] Without arms. The estimated cost of two 4 DoF arms is 5K USD, the total non-profit cost will be 15K USD.

transmissions between joints and actuators [21]. Boston Dynamics' hydraulic Atlas [22] excels in highly dynamic tasks, and the newly released electric Atlas [22] showcases simplified joint designs with a large range of motion. The robots from companies are well-designed and well-tested, but unfortunately, most of them are not available for researchers in labs or do not provide access to modify or improve the low-level system.

Humanoid and Quadruped Control
Controlling both humanoid and quadruped robots presents significant challenges in the field of robotics. Utilizing various control approaches, from heuristic-based methods to model-based control, both humanoids and quadrupeds have been equipped with stable movement abilities. Recently, learning-based approaches have demonstrated promising capabilities for both types of robots, ranging from locomotion to manipulation. Dynamic locomotion has been shown for both humanoids and quadrupeds, with examples such as walking on rough terrain, resisting large disturbances, running, and even performing complex motions like parkour. These works often rely on complex neural networks and training pipelines to improve expressiveness, or they utilize a history of state-action pairs for online adaptation, which helps reduce the sim-to-real gap in deployment. However, performing dynamic motions with a simple algorithm and architecture remains challenging.

Moreover, previous research often includes broad distributions of domain randomization to account for robustness and to mitigate the imprecise models associated with complex transmissions. Unfortunately, excessive randomization can hinder successful policy learning or lead to overly conservative strategies. Despite the progress in controlling full-scale humanoid and quadruped robots, learning control policies for smaller-scale systems pose additional challenges, particularly due to shorter-legged designs. These designs, which are common in quadrupeds as well as humanoids, lead to limited dynamic motion capabilities. For example, teaching miniature robots to perform tasks like soccer-playing has historically been met with challenges, such as using flat foot designs and servo motors, which restrict the robots' ability to perform dynamic movements. In contrast, our design utilizes smaller feet and more powerful actuators, enabling more dynamic and agile motions, albeit with increased control challenges.

3 Design for Learning-based Control

In this section, we will introduce our humanoid robot design. First, we provide an overview of the system design, and then, we will explain the motivations and our solutions behind the design choices tailored for learning-based control algorithms.

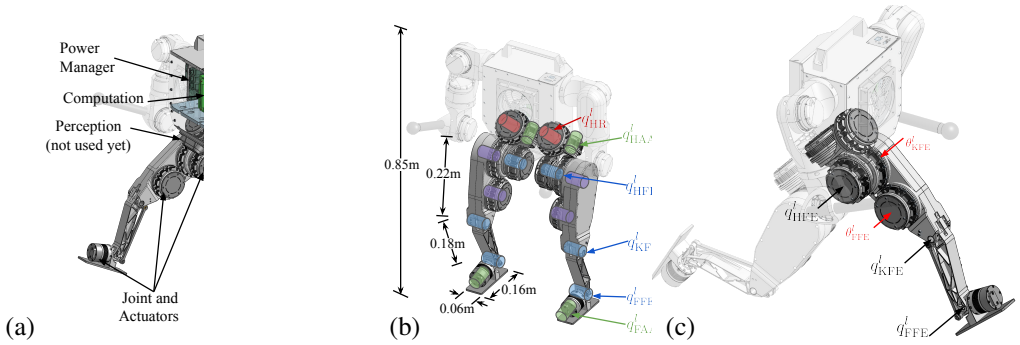


Figure 2: Overview of design: (a) main components, (b) joints and key dimensions, (c) key actuators and joints of the left leg.

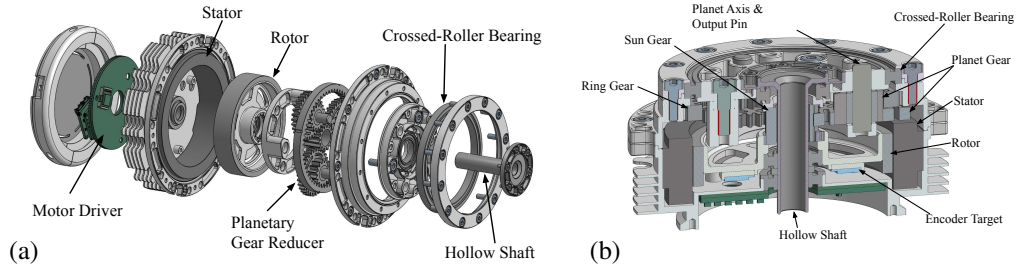


Figure 3: (a) Exposed view and (b) cross view of one of our custom actuators.

3.1 System Overview

The Syntrynos Robotics Unit is a 16 kg, fully electric mid-scale robot designed for advanced research in robotics. The main component is shown in Figure 2(a). The robot has a torso and two 6-DoF legs, with a thigh length of 220 mm, a calf length of 180 mm, and a total height of 0.85 m in a nominal standing configuration, resembling the body shape of a 5-year-old child.

Inside the torso, a computer, a power management board, and an affordable IMU sensor are installed. In addition, two easily replaceable batteries are mounted in a protected compartment within the torso. Each leg is equipped with 6 actuators for the 6 joints, most of which are directly attached to the link and act as a joint. Although two 4-DoF arms were designed, they were omitted to simplify the focus on locomotion abilities. To adapt to different torque requirements on each joint, Syntrynos Robotics developed 4 types of actuators, categorized by motor size, and 2 types of motor drivers for each leg, as shown in Table 2 and Figure 10. These high-performance actuators allow the robot to execute highly dynamic maneuvers.

The communication system is another critical component of the Syntrynos design. To enable precise communication with minimal latency, the system employs a high-bandwidth EtherCAT protocol. Syntrynos has developed custom EtherCAT clients for both motor drivers and the IMU. The onboard PC runs the EtherCAT master and communicates with peripherals at frequencies ranging from 1 kHz to 4 kHz. USB and ethernet connections are also supported for interfacing external sensors, such as depth cameras, lidar, or other devices. For user development and debugging, a router inside the torso provides both wired and wireless connections to the onboard PC.

Nearly every part of the robot is custom-designed and built, including the actuators, mechanical components, motor drivers, IMU, communication system, and power management board. This comprehensive understanding of the entire system allows Syntrynos Robotics to explore new control strategies with a narrow sim-to-real gap, while meeting specific hardware requirements for learning-based algorithms.

Table 2: Custom Actuator Specifications.

Actuator	5013	8513	8518	10413
Mass (g)	251	756	856	1011
Gear Ratio	9:1	9:1	9:1	9:1
Hollow Shaft	✗	✓	✓	✓
Diameter × Thickness (mm)	54.6 × 53	104 × 50	104 × 55	123 × 50
Peak Torque (Nm)	9.7	45.3	62.6	81.1
Sustained Torque (Nm)	4.59	18.9	26.1	34.2
Max. Speed at 48V (rad/s)	83.7	40.7	29	27.9
Max. Power (W)	220	570	730	890
Rotor Inertia (kgm^2)	6.1e-6	6.9e-5	9.4e-5	1.5e-4

algorithms, namely, being **Simulation-Friendly**, **Reliable and Low-cost**, **Experiment-Friendly**, and **Anthropomorphic**. We will provide more detail on each of these next.

3.2 Simulation-Friendly

Motivation. Since the dominant trend of modern learning-based locomotion policies leverages model-free reinforcement learning with massively parallelizable simulators as the learning platform, a key consideration of our robot is its simulation cost. For example, while designing transmission linkages with unilateral springs may reduce the load for joint motors, and absorb large impacts, the resulting mechanism involves solving extra dynamical equations that are notoriously hard to simulate and result in high computation costs for parallelism. Furthermore, as most simulators typically model robots with multi-rigid-body dynamics, some can only apply torque directly in joint space without considering actuator transmissions, while others require much more computation to solve the closed kinematic chains involved in the transmissions. However, actuator and transmission factors that can significantly alter the actuation dynamics during highly dynamic tasks, such as torque, velocity, position limits, sensor noise, friction, and inertia of the linkage and rotor, are very challenging to accurately and efficiently map and randomize in joint space. Additionally, more computation and smaller timesteps are required to simulate communication delays [48, 49, 50], motor/actuator dynamics [51], and inaccurate execution rates [37], which further slows down the simulation.

Our Approach. To avoid these difficulties, we opt to remove all flexible or energy-absorbing components, such as springs or dampers, as well as any closed kinematic chains from the robot's kinematic chain and use the simplest actuator-joint transmissions. As illustrated in Figure 3, all actuators are equipped with a cross roller bearing, so that the actuators can be directly mounted and used as joints. As a result, rotor inertia can be easily simulated by adding armature to the diagonal of the joint mass matrix, and other actuator factors can be modeled the same as the joint. One exception is the FFE joint shown in Figure 2, where a linkage transmission is employed to provide large torques, resulting in a coupled but linear joint-actuator mapping for KFE and FFE. This design allows us to treat the actuator as a joint in simulation. In addition, the selection of a planetary gearbox with a QDD gear ratio in our actuators introduces only minor friction uncertainties which are easy to model in joint space. By combining these designs during training, we can focus solely on joint simulation without considering actuator dynamics. To avoid simulating system latency, we use EtherCAT for communications. This ensures a negligible maximum latency ranging from 0.5 ms to 2 ms[1]. The motor torque control bandwidth is set to 1 kHz, allowing the actuator to be simulated as a torque source without delay. These designs enable our robot to achieve an accurate simulation at an efficiency of more than 90,000 simulation steps per second on an NVIDIA A4500 GPU.

Table 3: Cost of Each Component in Small Quantity Production.

Module	Actuator				Sensor	Misc		Off-the-shelf		Total
	5013	8513	8518	10413	IMU	Torso	Leg	PC	Battery	
Cost (USD)	422	570	639	676	50	410	974	347	153	9955
Quantity	2	6	2	2	1	1	2	1	2	-

3.3 Reliable and Low-Cost

Motivation. In the past, humanoid locomotion research required high-end robots, accurate sensors, careful protection, and lengthy repairs, limiting the field's development.

In order to accelerate the field further and to make a change, our robot must be reliable and accessible, meaning that it should be durable for repeated experiments and of low cost. A more accessible and reliable robot also paves the way for scaling up humanoid robot learning in real-world settings.

Our Approach. In order to improve durability, we build the robot with high-performance materials as opposed to [52, 11, 53, 54, 55]. We use 7075 and 6061 aluminum for building most of the main components, and SKD11 steel for the gearbox and linkage, allowing the robot to survive heavy impacts with lightweight structures. The endurance of electrical cables for power and signals is a key factor for the reliability of the robot, where contact with the environment creates tearing due to friction and vibrations that post significant challenges for cable durability. To overcome this, we opt to leverage hollow shaft designs for most of the actuators as shown in Table 2, where power and communication cables cross between the two moving bodies through the hollow shaft axis of the joint, minimizing the tearing caused by joint movements. Furthermore, the usage of custom QDD actuators allows us to estimate the joint torque without adding strain gauges. With reliable joint torque sensing, a generalized momentum observer [56] can be used to estimate the contact wrench of each foot without requiring contact sensors or force/torque sensors, which further improves the reliability of the robot.

The fully customized hardware allowed us to minimize the robot's cost, as shown in Table 3. With learning algorithms, we typically gain enhanced robustness against hardware inaccuracies, allowing for cheaper sensors and further cost reductions. Thus, unlike most previous works [7, 57, 15, 4] where the IMU costs around USD 1,000, we can utilize a cell phone level IMU ICM42688 that costs less than one dollar[2]. These designs help cut the cost down to USD 10,000 for the whole robot without arms. Note that most costs shown in Table 3 will decrease with scaled-up production. The only non-custom components are the computers (Intel i7-1255U) and batteries (DJI TB50), sourced commercially for performance and safety.

3.4 Experiment-Friendly

Motivation. In the past, the size and weight of humanoid robots are especially troublesome for experiments. Traditional full-scaled humanoids are often heavier than a person of the same size, which means handling the robot requires at least two or three people with the help of gantries. More importantly, experimenting with such robots with high torque actuators (≈300 Nm) is dangerous and may result in severe injuries to people nearby.

Our Approach. By properly choosing the robot size and the custom lightweight materials, we reduced the weight to only 16 kg, which allows us to do experiments with only one robot operator for indoor environments, and with an optional cameraman in outdoor environments, including commanding the robot, collecting data, taking video, and sometimes resetting the robot from failure. All of the experiments reported in this work are done with this setup.

[1]The exact latency depends on the selected frequency: 2 ms at 1 kHz and 0.5 ms at 4 kHz.

[2]For sensor IC itself, net cost of IMU Module shown in Table 3.

3.5 Anthropomorphic

Motivation. The advantage of using an anthropomorphic design is significant: it allows for higher static stability and human-like motions by having similar dominant DoFs as human bodies. This results in wider applicability, richer task selection, and easier learning from widely available human demonstrations.

Our Approach. The dominant motion of a human leg [58], while we can model a foot contact with the ground as a 6 DoF contact wrench [59]. Our robot uses an anthropomorphic design with 6 DoFs per leg, which replicates the common modeling of DoFs human legs have. Compared to [5, 15, 17, 16, 18], providing actuation on the roll direction of the ankle joint improves the robot's stability in challenging static poses, such as when manipulating distant objects, and enables it to potentially balance on one foot. Furthermore, each joint limit is designed to closely align with the corresponding physical limits of human bodies. This allows us to provide further protection on the hardware while ensuring enough ranges for imitating human motions.

4 A Minimally Composed Learning-based Controller

With a humanoid platform designed for learning-based control, we are able to achieve robust and agile locomotion with a minimally composed RL controller. In this section, we first introduce the design of the RL controller. Then, we elaborate on how our humanoid platform enables the narrowing of the sim-to-real gap for the RL controller.

4.1 Reinforcement Learning Formulation

We formulate our tasks as Markov Decision Processes (MDPs) and leverage RL to solve them due to their promising performance in humanoid control. We create a minimally composed learning-based controller by doing the following. We formulate the MDP with minimal observation and action spaces. Specifically, we only use immediate state feedback as actor input, without formulating a short or long history [37, 60] or teacher-student training [61, 62] to estimate environment parameters. Similarly, we opt out of pre-defined phase signals [28] or reference motion [60] to reduce human biases. The immediate state feedback includes raw proprioceptive readings (base angular velocity ω, projected gravity vector $\mathbf{g}$, joint positions $\mathbf{q}$, velocities $\dot{\mathbf{q}}$), base linear velocity $\mathbf{v}$ from a state estimator [63], velocity commands $\mathbf{v}^c_{x,y}$ and ω^c_z, and the previous action. Likewise, the action space consists solely of the desired joint positions $\mathbf{q}^d$, which are converted into torques τ directly by a PD controller on the motor driver.

We also design the architecture of the actor-critic with the most basic multilayer perceptron (MLP) networks only. Specifically, each network has hidden sizes of $[512, 256, 128]$ neurons and ELU activation. The policy is optimized via PPO [64] and trained in Isaac Lab [65]. The RL policy executes at 50 Hz, the state estimator at 1 kHz, and the PD controller at 25 kHz.

This minimally-composed RL controller facilitates the validation of the adequacy of our hardware design for learning-based control. Without the ability to do online system identification (through the I/O history) or reference motion guidance, our policy relies on the synergy of the hardware and learning algorithm to achieve a narrow sim-to-real gap, ensuring that the robust and agile locomotion performance in training can be fully demonstrated on the real-world robot. Additionally, it serves as a competent baseline for other algorithms developed on our platform.

