Masked Sensory-Temporal Attention for Sensor Generalization in Quadruped Locomotion

Dikai Liu[1,2] Tianwei Zhang[2] Jianxiong Yin[1] Simon See[1,3]

***Abstract*— With the rising focus on quadrupeds, a generalized policy capable of handling different robot models and sensor inputs becomes highly beneficial. Although several methods have been proposed to address different morphologies, it remains a challenge for learning-based policies to manage various combinations of proprioceptive information. This paper presents Masked Sensory-Temporal Attention (`MSTA`), a novel transformer-based mechanism with masking for quadruped locomotion. It employs direct sensor-level attention to enhance the sensory-temporal understanding and handle different combinations of sensor data, serving as a foundation for incorporating unseen information. `MSTA` can effectively understand its states even with a large portion of missing information, and is flexible enough to be deployed on physical systems despite the long input sequence.**

Fig. 1. Commonly seen low-level sensors on a quadrupedal robot. However, actual sensor set is still different across models, and sensor degradation can cause part of sensor data to be unreliable or even unavailable. With `MSTA`, we create a generalized model to enhance the understanding of sensor information to handle variable sensor input for quadruped locomotion.

I. Introduction

Benefiting from the rapid advancements of deep reinforcement learning (RL) technology [1]–[4], quadrupedal robots have showcased their capability to navigate in diverse complex terrains. With the increasing availability of affordable quadruped robots on the market, there is a growing interest in developing general-purpose locomotion policies that can fit all types of quadrupedal devices. Unfortunately, existing learning-based locomotion policies are trained for specific models, observation spaces, and tasks, making it challenging to transfer or generalize to other unseen robots or scenarios.

Recently, researchers have developed some generalized policies for quadruped locomotion, such as GenLoco [5] and ManyQuadrupeds [6], which have the ability to adapt to diverse morphologies. However, these methods still depend on a fixed observation space input for generating latent space representations. They become ineffective when facing the following situations: (1) deployment on quadrupeds with a different sensor set; (2) unreliable sensor data due to wear and tear, (3) adapting to a new task with new input. Since sensory feedback is interrelated and each sensor plays a critical role at different stages of the locomotion [7], a policy with a deep understanding of proprioceptive information to handle flexible inputs is desired, to enhance the generalization, flexibility, and extensibility.

One promising solution is self-attention-based transformers [8], which have demonstrated exceptional capabilities in understanding complex sequential information of arbitrary lengths. They have been widely used in robotics to enhance various tasks with multimodal processes [9]–[12]. However, due to the complex model structures and vast parameters, robots driven by transformers often run at very low frequency [10], [11], or depend on external high-power computing platforms [13]. For locomotion tasks, the observation-action data are commonly encoded at the timestep level [14]–[17], which is straightforward and efficient for producing joint commands in an end-to-end manner. However, it limits the transformer's direct access to sensory information and still relies on fixed sensor input as they are hidden behind linear projection, thereby constraining its in-context understanding capability and multimodal nature of the data.

To address the above limitations, we propose Masked Sensory-Temporal Attention (`MSTA`), a novel transformer-based model for end-to-end quadruped locomotion control. It achieves sensor input generalization with its multimodal nature, while still being directly deployable on physical systems. Specifically, in `MSTA`, all sensory data are discretized and tokenized to form a long proprioceptive information sequence. Inspired by the work [18] on learning spatiotemporal information in video understanding, a random mask is applied to remove a portion of the observation during training. This significantly enhances the model's sensory-temporal understanding, to better handle different combinations of sensor data and serve as a foundation for incorporating unseen data. Additionally, it aids in identifying the most essential sensory information, thereby reducing the computational power required for physical deployment.

We conduct extensive experiments in the simulation and physical world. Evaluation results demonstrate that `MSTA` can efficiently handle incomplete sensory information, even with half of the data missing. It is also robust against unseen data, making it a solid foundation for further extensions. With direct sensory-temporal attention, the model

[1] NVIDIA AI Technology Centre (NVAITC); e-mail: {dikail,jianxiongy,ssee}@nvidia.com
[2] College of Computing and Data Science, Nanyang Technological University, Singapore; e-mail: dikai001@e.ntu.edu.sg, tianwei.zhang@ntu.edu.sg
[3] also with Nanyang Technological University and Coventry University

is flexible enough to mix-and-match desired information for finetuneing, meeting the requirement for different end-to-end quadruped locomotion control in the physical world.

