Deep Meta-Learning Energy-Aware Path Planner for Unmanned Ground Vehicles in Unknown Terrains

Marco Visca, Roger Powell, Yang Gao and Saber Fallah

Abstract—This paper presents an adaptive energy-aware prediction and planning framework for vehicles navigating over terrains with varying and unknown properties. A novel feature of the method is the use of a deep meta-learning framework to learn a prior energy model, which can efficiently adapt to the local terrain conditions based on small quantities of exteroceptive and proprioceptive data. A meta-adaptive heuristic function is also proposed for the integration of the energy model into an A* path planner. The performance of the proposed approach is assessed in a 3D-body dynamic simulator over several typologies of deformable terrains, and compared with alternative machine learning solutions. We provide evidence of the advantages of the proposed method to adapt to unforeseen terrain conditions, thereby yielding more informed estimations and energy-efficient paths, when navigating on unknown terrains.

Index Terms—Unmanned ground vehicles, driving energy prediction, path planning, deep meta-learning, machine learning

I. INTRODUCTION

UNMANNED ground vehicles (UGVs) are well-suited to the long-term exploration of unknown challenging environments. From military applications to search and rescue activities and planetary exploration, UGVs can be crucial to extend the range of operations to otherwise precluded or dangerous areas. In this context, driving energy consumption plays a major role in the success and the efficiency of the mission [1] [2]. Long-term exploration UGVs spend most of their time moving and have often limited onboard power. Therefore, enabling the autonomous platform with accurate driving energy prediction and planning can be beneficial to minimize the time spent recharging, allowing for coverage of longer distances and reducing the cost of the operations.

Two main factors are responsible for driving energy consumption: the geometry of the terrain and its terramechanical properties, which denote the mechanical characteristics of the terrain and its response to vehicular loading and shear stress [3]. Particularly, in off-road applications, numerous types of terrains with different characteristics, ranging from snow, mud, sand,

and countless intra-classes, may be encountered. However, the main challenge with planning energy-efficient paths in natural environments is that the terrain properties are *a priori* unknown, can constantly vary, and are cumbersome to estimate in real-time. Moreover, the wheel-terrain interaction and its effect on driving energy can be challenging to model in natural terrains. Indeed, state-of-the-art terramechanical models are often based on the estimation of semi-empirical parameters, and demand excessive computational workload for the real-time implementation on board the autonomous platform [4].

In this context, deep neural networks (DNNs) can represent a convenient asset, as they do not require explicit domain knowledge to be integrated into the prediction algorithm and they can be efficiently implemented into parallel computing architectures [5]. Artificial neural networks have demonstrated remarkable capabilities to improve the autonomy of robotic vehicles. This includes terrain classification [6] [7] [8] [9], slip prediction [10], tracking control [11], and terrain properties estimation [12] among the others. However, performances in such tasks are normally evaluated after extensive incremental tests over large datasets which must be representative of the final operational environment. This makes the standard application of deep learning methods non-trivial in the context of wheel-terrain interaction for off-road robotics. Namely, the collection of a representative and sufficiently large training dataset is often unmanageable due to the unknown and ever-changing properties of the encountered terrains.

To address the aforementioned challenges, we devise an algorithm that alleviates the need to learn a single global model by allowing it to be automatically adapted to different scenarios based on small numbers of recent observations (see Figure 1). Previous works suggest one possible strategy to achieve rapid learning from sparse data, which is referred to as meta-learning [13]. Meta-learning takes inspiration from how humans and animals almost instantaneously adapt in a variety of contexts (e.g. adaptation to unforeseen physical changes). Such rapid adaptation is possible as learning never occurs from scratch, but previous experience is used as a prior to learn faster [14].

The second aspect considered in this paper is that of energy-aware path planning optimization. In line with existing methods, we exploit an A* graph search optimization approach. However, while most of the previous works have considered deterministic energy models, little attention has been given to the integration of neural-network-based energy predictors into planning frameworks and their effects on the planning computational efficiency and optimality performance.

The contributions of this paper can be summarised as follows:

1) We extend the deep-meta-learning framework to the prob-

Manuscript received June 20, 2021; accepted x x, 202x.

This work has been carried out within the framework of the EUROfusion Consortium and has received funding from the Euratom research and training programme under grant agreement No 633053. The views and opinions expressed herein do not necessarily reflect those of the European Commission.

Marco Visca and Saber Fallah are with the Connected and Autonomous Vehicles Lab (CAV Lab), University of Surrey, Guildford, GU2 7XH, UK. Email: {m.visca, s.fallah}@surrey.ac.uk

Yang Gao are with the Space Technology for Autonomous and Robotic Laboratory (STAR LAB), Surrey Space Centre, University of Surrey, Guildford, GU2 7XH, UK. Email: yang.gao@surrey.ac.uk

Roger Powell is in the Cybernetics Group, Remote Applications in Challenging Environments, UK Atomic Energy Authority, Culham Science Centre, OX14 3DB.

Fig. 1: The overall schematic of the proposed framework. An initial set of datasets is collected from different terrain types. The datasets are used to form the prior knowledge of a deep neural network trained in a meta-learning fashion. Then, the network is deployed into the UGV, which navigates into new environments with unknown and varying terrain properties. Small numbers of new data are gathered and exploited to update the meta-learning energy model. Finally, the meta-adaptive energy predictor is integrated into the A* path planner, enabling energy-aware path planning optimization. The implementation of our method is made available at [META-UGV](#).

lem of learning and adapting the energy prediction model of an UGV, which traverses terrains with unknown and varying terramechanical properties (Section III). Within this framework, a deep neural network is trained to implicitly pursue a two-fold objective: (1) gathering prior knowledge shared across different terrains, and (2) retrieving, from few data, terrain-specific representations to be used for fast adaptation.

2) We propose a method to integrate the adaptive energy estimator into an A* path planner (Section IV). Particularly, in our approach, we exploit the meta-learning formulation to adapt both the energy predictor and the A* heuristic function, based on the most recent local data.

3) We provide evidence of the model performance in simulation over several typologies of terrains, ranging from snow, dry sand, sandy loams, compact and wet clay, and Martian and Lunar simulants (Section V).

4) We compare the performance of our method with alternative machine learning solutions (Section VI). Specifically, we show that the meta-learning formulation enables broader generalization, in the low-data regime, than standard non-adaptive models.

5) We compare our method with alternative graph search heuristic methods used in driving energy optimization problems (Section VII). In this way, we demonstrate the potential benefit of our approach, when navigating over unknown terrains, to provide more informed estimations and more energy-efficient paths than a non-adaptive energy predictor (Section VII-A), and to reduce the computational time of planning while retaining close-to-optimal solutions compared with a non-adaptive admissible heuristic function (Section VII-B).

II. RELATED WORKS

In the context of autonomous robotic navigation in challenging scenarios, several works have considered the environment as composed of a finite set of known terrain classes [6] [9]. Within this framework, terrain classifiers can be used to identify the type of terrain, while sensors such as LIDAR or stereo vision provide the geometric information. Hence, different energy models can be adopted, such as semi-empirical functions [15] [16] [17], look-up tables [18], or neural networks [19] [20] to link the terrain geometry to the energy consumption for each terrain type. The main drawback of these methods is that they cannot account for terrains with unknown properties.

Given the challenges to model all possible terrain variations in advance, online adaptation algorithms can represent a useful asset. Particularly, most of the existing contact models, having a suitable computational workload for real-time planning, are based on the Bekker-Wong theory, a semi-empirical method that combines physics considerations with the evaluation of experimental soil-dependent parameters [3] [4]. Several methods have been proposed to estimate online the Bekker's parameters exploiting proprioceptive data [21] [22]. Then, once the parameters have been identified, the Bekker's equations can be used to infer contact forces, torques, drawbar pull, as well as energy consumption. However, the Bekker's parameters have been proved to be highly sensitive to the test setup, resulting in significant uncertainty about their measured values [23] [24]. Moreover, the Bekker theory is based on uniaxial pressure-sinkage and symmetric plat-shear tests, while its extension to tridimensional and generally more complex wheel-soil contact geometry is not fully justified [25]. Finally, the semi-empirical nature of the Bekker theory has demonstrated, at best, only approximate modeling of the actual mechanical behavior of many types of terrain [26].