4.2 Closing the Sim-to-Real Gap

Hardware Side. We focus on closing the sim-to-real gap through hardware design choices. The main factors of the sim-to-real gap, aside from sensor noise, are modeling errors and command execution rate, accuracy, and delay [50, 48, 51]. To reduce modeling errors, we install actuators directly as joints or design a linear joint-actuator mapping, avoiding the simulation of structures that are

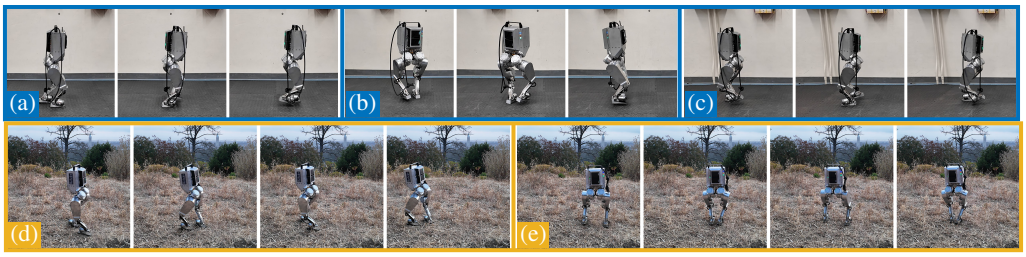


Figure 4: Omnidirectional Walking. (a-c) The robot walks forward, turns in place, and walks backward in the lab environment. (d, e) The robot walks forward and sideways in the wild.

likely to result in inaccurate modeling. To improve command execution, we employ high bandwidth torque control that leads to a precise execution rate, and transparent QDD actuator dynamics so that the commanded torque is accurately tracked and has negligible communication latency. All of these lead to less discrepancy between the hardware and the simulated dynamics.

Design-enabled Accurate Domain Randomization. While most of the learning controllers rely on domain randomization, extensive domain randomization slows down training and results in conservative policies [42]. To avoid this while still preserving a robust policy, in this work, we leverage a different approach aimed at providing accurate domain randomizations given the hardware design. For a humanoid robot performing locomotion tasks, we identify two sources of uncertainties: uncertainty in the robot physics property, e.g., the mass of each link, and that in performing tasks, e.g., contact with the environment.

For hardware uncertainty, our detailed design allows us to obtain a small and accurate range of parameter variations. Specifically, we use CAD to retrieve accurate mechanical parameters like rotor inertia and conduct simple experiments to characterize the friction of each actuator separately. This demonstrates the benefits of an in-house-built robot, as obtaining such detailed hardware parameters for commercial robots would be difficult.

For uncertainty in contact with the environment, we apply a wide range of domain randomization to cover as many real-world environment conditions as possible. This includes ground friction, restitution, and external perturbation forces from obstacles and unstable ground conditions.

Unlike previous work [48, 66, 60], we opt not to randomize properties that cannot be identified in these two categories, such as a general "motor strength" ratio or PD gains, which were often used as a "lazy approach" to approximate actuation uncertainties. However, because it is hard to accurately analyze the range of uncertainties with PD approximation, prior works rely on heuristics, which can lead to unnecessarily large ranges of domain randomization, which we aim to avoid.

As we will show later, with design-enabled accurate domain randomizations, we can achieve robust and agile locomotion skills when zero-shot transferring to robot hardware, even with a minimally-composed RL controller.

5 Experimental Validation

In our experiments, we aim to validate how our humanoid design facilitates learning locomotion control from three aspects: (1) The effectiveness of our minimally-composed RL controller in learning humanoid locomotion tasks. (2) The sim-to-real gap for the minimal RL algorithm with our adequate hardware design. (3) The hardware reliability of the robot.

Figure 5: Walking on Various Terrains. (a) The robot walks on eight different types of terrain. (b) The robot climbs a relatively steep and narrow unpaved trail covered with dust and rocks. (c) The robot walks on an uneven pathway. (d) The robot makes a turn on rocky stairs.

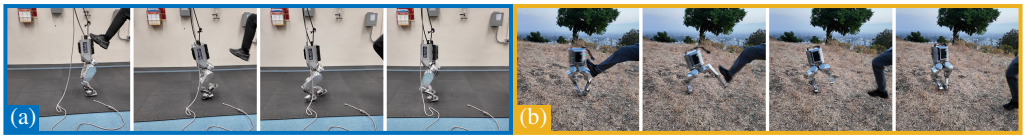


Figure 6: Disturbance Rejection. The robot is able to recover from large external perturbations, such as being kicked (a) from behind while walking in the lab, and (b) from the side while walking in the wild.

5.1 Learning Control Performance

Compared to previous works leveraging advanced architectures, in this work, we emphasize how our minimal design that puts a specific focus on adapting learning-based control algorithms facilitates us to achieve robust and agile locomotion performance with a basic RL controller introduced in Sec. 4.

Omnidirectional Walking. We train our robot to perform omnidirectional locomotion by following linear velocity commands in sagittal and lateral directions as well as angular velocity commands in yaw. In Figure 4, we show examples of walking forward, backward, and turning left and right. In the following paragraphs, we focus on demonstrating the performance of this omnidirectional controller on various terrains and against external perturbations.

Walking on Various Terrains. Perhaps the best demonstration of the advanced performance of a humanoid is its capability to traverse various everyday environments robustly. As shown in Figure 5(a), our robot is able to walk robustly on diverse outdoor terrains, such as grass fields, brick

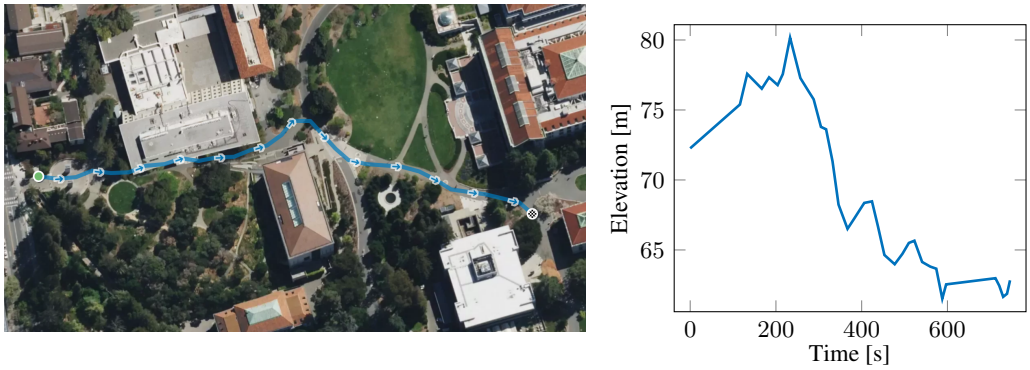


Figure 7: Recorded GPS visualization of a long distance walking.

sidewalks, unpaved trails, asphalt roads, bridges, concrete roads, running tracks, and tiled surfaces, as well as stairs and inclines.

Among these environments, we would like to emphasize the two most challenging terrains. First, as shown in Figure 5(b) and the accompanying video, we are surprised to find that our robot is able to climb a relatively steep and narrow unpaved trail covered with dust and rocks. This trail is a bit steep to climb even for adults, let alone our robot which resembles only a 5-year-old child in size. Specifically, the incline of the trail is on average 20 degrees, higher than the upward pitch range of the ankle so that it has to go backward to be able to step firmly on the ground with the torso in the upright position. Despite this, our robot is able to walk stably, make turns, and recover from stepping on loose rocks.

Second, as shown in Figure 5(c), we often find uneven pathways with noticeable gaps and changes in height between the slabs in urban environments. These gaps and slippery slabs require extra attention from children and aged individuals and sometimes cause them to fall over. On this challenging terrain, our robot is able to navigate both forward and backward inside the small pathway across changes in stair heights and recover from slipping.

In order to further demonstrate uneven terrain, we create a set of rocky stairs with step heights of 4 cm (10% of full leg length) and find that our robot is able to traverse the stairs smoothly and make turns on them, as seen in Figure 5(d). Being able to handle these challenging terrains shows an advanced performance on locomotion control for our humanoid, even with such a basic RL controller, attributed to the careful adaptations for learning-based control algorithms in the hardware design.

Disturbance Rejection. A crucial test of the robustness of the policy and the reliability of the hardware is the ability to recover from external perturbations. We exert instantaneous force randomly by kicking different parts of our robot while it is stepping in place. As shown in Figure 6, this perturbation causes a significant deviation from the nominal walking pose, making the robot almost fall over. Nevertheless, our robot is able to respond immediately, regain its stability from the perturbation within a few steps, and resume stepping.

In addition to the flat ground in the controlled lab environment, we repeat this test in outdoor environments, such as on uneven grass terrains. In these conditions, our robot is also able to recover from heavy external forces, as shown in Figure 6(b). This further showcases the robustness of our humanoid robot in real-world scenarios.

Long Distance Walking With the ability to traverse terrains and reject perturbations, the robot is able to perform relatively long-distance walking for several hundred meters over multiple terrains. As shown in Figure 7, the robot rambles freely on the campus of UC Berkeley for 10 minutes, traversing a total distance of 364 m with uphills and downhills. Furthermore, the robot is able

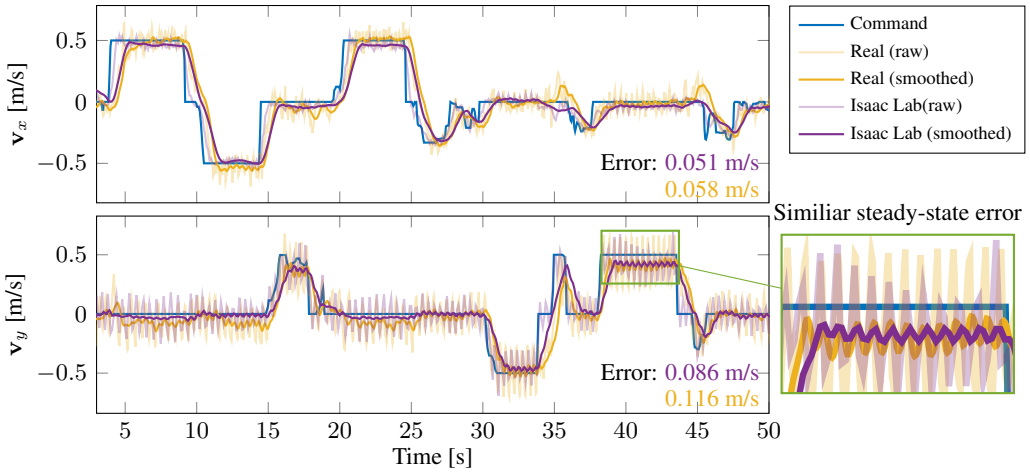


Figure 8: Sim-to-real gap evaluation. We show trajectories for commanded (blue) and actual (yellow) base linear velocity. The actual value is smoothed by a moving average filter to better illustrate the steady-state error.

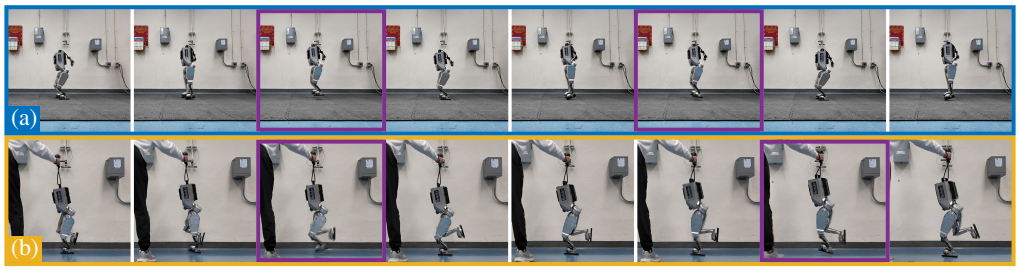


Figure 9: Hopping with (a) both legs and (b) a single leg, with noticeable flight phases. Being able to accomplish dynamic tasks with a simple RL controller shows the small sim-to-real gap of the hardware design. The purple frames indicate that the robot is in the flight phase.

to climb steadily along the rough terrain shown in Figure 5(b) for more than 5 minutes non-stop, covering 96 m in distance and an elevation gain of 10.5 m. The video of the campus walking can be seen at `https://youtu.be/STbB12-oc_w` and the video of walking on rough terrain is at `https://youtu.be/Z2Bzslmu7DA`.

5.2 Evaluation of Sim-to-Real Transfer

Because the majority of learning-based algorithms are trained entirely in simulation, the sim-to-real gap becomes a critical component of the performance of learning-based controllers in the real world. We demonstrate the small sim-to-real gap of our robot in two aspects: (i) A quantitative analysis of the locomotion task metrics. (ii) The ability to perform highly dynamic locomotion tasks.

First, we present a quantitative analysis of the sim-to-real transfer by plotting the tracking performance with random velocity commands given by the operator. As shown in Figure 8, our robot is able to follow the rapidly changing command closely in both lateral and sagittal directions with small steady-state errors. Over a 60-second trial, the average tracking error in the sagittal direction is 0.051 m/s in simulation and 0.058 m/s on hardware. In the lateral direction, the error is 0.086 m/s in simulation and 0.1156 m/s on hardware, respectively. Note that our RL controller is unable to perform online system identification or adaptation as it does not have access to the history during either training or deployment. Thus, these small differences in tracking errors indicate that the gap between the simulation MDPs during training and the MDPs of the real-world deployment is indeed small, which confirms the narrow sim-to-real gap for our hardware design.

Second, we showcase the ability to perform highly dynamic motions by demonstrating a hopping controller trained with the same settings as in Sec. 4 except for the rewards. As shown in Figure 9(a), our robot can perform omnidirectional hops, accelerate, and decelerate while maintaining balance. Notably, the robot further demonstrates exceptional agility by being able to perform hops using only one leg in Figure 9(b), a highly challenging feat. Although a safety rope is used and minor balance assistance is needed during single-leg hopping experiments, the rope is mostly slack, and the robot is able to maintain its balance on its own. Compared to complex algorithm designs in prior works, this further shows that the hardware design facilitates us to perform agile motions with simple algorithmic design.

5.3 Hardware Reliability

Lastly, hardware reliability against ground impacts is vital for learning-based approaches. Throughout this work, we recorded a total of 38 times of our robot falling over on various terrains including concrete pavements and unpaved roads, shown in Table 4 in the Appendix. Thanks to the reliable and lightweight design, we did not experience any damage to the hardware itself except for two failures caused by loose screws and glue. In most fallovers, we are able to reset the robot and resume the control policy within 3 to 5 seconds. The ability to reset easily and rapidly not only relieves the burden of experiments but more importantly, is necessary for the ultimate goal of scalable real-world deployment.

6 Limitations

Major limitations of this work include the omission of arms for simplicity since the main research topic of mid-scale humanoids still focuses on locomotion tasks. The range of motion, backlash, weight, and mechanical strength, will be further improved after a few hardware iterations. To further minimize the sim-to-real gap for more dynamic motion, detailed system identification for torque-current non-linear mapping near saturated torque should be performed. Motor region of work [67], and heat protection should be simulated during training. In the future, the platform will be equipped with two 4 or 6 DoFs arms and enough power to perform dynamic tasks such as backflipping.

7 Conclusion

This work introduced the Syntrynos Robotics Unit, a reliable and low-cost research platform for learning-based bipedal locomotion control with a narrow sim-to-real gap. Our in-house-built robotic unit is designed to accommodate advanced learning-based control algorithms, featuring low simulation complexity, human-like ranges of motion, and high reliability against falls and impacts. Constructed with lightweight materials, the unit significantly reduces the challenges of conducting hardware experiments.