II. Related Work

A. Sim-to-Real Policy Learning in Legged Locomotion

Reinforcement learning (RL) has gained significant attention in developing robotic controllers for tasks such as legged locomotion [1]–[4], [15], eliminating the need for extensive prior knowledge. With the advance of robotic simulation, RL-based locomotion is often trained in virtual environments [4], [19], [20] with diverse terrains [20] and randomized environmental factors [3], [4] to improve the policy robustness. This technique is commonly known as domain randomization (DR). Training in simulators also provides rich information, some of which is not easily accessible in the real world (i.e., privileged information). To better interpret such information and bridge the sim-to-real gap when deploying policies to the physical world, system identification is commonly used to transfer knowledge to a deployable student policy. For instance, Lee et al. [3] employed action imitation to infer teacher behaviors using historical proprioceptive data. Kumar et al. [4] further developed a two-stage adaption framework for faster and more robust online transfer, which has become the foundation for many subsequent works [15], [21]. Another approach combines transfer loss with RL loss for joint optimization [16], [22] to allow the student to explore with teacher guidance to the maximization of reward return.

B. Transformer in Robotics

The transformer-based models have been introduced to solve robotic tasks. For instance, Decision Transformer [23] converts states, actions, and rewards into embeddings using an encoder. Trajectory Transformer [24] uses the complete discretized trajectory for language model-like autoregressive prediction. Building on these frameworks, Gato [9] was developed to serve as a general agent for hundreds of tasks, including real-world robotic manipulation. Vision Language Models (VLM) [10] and Vision Language Action Models (VLA) [11], [12] use transformers as interfaces for scene and language understanding to provide high-level commands for robotic control and human-robot interaction but due to the enormous model size, they often run at only 5 Hz [10], [11].

For legged locomotion, which requires real-time control, some work involves outputting high-level commands as an interface. For instance, Yang et al. [25] developed a transformer model for vision-based locomotion, which outputs high-level velocity commands and relies on a dedicated low-level controller for motor control. Similarly, Tang et al. [26] uses gait pattern as the interface for a low level controller. The external controller often requires additional design and training, and to bring transformers to direct motor control, Lai et al. [15] proposed TERT, which utilizes historical observation-action pairs to generate target motor commands directly. Barkour [14] uses a similar architecture to merge multiple specialist policies into a single locomotion policy. Radosavovic et al. [16] applied a similar method to the bipedal locomotion task and later reformulated it as a next token prediction problem [17]. Recently, Sferrazza et al. proposed BoT [27], an embodiment-aware transformer network with body-induced bias based on the embodiment graph with embody-specific masking. However, to achieve real time onboard inference, end-to-end controllers relies on fixed information on each timestep and node for ecoding, exhibiting inflexibility of handling different inputs.

III. Preliminary

We adopt the two-stage teacher-student transfer approach from TERT [15] as the basis, which utilizes a well-trained teacher policy through RL with privileged information.

A. Simulation Environment

We implement the simulation environment based on Isaac Gym and its open-source library IsaacGymEnvs [19] to enable massive parallel training.

Terrain and Curriculum. We adopt the terrain curriculum from [20] with five terrain types (smooth slope, rough slope, stairs up, stairs down, discrete obstacle) and difficulty curriculum. The agent progresses and regresses the level based on the episode cumulative tracked linear reward.

Domain Randomization. To enhance the robustness of the policy, DR is used in the simulation following [20], [28]. We sample the commanded longitudinal and lateral velocity from [-1.0, 1.0] m/s, and horizontal angular velocity first calculated based on sampled heading and capped at [-1.0, 1.0] rad/s. Due to the significant computation required for transformer, a system delay is added [28].

Observations and Actions. The privilege observation e_t for teacher training contains ground-truth data gathered from simulation, including base linear and angular velocity, orientation, surrounding height map and randomized parameters as described above. For proprioceptive information, we use three commonly seen low-level sensors from quadrupeds. i.e., joint encoders, IMU and foot contact sensors. These sensors can provide five sensory data, including joint position $q \in \mathbb{R}^{12}$, joint velocity $\dot{q} \in \mathbb{R}^{12}$, angular velocity $\omega \in \mathbb{R}^3$, gravity vector $g \in \mathbb{R}^3$ and binary foot contact $c \in \mathbb{R}^4$. Furthermore, the randomly sampled user command target $cmd = [v_x, v_y, \omega_z]$ and actions from previous step $a_{t-1} \in \mathbb{R}^{12}$ are added, resulting in an observation of $o_t \in \mathbb{R}^{49}$ for each step. To gather the temporal information, a list of historical proprioceptive information $[o_0, o_1, \cdots, o_T]$ from past $T = 15$ steps is stored. Thus, full observation is in the $\mathbb{R}^{49\times15}$ space. Both the teacher and student output the desired joint position a_t, which is further processed by a PD controller for the output torque $\tau = K_p(\hat{q} - q) + K_d(\hat{\dot{q}} - \dot{q})$, with base stiffness and damping set to 30 and 0.7 respectively and the target joint velocity $\hat{\dot{q}}$ set to 0.