To overcome the limitations of Bekker's models, machine learning algorithms have been used to directly learn energy models from data. Martin et al. [27] proposed to monitor the motor power usage of a wheeled robot as it traverses an unknown terrain over several loops. Then, a gaussian process was used to build an energy map, while exploration was encouraged to converge to minimum energy tours of the environment. However, while their formulation allows for path adaptation when the robot performs repetitive tours over previously observed environments, it cannot make inference for unobserved trajectories, making it impractical for robots that are exploring new terrains.

In [28], a deep learning method has been proposed which exploits a 2D convolutional neural network to infer driving energy consumption from RGB and depth images. However, predicting energy consumption from images makes strong assumptions on the correlation between visual appearance and terrain properties. In most unknown scenarios, images can be ambiguous, as terrains with similar visual aspects might, in turn, require very different energy expenditures. Moreover, their proposed architecture makes use of a global energy model, which requires prior training on large and often expensive datasets, while possible adaptation is not considered.

Meta-learning represents a promising strategy to enable fast inference and adaptation based on limited amounts of data [29]. Several deep meta-learning algorithms have been proposed in recent years, which have demonstrated state-of-the-art capabilities in a variety of settings ranging from one-shot imitation learning [30] and multi-task reinforcement learning [31]. Other works have proposed the use of meta-learning for the online adaptation of robotic platforms [32] [33] [34]. Similar to our work, Nagabandi et al. [35] proposed a meta-learning approach to control a legged-robot by learning and adapting its dynamical model when faced with different unforeseen conditions (e.g. broken leg, varying payload). However, their method does not address the problem of energy consumption prediction and its integration into a planning optimization framework.

Some recent works have proposed the use of neural networks to model complex wheel-terrain interactions, their effect on energy consumption, as well as their integration into an energy-aware path planning algorithm [19] [20]. However, in those works prediction and planning are performed on a single terrain type with known terramechanical properties, while possible generalization and adaptation are not considered.

To the best of the authors' knowledge, this work is the first to propose an integrated energy-aware deep meta-learning framework to predict, adapt, and plan over terrains with unknown and varying terramechanical properties.

III. METHODOLOGY - ENERGY PREDICTION MODEL

A. Meta-Learning Preliminaries

Meta-learning is concerned with learning algorithms that are more efficient than learning from scratch. Specifically, meta-learning leverages knowledge gradually gathered across previous tasks to learn a common prior which enables fast inference for new tasks. For this reason, meta-learning is often referred to as "learning to learn". Formally, while in standard supervised learning we aim at learning a function f_θ to minimize a loss $\mathcal{L}$ over a single large dataset $\mathcal{D}$ from a task, in meta-learning, we minimize the expected cost over a distribution $\rho(\mathcal{D})$ of several small datasets from different tasks. That is:

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \mathbb{E}_{\mathcal{D} \sim \rho(\mathcal{D})} [\mathcal{L}(\mathcal{D}, \theta)] \quad (1)$$

We note that the tasks must be drawn from the same distribution ρ , which means that they must share some common underlying structure. In this way, the meta-learning algorithm can be trained to capture this structure and leverage it for learning faster on similar new tasks. To enable adaptation from few examples, each training dataset $\mathcal{D}$ is additionally split into a meta-training dataset $\mathcal{D}_{tr}$ and meta-test dataset $\mathcal{D}_{ts}$, both of them composed of (x_i, y_i) pairs from the same task. In this way, the optimization in Equation (1) can also be expressed as:

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \mathbb{E}_{\mathcal{D}=\{\mathcal{D}_{tr}, \mathcal{D}_{ts}\} \sim \rho(\mathcal{D})} \sum_{i=1}^J [\mathcal{L}(f_\theta(x_i^{ts}, \mathcal{D}_{tr}), y_i^{ts})] \quad (2)$$

That is, for all the tasks drawn from the distribution ρ , we aim to find the parameters θ of a function f_θ such that, given few (e.g. K) examples from the meta-training dataset $\mathcal{D}_{tr} = \{(x, y)_{1:K}^{tr}\}$, we can successfully predict new pairs from the same task on

the meta-test dataset $\mathcal{D}_{ts} = \{(x, y)_{1:J}^{ts}\}$. Several meta-learning algorithms have been proposed in the literature to solve the optimization problem in Equation (2). In this paper, we opted for a black-box approach, which means that f_θ is modelled by a single neural network trained in an end-to-end fashion to predict y_i^{ts} from x_i^{ts} and $\mathcal{D}_{tr}$. This choice is motivated by the simplicity and high representational power of black-box approaches, which have shown state-of-the-art performances in a variety of settings [29]. Therefore, the system aims at modelling the predictive distribution:

$$p(y_i^{ts} | x_i^{ts}, \mathcal{D}_{tr}, \theta) = p(y_i^{ts} | x_i^{ts}, \{(x, y)_{1:K}^{tr}\}, \theta) \quad (3)$$

We observe that, although in the black-box approach f_θ is represented by a single set of parameters θ , the meta-learning optimization can be seen as pursuing a two-fold objective. This can be noted by rearranging p in Equation (3) as composed of two separate distributions p_1 and p_2 with parameters θ_1 and θ_2 . Specifically:

$$p(y_i^{ts} | x_i^{ts}, \mathcal{D}_{tr}, \theta) = p_1(y_i^{ts} | x_i^{ts}, p_2(h | \mathcal{D}_{tr}, \theta_2), \theta_1) \quad (4)$$

In this way, p_2 can be interpreted as a function that must capture from the few examples in $\mathcal{D}_{tr}$ the latent characteristics of the new task, which are represented by the hidden vector h . Meanwhile, p_1 can be seen as the task-independent underlying structure, which is shared by all the tasks, and which can be adapted to the task-specific objective by observing h .

B. Meta-Learning for Adaptive Energy Prediction

In this section, we illustrate the proposed extension of the meta-learning framework to the problem of adapting the driving energy model of an UGV traversing unknown terrains. In this context, each new task and corresponding dataset $\mathcal{D}$ can be seen as a new energy modeling problem of a terrain with previously unseen terramechanical properties. Specifically, we assume that the only available input feature x to perform the estimation is the geometric information of the terrain. Indeed, this can be conveniently estimated at distance with exteroceptive sensors, such as stereo-cameras or LIDAR. Conversely, no other terramechanical characteristic is explicitly given to the network, whilst it is treated as the unknown property that varies across tasks. Practically, training can be achieved by assuming to have access to a collection of small datasets of geometry-energy pairs, each one of them coming from a different type of terrain. Then, a neural network f_θ is fed with few examples of geometry-energy pairs $\mathcal{D}_{tr} = \{(x, y)_{1:K}^{tr}\}$ from the meta-training dataset, and additional geometries $\{x_{1:J}^{ts}\}$ from the meta-test dataset, all of them coming from one of the small datasets $\mathcal{D}$ (i.e. from the same terrain type). Finally, the network is trained to implicitly retrieve the properties of that terrain from the example pairs $\mathcal{D}_{tr}$ (p_2 in Equation (4)), and to exploit this information to predict the energy consumption y_i^{ts} relative to x_i^{ts} (p_1 in Equation (4)).

1) *Network Architecture*: Figure 2 illustrates our proposed neural network architecture. The bottom part of the neural network is a Long-Short Term Memory (LSTM) layer with 128 units. This choice is motivated by the proven capability of neural networks with memory capacities of meta-learning [36]. For example, LSTM neural networks have demonstrated

Fig. 2: The neural network architecture.

capabilities to learn unforeseen quadratic functions with a low number of data samples [37]. In our setup, each intermediate cell i of the LSTM layer is presented with a geometry-energy example pair $(x_i, y_i)^{tr}$. In this way, each LSTM cell implicitly captures the terramechanical properties of the terrain from the i example pairs provided up to that point and passes its current estimate h_i to the next cell. Hence, the estimate can be refined within the cells as more examples are provided. Experimentally, we determine as $K = 3$ the number of example pairs beyond which minimal improvement in prediction error is observed. The top part of the neural network is a stack of two Fully Connected (FC) layers with relu activation, and 64 and 1 units respectively. The FC stack takes as input the LSTM hidden vector h_i and a new geometry x_i^{ts} , and outputs the corresponding predicted energy $\hat{y}_i^{ts}$.