The Syntrynos platform excels in performing robust outdoor experiments across various terrains and ground conditions with a minimally designed reinforcement learning (RL) algorithm, further demonstrating its efficacy in learning-based control and maintaining a small sim-to-real gap. Without relying on historical data or phase signals, our control policy is able to withstand large, random external disturbances and execute omnidirectional locomotion over difficult terrains. Notably, the unit showcases the ability to cover long distances, climb steadily along steep and narrow unpaved trails, and hop on a single leg—a highly dynamic and challenging feat.

As a reliable and cost-effective research platform, the Syntrynos Robotics Unit's ultimate goal is scalable deployment for learning and real-world applications.

Acknowledgments

This work was supported in part by Brian Edmondson. I would like to express my gratitude for the valuable discussions and assistance provided throughout the project.

References

[1] J. Englsberger, A. Werner, C. Ott, B. Henze, M. A. Roa, G. Garofalo, R. Burger, A. Beyer, O. Eiberger, K. Schmid, et al. Overview of the torque-controlled humanoid robot toro. In *2014 IEEE-RAS International Conference on Humanoid Robots*, pages 916–923. IEEE, 2014.

[2] P. Seiwald, S.-C. Wu, F. Sygulla, T. F. Berninger, N.-S. Staufenberg, M. F. Sattler, N. Neuburger, D. Rixen, and F. Tombari. Lola v1. 1–an upgrade in hardware and software design for dynamic multi-contact locomotion. In *2020 IEEE-RAS 20th International Conference on Humanoid Robots (Humanoids)*, pages 9–16. IEEE, 2021.

[3] N. G. Tsagarakis, D. G. Caldwell, F. Negrello, W. Choi, L. Baccelliere, V.-G. Loc, J. Noorden, L. Muratore, A. Margan, A. Cardellino, et al. Walk-man: A high-performance humanoid platform for realistic environments. *Journal of Field Robotics*, 34(7):1225–1259, 2017.

[4] Agility Robotics. Meet digit: The newest robot from agility robotics, 2024. URL https://agilityrobotics.com/products/digit.

[5] Unitree Robotics. Unitree h1, 2024. URL https://www.unitree.com/h1/.

[6] J. Ramos, B. Katz, M. Y. M. Chuah, and S. Kim. Facilitating model-based control through software-hardware co-design. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 566–572. IEEE, 2018.

[7] B. G. Katz. *A low cost modular actuator for dynamic robots*. PhD thesis, Massachusetts Institute of Technology, 2018.

[8] M. Chignoli, D. Kim, E. Stanger-Jones, and S. Kim. The MIT humanoid robot: Design, motion planning, and control for acrobatic behaviors. In *2020 IEEE-RAS 20th International Conference on Humanoid Robots (Humanoids)*, pages 1–8. IEEE, 2021.

[9] B. Katz, J. Di Carlo, and S. Kim. Mini cheetah: A platform for pushing the limits of dynamic quadruped control. In *2019 international conference on robotics and automation (ICRA)*, pages 6295–6301. IEEE, 2019.

[10] Y. Liu, J. Shen, J. Zhang, X. Zhang, T. Zhu, and D. Hong. Design and control of a miniature bipedal robot with proprioceptive actuation for dynamic behaviors. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 8547–8553. IEEE, 2022.

[11] F. Grimminger, A. Meduri, M. Khadiv, J. Viereck, M. Wüthrich, M. Naveau, V. Berenz, S. Heim, F. Widmaier, T. Flayols, et al. An open torque-controlled modular robot architecture for legged locomotion research. *IEEE Robotics and Automation Letters*, 5(2):3650–3657, 2020.

[12] A. B. Ghansah, J. Kim, K. Li, and A. D. Ames. Dynamic walking on highly underactuated point foot humanoids: Closing the loop between hzd and hlip. *arXiv preprint arXiv:2406.13115*, 2024.

[13] M. Hutter, C. Gehring, D. Jud, A. Lauber, C. D. Bellicoso, V. Tsounis, J. Hwangbo, K. Bodie, P. Fankhauser, M. Bloesch, et al. Anymal-a highly mobile and dynamic quadrupedal robot. In *2016 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 38–44. IEEE, 2016.

[14] A. Hattori. *Design of a high torque density modular actuator for dynamic robots*. PhD thesis, Massachusetts Institute of Technology, 2020.

[15] Agility Robotics. Cassie sets a guinness world record, 2022. URL https://agilityrobotics.com/news/2022/cassie-sets-a-guinness-world-record.

[16] T. Zhu. *Design of a highly dynamic humanoid robot*. University of California, Los Angeles, 2023.

[17] J. Li, J. Ma, O. Kolt, M. Shah, and Q. Nguyen. Dynamic loco-manipulation on hector: Humanoid for enhanced control and open-source research. *arXiv preprint arXiv:2312.11868*, 2023.

[18] A. SaLoutos, E. Stanger-Joncs, Y. Ding, M. Chignoli, and S. Kim. Design and development of the mit humanoid: A dynamic and robust research platform. In *2023 IEEE-RAS 22nd International Conference on Humanoid Robots (Humanoids)*, pages 1–8. IEEE, 2023.

[19] Unitree Robotics. Unitree g1, 2024. URL https://www.unitree.com/g1/.

[20] A. Wang, J. Ramos, J. Mayo, W. Ubellacker, J. Cheung, and S. Kim. The hermes humanoid system: A platform for full-body teleoperation with balance feedback. In *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, pages 730–737. IEEE, 2015.

[21] H. Khan, R. Featherstone, D. G. Caldwell, and C. Semini. Bio-inspired knee joint mechanism for a hydraulic quadruped robot. In *2015 6th International Conference on Automation, Robotics and Applications (ICARA)*, pages 325–331. IEEE, 2015.

[22] Boston Dynamics. Atlas, 2024. URL https://www.bostondynamics.com/atlas.

[23] M. H. Raibert. *Legged robots that balance*. MIT press, 1986.

[24] S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, and H. Hirukawa. The 3d linear inverted pendulum mode: A simple modeling for a biped walking pattern generation. In *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No. 01CH37180)*, volume 1, pages 239–246. IEEE, 2001.

[25] S. Kuindersma, R. Deits, M. Fallon, A. Valenzuela, H. Dai, F. Permenter, T. Koolen, P. Marion, and R. Tedrake. Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot. *Autonomous robots*, 40:429–455, 2016.

[26] Y. Ding, C. Khazoom, M. Chignoli, and S. Kim. Orientation-aware model predictive control with footstep adaptation for dynamic humanoid walking. In *2022 IEEE-RAS 21st International Conference on Humanoid Robots (Humanoids)*, pages 299–305. IEEE, 2022.

[27] C. Khazoom, S. Hong, M. Chignoli, E. Stanger-Jones, and S. Kim. Tailoring solution accuracy for fast whole-body model predictive control of legged robots. *arXiv preprint arXiv:2407.10789*, 2024.

[28] J. Siekmann, Y. Godse, A. Fern, and J. Hurst. Sim-to-real learning of all common bipedal gaits via periodic reward composition. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7309–7315. IEEE, 2021.

[29] I. Radosavovic, T. Xiao, B. Zhang, T. Darrell, J. Malik, and K. Sreenath. Real-world humanoid locomotion with reinforcement learning. *Science Robotics*, 9(89):eadi9579, 2024.

[30] R. P. Singh, Z. Xie, P. Gergondet, and F. Kanehiro. Learning bipedal walking for humanoids with current feedback. *IEEE Access*, 2023.

[31] A. Tang, T. Hiraoka, N. Hiraoka, F. Shi, K. Kawaharazuka, K. Kojima, K. Okada, and M. Inaba. Humanmimic: Learning natural locomotion and transitions for humanoid robot via wasserstein adversarial imitation. *arXiv preprint arXiv:2309.14225*, 2023.

[32] I. Radosavovic, B. Zhang, B. Shi, J. Rajasegaran, S. Kamat, T. Darrell, K. Sreenath, and J. Malik. Humanoid locomotion as next token prediction. *arXiv preprint arXiv:2402.19469*, 2024.

[33] X. Cheng, Y. Ji, J. Chen, R. Yang, G. Yang, and X. Wang. Expressive whole-body control for humanoid robots. *arXiv preprint arXiv:2402.16796*, 2024.

[34] T. He, Z. Luo, W. Xiao, C. Zhang, K. Kitani, C. Liu, and G. Shi. Learning human-to-humanoid real-time whole-body teleoperation. *arXiv preprint arXiv:2403.04436*, 2024.

[35] J. Dao, H. Duan, and A. Fern. Sim-to-real learning for humanoid box loco-manipulation. *arXiv preprint arXiv:2310.03191*, 2023.

[36] Z. Fu, Q. Zhao, Q. Wu, G. Wetzstein, and C. Finn. Humanplus: Humanoid shadowing and imitation from humans. *arXiv preprint arXiv:2406.10454*, 2024.

[37] J. Siekmann, K. Green, J. Warila, A. Fern, and J. Hurst. Blind bipedal stair traversal via sim-to-real reinforcement learning. *arXiv preprint arXiv:2105.08328*, 2021.

[38] X. Gu, Y.-J. Wang, X. Zhu, C. Shi, Y. Guo, Y. Liu, and J. Chen. Advancing Humanoid Locomotion: Mastering Challenging Terrains with Denoising World Model Learning. In *Proceedings of Robotics: Science and Systems*, 2024.

[39] B. van Marum, A. Shrestha, H. Duan, P. Dugar, J. Dao, and A. Fern. Revisiting reward design and evaluation for robust humanoid standing and walking. *arXiv preprint arXiv:2404.19173*, 2024.

[40] D. Crowley, J. Dao, H. Duan, K. Green, J. Hurst, and A. Fern. Optimizing bipedal locomotion for the 100m dash with comparison to human running. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 12205–12211. IEEE, 2023.

[41] Z. Zhuang, S. Yao, and H. Zhao. Humanoid parkour learning. *arXiv preprint arXiv:2406.10759*, 2024.

[42] Y. Chebotar, A. Handa, V. Makoviychuk, M. Macklin, J. Issac, N. Ratliff, and D. Fox. Closing the sim-to-real loop: Adapting simulation randomization with real world experience. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8973–8979. IEEE, 2019.

[43] T. Haarnoja, B. Moran, G. Lever, S. H. Huang, D. Tirumala, J. Humplik, M. Wulfmeier, S. Tunyasuvunakool, N. Y. Siegel, R. Hafner, et al. Learning agile soccer skills for a bipedal robot with deep reinforcement learning. *Science Robotics*, 9(89):eadi8022, 2024.

[44] A. Parmiggiani, M. Maggiali, L. Natale, F. Nori, A. Schmitz, N. Tsagarakis, J. S. Victor, F. Becchi, G. Sandini, and G. Metta. The design of the icub humanoid robot. *International journal of humanoid robotics*, 9(04):1250027, 2012.

[45] D. Gouaillier, V. Hugel, P. Blazevic, C. Kilner, J. Monceaux, P. Lafourcade, B. Marnier, J. Serre, and B. Maisonnier. Mechatronic design of nao humanoid. In *2009 IEEE international conference on robotics and automation*, pages 769–774. IEEE, 2009.

[46] I. Ha, Y. Tamura, H. Asama, J. Han, and D. W. Hong. Development of open humanoid platform darwin-op. In *SICE annual conference 2011*, pages 2178–2181. IEEE, 2011.

[47] A. Nikkhah, A. Yousefi-Koma, R. Mirjalili, and H. M. Farimani. Design and implementation of small-sized 3d printed surena-mini humanoid platform. In *2017 5th RSI International Conference on Robotics and Mechatronics (ICRoM)*, pages 132–137. IEEE, 2017.

[48] J. Tan, T. Zhang, E. Coumans, A. Iscen, Y. Bai, D. Hafner, S. Bohez, and V. Vanhoucke. Sim-to-real: Learning agile locomotion for quadruped robots. *arXiv preprint arXiv:1804.10332*, 2018.

[49] Z. Li, X. Cheng, X. B. Peng, P. Abbeel, S. Levine, G. Berseth, and K. Sreenath. Reinforcement learning for robust parameterized locomotion control of bipedal robots. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2811–2817. IEEE, 2021.

[50] Z. Xie, P. Clary, J. Dao, P. Morais, J. Hurst, and M. Panne. Learning locomotion skills for cassie: Iterative design and sim-to-real. In *Conference on Robot Learning*, pages 317–329. PMLR, 2020.

[51] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter. Learning agile and dynamic motor skills for legged robots. *Science Robotics*, 4(26):eaau5872, 2019.

[52] K.-S. Labs. K-scale, 2024. URL https://kscale.dev/.

[53] K. Urs, C. E. Adu, E. J. Rouse, and T. Y. Moore. Design and characterization of 3d printed, open-source actuators for legged locomotion. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1957–1964. IEEE, 2022.

[54] T.-G. Song, Y.-H. Shin, S. Hong, H. C. Choi, J.-H. Kim, and H.-W. Park. Drpd, dual reduction ratio planetary drive for articulated robot actuators. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 443–450. IEEE, 2022.

[55] A. J. Fuge, C. W. Herron, B. C. Beiter, B. Kalita, and A. Leonessa. Design, development, and analysis of the lower body of next-generation 3d-printed humanoid research platform: Pandora. *Robotica*, 41(7):2177–2206, 2023.

[56] S. Haddadin, A. Albu-Schaffer, A. De Luca, and G. Hirzinger. Collision detection and reaction: A contribution to safe physical human-robot interaction. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3356–3363. IEEE, 2008.

[57] Boston Dynamics. Spot - the agile mobile robot, 2024. URL https://bostondynamics.com/products/spot.

[58] S. Kudo, M. Fujimoto, T. Sato, and A. Nagano. Optimal degrees of freedom of the lower extremities for human walking and running. *Scientific Reports*, 13(1):16164, 2023.

[59] S. Caron, Q.-C. Pham, and Y. Nakamura. Stability of surface contacts for humanoid robots: Closed-form formulae of the contact wrench cone for rectangular support areas. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5107–5112. IEEE, 2015.

[60] Z. Li, X. B. Peng, P. Abbeel, S. Levine, G. Berseth, and K. Sreenath. Reinforcement learning for versatile, dynamic, and robust bipedal locomotion control. *arXiv preprint arXiv:2401.16889*, 2024.

[61] A. Kumar, Z. Fu, D. Pathak, and J. Malik. Rma: Rapid motor adaptation for legged robots. In *Robotics: Science and Systems*, 2021.

[62] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter. Learning quadrupedal locomotion over challenging terrain. *Science robotics*, 5(47):eabc5986, 2020.

[63] T. Flayols, A. Del Prete, P. Wensing, A. Mifsud, M. Benallegue, and O. Stasse. Experimental evaluation of simple estimators for humanoid robots. In *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*, pages 889–895. IEEE, 2017.

[64] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[65] M. Mittal, C. Yu, Q. Yu, J. Liu, N. Rudin, D. Hoeller, J. L. Yuan, R. Singh, Y. Guo, H. Mazhar, A. Mandlekar, B. Babich, G. State, M. Hutter, and A. Garg. Orbit: A unified simulation framework for interactive robot learning environments. *IEEE Robotics and Automation Letters*, 8(6):3740–3747, 2023. doi:10.1109/LRA.2023.3270034.

[66] X. B. Peng, E. Coumans, T. Zhang, T.-W. Lee, J. Tan, and S. Levine. Learning agile robotic locomotion skills by imitating animals. *arXiv preprint arXiv:2004.00784*, 2020.

[67] Y.-H. Shin, T.-G. Song, G. Ji, and H.-W. Park. Actuator-constrained reinforcement learning for high-speed quadrupedal locomotion. *arXiv preprint arXiv:2312.17507*, 2023.