Reward Function for RL. The reward functions are designed to encourage the agent to follow the commanded velocity. Following [4], [20], [28], we primarily penalize the linear and angular movement along other axes, large joint acceleration and excessive power consumption.

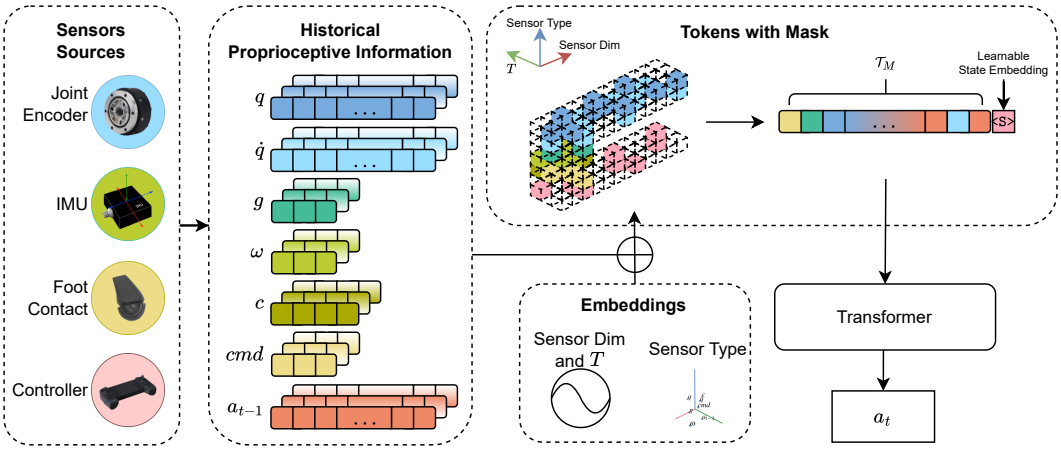

Fig. 2. Overview of our MSTA. We gather proprioceptive information from commonly seen low-level sensors for discretization and tokenization. Similar to video understanding, we add additional embedding in three dimensions: sensor type, sensor dim and time. Before being passed to the transformer, a random mask is applied to partially remove the information and a learnable state embedding <S> is used to consolidate the information for action prediction. The target joint position output is passed to the PD controller for direct joint control.

B. Teacher Policy and Training

We implement a teacher policy following [4]. The teacher first encodes the privilege information e_t, with a factor encoder μ into a latent space l_t, which is then combined with the latest observation-action pair o_t for the teacher policy $\hat{\pi}$ to output the desired joint position $\hat{a}_t$:

$$\hat{l}_t = \mu(e_t), \quad \hat{a}_t = \hat{\pi}(\hat{l}_t, o_t) \tag{1}$$

The μ and π networks are implemented as MLP with hidden layers of $[512, 256, 128]$ and $[256, 128]$, respectively. The teacher policy is trained with PPO [29] directly to maximize the reward return and is shared across all student transfers at later stages for a fair comparison.

IV. Methodology

We present MSTA, a novel transformer-based model to generate a generalized understanding of low-level proprioceptive information for quadruped locomotion in complex terrains to handle different sensor set equipped on various robot models or when the sensors are damaged and not available. Unlike previous works [15]–[17], where each observation-action pair is processed at the timestep level, we treat each sensor modality individually so that the transformer can learn at the lowest level possible. With this foundational understanding, our model is capable of handling different combinations of sensor inputs, enabling better generalization and flexibility. It can potentially be extended to incorporate high-dimensional sensors for more complex tasks. Fig. 2 shows the overview of MSTA.

Sensory-Action Data Tokenize. To learn in-context information at the lowest sensor level, each modality is encoded individually. Instead of the linear projection used in previous transformer-based locomotion controllers [15]–[17], where all sensory observations are merged, individual continuous sensor and control data are mapped to tokens directly. Following previous works [9]–[11], we pass the normalized data through an encoder to discretizes the value into 256 bins, which are further mapped into a learnable embedding space with $d = 128$ dimensions. Compared to timestep level encoding, in this way, the most information is preserved for in-context understanding by transformer.