2) *Training Procedure*: Algorithm 1 illustrates the training procedure for the proposed neural network according to the meta-learning framework. At each training step, for each element in a mini-batch, we randomly select a terrain type, and its corresponding dataset $\mathcal{D}$, from a collection of datasets of terrains with different terramechanical properties. Then, $K + K$ geometry-energy pairs (x_i, y_i) are randomly sampled respectively for meta-training and meta-test. First, the meta-training pairs $\{(x, y)_{1:K}^{tr}\}$ are input to the LSTM layer. Then, each one of the K hidden states h_i is concatenated with a geometry x_i^{ts} from the meta-test dataset and fed to K instances of the FC stack (having shared weights). The labels for the outputs are $\{y_{1:K}^{ts}\}$. Hence, the loss for each batch is computed as in Line 9 of Algorithm 1. In this way, each intermediate LSTM cell must provide an expressive latent representation h_i , using the examples available up to that cell, to solve the meta-learning problem. This can be advantageous in the context of mobile robotics, as it enables updated energy inference after each new collected data point. Moreover, we observe that the example pairs are not fed to the network in a specific temporal order, whilst they are sampled at random from the available datasets. This ensures that the LSTM retrieves terrain properties from any example pairs, without biasing its prediction to specific geometric sequences.

Finally, the parameters of the network are trained by means of stochastic gradient descent. In all our experiments, we train the network for 60 epochs, with mini-batch size $N = 32$, learning rate $\alpha = 10^{-4}$, and RMSprop optimizer [38].

Algorithm 1 Meta-Learning Training Procedure

Input: Collection of datasets $\mathcal{D}$ from different terrain types

Input: Distribution ρ over $\mathcal{D}$

Input: Mini-batch size N , number of pairs K

Input: Neural Network f_θ , learning rate α

```

1: Randomly Initialize  $\theta$ 
2: for  $t=1, \dots$  do
3:   for  $j=1, \dots, N$  do
4:     Sample  $\mathcal{D}_j \sim \rho(\mathcal{D})$ 
5:     Randomly sample  $K (x_i, y_i)^{tr} \sim \mathcal{D}_j$ 
6:     Randomly sample  $K (x_i, y_i)^{ts} \sim \mathcal{D}_j$ 
7:      $\hat{y}_{1:K}^{ts} = f_\theta([(x, y)_{1:K}^{tr}, x_{1:K}^{ts}])$ 
8:   end for
9:    $\mathcal{L} = \frac{1}{NK} \sum_{j=1}^N \sum_{i=1}^K (y_i^{ts} - \hat{y}_i^{ts})^2_{\mathcal{D}_j}$ 
10:   $\theta \leftarrow \theta - \alpha \nabla_{\theta} \mathcal{L}$ 
11: end for
Output:  $\theta$ 

```

Fig. 3: The geometry estimation pipeline along a trajectory.

3) *Adaptation Procedure*: Upon training completion, example pairs from new terrains can be leveraged to adapt to the local conditions. For example, if one pair is available, only the first LSTM cell must be filled in with $(x_1, y_1)^{tr}$. Hence, h_1 is used as input to the FC stack for the energy estimation of new terrain geometries x_i^{ts} .

C. Input-Output Features

1) *Geometry*: The geometry x of each segment is estimated by analysing the terrain point cloud in front of the vehicle with standard geometric features extraction techniques [39]. Specifically, we select the pitch and roll inclinations of the terrain as the most energy-relevant geometric features. In our approach, the trajectory pitch and roll values are estimated by finding the best-fitting plane over a vehicle-sized terrain patch, and sliding the patch by incremental steps (6 cm long) in the trajectory direction (see Fig. 3). Then, we estimate the pitch and roll for each patch and we compute their mean and standard deviation values. In this way, possible slope variation is better accounted for along the path. Therefore, each geometric input feature can be expressed as a four-dimensional vector $x_i = [p_i^m, r_i^m, p_i^{std}, r_i^{std}]$, whose values are respectively the pitch and roll mean values, and the pitch and roll standard deviations along the trajectory.

2) *Energy*: The energy consumption y is derived by direct measurement of the motor torque τ and angular speed ω .

Fig. 4: (a) The 5 elementary trajectories of the state-lattice space and (b) diagram example of A* optimisation.

Simulated sensors onboard the vehicle (see Section V-B) collect these two variables with a constant sampling rate $\Delta t = 2$ ms over the time interval $[t_0, t_f]$ needed to traverse each segment. Hence, the energy consumption is computed as follows:

$$y_i = \sum_{t_0}^{t_f} \begin{cases} \tau_t \omega_t \Delta t & \tau_t \omega_t \geq 0 \\ 0 & \tau_t \omega_t < 0 \end{cases} \quad (5)$$

It is noted that Equation (5) represents the mechanical formulation of energy consumption without considering motor loss components or energy recovery. While this can be considered as a simplified assumption, a thorough study of specific motor characteristics is out of the scope of this work. Moreover, as our method is completely model-agnostic, this formulation can be updated with minimal effort to any available motor specifications.

IV. METHODOLOGY - ENERGY-AWARE PATH PLANNER

A. Path Planning Preliminaries

1) *State-Lattice Space*: Motion planning is performed in a state-lattice space, a well-known approach to the problem of differentially constrained planning [40]. State lattice is a search graph where vertices representing kinematic states of the vehicle are connected by edges representing trajectories that satisfy its kinematic constraints. In this way, planning and cost estimation can be achieved directly over feasible trajectories, thereby considering the actual vehicle mobility constraints [20]. In our application, we define a set of 5 elementary trajectories 2.7 m long, and with curvature uniformly spaced in $[-0.13, 0.13] \text{ m}^{-1}$, according to the mobility capability of our vehicle (see Fig. 4a).

2) *A* Algorithm*: The A* graph search optimization is used in the state-lattice space [41]. This choice is motivated by the simplicity and effectiveness of A* to optimize paths according to complex cost functions, making it well-suited to the mobile robots energy-aware path planning [15] [20]. Moreover, A*-like graph search methods have the advantage to guarantee bounded path optimality, and to reduce the computational cost of planning by making use of heuristics [42]. Fig. 4b shows a diagram of the A* planning process in the state-lattice space. In the diagram, the start position is labelled as node A and added to a list called OPEN, which contains all nodes that have been found but not yet expanded. At each iteration of A*, a new node in the OPEN list is chosen according to its priority and expanded for the set of all elementary trajectories. In A* the priority is expressed as:

$$P(n) = C(n_p, n) + H(n, g) \quad (6)$$

The priority P of a node n is given by the energy cost $C(n_p, n)$ of moving from the parent node n_p to n , and the heuristic estimation $H(n, g)$ of the remaining cost from n to the goal g . The new node is then chosen as the one having the lowest P . The search continues until a goal node is retrieved from the OPEN list. At this point, by traversing backward the stored parent-child relationships, the path and the associated cost are found from the start to the goal.

3) *Heuristic Function*: We observe that the heuristic term in Equation (6) is not required to represent a true energy value, but is used to guide the search in promising directions. Namely, the choice of the heuristic function can affect the guarantee of A* to converge to an optimal solution and the computational efficiency of the search. The following definitions are given:

Optimality: a path in a search graph from a node n to a goal g is defined *optimal* if its cost $C^*(n, g)$ is not greater than the cost $C(n, g)$ of any other feasible path from n to g .

Admissibility: a heuristic function $H(n, g)$ is defined *admissible* if it never overestimates the optimal true cost of reaching the goal $C^*(n, g)$. That is: $H(n, g) \leq C^*(n, g)$.