[68] N. Rudin, D. Hoeller, P. Reist, and M. Hutter. Learning to walk in minutes using massively parallel deep reinforcement learning. In *Conference on Robot Learning*, pages 91–100. PMLR, 2022.

[69] P. A. Houglum and D. B. Bertoti. *Brunnstrom's clinical kinesiology*. FA Davis, 2011.

[70] N. P. Hamilton. *Kinesiology: Scientific basis of human motion*. Brown & Benchmark, 2011.

[71] P. Ball and G. Johnson. Technique for the measurement of hindfoot inversion and eversion and its use to study a normal population. *Clinical Biomechanics*, 11(3):165–169, 1996.

Appendix

A Reward Function

In this section, we provide the detailed reward functions used to train our policy.

A.1 Walking

The reward function design for walking has four parts. The first part includes tracking terms, implemented as the L^2 norm of the difference between the desired and actual linear velocities in the sagittal and lateral directions, as well as the angular velocity in the yaw.

The second part is the smoothing terms where we penalize non-zero values in the linear velocity in the vertical direction and angular velocities in both roll and pitch. The joint torques and action rates are also penalized. These terms help improve the smoothness of the policy.

Furthermore, we regularize the hip and knee joints with respect to their nominal positions and body orientation with upright orientation. We also set a soft limit for the actuators, over which the actions will be penalized. These regularization terms are beneficial in preventing aggressive and dangerous motions the policy might learn.

Lastly, we include gait quality terms necessary for exhibiting reasonable walking gaits. These terms encourage feet to stay longer in the air [68], to not slip on the ground [65], and to keep contact forces under a threshold to protect the gearboxes and other hardware.

A.2 Hopping

The reward function for hopping is slightly modified from the walking task. First, instead of penalizing vertical linear velocity, we encourage positive linear velocity in vertical direction using a ReLU function, namely, $r_{vz} = \text{ReLU}(\mathbf{v}_z)$. Second, we do not limit knee joints and hip joints in pitch as they are necessary in providing a large upward acceleration in hopping. Additionally, in single-leg hopping, we penalize the in-air leg contacting with the ground. Apart from these, the other terms stay the same as the walking task.

B Outdoor Failure Counts Throughout the Project

Throughout the entire project, we record the failure counts over different terrains as proof of the durability of the robotic hardware. Note that this represents failures during testing and debugging of the hardware, but not the experiments presented above.

Table 4: Number of **Recorded** Falls on Different Surfaces.

Surface	Stone Brick Road	Grassland	Running Track	Unpaved Road
Number	6	14	3	15

C Joint Ranges of the Hardware

As discussed in Sec. 3.5, our hardware follows an anthropomorphic design to approach the range of human movements as much as possible. The ranges for each of the 6 DoFs are recorded in the table below, The joint names and their definitions are as follows:

- HR: Hip Rotation
- HAA: Hip Abduction/Adduction
- HFE: Hip Flexion/Extension

Table 5: Comparsion Between Ranges of Motion Humand Joint and Proposed Robot (right leg). Data from [69], [70], [71].

Joint Names	HR	HAA	HFE	KFE	FFE	FAA
Human [°]	$[-50, 40]$	$[-40, 20]$	$[-110, 30]$	$[0, 150]$	$[-20, 50]$	$[-30, 18]$
Proposed [°]	$[-35, 35]$	$[-35, 35]$	$[-100, 30]$	$[0, 120]$	$[-30, 70]$	$[-30, 30]$
Coverage Rate	77.8%	91.6%	92.9%	80.0%	100.0%	100.0%

- KFE: Knee Flexion/Extension
- FFE: Foot Flexion/Extension
- FAA: Foot Abduction/Adduction

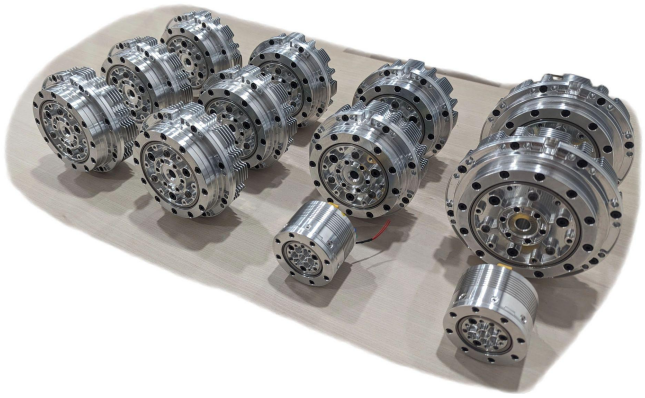


Figure 10: Totally 12 Actuators Used in the Robot.

D Dynamics Randomization Details

As discussed in Sec. 4.2, we designed dynamics randomization carefully to best fit the actual hardware. The ranges are summarized in Table 6 below,

Dynamics Terms	Friction	Restitution	Base Mass	Linkage Mass	Joint Friction	Joint Armature	Default Joint Pos
Low	0.2	0.0	-1.0	x0.9	x0.9	x1.0	-0.05
High	1.25	0.1	+1.0	x1.1	x1.1	x1.05	0.05

Noise Terms	Lin Vel	Ang Vel	IMU	Hip Joints Pos	KFE Pos	FFE Pos	FAA Pos	Joints Vel
Range (±)	0.1	0.2	0.05	0.03	0.05	0.08	0.03	1.5

Table 6: List of domain randomizations. After system identification on the in-house designed hardware, we provide a small range of 6 dynamics parameters and 8 noise terms that minimize the sim-to-real gap.

Sentinel-X Drone

Close Proximity Security Drone

Key Features

High-Resolution Cameras

- 4K Ultra HD: Ensures detailed and sharp visuals for real-time surveillance.
- Low-Light Sensitivity: Effective even in low-light conditions for 24/7 monitoring.
- Real-Time Visual Surveillance: Immediate feedback and live feeds to the secured app or autonomous mode.
- Optical Sensors & Image Processors: Advanced image processing for clear and reliable footage.

Radar System

- Long-Range Detection: Tracks and identifies threats from a distance, even in challenging conditions.
- Motion Tracking: Real-time detection and movement analysis of intruders.
- Intruder Identification & Tracking: Automatically locks on to and follows potential threats.
- Radar Sensors & Signal Processors: Ensures efficient and accurate threat analysis in all weather conditions.

Environmental Sensors

- Temperature, Humidity, and Gas Detection: Monitors environmental changes for proactive threat detection.
- Threat Detection: Identifies hazardous gases, unusual temperatures, and other environmental anomalies.
- Gas Sensors, Temperature Probes, and Humidity Sensors: Provide comprehensive environmental feedback.
- Secured App or Autonomous Modes: Can be monitored or operate autonomously.

Immobilization Mechanism

- Electromagnetic Locks & Pneumatic Restraints: Securely immobilize potential threats.

- Threat Neutralization: Safely restrain and prevent further actions from intruders.

- Electromagnetic Actuators & Pneumatic Controllers: Advanced mechanics ensure smooth and reliable operation.

Deterrent Systems

- Strobes, Sirens, and Non-Lethal Weapons: Designed to disorient and incapacitate intruders.

- Threat Deterrence: Visual and audio systems coupled with non-lethal means of incapacitation.

- LED Strobes, Loudspeakers, & Non-Lethal Weapon Controllers: Robust systems for threat management.

A DEEP NEURAL NETWORK FOR ESTIMATING DEPTH FROM STEREO

Nikolai Smolyanskiy, Alexey Kamenev, Stan Birchfield

NVIDIA

Project Redtail

GTC 2018

AGENDA

Why Deep Learning for depth computation?

Our end-to-end stereo depth DNN

Supervised, unsupervised, semi-supervised training

Our stereo DNN vs mono DNN vs traditional stereo

Inference runtime

Performance metrics

WHY DL FOR DEPTH COMPUTATION?

Accurate depth is needed in

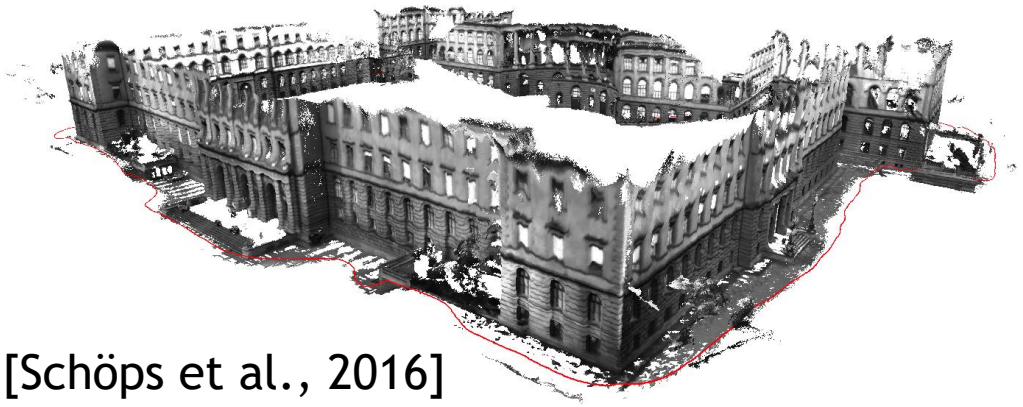
3D reconstruction

Robotic manipulation

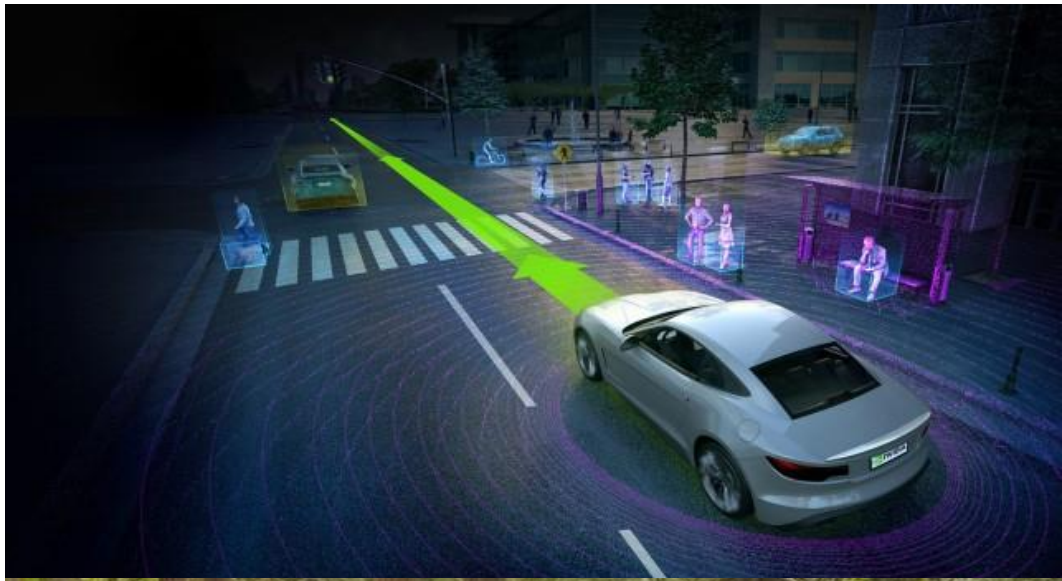
Robotic navigation

Self-driven cars

Augmented reality



[Schöps et al., 2016]



[Smolyanskiy et al., IROS 2017]

WHY DL FOR DEPTH COMPUTATION?

DNNs are more accurate than CV stereo and more practical than Lidars

LIDARs are accurate, but bulky, expensive, have narrow angle and run at 10 FPS

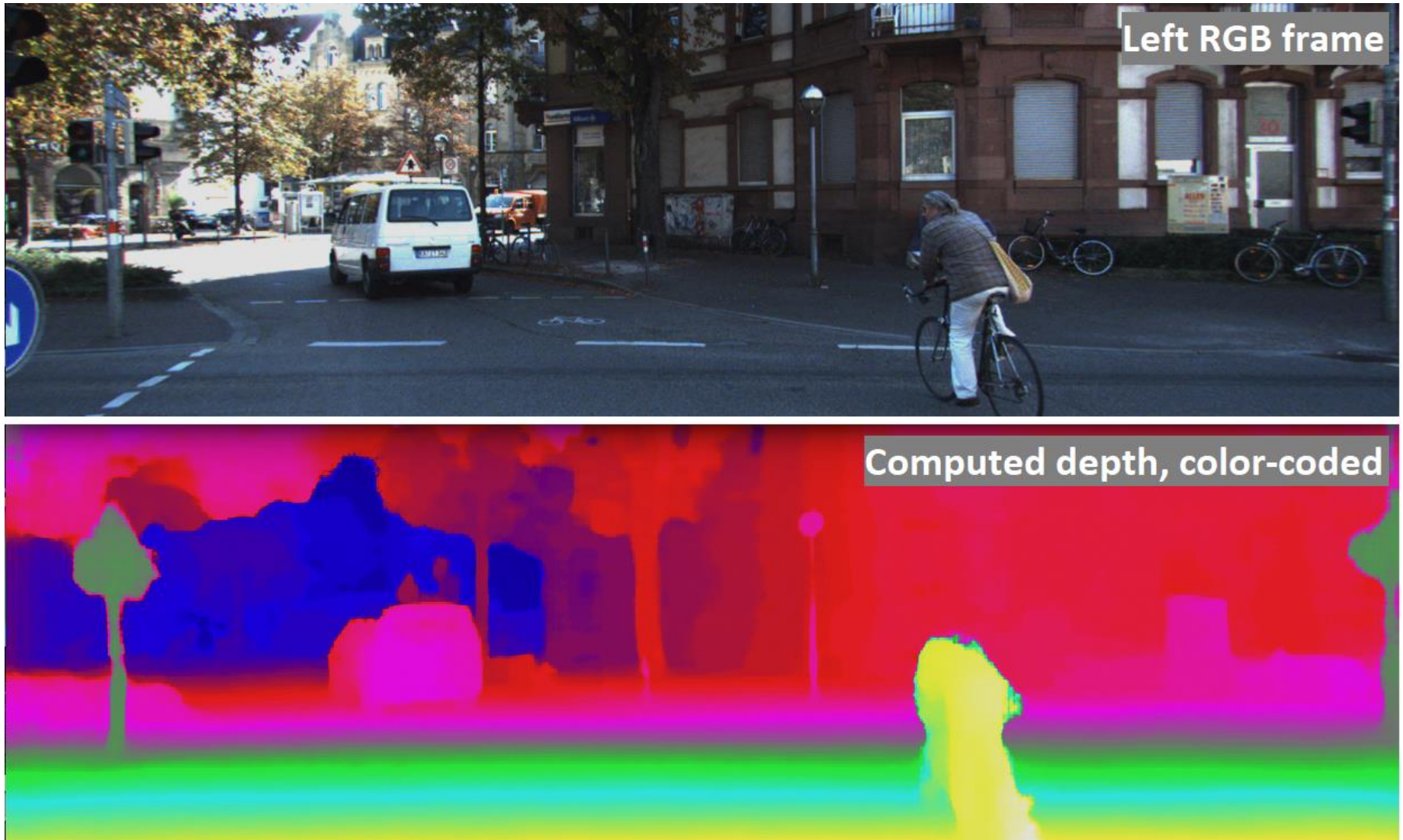
Traditional stereo matching methods are inaccurate

- [Mroz and Breckon, 2012] http://breckon.eu/toby/demos/autostereo/

Deep learning methods provide dense accurate depth and are only bound by compute

WHY DL FOR DEPTH COMPUTATION?