Positional Embedding and Sensor Type Embedding. We view the encoded information in a three-dimensional way, sensor type, sensor channel and timestep. To distinguish proprioceptive information from different sources with temporal relations, two additional embeddings are added. The first one is a fixed 2D sin-cos position embedding e_p [30] applied on the channel dimension and time axis of each sensor. For instance, $e_P^{i,t}$ means the embedding added to the i-th channel at timestep t This allows the model to handle sensors with varying lengths of dimensions and historical time windows directly and be easily extendable. To accommodate the multimodal nature of the sensory data, another learnable embedding e_S is add to indicate each sensor type. This enables easy mix and match of information from different sensors without concerns about the order or placeholders. When new sensors are added, a new sensor embedding can be trained and added in. Thus, for the embedding of i-th channel of a sensor at timestep t, with original encoded token embedding $e_t^{a,i}$, the finial value in the sequence $\mathcal{T}$ is:

$$\mathcal{T}_t^{a,i} = e_t^{a,i} + e_P^{i,t} + e_S^a \tag{2}$$

Random Masking. Inspired by the use of masking in image and video understanding [18], [31] with autoencoders to improve vision understanding, we create a binary mask M based on the target ratio α to randomly mask out portions of the collected sensory, which are directly removed tokens from the original sequence:

$$\mathcal{T}_M = \{e_i \in \mathcal{T} : M_i = 1\} \tag{3}$$

Since only part of the sensory data are visible to the network, the model is required to infer and reconstruct

the missing information from them, thereby enhancing its understanding of the relationships between different sensory inputs. Furthermore, random masking significantly reduces the training time and computational resources required. With sensor level tokens, the input sequence length grows from T to $49T$ for observation, and as the complexity of self-attention is necessarily quadratic in the input length [32], the added overhead is enormous, and masking makes it more feasible to run during massive parallel training.

Transformer Model. We implement a vanilla transformer model to process the generated tokens. The model consists of multihead self-attention blocks with an MLP ratio of 2.0. An additional learnable state embedding <S> is added to the end of the masked sequence $\mathcal{T}_M$ to consolidate the processed information [33], which is subsequently projected into the action space with an MLP network π:

$$l_t = \texttt{MSTA}([\mathcal{T}_M, \texttt{<S>}]), \quad a_t = \pi(l_t) \tag{4}$$

Teacher-Student Transfer. Following TERT [15], we train MSTA with a two-stage transfer strategy. In the first offline pretraining stage, trajectory is gathered by unrolling the well-trained teacher policy while the student will predict the next actions. This is to ensure that the student can produce reasonable actions during the second online correction stage to overcome the gap of distribution shift by training on its own trajectory. We minimize the loss for action prediction:

$$\mathcal{L} = \|a_t - \hat{a}_t\|^2 \tag{5}$$

V. Experiments and Results

We design and conduct various simulation experiments to evaluate the effectiveness of the proposed MSTA, and its generalization ability for different sensor data. We mainly adopt three metrics: linear velocity tracking return per step, angular velocity tracking return per step, and total final reward return. They indicate how the agent can conduct the task following users' commands and the overall performance. All reported results are averaged over 5000 trails with five terrain types and different levels. They are normalized on the basis of respect teacher data for easy comparison.

A. Impact of Mask Ratio

First, we investigate the maximum portion of missing data that MSTA can handle to reconstruct robot states. During each the transfer stages, we set the masking ratio to 0%, 25%, 50% and 75% independently. Fig. 3 shows the resultant heatmap matrix. When trained without masking, despite the model having very good performance with all the information available, it suffers from missing data and cannot efficiently reconstruct the status. We can also see that the performance is more dependent on the masking ratio in the second stage than that in the first stage. This is because in the second transfer stage, the student is interacting with the environment to reduce the gap caused by missing information and observation shift. In contrast, the mission of the first stage is to generate a usable policy that outputs reasonable actions so the agent does not fail dramatically and has the chance in the second stage to generate high-quality trajectories for optimization, which is achievable even with a masking ratio of 75%. This also demonstrates the importance of using two-stage transfer.

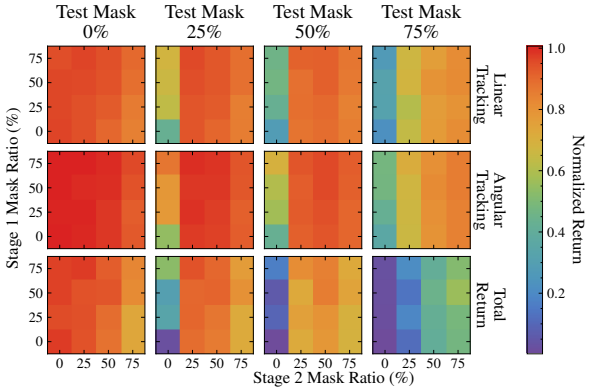

Fig. 3. Heatmap matrix for the performance of models that are trained with different combinations of mask ratios. The three rows from top to bottom represent the linear velocity tracking, angular velocity tracking and total reward return respectively. The four columns denote different masking ratios applied during testing. For each sub-figure, the y-axis is the masking ratio applied during the offline pretraining stage and the x-axis is the masking ratio applied during the online correction stage.