In graph search methods, optimality is ensured only for admissible heuristics. However, not all admissible heuristics have the same computational efficiency. If the heuristic underestimation is excessive, its relevance to direct the algorithm in promising directions is minimal. As a consequence, the planner has to expand a higher number of nodes prior to finding an optimal solution, leading to a lower computational efficiency. Conversely, if the heuristic overestimates the optimal energy required to reach the goal, the search more rapidly converges to a feasible solution, but at the cost of optimality guarantee. Therefore, the ideal case would be for the heuristic to exactly estimate the minimum cost of remaining energy, that is $H(n, g) = C^*(n, g)$. However, obtaining such a heuristic is non-trivial for most of the applications. Practically, the heuristic function is often devised, in energy-aware path planning problems, to represent a marginal underestimation of the true cost [15] [16]. However, as in our application the vehicle moves over unknown terrains, the energy model for the true cost can constantly change. As a consequence, traditional fixed heuristic functions might result in degraded planning performance. Therefore, to preserve the computational efficiency of planning, a method must be devised capable of adapting the heuristic function together with the true cost.

B. Meta-Learning Integration into Path Planner

We propose a novel approach to the aforementioned energy-aware path planning problems. The diagram of the procedure is illustrated in Fig. 5. First, we define each geometry-energy pair (x_i, y_i) , to be fed to the network of Section III-B, by splitting each edge of the lattice space (n_p, n) (2.7 m long) in three equal segments l (0.9 m long). This is done to collect multiple example pairs from each trajectory, thereby obtaining more refined information over shorter portions of terrain. Hence, the vehicle starts moving, according to a previously planned path, and collects (x_i, y_i) pairs with the method described in Section III-C. Upon path completion, a new planning procedure is started. The three (x_i, y_i) pairs from the most recent trajectory are retrieved from memory and fed to the LSTM layer of the

model described in Section III-B. In this way, the terrain-specific properties of the last trajectory can be recovered (represented by the hidden vector h). Then, the energies for the upcoming edges in the graph $C(n_p, n)$ and for the heuristic $H(n, g)$ are estimated by assuming the same properties h of the last trajectory. This can be considered as a reasonable assumption, provided that re-planning is performed often, and that the properties of the terrain do not change with excessive frequency. Hence, the cost of each edge $C(n_p, n)$ is computed as follows:

- Each edge (n_p, n) is divided in three consecutive segments;
- The three geometric values of the three segments are estimated with the method described in Section III-C;
- Each geometry is concatenated with h , and fed to the FC stack to predict the corresponding energy consumption;
- $C(n_p, n)$ is defined by summing up the energy estimates of the three segments.

Meanwhile, the heuristic $H(n, g)$ is computed as follows:

- With the aim of generating an underestimating heuristic, we assume that the goal can be reached on a straight line and that the remaining terrain to the goal can be approximated with a perfectly planar ramp, without roll and variance components. This is represented by a heuristic geometric vector $x_h = [\theta, 0, 0, 0]$, where θ is the relative inclination in degree between n and g ;
- x_h is concatenated with h , and fed to the FC stack to predict the heuristic energy over a unitary segment H_u ;
- The total heuristic for remaining energy is defined as $H(n, g) = H_u * \frac{d}{l}$, where d is the euclidean distance in meters between n and g , and l is the length of a unitary segment.

It is noted that, in our formulation, an analytical guarantee of the admissibility of the heuristic cannot be provided since the heuristic is processed by a neural network, which is analytically intractable. Nevertheless, our approach can be advantageous in that, both the true cost and the heuristic function constantly adapt their estimates based on the most recent terrain properties h . In this way, the heuristic remains informative as the vehicle navigates over unknown terrains, thereby improving the efficiency of planning. We provide evidence of this in Section VII-B.

V. EXPERIMENTAL SETUP

A. Simulation Environment

The proposed method is developed and tested using a physical simulator implemented in Python and based on the Chrono [43] implementation of the Soil Contact Model (SCM) [44]. Specifically, SCM extends the Bekker-Wong theory to 3D-bodies dynamic simulators, enabling more realistic modeling of arbitrary shaped wheel-terrain interactions in deformable terrains and retaining a more efficient computational workload than alternative Finite Element or Discrete Element methods [4] [45]. Practically, terrains with different behaviors can be conveniently modeled in SCM by the setting of 6 terrain-dependent parameters: exponent of sinkage n , cohesive modulus K_c , frictional modulus K_ϕ , cohesion limit c , angle of internal friction ϕ , Janosi-Hanamoto coefficient J [3] [46].

In this work, 22 different terrain types are considered, whose parameters are derived from [3] and [47]. They range from different types of snow, dry sand, sandy loams, compact and wet clays, and Martian and Lunar simulants. The complete list

Fig. 5: Meta-learning integration into the path planner.

with the corresponding parameters is provided at [META-UGV](#). The terrain types are divided into five macro-categories each one of them having substantially different properties, thereby requiring different energy expenditures to be traversed. They are: (1) very loose frictional, (2) high moisture content clays, (3) loose frictional, (4) compact frictional, and (5) dry clays. Moreover, different intra-classes are present within each macro-category, having more closely related, but not identical, energy models.

B. Vehicle Model

The vehicle model is based on the features of the High Mobility Multipurpose Wheeled Vehicle (HMMWV) M966 as provided by the Chrono physics engine [48]. The vehicle has a total mass of 3000 kg, a wheelbase of 3.4 m, and a wheel track of 1.9 m. It has a double-wishbone suspension on both axles, pitman-arm steering, and all wheels can be driven. It has a three-gear automatic gearbox with a torque converter and a realistic engine subsystem. The vehicle is subjected to a PID controller for the acceleration-brake pedals and a P controller for the wheel steering. Both controllers run at a frequency of 500 Hz.

C. Data Collection

The vehicle is driven over unstructured maps at a constant speed of 1 m/s for a total of 70.4 km (3.2 km for each of the 22 terrain types described in Section V-A). This choice is motivated by the often low speed of robotic vehicles in natural environments [2] [27]. The elevation maps are randomly generated with a Perline-noise algorithm described in [49]. Figure 6 illustrates some examples of the unstructured maps with different terramechanical properties in the Chrono simulator. Then, the geometry-energy pairs are collected with the method described in Section III-C. The final dataset is composed of 22 different datasets, each one of them from a different terrain type and made of 3530 geometry-energy pairs.

D. Comparative Methods for Energy Prediction

A description is provided of alternative machine learning approaches, which are used to compare the generalization and adaptation performance of the proposed meta-learning method. All the methods exploit the same training and validation datasets, while they only differ in the model architecture and training procedure.

Fig. 6: Examples of unstructured maps with different terrain properties in SCM simulator. Colours indicate the sinkage depth.

1) *Meta*: is the proposed deep meta-learning neural network as described in Section III-B.

2) *Global Model (GM)*: is a standard non-adaptive approach in which a neural network is trained to predict energy consumption in a supervised learning fashion. Specifically, GM takes as input the terrain geometry x and its terramechanical properties t (i.e. the 6 parameters described in Section V-A). Therefore, the network is in principle provided with all the information required to make an accurate energy prediction. However, while this is possible in simulation, the terramechanical properties of a terrain are often *a priori* unknown and difficult to estimate online, thereby representing the first limitation of this method. Nevertheless, in this work, we assume to have full knowledge of those properties, while focusing our analysis on the generalization capabilities of a non-adaptive global approach under limited amounts of data. The network model is composed of 3 fully connected layers with relu activation and respectively 128, 6, and 1 units. All the input features are standardized before being fed to the network. We observe that the GM architecture is similar to the FC stack of the Meta approach. Indeed, they are both models trained to infer energy consumption from the terrain geometry and some representation of terrain properties. However, this representation is assumed *a priori* and fixed in GM. Conversely, the meta-learning formulation aims at learning this representation implicitly from example pairs and in such a way to enable fast adaptation.