Our stereo depth DNN produces accurate and clean depth



OUR DEEP LEARNING APPROACH

We were inspired by 2 DNN architectures

GC-Net: "End-to-End Learning of Geometry and Context for Deep Stereo Regression" [Kendall et al. 2017, Skydio]

Monodepth: "Unsupervised Monocular Depth Estimation with Left-Right Consistency" [Godard et al. 2017]

Our contributions:

- We can train in supervised, unsupervised and semi-supervised modes
- Simpler than GC-Net architecture: we use ELU and no batch norm.
- Novel "machine learned" argmax
- A smaller version runs in near real-time on desktop GPUs
- Our custom inference runtime allows running on Jetson TX2

DEPTH FROM DISPARITY

Short review of stereo methods

Passive stereo techniques compute depth from disparity

Disparity is a distance between corresponding points on epipolar lines

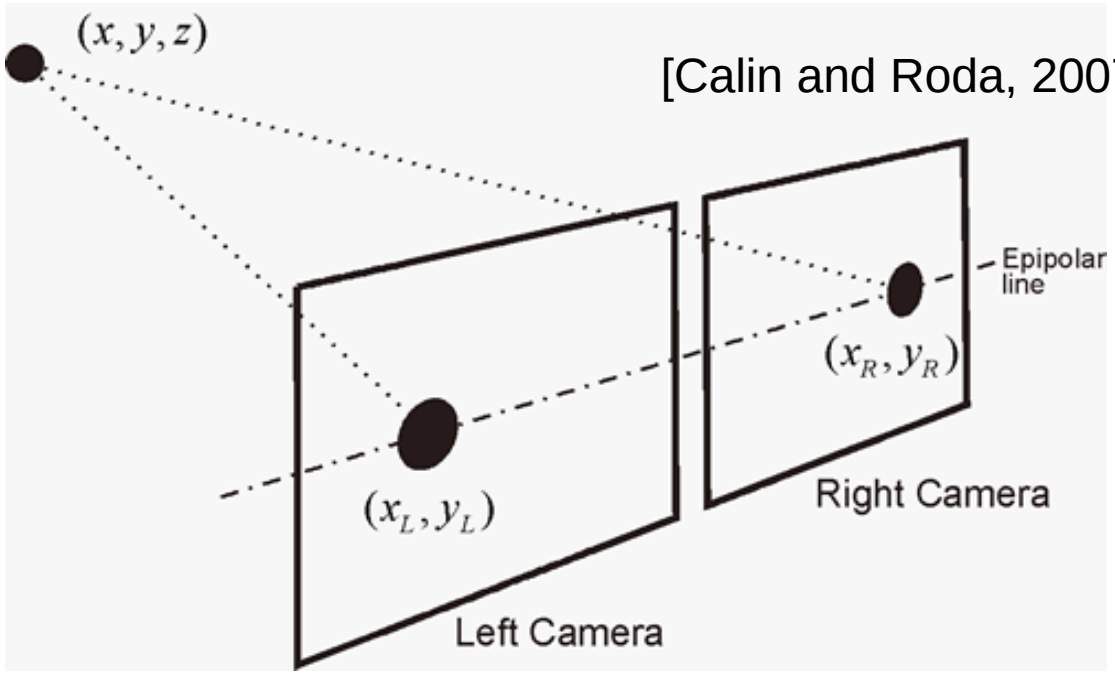
Depth is inversely proportional to disparity:

$$depth = \frac{Bf}{x_L - x_R}$$

$where$:

B $-$ $camera\ baseline\ in\ meters$

f $-$ $focal\ length\ in\ pixels$



[Calin and Roda, 2007]

STEREO MATCHING VIA COST VOLUME

Stereo matching via exhaustive search

We can build a 3D volume of match costs for all pixels for all disparities

Then a disparity for a given pixel can be computed as:

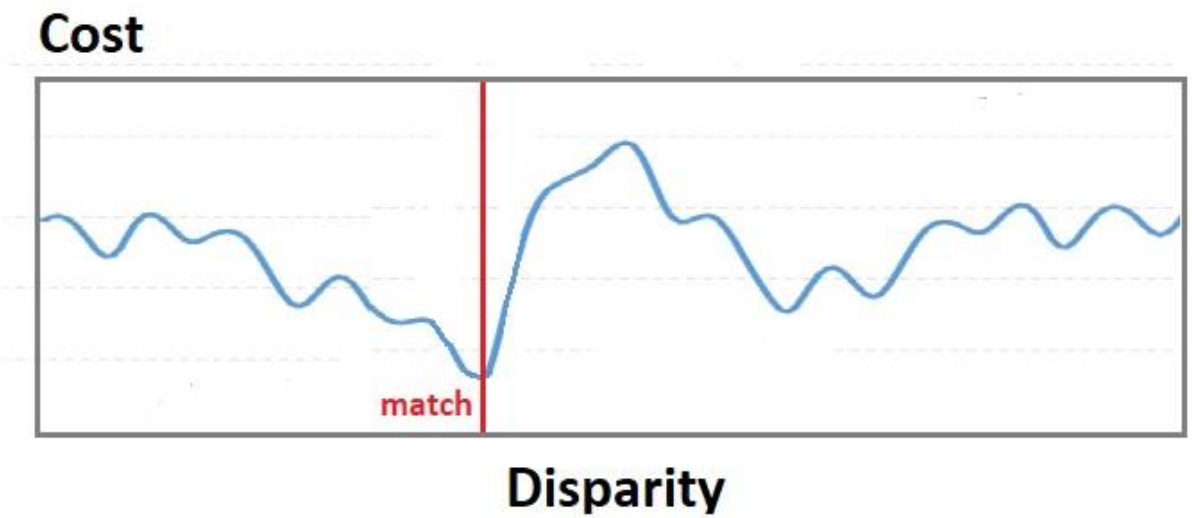
$$disparity = argmax(p_i);\ p\ is\ a\ pdf\ built\ from\ costs\ for\ this\ pixel\ (e.g.\ via\ softmax)$$

Argmax is not differentiable. We can use "soft-argmax" instead [Kendall et al., 2017]

$$disparity = \sum_i p_i d_i\ ;\ where\ d_i - disparity\ level$$

Soft-argmax is differentiable and

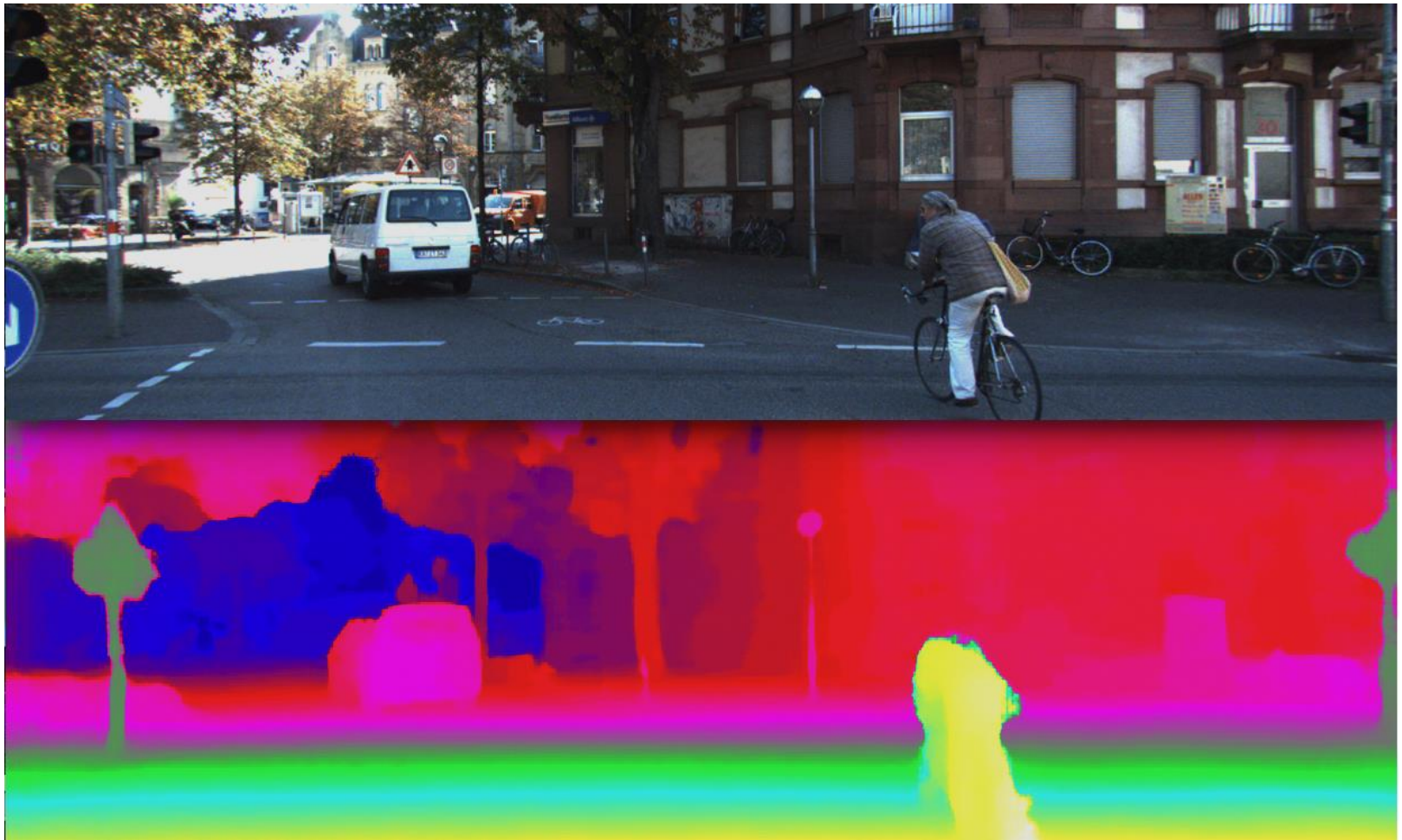
can be used to train DNNs



VIDEO DEMO

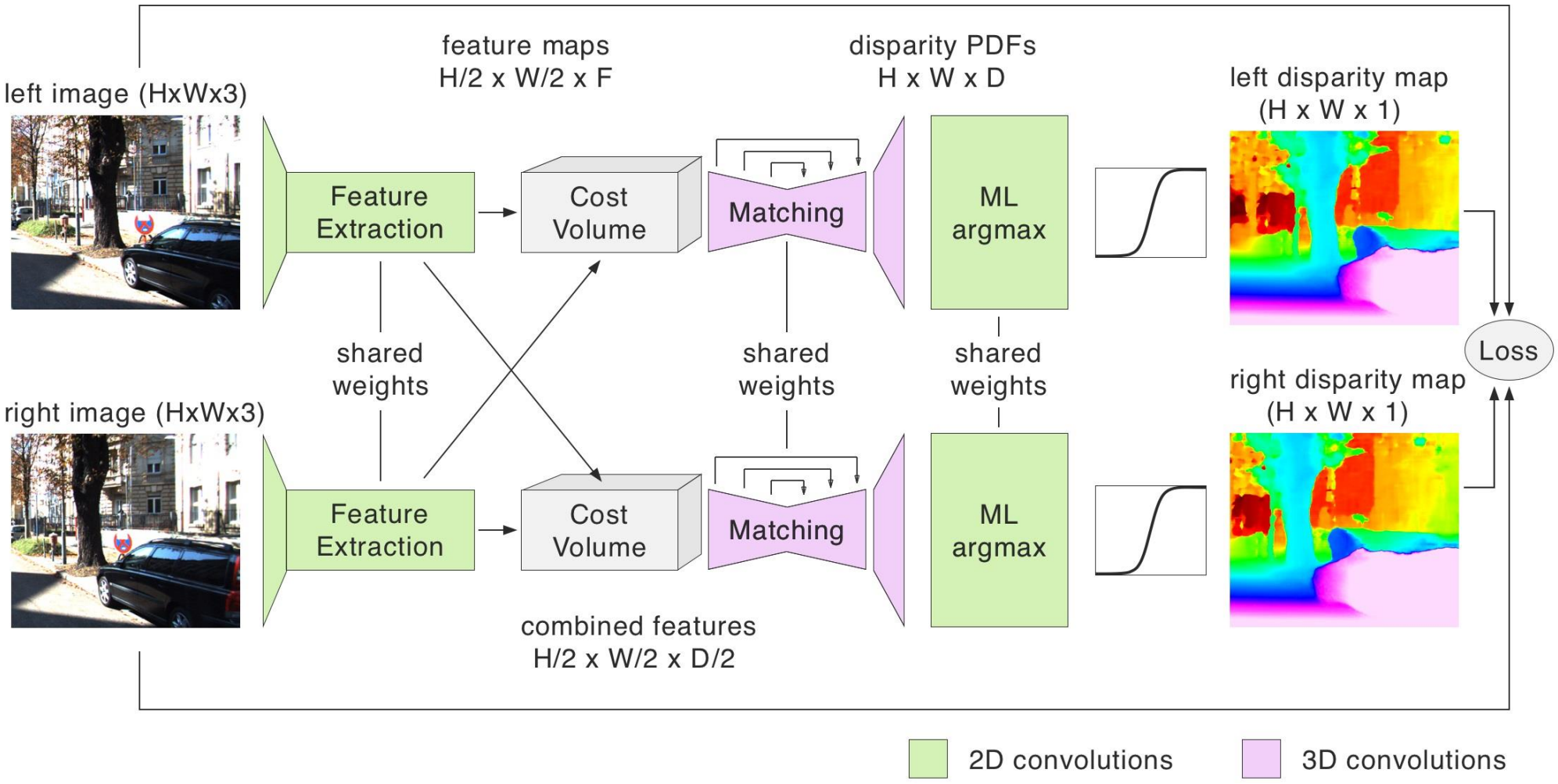
This video demonstrates our stereo DNN depth results

https://youtu.be/0FPQdVOYoAU



STEREO DNN ARCHITECTURE

Our architecture mimics traditional stereo pipeline



DIFFERENT SIZE MODELS

We created several models to test performance

Large: Resnet18 like for 2D input -> cost volume -> 3D encoder/decoder -> soft-argmax

Large: Resnet18 like for 2D input -> cost volume -> 3D encoder/decoder -> ML-argmax

Small: AlexNet as 2D input -> small cost volume -> small 3D encoder/decoder -> soft-argmax

Variations:

- Use 1 tower instead of 2 for training
- Use correlation instead of feature concatenation in cost volume
- Use different constraints in the loss

LOSS FUNCTION

Has unsupervised photometric terms and supervised L2 disparity terms

$$L = \lambda_1 E_{image} + \lambda_2 E_{lidar} + \lambda_3 E_{lr} + \lambda_4 E_{ds}, \quad (1)$$

where

$$E_{image} = E^l_{image} + E^r_{image} \quad (2)$$

$$E_{lidar} = |d_l - \bar{d}_l| + |d_r - \bar{d}_r| \quad (3)$$

$$E_{lr} = \frac{1}{n}\sum_{ij} |d^l_{ij} - \tilde{d}^l_{ij}| + \frac{1}{n}\sum_{ij} |d^r_{ij} - \tilde{d}^r_{ij}| \quad (4)$$

$$E_{ds} = E^l_{ds} + E^r_{ds} \quad (5)$$

LOSS FUNCTION

Continued

$$E^l_{image} = \frac{1}{n}\sum_{i,j} \alpha \frac{1 - SSIM(I^l_{ij}, \tilde{I}^l_{ij})}{2} + (1-\alpha)|I^l_{ij} - \tilde{I}^l_{ij}|$$

$$E^l_{ds} = \frac{1}{n}\sum_{i,j} |\partial_x d^l_{ij}| e^{-\|\partial_x I^l_{i,j}\|} + |\partial_y d^l_{ij}| e^{-\|\partial_y I^l_{i,j}\|}$$