Comparing the performance of these models, we choose the one trained with the masking ratio of 75% in the first stage and 50% in the second stage, which can well balance the resource requirement and agent performance.

B. Comparison with Baselines

We compare MSTA with two baselines. The first is RMA [4], which is implemented with TCN [34] to capture temporal information. The second is TERT [15], a transformer-based framework with linear projection for observations and actions in two favors: concatenated single token and separate tokens for states and action, resulting in T and $2T$ tokens respectively [17]. To evaluate the masking mechanism in our method, we replace the selected observation in MSTA with a learnable representation instead of removing them. To further evaluate the importance and capability of the transformer structure, we replace it with a GRU [35] model. We expand the missing information testing to TERT. However, since the observations and actions in TERT are encoded through linear projection before passing to the transformer, it is impossible to directly remove any input. Thus, the the same learnable masks method is applied to TERT.

All variations of MSTA, TERT and other baselines are trained with the same two-stage transfer, sharing a common well-trained teacher network, and we apply a testing mask of up to 50%, as identified in Section V-A.

Tab. I shows the comparison results. When fully optimized with teacher-student transfer, the performance of all fully trained vanilla policies with complete observations is very close, often within just 2% difference. When faced with incomplete information, transformer-based MSTA can have a better understanding of the data and reconstruct the robot state more accurately than the GRU-based network, even

TABLE I

COMPARISON RESULTS ON DIFFERENT TERRAIN TYPES IN TERMS OF LINEAR VELOCITY TRACKING, ANGULAR VELOCITY TRACKING AND TOTAL REWARD RETURN FOR ALL THE VARIATIONS OF TRAINED MODELS.

Terrain	Metric	Ours			RMA	TERT		GRU			TERT w/ Mask			Ours w/ Learnable		
		0%	25%	50%		Concat	Seperate	0%	25%	50%	0%	25%	50%	0%	25%	50%
Smooth Slope	Linear Tracking	1.00	0.99	0.99	1.00	1.01	1.00	1.00	0.96	0.78	0.95	0.98	0.99	0.21	0.93	0.99
	Angular Tracking	1.02	1.02	1.00	1.02	1.02	1.02	1.02	1.01	0.97	0.99	1.00	0.99	0.35	0.97	0.99
	Total Reward	1.01	1.01	0.99	1.02	1.02	1.02	1.01	0.99	0.84	0.95	0.99	0.98	0.25	0.92	0.99
Rough Slope	Linear Tracking	0.97	0.94	0.95	0.99	1.00	0.98	0.96	0.90	0.74	0.91	0.95	0.93	0.21	0.87	0.95
	Angular Tracking	1.01	1.00	0.98	1.02	1.02	1.01	1.01	1.00	0.95	0.96	0.97	0.94	0.33	0.94	0.96
	Total Reward	0.98	0.95	0.95	1.00	1.01	0.99	0.97	0.92	0.79	0.90	0.94	0.89	0.19	0.80	0.91
Stairs Up	Linear Tracking	0.93	0.91	0.91	0.94	0.95	0.95	0.90	0.91	0.69	0.82	0.86	0.85	0.26	0.81	0.85
	Angular Tracking	1.00	0.98	0.97	0.99	1.00	1.01	0.99	0.98	0.94	0.95	0.95	0.92	0.71	0.94	0.93
	Total Reward	0.95	0.91	0.87	0.95	0.99	1.00	0.90	0.87	0.72	0.80	0.81	0.73	0.54	0.74	0.72
Stairs Down	Linear Tracking	0.94	0.93	0.93	0.96	0.97	0.95	0.92	0.93	0.74	0.86	0.91	0.92	0.71	0.94	0.93
	Angular Tracking	1.00	0.99	0.99	1.00	1.01	1.01	0.99	0.98	0.95	0.95	0.96	0.92	0.70	0.93	0.93
	Total Reward	0.94	0.91	0.91	0.95	1.00	0.98	0.89	0.90	0.77	0.83	0.86	0.77	0.49	0.75	0.74
Discrete	Linear Tracking	0.96	0.96	0.94	0.97	0.99	0.98	0.93	0.88	0.75	0.87	0.91	0.89	0.25	0.87	0.85
	Angular Tracking	1.01	1.01	0.99	1.01	1.02	1.02	1.01	0.99	0.95	0.96	0.97	0.94	0.37	0.94	0.94
	Total Reward	1.00	0.97	0.96	0.99	1.04	1.02	0.94	0.89	0.83	0.88	0.91	0.80	0.03	0.80	0.76
Average	Linear Tracking	0.96	0.95	0.94	0.97	0.98	0.97	0.94	0.92	0.74	0.88	0.92	0.92	0.23	0.87	0.91
	Angular Tracking	1.01	1.00	0.99	1.01	1.01	1.01	1.00	0.99	0.95	0.96	0.97	0.94	0.49	0.95	0.95
	Total Reward	0.97	0.95	0.94	0.98	1.01	1.00	0.94	0.92	0.79	0.87	0.90	0.83	0.30	0.80	0.82