3) *Separate Model (SM)*: consists of a collection of independent neural networks trained separately on the different training datasets. Specifically, each network takes as input the terrain geometry x and is trained to predict the energy consumption of one of the terrain types in a supervised learning fashion. Therefore, upon training completion, each network can effectively model the geometry-energy relationship of its specific training terrain. Each neural network model is composed of 3 fully connected layers with relu activation and respectively 16, 8, and 1 units. When testing on new terrains, previous example pairs are leveraged to select which of the models to use. Specifically, the following approach is adopted: (1) $K(x_i, y_i)$ example pairs are observed from the terrain to test, (2) all the x_i are fed to all the separate models, (3) the best model is chosen as the one having the most accurate prediction (in terms of mean squared error) compared with the actual values y_i , and (4) the best model is used to predict the energy of new terrain geometries. In this way, the Meta and SM approaches can be compared when they are provided with the same previous example pairs.

VI. EXPERIMENTS - META-LEARNING PERFORMANCE

In this section, the prediction performance of the meta-learning approach, prior integration into the path planner, is

TABLE I: The overall validation performance for different numbers of example pairs (averaged over 10 trials).

Experiment 1				
Example Pairs	0	1	2	3
MSE [k^2J^2]				
GM	225.82	-	-	-
SM	-	55.71	45.76	41.83
Meta	-	49.80	35.36	32.79
R2				
GM	-0.104	-	-	-
SM	-	0.751	0.798	0.818
Meta	-	0.785	0.850	0.862

tested. Specifically, in the following experiments, part of the 22 datasets will be used for training, while the others will be employed for validation to assess the adaptation performance on new terrain types. Furthermore, as meta-learning is concerned with finding a common structure across different datasets, the proper selection of the training tasks can be critical. Particularly, both training and validation datasets must come from the same distribution. For example, if all the training datasets belong to different types of compact sand, meta-learning will be able to adapt to new classes of compact sands, but it will not be able to generalize to completely different terrain classes, such as deep snow. For this reason, the training datasets are selected, in the following experiments, to be uniformly distributed among the 5 macro-categories described in Section V-A.

A. Generalization Performance

In the first experiment, we test the generalization performance of Meta compared with GM and SM. Specifically, we perform 10 test trials with different training-validation splits. For each trial, the training dataset is defined by selecting at random $2 + 2 + 2 + 3 + 1 = 10$ datasets respectively from each macro-category from 1 to 5. Hence, the remaining 12 datasets are used for validation.

Table I summarises the overall validation performance, averaged over the 10 trials, and for different numbers of example pairs (from 0 to 3). We observe that, even assuming to have perfect knowledge of the terrain properties, the prediction error of GM remains excessively high on the validation datasets, with a loss of $225.82 \text{ k}^2\text{J}^2$ and r2 score of -10.4% . This is due to the limited number of terrain types used during training, which are not sufficient to enable learning of a global energy model without adaptation. Therefore, while the GM approach can be in principle a valid solution if considerably more data from several terrain types were available, it largely overfits the training dataset in the low data regime. On the contrary, both SM and Meta show considerably better performance and the

Fig. 7: The effect of the training datasets' reduction. The results refer to the average MSE and r2 score with 3 example pairs over 10 test trials. The overall performance is labelled above the charts (green bars).

capability to improve their predictions as more examples are given. Indeed, SM aims at learning a much simpler geometry-energy relationship for specific terrain types, which require considerably fewer data to be captured. However, its main limitation is that its performance on the validation datasets is strictly dependent on the similarity with the training data, while possible adaptation is not considered. Conversely, Meta efficiently exploits example pairs to implicitly capture terrain-specific properties and adapt its energy estimation accordingly. As a result, Meta demonstrates consistently better performance than SM, which provides evidence of its improved generalization capabilities. For instance, the loss and r2 score with 3 example pairs are respectively $41.83 \text{ k}^2\text{J}^2$ and 81.8% for SM, and $32.79 \text{ k}^2\text{J}^2$ and 86.2% for Meta.

B. Effect of Training Dataset Reduction

In this section, we further test the capability of Meta to generalize to an increasingly larger set of unknown terrain types, while the classes experienced during training are reduced. Its performance is compared with the SM approach, while GM, given the already poor performance in the previous experiment, is excluded from this and all the following analyses. Two experiments (called Experiment 2 and 3) are performed in which the training datasets are progressively reduced. Specifically, for the two experiments, the training datasets are defined respectively by selecting at random $1+1+1+3+1 = 7$, and $1+1+0+1+1 = 4$ datasets from the 5 macro-categories, while all the remaining 15 and 18 datasets are used as validation. Moreover, as with the experiment in Section VI-A, both the experiments are repeated over 10 test trials with random training-validation splits.

The performance of the methods with 3 example pairs is summarised in Fig. 7. We observe that, as the number of training datasets decreases, SM degrades its prediction more rapidly than Meta. For instance, the overall r2 score (see the green bars on the right graph) drops by 6.6% and 12.8% for SM (from 81.8% to 75.2% and 69.0%), while decreases by 2.5% and 3.8% for Meta (from 86.2% to 83.7% and 82.4%). Furthermore, Fig. 7 shows the performance across the three experiments for each one of the 5 terrain macro-categories. We note that the categories 1 and 2 (i.e. very loose frictional soils, and high moisture content clays) have the highest prediction errors for both Meta and

SM. This can be explained by the more challenging nature of these terrains. Nevertheless, Meta retains consistently better performance for all the categories. This highlights the fact that the performance of SM depends more strongly on the similarity between the training and validation scenarios. Therefore, while SM can be well-suited for known operational environments, made of a finite set of known terrain classes, it fails to generalize to unforeseen terrain types. Conversely, the Meta approach leverages the training datasets in a different manner, which enables more efficient interpolation and adaptation, thereby leading to better generalization across a larger set of previously unknown scenarios. Finally, in Experiment 3 we also test the performance of Meta by reducing the training samples to 10% of the original training datasets (Exp3 10% in Fig. 7). This means that, while the same number of training datasets is maintained, the samples of each one of them are reduced from 3.2 km to 320 m. Therefore, the total training samples used for Exp3 10% corresponds to 1280 m of traverse. A limited degradation in performance is observed. This provides insights on the capability of Meta to retain good adaptation with a limited amount of data, which can be a crucial aspect in real-world applications.

VII. EXPERIMENTS - PATH PLANNING INTEGRATION

In this section, the performance of the meta-learning integration into the energy-aware path planner is tested. All experiments are performed by adopting the same training-validation strategy of Experiment 3 in Section VI-B. This means that the neural networks are trained on 4 terrain datasets, integrated into the A* path planner, and tested on new scenarios selected at random from the remaining 18 terrain types. Moreover, to further increase the uncertainty of the terrain properties encountered during testing, random noise is added to each one of the 6 terramechanical parameters (with noise bounded to $\pm 5\%$ of the value of each parameter). All methods are tested on an Intel Core i9-9940X CPU and on a GeForce RTX 2080 Ti GPU.

A. Effect of Terrain Transition Frequency

The prediction performance of Meta integrated into the A* path planner is tested, and compared with SM. A total of 5 traverses is performed, each 7.29 km long, in which the vehicle

TABLE II: Path Planner Results - The overall traverses' performance.

Model	Planning Performance			Prediction Performance			
	Nodes	Node Time [s]	Tot Time [s]	Predicted [kJ]	Real [kJ]	MSE [kJ ²]	R2
Meta	30620	0.148	4532	795.603×10^3	814.600×10^3	51.50	0.697
SM	33678	0.170	5725	751.480×10^3	816.934×10^3	77.51	0.556

Fig. 8: The effect of the terrain transition frequency on the prediction performance.

plans paths according to the two approaches. After each one of the 5 traverses, the frequency with which the terrain properties vary is progressively increased. On the contrary, energy prediction and replanning are performed for all experiments at fixed intervals of 8.1 m. In this way, the impact on the prediction performance of different frequencies of terrain variation can be analyzed. Figure 8 summarises our findings. As expected, the prediction performance of both methods decreases as the terrain variation is more frequent. Conversely, for periods higher than 81 m, both methods have similar performance to the validation results (see Experiment 3 in Section VI-B), where no terrain transition occurred. This confirms the viability of our method for terrains whose properties do not change with excessive frequency (81 m or higher), compared with the frequency of replanning (8.1 m).