LOSS FUNCTION

Continued

$$\tilde{I}^l = w_{rl}(I_r, d_l) \quad (6)$$

$$\tilde{I}^r = w_{lr}(I_l, d_r) \quad (7)$$

$$\tilde{d}^l = w_{rl}(d_r, d_l) \quad (8)$$

$$\tilde{d}^r = w_{lr}(d_l, d_r) \quad (9)$$

$$w_{lr}(I, d) = (x, y) \mapsto I(x - d(x, y), y) \quad (10)$$

$$w_{rl}(I, d) = (x, y) \mapsto I(x + d(x, y), y) \quad (11)$$

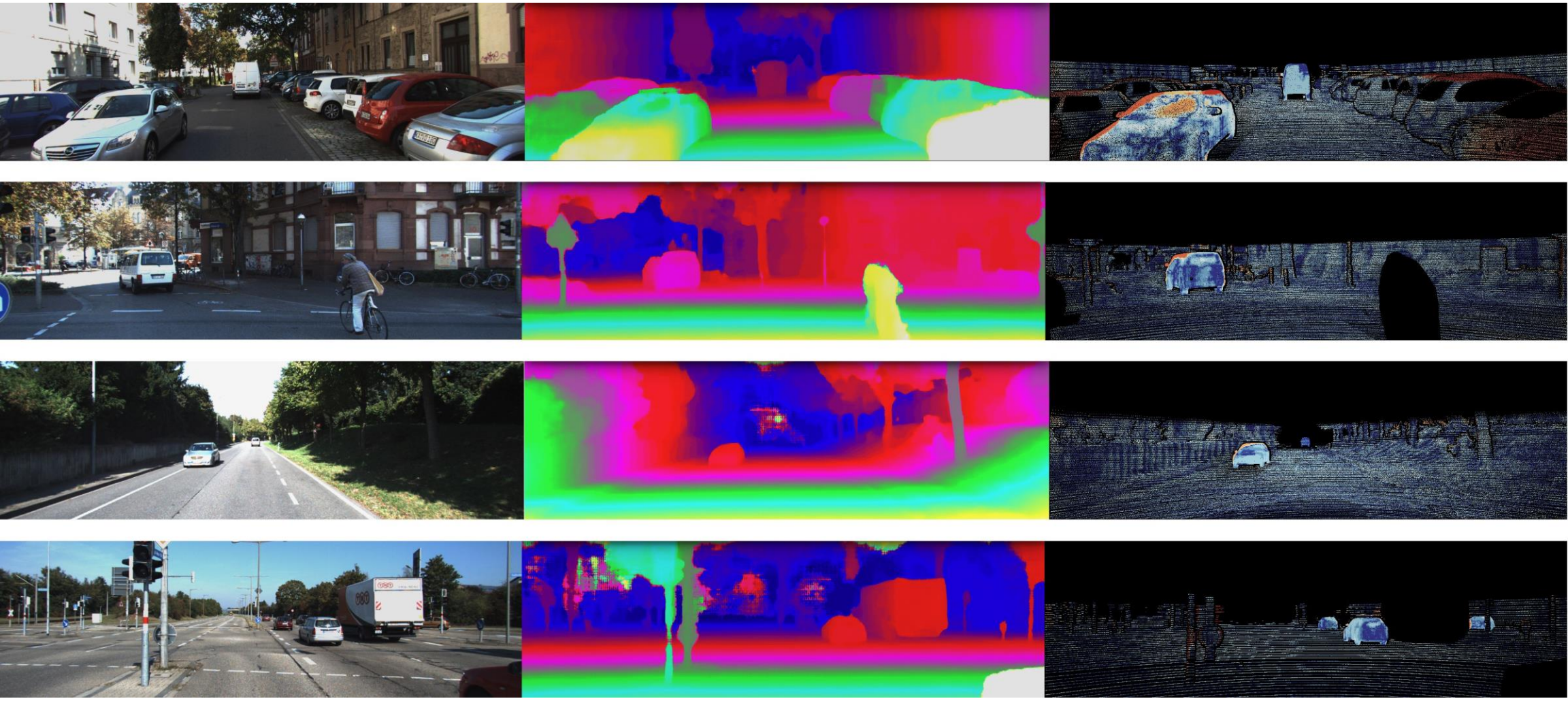
$$SSIM(x, y) = \left(\frac{2\mu_x \mu_y + c_1}{\mu_x^2 + \mu_y^2 + c_1} \right) \left(\frac{2\sigma_{xy} + c_2}{\sigma_x^2 + \sigma_y^2 + c_2} \right) \quad (12)$$

COST VOLUME CREATION

Model code in TensorFlow

```
def cost_volume_left_block(self, left, right, max_disp_steps, scope_name) :

    height = int(left.shape[1])
    width = int(left.shape[2])
    depth = int(left.shape[3])

    with tf.variable_scope(scope_name) as scope:
        right_padded = tf.pad(right,
            [[0, 0], [0, 0], [max_disp_steps-1, 0], [0,0]], "CONSTANT")
        right_disp = tf.extract_image_patches(right_padded,
            [1,height,width,1], [1,1,1,1], [1,1,1,1], padding="VALID")
        right_disp = tf.squeeze(right_disp, axis=1)
        disparity_dim = int(right_disp.shape[1])
        right_disp = tf.reshape(right_disp, [-1, disparity_dim, height, width, depth])
        right_disp = tf.reverse(right_disp, [1])

        left_disp = tf.expand_dims(left, axis=1)
        left_disp = tf.tile(left_disp,[1,disparity_dim,1,1,1])

        cost_volume = tf.concat([left_disp, right_disp], axis=4)

    return cost_volume
```

RESULTS



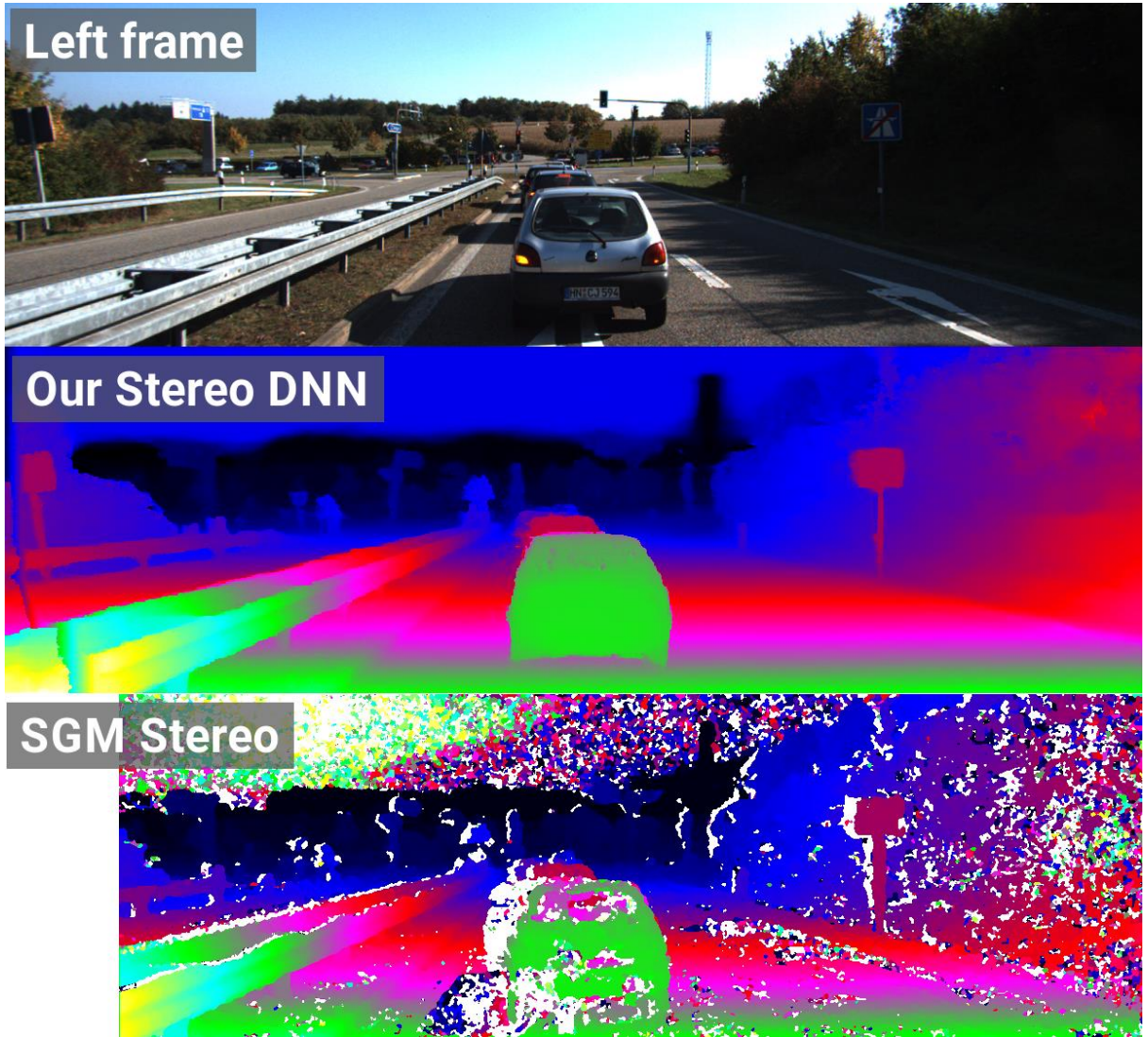
Left RGB frames

Computed depth, color-coded

Error maps |depth-lidar|

OUR STEREO DNN VS SEMI-GLOBAL MATCHING

Stereo DNN creates cleaner and more accurate depth



MONO DNN VS OUR STEREO DNN

Question: Can you guess what is the real geometry here?

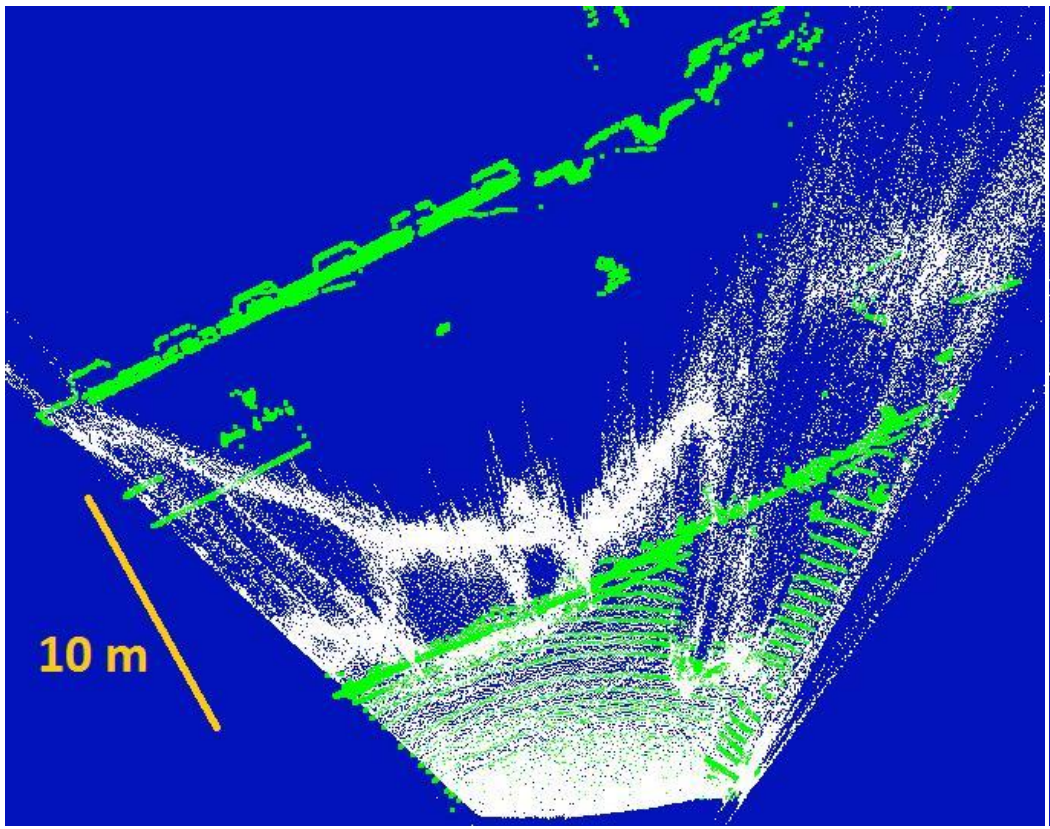
What is the distance to the fence (B)?

What is the distance from the fence (B) to the building (D)?

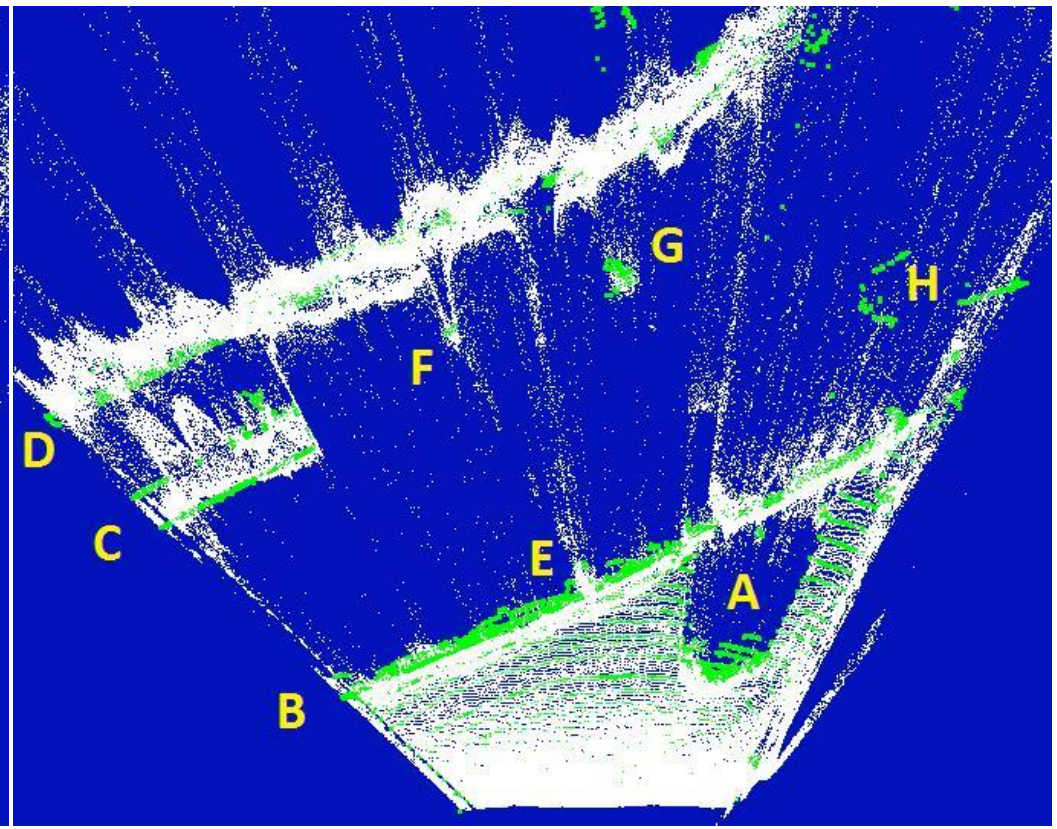
VIDEO DEMO (CONTINUED)