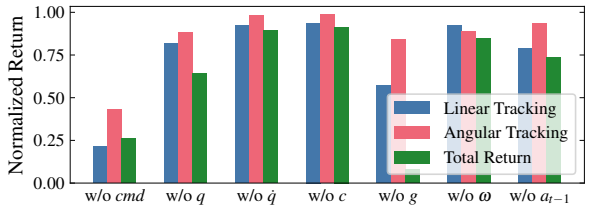

Fig. 4. Performance with certain sensory feedback completely removed.

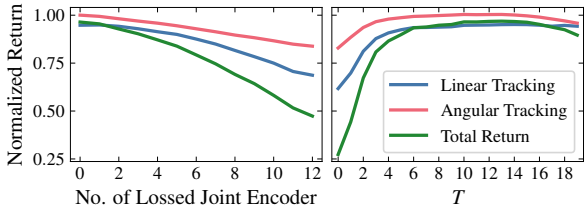

Fig. 5. Performance with various setups: **Left** certain numbers of joint encoders are masked out; **Right** different history time window T is applied.

with only half of the information. When using a learnable representation mask, with MSTA or TERT, the agent underperforms to the vanilla removing mask, especially with full observation, showing that a direct removing mask has an advance in both better performance and less resource required. Although we can hack the linear projection in the TERT network to take in missing information, it is not comparable to direct sensor level tokenzation and attention for sensory information understanding.

C. Generalization, Robustness and Flexibility

While achieving state-of-the-art performance, MSTA offers additional benefits of generalization and flexibility to customize the model after training or even on the fly to fit the deployment requirement. Quadrupeds are equipped with different sensor sets, and sensor damage can cause certain channels or the entire sensor to be unavailable during deployment, which required the robustness against missing information to handle. Furthermore, we can balance the performance and required computation power by using a shorter sequence based on the insights from in-context sensory information understanding.

Important Sensory Feedback. To understand the importance of each sensory feedback, we further investigate the impact of removing each sensor completely from the observation and the results are shown in Fig. 4. It is clear that certain feedback like $\dot{q}$, c, ω and even a_{t-1} are quite redundant and a well trained transformer-based MSTA can easily compensate the missing information from other sources, while the other sensor data are more critical.

Missing of Sensor Dimension. Some proprioceptive information has multiple channels, such as joint encoders and force senors. This means that these sensors can also be damaged independently due to wear and tear from daily operations and it is not easy to have a redundant sensor. Among these sensors, joint encoders are the source of both q and $\dot{q}$ for the observation. From previous analysis, missing of joint information can be crucial. We investigate the scenario where only a few encoders are dead or the data are compromised and need to be excluded. We conduct the test by masking certain numbers of joint encoders and for each masked joint, the related q and $\dot{q}$ are removed completely from the observation. The results are shown in Fig. 5. The loss of the joint encoder can have a great impact on the performance as the related information is very essential for quadruped locomotion. However, our transformer model can still handle multiple missing encoders before large performance degradation.

Time Window. Another special masking is to completely remove some timesteps, the default window, $T = 15$, is equivalent to past 0.3s. We check whether such a long sequence of information is necessary by applying different time windows without masking. The results are shown in Fig. 5. It is clear that MSTA can efficiently extract and reconstruct the robot state for actions even with only 7 steps

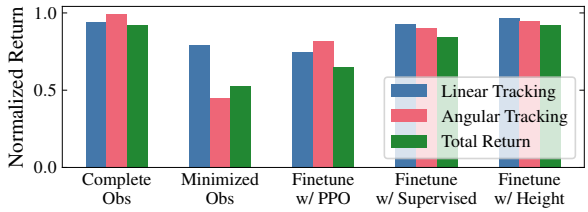

Fig. 6. Performance using minimized observations with finetuning and extension of height map.

of past information. Interestingly, given a longer timeframe like $T = 20$, which the transformer has never seen during training, MSTA is still robust and not affected by such unknown information.

Minimized Observation and Fine-tuning. It is not feasible to infer the transformer with full observation space and long windows with the limited computation power onboard. From the previous analysis, we have identified the important sensors and the minimal history length required. We further explore the feasibility of creating a minimized observation policy based on the information. Using an observation with only cmd, q, g and a_{t-1} with a window of $T = 7$, we are essentially removing 71% of the tokens from the complete training observation space.