Figure 9 illustrates an example of planning adaptation as the vehicle transitions on a new terrain. In the example, the vehicle initially navigates over a compact soil, characterized by low energy requirements. Then, it moves on a more energy-demanding loose soil (see Fig. 9b). In the top diagram of Fig. 9a, the trajectories planned by Meta and SM are presented, with replanning occurring at fixed intervals of 8.1 m (white circles from 1 to 6). In the bottom graph of Fig. 9a, each point represents the energy consumption (true and predicted) over the previous 2.7 m of traverse. Before the terrain transition (path from 1 to 3) Meta accurately predicts the energy consumption of the planned paths, while SM has higher prediction errors. During the terrain transition (path from 3 to 4) both models provide inaccurate estimates. Indeed, the last example pairs used to infer energy do not match with the new terrain conditions. At the next planning step (node 4), the most recent example pairs from the new terrain are exploited. As with previous results, the Meta approach updates the energy estimates more effectively, while SM demonstrates degraded performance.

In Table II, the overall performance over the 5 traverses is illustrated. We observe that the average time needed to expand a node takes longer in SM than in Meta with an average node expansion time of respectively 0.170 s and 0.148 s. Indeed, in

Fig. 9: (a) An example of energy prediction and planning before, during, and after a terrain transition. (b) The UGV during the terrain transition.

SM, all the models must be tested and compared independently, before selecting the most suitable for the current terrain. Conversely, a single feed-forward computation is performed by Meta, thereby leading to a lower computational workload. More importantly, we observe that Meta provides solutions that require lower driving energy consumption, and with considerably higher prediction accuracy.

B. Effect of Heuristic Function

In this section, a description is provided of alternative versions of the meta-learning integration into the A* path planner described in Section IV-B. Those versions are devised to test the *optimality* and *computational efficiency* of the proposed meta-adaptive heuristic function. Specifically, all the approaches estimate the true cost $C(n_p, n)$ by using the proposed meta-learning method, while they differ in the choice of the heuristic function $H(n, g)$ and/or the planning strategy.

1) *Meta-Adaptive*: is the method described in Section IV-B, with the heuristic function $H(n, g)$ meta-adapting according to the current terrain properties.

2) *Meta-Optimal*: uses a fixed and admissible heuristic function, with a similar implementation of the method described in [20]. Specifically, the heuristic is defined as $H(n, g) =$

TABLE III: Path Planner Results - The effect of the heuristic function and the target distance.

Targets		720							
Target Distance		8.1 m							
Model	Planning Performance				Prediction Performance				
	Nodes	Node Time [s]	Tot Time [s]	Optimal Paths	Predicted [kJ]	Real [kJ]	MSE [kJ ²]	R2	
Meta-Adapt.	12715	0.156	1991	678	119.732×10^3	130.493×10^3	26.35	0.867	
Meta-Opt.	18298	0.126	2306	720	119.694×10^3	131.045×10^3	32.00	0.845	
Meta-ARA*	14262	0.127	1810	650	120.501×10^3	131.274×10^3	28.59	0.859	
Target Distance		10.8 m							
Model	Planning Performance				Prediction Performance				
	Nodes	Node Time [s]	Tot Time [s]	Optimal Paths	Predicted [kJ]	Real [kJ]	MSE [kJ ²]	R2	
Meta-Adapt.	40980	0.153	6274	620	163.564×10^3	176.738×10^3	26.04	0.861	
Meta-Opt.	64438	0.125	8038	720	163.337×10^3	177.232×10^3	27.34	0.856	
Meta-ARA*	47637	0.125	5941	596	163.887×10^3	178.213×10^3	25.54	0.862	

$\max[0, (G\theta + \beta)d]$, where θ and d are respectively the relative inclination in degree and euclidean distance in meters between n and g , while G and β are heuristic parameters set to provide globally optimistic energy estimates for all available terrain types. In this way, the admissibility of the heuristic can be guaranteed for all terrains, thereby ensuring the convergence of this method to a predicted optimal solution. Nevertheless, as the heuristic is fixed and cannot adapt to varying terrain properties, the underestimation might be excessive in certain cases, leading to reduced computational efficiency. In our application, G and β are set respectively to 1.41 kJ/m, and -4 kJ/m.

3) *Meta-ARA**: uses the same heuristic of Meta-Optimal but adopts Anytime Repairing A* (ARA*) in place of the standard A* [50]. ARA* is an anytime planning algorithm, which can provide a more rapid, but likely sub-optimal, first solution, by inflating $H(n, g)$ by an inflation factor $\epsilon > 1$. Then, ϵ can be progressively reduced, within the available time, to refine the initial solution, and eventually converge to the solution provided by Meta-Optimal. In our experiments, the maximum planning time of Meta-ARA* is set to be equal to the time needed by Meta-Adaptive to converge to a solution (note: if no solution is found by Meta-ARA* in this time, Meta-ARA* is run for as long as needed to find at least one solution). Hence, the best solution found by ARA*, up to that moment, is selected. In this way, we can compare different degradations of path optimality of Meta-Adaptive and Meta-ARA*, when their searches are restricted to the same amount of time.

The planning performance of the three alternative heuristic methods is tested, by conducting statistical analysis over 720 random start-goal positions within the different test environments. Specifically, for each start-goal position: (1) the vehicle navigates for 2.7 m, according to a breadth-first search approach [42], and collects the three most recent geometry-energy pairs, and (2) the three example pairs are used by the three alternative methods to plan a path to the next target. In this way, the planning performance of the three algorithms can be tested under identical initial conditions (i.e. both start-goal, and previous example pairs). The first part of Table III summarises our findings for 8.1 m distant targets. Expectedly, the three approaches have similar prediction performance (as the cost $C(n_p, n)$ is estimated with the same method, i.e. Meta), while marginal variations occur due to different planned trajectories. On the contrary, the three methods substantially

differ in their planning performance. For example, we observe that the average time needed to expand a node in Meta-Adaptive is higher than in Meta-Optimal and Meta-ARA*, with node expansion times of respectively 0.156 s, 0.126 s, 0.127 s. This can be explained by the fact that, in Meta-Adaptive, the heuristic function involves the feed-forward computation of a neural network, thereby requiring a higher computational workload than the other two methods. Nevertheless, considerably fewer nodes must be expanded by Meta-Adaptive to converge to a solution. Specifically, the total number of expanded nodes for Meta-Adaptive and Meta-Optimal is of respectively 12 715, and 18 298, thereby leading to a total planning time of respectively 1991 s and 2306 s. Therefore, Meta-Adaptive requires on average 13.7 % less time than Meta-Optimal to converge to a solution. Moreover, we observe that Meta-Adaptive converges to the same solution than Meta-Optimal in 678 of the 720 planning procedures, while it provides slightly more energy-demanding solutions in the remaining cases. Specifically, the total predicted energy consumption over the planned paths by Meta-Adaptive and Meta-Optimal is of respectively 119.732×10^3 kJ, and 119.694×10^3 kJ. Therefore, the overall Meta-Adaptive degradation in terms of path optimality is of 0.03 %. This shows evidence of the capability of an adaptive meta-heuristic to retain close-to-optimal solutions while reducing the computational time of planning. Furthermore, Meta-Adaptive retains better solutions than Meta-ARA*, given the same amount of maximum planning time. Specifically, in Meta-ARA* only 650 planning procedures provide an optimal solution, with a total predicted energy consumption of 120.501×10^3 kJ, and resulting optimality degradation of 0.67 %.