Answer: Mono DNN, Stereo DNN, LIDAR point clouds for that street view

https://youtu.be/0FPQdVOYoAU

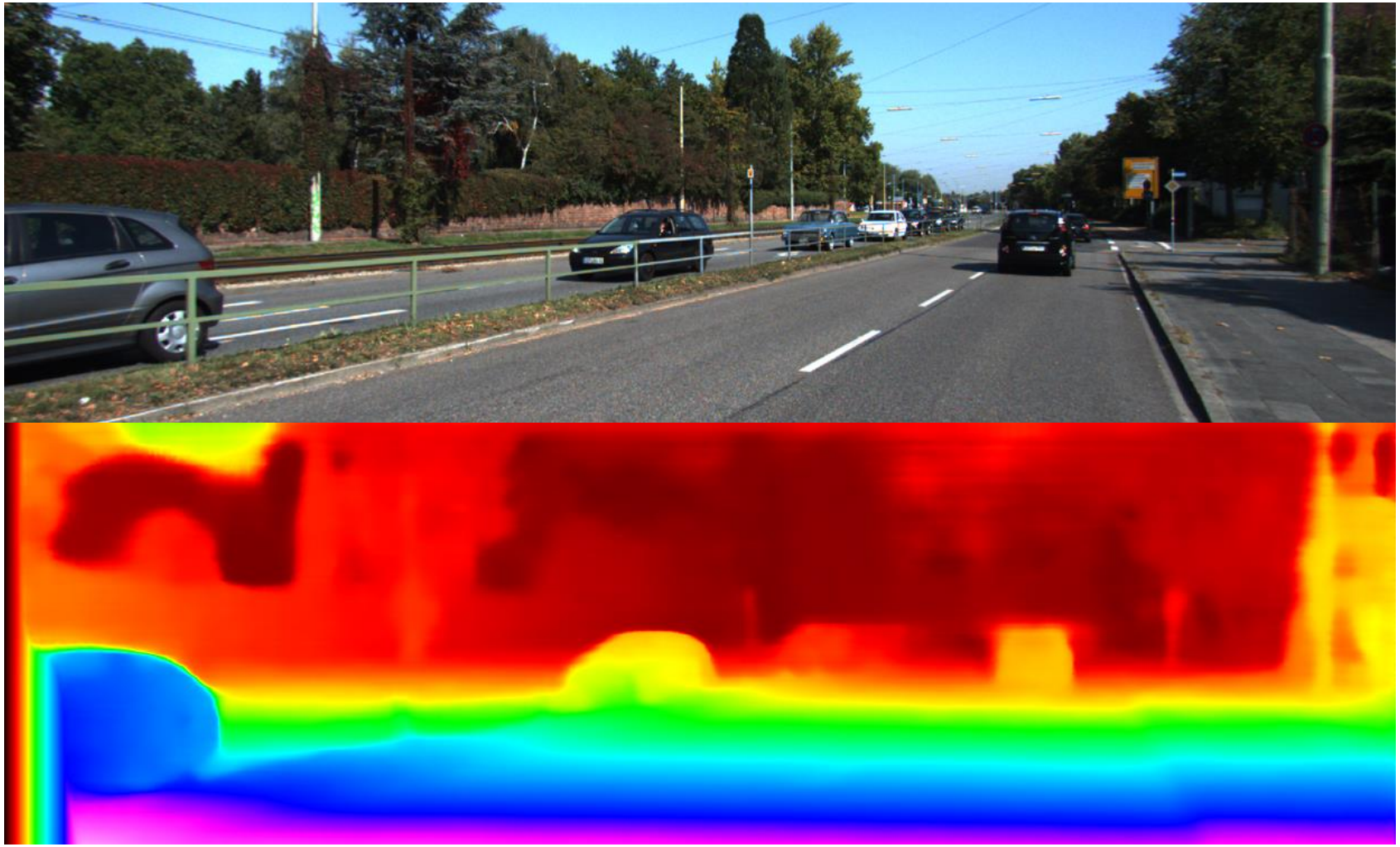


Mono DNN point cloud (white), LIDAR (green), top-down view

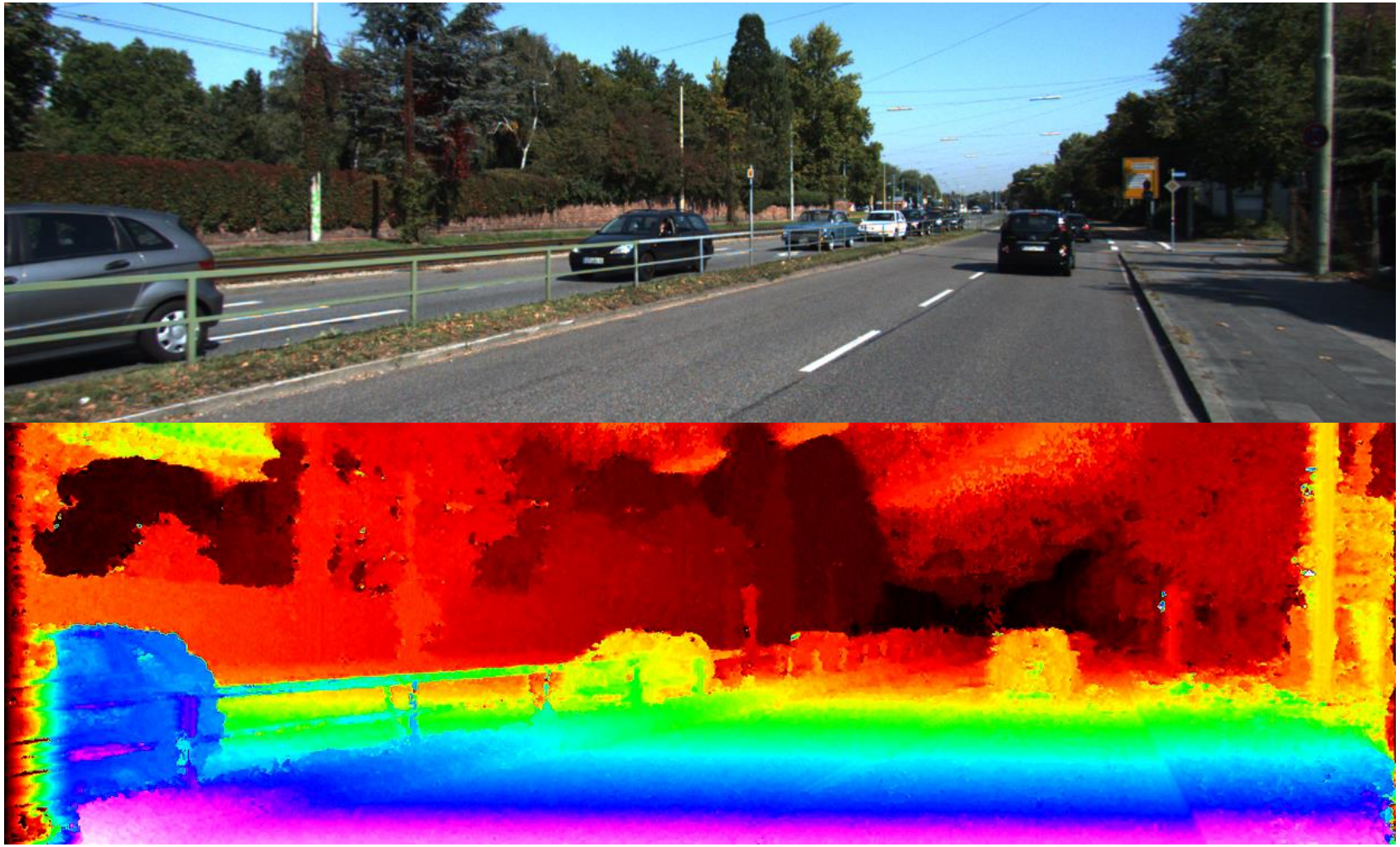


Our stereo DNN point cloud (white), LIDAR (green), top-down view

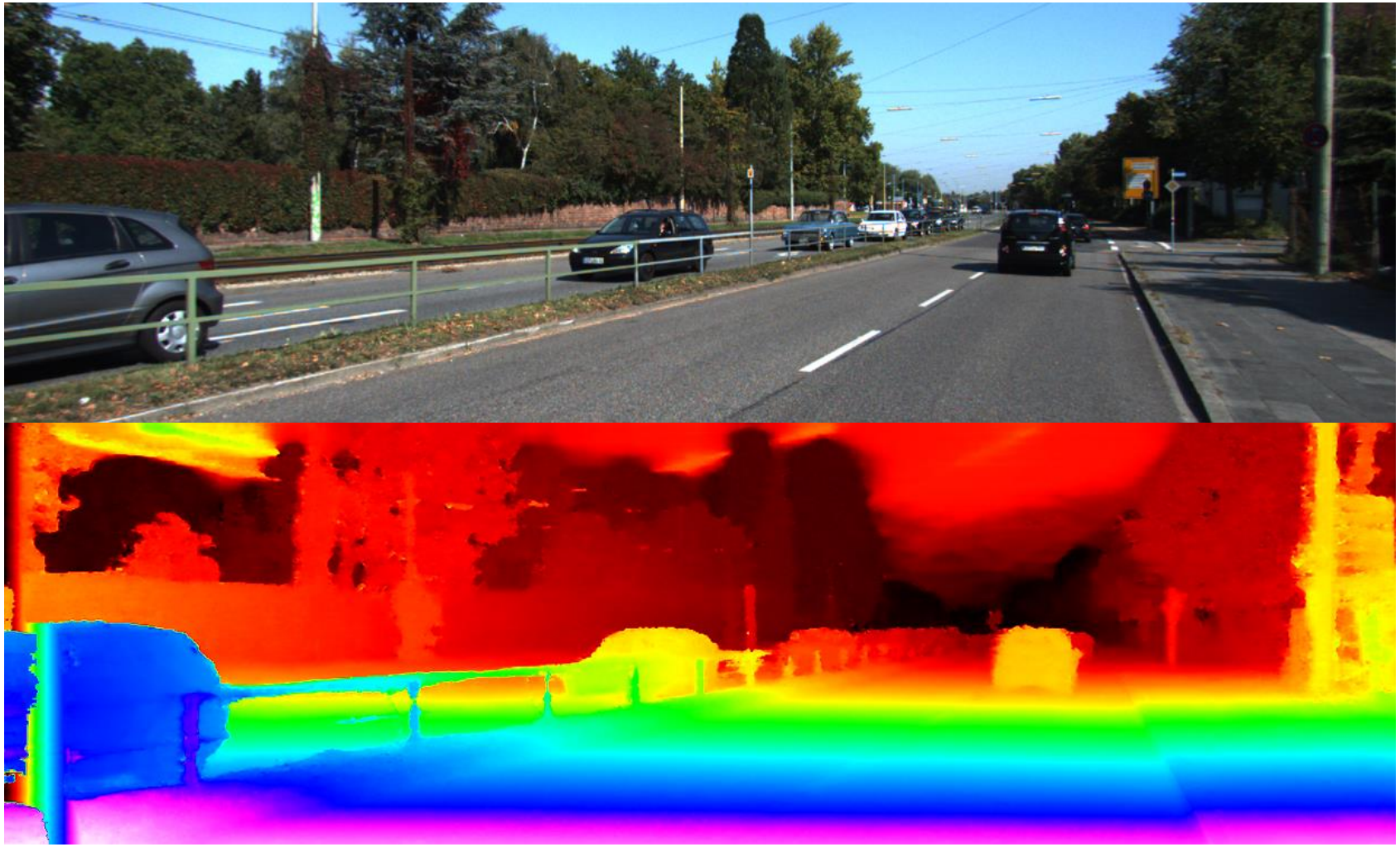
SUPERVISED BY LIDAR



UNSUPERVISED

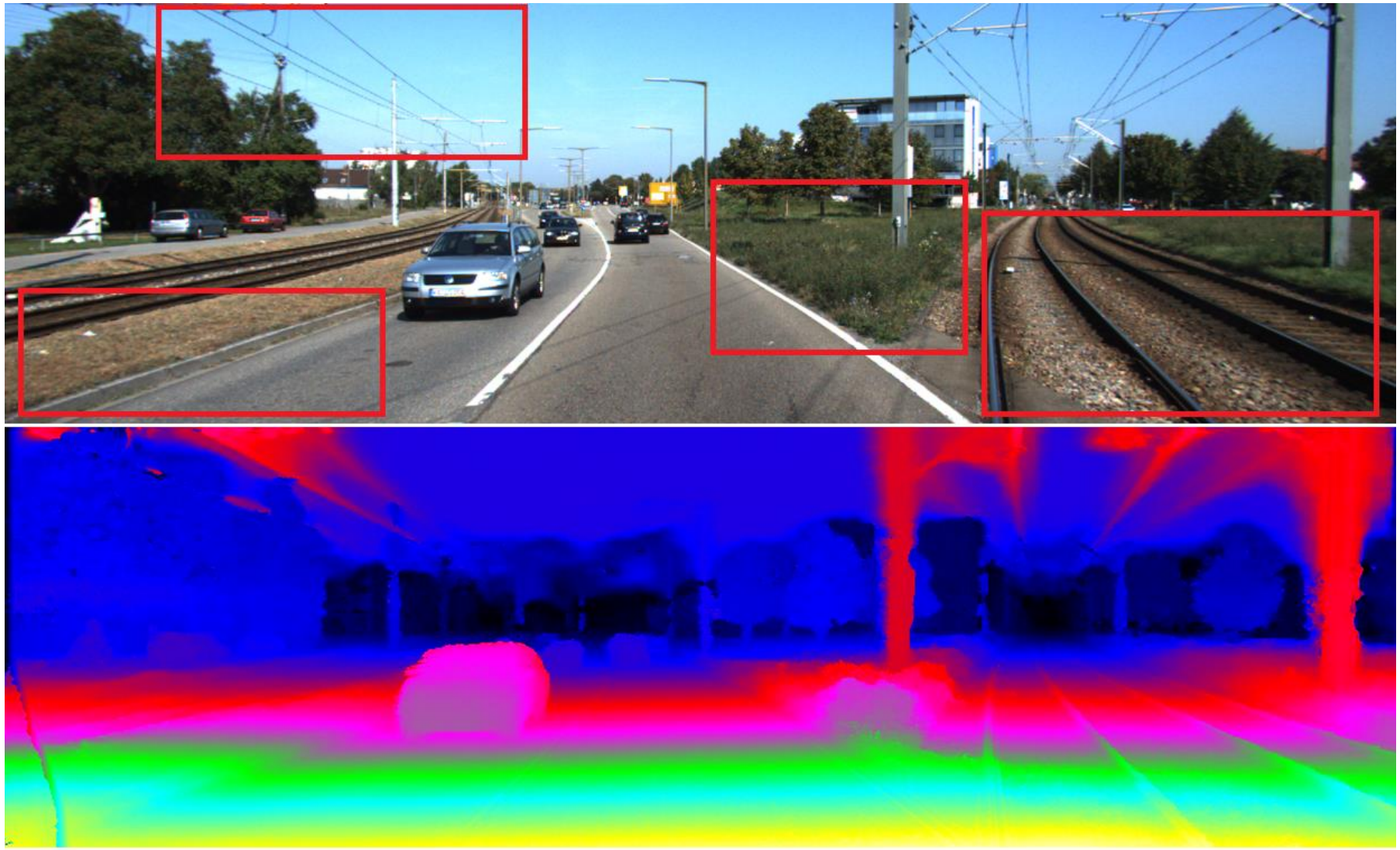


SEMI-SUPERVISED

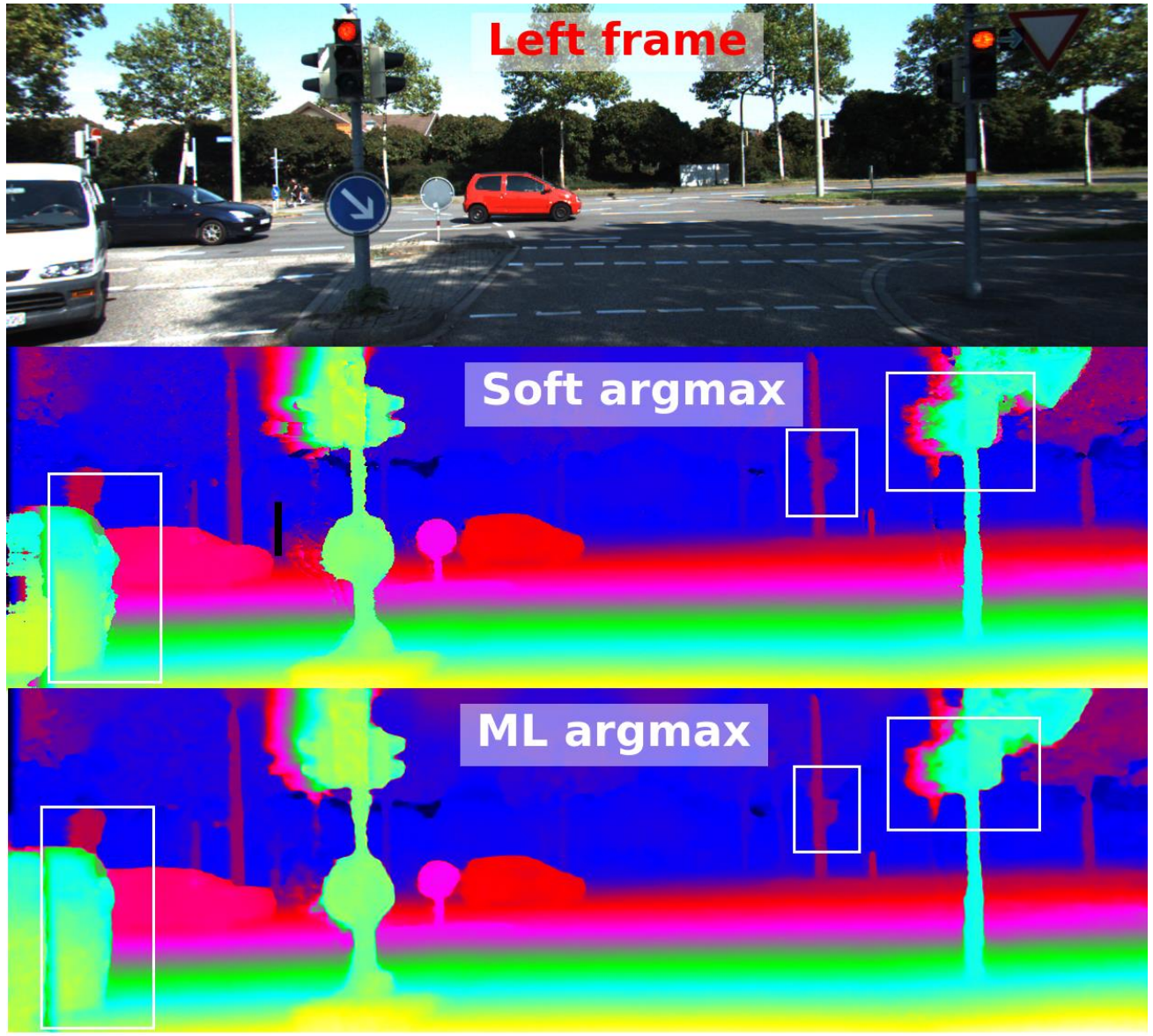


FINE GRAIN DETAILS

Stereo DNN is capable of capturing wires, rails, curbs, grass, etc.

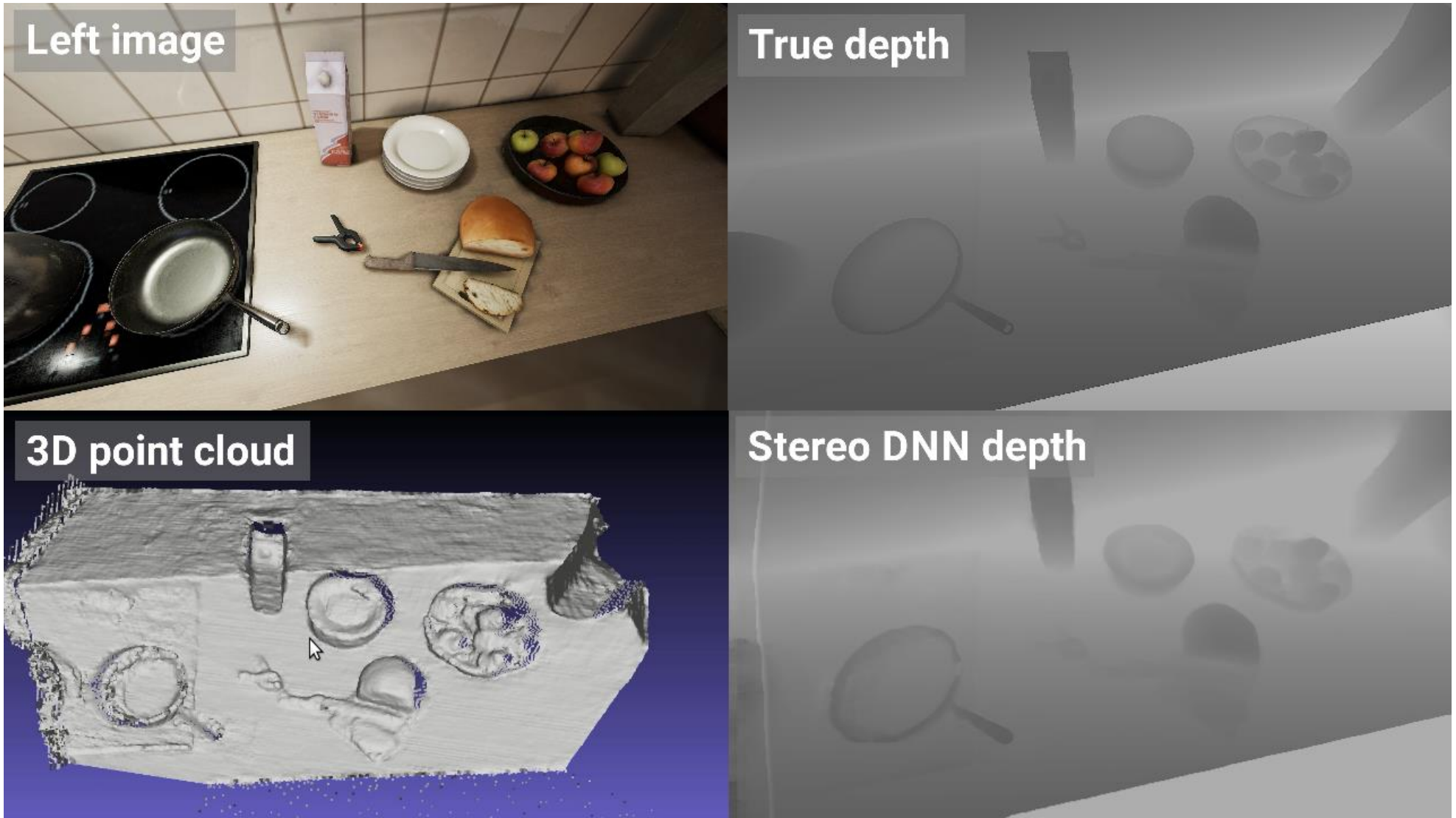


SOFT ARGMAX VS ML-ARGMAX



SYNTHETIC MODEL

We also trained and tested on synthetic 3D scenes



COMPARING MODELS ON KITTI 2015

This table shows our **KITTI D1 error**: % of pixels where disparity error is more than 3 pixels close range or more then than 5% further out

We show D1 error for models trained on **raw dataset** (sparse LIDAR, 29K frames)

Model / Training mode	Supervised w/ Lidar	Unsupervised Photometric	Semi-supervised Lidar + Photo
Mono Depth		32.8%	
Correlation based	14.6%	13.3%	12.9%
Our stereo DNN	15.0%	12.9%	8.8%

Semi-supervised mode with Lidar + Photo yields better results

COMPARING MODELS ON KITTI 2015

Numbers in red are for models trained on 200 scenes with **densified LIDAR depth**

Most papers report models trained on 200 scenes with densified LIDAR depth

When we fine-tune on those 200 dense scenes, we are in top 10 KITTI 2015 stereo

Model	size	Lidar + photo D1 error
No bottleneck	0.2M	14.5%
Correlation	2.7M	12.9%
Small	1.8M	9.8%
Tiny (near real-time)	0.5M	11.9%
Single tower	2.8M	10.1%
Resnet18 based (our baseline)	2.8M	8.8%
ML-argmax	3.1M	8.7%
ML-argmax + dense depth	3.1M	3.5%
Resnet18 based + dense depth	2.8M	3.4%

KITTI 2015 BENCHMARK

Most papers report models trained on 200 scenes with densified LIDAR depth

When we fine-tune on those 200 dense scenes, we are in top 10 KITTI 2015 stereo

Model	D1-background	D1-foreground	D1-All
DispNetC	4.3%	4.4%	4.3%
SGM-Net	2.7%	8.6%	3.7%
GC-Net	2.2%	6.2%	2.9%
CRL	2.5%	3.6%	2.7%
L-ResMatch	2.7%	7.0%	3.4%
Ours (no-finetuning)	3.2%	14.8%	5.1%
Ours (finetuned on dense 200)	**2.7%**	**6.0%**	**3.4%**

INFERENCE RUNTIME

