8 Appendix

8.1 Details of robotic environments

We consider 5 tasks (four from DM Control suite [55] and one task on Cassie-walking [56]) for the low dimensional proprioceptive sensory failure, 5 tasks for high-dimensional modality failures from the ManiSkill2 [59] suite as shown in Figure 4. The dimensionality of the observation space and action space is mentioned in Table 4.

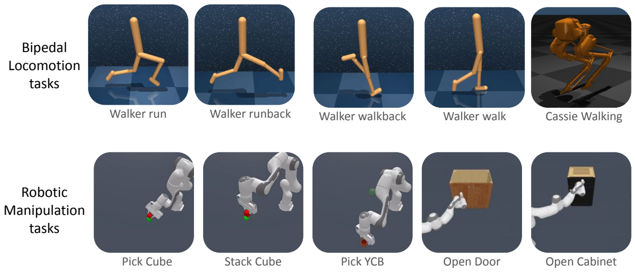

Figure 4: *Tasks* – We consider 5 bi-pedal locomotion tasks (top) and 5 robotic manipulation tasks (bottom). We test the low-dimensional sensory failures (e.g., the robot's proprioception) in the locomotion task suite and the high-dimensional sensory failures (e.g., cameras) in the manipulation task suite.

Task	**Observation dim**	**Observation type**	**Action dim**
Walker-Walk	24	Proprioception	6
Walker-Run	24	Proprioception	6
Walker-WalkBack	24	Proprioception	6
Walker-RunBack	24	Proprioception	6
Cassie-Walk	39	Proprioception	10
Pick Cube	$(64 \times 64 \times 3)$	RGB & Depth (TP, FP)	4
Pick YCB	$(64 \times 64 \times 3)$	RGB & Depth (TP, FP)	7
Stack Cube	$(64 \times 64 \times 3)$	RGB & Depth (TP, FP)	4
Open Door	$(128 \times 128 \times 3)$	RGB & Depth (TP, FP)	11
Open Drawer	$(128 \times 128 \times 3)$	RGB & Depth (TP, FP)	11

Table 4: State and Action spaces of each task in the locomotion and manipulation suite.

8.2 Details on Cassie bipedal walking experiment

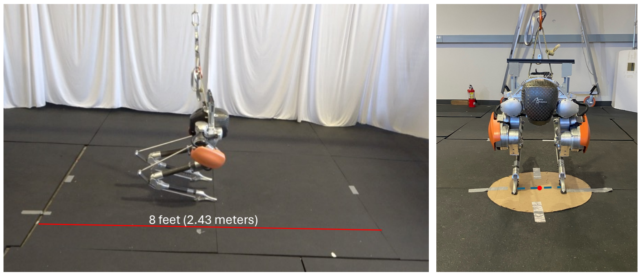

Figure 5: *Measuring Commandability of locomotion policy on Cassie* (left) commadability in the `x-direction` and (right) `angular drift`

Setting: We consider sensory failures in 4 joint links of Cassie, a bipedal robot. Specifically, we simulate the failure in `shin` and `tarsus` links on both the legs. We train the policy following the reward function design in [12] and train the model in 2 stages. The first stage considers dynamics and environment state variable randomization, and the second stage involves learning a robust policy under random external forces (of 30N) with a probability of 0.1 acting on Cassie. We follow this

training procedure from [12]. Each of the stages is trained for 500M steps each.
Metrics: It is important to measure a set of quantifiable metrics to measure the progress in real-world robot deployments. For this we adopt two metrics from [12] that measure the *commandability* of the robot in `x-direction` and `angular drift`. The evaluation setup is shown in Figure 5.
Commandability in x-direction. We consider a distance of 8 feet (or 2.43 meters) as shown in Figure 5 and command Cassie to travel at a speed of 0.25 m/s. For a near-optimal policy, it takes about 9.72 seconds. We then measure how long it takes for an RME based policy to travel the distance and report it in Table 3.
Commandability to measure angular drift. We consider a circular region with a diameter of 50cm and initialize Cassie at the center of the circle (center of Cassie is defined as the mid-point of line joining the feet of Cassie as shown by the ● in Figure 5). We command an angular speed of 15 degree/s and measure the time taken to complete a single rotation (360°). We consider the trial successful if the center of Cassie is withing 10 cm from the circumference of the circle. For an optimal policy it takes 24 seconds and we compare this against RME based policy with sensory failures to see how much additional time it takes for our policy (Table 3).