When directly deployed with such mask, the policy cannot perform well due to all the missing information. To restore the performance, we freeze the transformer for fast fine-tuning of the projection layers and test both the vanilla PPO [29] and supervised learning with online correction [15]. The performance of the policies is shown in Fig. 6. While both algorithms can help improve the performance of the policy with only minimized observations, supervised learning gives larger boost. Training with the teacher has been identified as one major approach to achieve quadruped locomotion on challenging terrains [4], [15]. Although our foundation with the transformer can provide a solid start point of student policy, additional work is still needed for pure RL-based fine-tuning to reduce the dependence on privilege information.

Extension with New Information. In previous analysis, MSTA is robustness against new timestep information. When extending the capability for quadrupeds, additional sensors such as cameras and LiDAR are often needed. We assess the model's capability of handling previously unseen information, which can be appended into $\mathcal{T}$ as new tokens. For instance, we tokenize the height map information using a vanilla MLP encoder and directly extend it to our minimized observation agent for fine-tuning. The performance of the extended agent is shown in Fig. 6. Height information significantly aids in navigating challenging terrains, such as staircases, and improves the overall locomotion performance even with minimal observations and new encoder needed to be trained. To take the test to an extreme, we added 256 randomly generated dummy tokens, equivalent to a camera frame with ViT [33] before processing, and the agent can still produce explore, which is crucial for two-stage knowledge transfer. This demonstrates that the model can be used as a solid foundation for further extension with high-dimensional information by direct deployment in virtual environments to gather new trajectories. Please refer to the supplementary video for more information.

Fig. 7. Deployment in the physical world on Unitree A1 with minimized observations with zero-shot transfer.

D. Physical Deployment

We successfully deploy the trained policy, exported with JIT, directly on a Unitree A1 robot equipped with a Jetson AGX Orin Developer Kit. The Jetson acts as both the main processor and a payload. No further model optimization is required for a zero-shot transfer. With the onboard processing power, the policy can run at 150Hz with our minimized observation, meeting the requirements for real-time deployment and allowing room for further extension with high-dimensional sensors. However, we notice that the JIT model will have slight difference in output compared to the original model, indicating additional work is needed for better portability. Fig. 7 shows some snapshots from the deployment test. Please refer to the supplementary video for more information.

VI. Conclusion

This paper introduces MSTA, a transformer-based model for quadruped locomotion. It leverages the masking technique and direct sensor-level attention to enhance the understanding and generation of sensory information input. We evaluate the robustness of MSTA with different combinations of proprioceptive information and demonstrate its capability to compensate for missing data and handle unseen information. Finally, we show that MSTA is efficient to be deployed on a physical robot without any additional optimization.

Attention in the full sensory-temporal observation space is computationally intensive and time-consuming. Although using masking can significantly reduce the resources needed, it still takes hours for knowledge transfer, which is considerably longer than existing methods like TCN and temporal-level attention. Additionally, fine-tuning the model with pure reinforcement learning remains challenging, necessitating a more efficient knowledge transfer solution to leverage privileged information effectively. While the policy demonstrated its capability to handle missing data, an addition module is needed to detect and mask out the defected sensors. These will be our future work.

REFERENCES

[1] 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.

[2] 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*, vol. 4, no. 26, p. eaau5872, 2019.

[3] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning quadrupedal locomotion over challenging terrain," *Science robotics*, vol. 5, no. 47, p. eabc5986, 2020.

[4] A. Kumar, Z. Fu, D. Pathak, and J. Malik, "Rma: Rapid motor adaptation for legged robots," *arXiv preprint arXiv:2107.04034*, 2021.

[5] G. Feng, H. Zhang, Z. Li, X. B. Peng, B. Basireddy, L. Yue, Z. Song, L. Yang, Y. Liu, K. Sreenath *et al.*, "Genloco: Generalized locomotion controllers for quadrupedal robots," in *Conference on Robot Learning*. PMLR, 2023, pp. 1893–1903.

[6] M. Shafiee, G. Bellegarda, and A. Ijspeert, "Manyquadrupeds: Learning a single locomotion policy for diverse quadruped robots," *arXiv preprint arXiv:2310.10486*, 2023.

[7] W. Yu, C. Yang, C. McGreavy, E. Triantafyllidis, G. Bellegarda, M. Shafiee, A. J. Ijspeert, and Z. Li, "Identifying important sensory feedback for learning locomotion skills," *Nature Machine Intelligence*, vol. 5, no. 8, pp. 919–932, 2023.