Project Redtail has runtime inference lib on GitHub

The library implements operations currently not available in TensorRT

Operations are implemented as custom TensorRT plugins

To run, use 2-step process:

- Convert TensorFlow binary model to TensorRT C++ API code
- Use generated C++ code in your TensorRT inference code

Note: TensorFlow runtime is NOT required to run our stereo DNN

INFERENCE RUNTIME

Our custom TensorRT plugins

3D convolutions and transposed 3D convolutions aka deconvolutions

- TensorFlow and cuDNN have different implementations of 3D convolution
- TensorFlow’s 3D convolution can be represented in cuDNN by reshaping and proper stride/padding calculation

INFERENCE RUNTIME

Our custom TensorRT plugins

ELU activation function

Cost volume plugin (stereo DNN specific)

Multidimensional soft-argmax plugin

Auxiliary plugins necessary for stereo DNN model:

- Tensor transforms/transpose
- Padding (due to asymmetric padding in TensorFlow)
- Slicing (same reason as for padding)

INFERENCE PERFORMANCE

On Different NVIDIA GPUs

Model	Resolution	D1 error	Titan XP performance (ms)		Jetson TX2 perf (ms)
		%	TensorFlow	TensorRT	TensorRT
ResNet-18	1025x321	3.4	950	650	11000
NVSmall	1025x321	9.8	800	450	7800
NVTiny	513x161	11.1	75	40	360

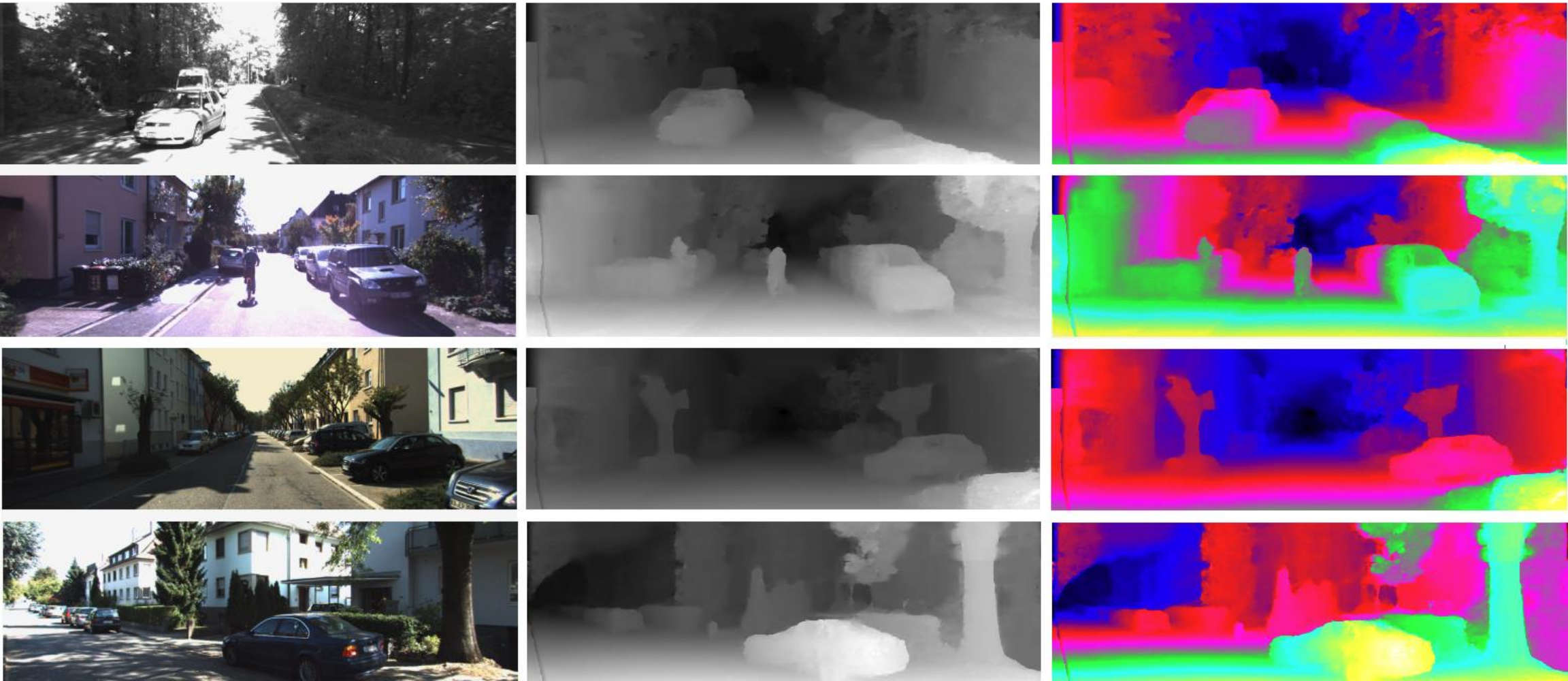
Notes:

- D1 error for ResNet-18 was measured on KITTI 2015 benchmark 200 test images. The model was fine-tuned on 200 train images (with dense LIDAR) after training on full KITTI
- D1 error for NVSmall and NVTiny was measured on KITTI 2015 benchmark 200 training images. These models were trained on full KITTI (with sparse LIDAR)
- FP16: at the moment, 3D convolutions in cuDNN are not optimized for FP16

INFERENCE ON JETSON

NVTiny model runs at 3 FPS on TX2

Stereo DNN depth output with left frame input. Model: "NVTiny", semi-supervised, 512x160, 48 max disparity, trained on KITTI



CONCLUSIONS AND FUTURE WORK

We can train fairly accurate stereo DNN end-to-end

Stereo DNNs not only do matches, but also understand context

Better accuracy is needed around fine branches, poles, etc.

Cannot yet estimate depth of textureless objects at infinity

More work needed to run depth DNNs on embedded GPUs in real-time