We additionally test the three methods by increasing the distance of the 720 targets from 8.1 m to 10.8 m (see second part of Table III). As the distance from the targets rises, the branching dimensionality of the graph-search increases. As a consequence, an increased computational effort is required to converge to a solution. Nonetheless, we observe that the planning time of Meta-Adaptive increases by a lower factor than the one of Meta-Optimal. Specifically, the total amount of planning time needed by Meta-Adaptive and Meta-Optimal increases respectively by a factor 3.2 (from 1991 s to 6274 s) and 3.5 (from 2306 s to 8038 s). Furthermore, Meta-Adaptive preserves better optimality than Meta-ARA*. Specifically, the optimal planning searches decreases from 678 to 620 for Meta-

Adaptive and from 650 to 596 for Meta-ARA*. This confirms that the Meta-Adaptive heuristic is more informative, being capable of guiding the search in more promising directions, thereby leading to increased computational advantages as the dimensionality of the search-space rises.

A qualitative example of the planning procedure according to the three methods is illustrated in Figure 10. As with previous results, Meta-Adaptive demonstrates convergence to the same solution of Meta-Optimal, and in a fraction of the computational time. In contrast, Meta-ARA* is not capable to provide a solution, in this scenario, if constrained to the same amount of planning time as Meta-Adaptive. Therefore, Meta-ARA* is run until at least one solution is found, resulting nonetheless in a sub-optimal path.

VIII. CONCLUSION AND FUTURE WORKS

In this study, we presented an integrated deep meta-learning energy prediction and planning framework for UGVs in unknown terrains. We highlight the benefits of our method (1) to efficiently and rapidly adapt its energy estimates based on small quantities of recent observations and (2) to reduce the vehicle driving energy by integrating the adaptive model into a path planning optimization framework. Specifically, the superior generalization capabilities, in the low data regime, of meta-learning were demonstrated by comparing with alternative machine learning solutions. This can be of particular relevance in the context of autonomous energy-efficient exploration of unknown scenarios, for which the early collection of a comprehensive dataset is often impractical.

We also investigated the performance of different A* heuristic functions in terms of planning optimality and computational efficiency. We showed that the proposed meta-adaptive heuristic function enables convergence to close-to-optimal solutions while retaining lower computational time than a non-adaptive admissible heuristic.

We are extending this work in several directions. While in the current implementation prediction and planning were performed assuming that all environments were traversable at constant speed, future works may consider an extension of the adaptive framework to account for dynamic effects, different speeds, and untraversable paths. Moreover, while in this paper adaptation to new terrains were tested on simulation, using the SCM wheel-terrain interaction model, greater variations may, in reality, exist among different terrain types, thereby requiring increased training effort and data to be captured. Future works will require real-world tests to validate the results obtained during simulation.

REFERENCES

- [1] J. Park, Y. L. Murphey, and M. Abul Masrur, "Intelligent energy management and optimization in a hybridized all-terrain vehicle with simple on-off control of the internal combustion engine," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 6, pp. 4584–4596, 2016.
- [2] O. Lamarre and J. Kelly, "Overcoming the challenges of solar rover autonomy: Enabling long-duration planetary navigation," *CoRR*, vol. abs/1805.05451, 2018.
- [3] J. Y. Wong, *Theory of ground vehicles*. John Wiley & Sons, 2008.
- [4] R. He, C. Sandu, A. K. Khan, A. G. Guthrie, P. S. Els, and H. A. Hamersma, "Review of terramechanics models and their applicability to real-time applications," *Journal of Terramechanics*, vol. 81, pp. 3–22, 2019.

	Meta-Adaptive	Meta-Optimal	Meta-ARA*
Energy Real [kJ]	199.76	199.76	228.84
Energy Pred. [kJ]	160.24	160.24	166.04
Time Planning [s]	1.828	17.125	2.671
Expanded Nodes	8	97	16

Fig. 10: An example of the three planning solutions according to the three alternative heuristic methods.

- [5] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [6] B. Rothrock, R. Kennedy, C. Cunningham, J. Papon, M. Heverly, and M. Ono, "Spoc: Deep learning-based terrain classification for mars rover missions," in *AIAA SPACE 2016*, 2016, p. 5539.
- [7] F. Vulpi, A. Milella, R. Marani, and G. Reina, "Recurrent and convolutional neural networks for deep terrain classification by autonomous robots," *Journal of Terramechanics*, 2021.
- [8] J. Li, Z. Chen, J. Chen, and Q. Lin, "Diversity-sensitive generative adversarial network for terrain mapping under limited human intervention," *IEEE Transactions on Cybernetics*, pp. 1–12, 2020.
- [9] N. Anantrasirichai, J. Burn, and D. Bull, "Terrain classification from body-mounted cameras during human locomotion," *IEEE Transactions on Cybernetics*, vol. 45, no. 10, pp. 2249–2260, 2015.
- [10] R. Gonzalez and K. Iagnemma, "Slippage estimation and compensation for planetary exploration rovers. state of the art and future challenges," *Journal of Field Robotics*, vol. 35, no. 4, pp. 564–577, 2018.
- [11] S. Li, L. Ding, H. Gao, Y.-J. Liu, L. Huang, and Z. Deng, "Adp-based online tracking control of partially uncertain time-delayed nonlinear system and application to wheeled mobile robots," *IEEE Transactions on Cybernetics*, vol. 50, no. 7, pp. 3182–3194, 2020.
- [12] L. Wellhausen, A. Dosovitskiy, R. Ranftl, K. Walas, C. Cadena, and M. Hutter, "Where should i walk? predicting terrain properties from images via self-supervised learning," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1509–1516, 2019.
- [13] J. Vanschoren, "Meta-learning," in *Automated Machine Learning*. Springer, Cham, 2019, pp. 35–61.
- [14] Z. Xu, X. Chen, and L. Cao, "Fast task adaptation based on the combination of model-based and gradient-based meta learning," *IEEE Transactions on Cybernetics*, pp. 1–10, 2020.
- [15] V. Gruning, J. Pentzer, S. Brennan, and K. Reichard, "Energy-aware path planning for skid-steer robots operating on hilly terrain," in *2020 American Control Conference (ACC)*, 2020, pp. 2094–2099.
- [16] N. Gangath, C. Cheng, and C. K. Tse, "A constraint-aware heuristic path planner for finding energy-efficient paths on uneven terrains," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 3, pp. 601–611, 2015.
- [17] G. Huang, X. Yuan, K. Shi, Z. Liu, and X. Wu, "A 3-d multi-object path planning method for electric vehicle considering the energy consumption and distance," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–13, 2021.
- [18] S. Fallah, B. Yue, O. Vahid-Araghi, and A. Khajepour, "Energy management of planetary rovers using a fast feature-based path planning and hardware-in-the-loop experiments," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 6, pp. 2389–2401, 2013.
- [19] G. Sakayori and G. Ishigami, "Energy efficient slope traversability planning for mobile robot in loose soil," in *2017 IEEE International Conference on Mechatronics (ICM)*. IEEE, 2017, pp. 99–104.
- [20] M. Visca, A. Bouton, R. Powell, Y. Gao, and S. Fallah, "Conv1d energy-aware path planner for mobile robots in unstructured environments," 2021.
- [21] M. Cross, A. Ellery *et al.*, "Estimating terrain parameters for a rigid wheeled rover using neural networks," *Journal of Terramechanics*, vol. 50, no. 3, pp. 165–174, 2013.
- [22] J. Dallas, K. Jain, Z. Dong, L. Sapronov, M. P. Cole, P. Jayakumar, and T. Ersal, "Online terrain estimation for autonomous vehicles on deformable terrains," *Journal of Terramechanics*, vol. 91, pp. 11–22, 2020.
- [23] A. Gallina, R. Krenn, M. Scharringhausen, T. Uhl, and B. Schäfer, "Parameter identification of a planetary rover wheel-soil contact model