[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[9] S. Reed, K. Zolna, E. Parisotto, S. G. Colmenarejo, A. Novikov, G. Barth-Maron, M. Gimenez, Y. Sulsky, J. Kay, J. T. Springenberg *et al.*, "A generalist agent," *arXiv preprint arXiv:2205.06175*, 2022.

[10] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu *et al.*, "Rt-1: Robotics transformer for real-world control at scale," *arXiv preprint arXiv:2212.06817*, 2022.

[11] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn *et al.*, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," *arXiv preprint arXiv:2307.15818*, 2023.

[12] A. Padalkar, A. Pooley, A. Jain, A. Bewley, A. Herzog, A. Irpan, A. Khazatsky, A. Rai, A. Singh, A. Brohan *et al.*, "Open x-embodiment: Robotic learning datasets and rt-x models," *arXiv preprint arXiv:2310.08864*, 2023.

[13] Z. Fu, T. Z. Zhao, and C. Finn, "Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation," *arXiv preprint arXiv:2401.02117*, 2024.

[14] K. Caluwaerts, A. Iscen, J. C. Kew, W. Yu, T. Zhang, D. Freeman, K.-H. Lee, L. Lee, S. Saliceti, V. Zhuang *et al.*, "Barkour: Benchmarking animal-level agility with quadruped robots," *arXiv preprint arXiv:2305.14654*, 2023.

[15] H. Lai, W. Zhang, X. He, C. Yu, Z. Tian, Y. Yu, and J. Wang, "Sim-to-real transfer for quadrupedal locomotion via terrain transformer," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 5141–5147.

[16] I. Radosavovic, T. Xiao, B. Zhang, T. Darrell, J. Malik, and K. Sreenath, "Learning humanoid locomotion with transformers," *arXiv e-prints*, pp. arXiv–2303, 2023.

[17] 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.

[18] C. Feichtenhofer, Y. Li, K. He *et al.*, "Masked autoencoders as spatiotemporal learners," *Advances in neural information processing systems*, vol. 35, pp. 35 946–35 958, 2022.

[19] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa *et al.*, "Isaac gym: High performance gpu-based physics simulation for robot learning," *arXiv preprint arXiv:2108.10470*, 2021.

[20] 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*. PMLR, 2022, pp. 91–100.

[21] G. B. Margolis, G. Yang, K. Paigwar, T. Chen, and P. Agrawal, "Rapid locomotion via reinforcement learning," *The International Journal of Robotics Research*, vol. 43, no. 4, pp. 572–587, 2024.

[22] D. Liu, T. Zhang, J. Yin, and S. See, "Saving the limping: Fault-tolerant quadruped locomotion via reinforcement learning," *arXiv preprint arXiv:2210.00474*, 2022.

[23] L. Chen, K. Lu, A. Rajeswaran, K. Lee, A. Grover, M. Laskin, P. Abbeel, A. Srinivas, and I. Mordatch, "Decision transformer: Reinforcement learning via sequence modeling," *Advances in neural information processing systems*, vol. 34, pp. 15 084–15 097, 2021.

[24] M. Janner, Q. Li, and S. Levine, "Offline reinforcement learning as one big sequence modeling problem," *Advances in neural information processing systems*, vol. 34, pp. 1273–1286, 2021.

[25] R. Yang, M. Zhang, N. Hansen, H. Xu, and X. Wang, "Learning vision-guided quadrupedal locomotion end-to-end with cross-modal transformers," *arXiv preprint arXiv:2107.03996*, 2021.

[26] Y. Tang, W. Yu, J. Tan, H. Zen, A. Faust, and T. Harada, "Saytap: Language to quadrupedal locomotion," *arXiv preprint arXiv:2306.07580*, 2023.

[27] C. Sferrazza, D.-M. Huang, F. Liu, J. Lee, and P. Abbeel, "Body transformer: Leveraging robot embodiment for policy learning," *arXiv preprint arXiv:2408.06316*, 2024.

[28] I. M. A. Nahrendra, B. Yu, and H. Myung, "Dreamwaq: Learning robust quadrupedal locomotion with implicit terrain imagination via deep reinforcement learning," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 5078–5084.

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

[30] L. Beyer, X. Zhai, and A. Kolesnikov, "Better plain vit baselines for imagenet-1k," *arXiv preprint arXiv:2205.01580*, 2022.

[31] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 16 000–16 009.

[32] F. D. Keles, P. M. Wijewardena, and C. Hegde, "On the computational complexity of self-attention," in *International Conference on Algorithmic Learning Theory*. PMLR, 2023, pp. 597–619.

[33] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[34] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv preprint arXiv:1803.01271*, 2018.

[35] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.