- via a bayesian approach,” *Journal of Field Robotics*, vol. 31, no. 1, pp. 161–175, 2014.
- [24] M. Apfelbeck, S. Kuß, B. Rebele, and B. Schäfer, “A systematic approach to reliably characterize soils based on bevameter testing,” *Journal of Terramechanics*, vol. 48, no. 5, pp. 360–371, 2011.
- [25] A. Gallina, A. Gibbesch, R. Krenn, T. Uhl, and B. Schäfer, “Multibody simulation of planetary rover mobility in condition of uncertain soft terrain,” *Procedia IUTAM*, vol. 13, pp. 118–126, 2015.
- [26] J. B. Johnson, P. X. Duvo, A. V. Kulchitsky, C. Creager, and J. Moore, “Analysis of mars exploration rover wheel mobility processes and the limitations of classical terramechanics models using discrete element method simulations,” *Journal of Terramechanics*, vol. 73, pp. 61–71, 2017.
- [27] S. Martin and P. Corke, “Long-term exploration tours for energy constrained robots with online proprioceptive traversability estimation,” in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 5778–5785.
- [28] S. Higa, Y. Iwashita, K. Otsu, M. Ono, O. Lamarre, A. Didier, and M. Hoffmann, “Vision-based estimation of driving energy for planetary rovers using deep learning and terramechanics,” *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3876–3883, 2019.
- [29] H. Peng, “A comprehensive overview and survey of recent advances in meta-learning,” *ArXiv*, vol. abs/2004.11149, 2020.
- [30] C. Finn, T. Yu, T. Zhang, P. Abbeel, and S. Levine, “One-shot visual imitation learning via meta-learning,” in *Conference on Robot Learning*, PMLR, 2017, pp. 357–368.
- [31] M. Al-Shedivat, T. Bansal, Y. Burda, I. Sutskever, I. Mordatch, and P. Abbeel, “Continuous adaptation via meta-learning in nonstationary and competitive environments,” *arXiv preprint arXiv:1710.03641*, 2017.
- [32] X. Song, Y. Yang, K. Choromanski, K. Caluwaerts, W. Gao, C. Finn, and J. Tan, “Rapidly adaptable legged robots via evolutionary meta-learning,” *arXiv preprint arXiv:2003.01239*, 2020.
- [33] R. Kaushik, T. Anne, and J.-B. Mouret, “Fast online adaptation in robotics through meta-learning embeddings of simulated priors,” *arXiv preprint arXiv:2003.04663*, 2020.
- [34] L. Johannsmeier, M. Gerchow, and S. Haddadin, “A framework for robot manipulation: Skill formalism, meta learning and adaptive control,” in *2019 International Conference on Robotics and Automation (ICRA)*, IEEE, 2019, pp. 5844–5850.
- [35] A. Nagabandi, I. Clavera, S. Liu, R. S. Fearing, P. Abbeel, S. Levine, and C. Finn, “Learning to adapt in dynamic, real-world environments through meta-reinforcement learning,” *arXiv preprint arXiv:1803.11347*, 2018.
- [36] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap, “Meta-learning with memory-augmented neural networks,” in *International conference on machine learning*. PMLR, 2016, pp. 1842–1850.
- [37] S. Hochreiter, A. S. Younger, and P. R. Conwell, “Learning to learn using gradient descent,” in *International Conference on Artificial Neural Networks*. Springer, 2001, pp. 87–94.
- [38] T. Tieleman and G. Hinton, “Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude,” COURSE: Neural Networks for Machine Learning, 2012.
- [39] D. Sancho-Pradel and Y. Gao, “A survey on terrain assessment techniques for autonomous operation of planetary robots,” *JBIS - Journal of the British Interplanetary Society*, vol. 63, no. 5-6, pp. 206 – 217, May 2010.
- [40] M. Pivtoraiko, R. A. Knepper, and A. Kelly, “Differentially constrained mobile robot motion planning in state lattices,” *Journal of Field Robotics*, vol. 26, no. 3, pp. 308–333, 2009.
- [41] P. E. Hart, N. J. Nilsson, and B. Raphael, “A formal basis for the heuristic determination of minimum cost paths,” *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, July 1968.
- [42] R. Dechter and J. Pearl, “Generalized best-first search strategies and the optimality of A*,” *J. ACM*, vol. 32, no. 3, p. 505–536, Jul. 1985. [Online]. Available: <https://doi.org/10.1145/3828.3830>
- [43] A. Tasora, R. Serban, H. Mazhar, A. Pazouki, D. Melanz, J. Fleischmann, M. Taylor, H. Sugiyama, and D. Negrut, “Chrono: An open source multi-physics dynamics engine,” in *International Conference on High Performance Computing in Science and Engineering*. Springer, 2015, pp. 19–49.
- [44] F. Buse, R. Lichtenheldt, and R. Krenn, “Scm-a novel approach for soil deformation in a modular soil contact model for multibody simulation,” *IMSD2016 e-Proceedings*, 2016.
- [45] S. Taheri, C. Sandu, S. Taheri, E. Pinto, and D. Gorsich, “A technical survey on terramechanics models for tire–terrain interaction used in modeling and simulation of wheeled vehicles,” *Journal of Terramechanics*, vol. 57, pp. 1–22, 2015.
- [46] Z. Janosi and B. Hanamoto, “An analysis of the drawbar pull vs slip relationship for track laying vehicles,” ARMY TANK-AUTOMOTIVE CENTER WARREN MI, Tech. Rep., 1961.
- [47] L. Richter, N. Schmitz, S. Weiss, and Mer/Athena Team, “Inferences of Strength of Soil Deposits along MER Rover Traverses,” in *European Planetary Science Congress 2006*, Jan. 2006, p. 523.
- [48] ProjectChrono. Vehicle models. Wheeled vehicle models. HMMWV. [Online]. Available: https://api.projectchrono.org/vehicle_models.html
- [49] M. Visca, S. Kuutti, R. Powell, Y. Gao, and S. Fallah, “Deep learning traversability estimator for mobile robots in unstructured environments,” 2021.
- [50] M. Likhachev, G. J. Gordon, and S. Thrun, “Ara*: Anytime a* with provable bounds on sub-optimality,” *Advances in neural information processing systems*, vol. 16, pp. 767–774, 2003.

Marco Visca received the B.Sc. in Aerospace Engineering in 2016 and the M.Sc. in Mechatronic Engineering in 2018, both from Politecnico di Torino, Italy. He is currently working toward the Ph.D. degree in Robotics at University of Surrey, U.K.

His research interests include artificial intelligence, machine learning and their application to autonomous vehicles and robots in challenging environments.

Roger Powell is head of the Robotics and AI Group within the Cybernetics Department of RACE (Remote Applications in Challenging Environments), which is part of the UK Atomic Energy Authority (UKAEA). Prior to this appointment he was a Senior Lecturer at Brunel University, in the department of Electronic and Computer Engineering.

Yang Gao (S'00-M'03-SM'09) received the B.Eng. degree in electrical and electronics engineering and the Ph.D. degree in artificial intelligence, control and instrumentation from Nanyang Technological University (NTU), Singapore, in 2000 and 2003, respectively. She is the Professor of Space Autonomous Systems and Founding Head of the multi-award winning STAR LAB at the Surrey Space Centre, University of Surrey, Guildford, UK. She specializes in robotic vision, machine learning, and biomimetic mechanism for industrial applications in extreme environments. She brings over 20 years of R&D experience in solving robotic system problems, and is actively involved in development of real-world space missions, such as ESA's ExoMars, Proba3 and Lunar VMMO, UK's MoonLITE and Moonraker, and CNSA's Chang'E3.

Saber Fallah is the Director of the Connected Autonomous Vehicles Lab (CAV-Lab) at the Department of Mechanical Engineering Sciences at the University of Surrey, where he leads several research activities funded by the UK and European governments (e.g. EPSRC, Innovate UK, H2020, KTP) in collaboration with major companies active in autonomous vehicle and robot technologies. CAV-Lab provides a unique laboratory to design, develop and test the next generation of robotics and autonomous systems used for remote assembly and manufacture, highly automated transportation systems and missions in hazardous environments including space. CAV-Lab also provides expertise in distributed control systems, AI and machine learning, and predictive optimisation techniques. Prior to joining the University of Surrey, he was part of a cross-disciplinary team in Green Intelligent Transportation Systems research (University of Waterloo, Canada). His research interests include deep reinforcement learning, advanced control and prediction and their application to autonomous robot systems.