8.3 Architecture Details

RME **multimodal transformer**: For the `base` TD-MPC2 model (5M parameters), the multimodal transformer consists of 2 encoder blocks – each consisting of multi-head attention (MHA) and feedforward network [47]. We consider 16 heads for MHA module. The input to the transformer is the encodings from each modality and the `[CLS]` token which are all of 512 dimensions each.We also generate fixed 512 dimensional modality type embeddings and add them to the corresponding modality encodings. The hidden dimension of the feedforward network in each encoder block of transformer is 256.

Imputation Network: For the locomotion tasks, we consider a 3 layer MLP with LayerNorm [57] and Mish activation function [58] with the final output layer as linear layer. We train one imputation network for predicting each proprioceptive sensor value for 200 epochs with a learning rate chosen from {1e-5, 5e-5, 1e-4}. The network parameters are optimized by Adam [62] and the transitions from the replay buffer of RME are used to train the network with a 80-10-10 (`train-val-test`) split.

Decoder architecture for Informed TD-MPC2 baseline. We use the decoder structure from [41]. We show the architecture in Table 5.

Layer	Channels	Kernel Size	Norm/Activation
ConvTranspose2d	512	2	LayerNorm/SiLU()
ConvTranspose2d	256	2	LayerNorm/SiLU()
ConvTranspose2d	128	4	LayerNorm/SiLU()
ConvTranspose2d	32	4	LayerNorm/SiLU()
ConvTranspose2d	3	2	LayerNorm/SiLU()
Sigmoid()	-	-	-

Table 5: Decoder Architecture for Informer-TDMPC2 baseline.

8.4 Additional results for sensor failures in high-dimensional modalities (`Easy`)

In this section, we present the results for sensory failures in high-dimensional modalities where only one of the modalities is dropped during evaluation. Specifically, for all tasks either the RGB or the depth modalities from the first-person (FP) or the third-person (TP) cameras are dropped out. Since only one of the modality is dropped, we denote this setting by `Easy` as compared to only operating on a *single* modality (`Hard`) as shown in Table 2 of the main paper.

Task	Base Policy (No Sensor Failure)	Informed TD-MPC2			RME (w/o Modality Dropout)			RME (Ours)		
		max (sensor)	min (sensor)	mean	max (sensor)	min (sensor)	mean	max (sensor)	min (sensor)	mean
PickCube	1.0 ±0	0.79 ±0.17 (C-FP)	0.69 ±0.15 (D-FP)	0.72 ±0.05	0.30 ±0.10 (C-FP)	0.10 ±0.03 (C-TP)	0.17 ±0.11	0.91 ±0.06 (D-FP)	0.80 ±0.20 (D-TP)	0.86 ±0.05
PickYCB	0.83 ±0.30	0.65 ±0.08 (C-FP)	0.60 ±0.19 (C-TP)	0.62 ±0.02	0.40 ±0.11 (D-TP)	0.07 ±0.01 (C-TP)	0.18 ±0.14	0.75 ±0.01 (D-TP)	0.81 ±0.06 (D-FP)	0.79 ±0.04
StackCube	0.96 ±0.05	0.74 ±0.13 (C-FP)	0.51 ±0.21 (D-FP)	0.72 ±0.10	0.31 ±0.18 (C-FP)	0.02 ±0.03 (D-TP)	0.15 ±0.14	0.89 ±0.09 (C-FP)	0.71 ±0.20 (C-TP)	0.82 ±0.07
OpenDrawer	0.81 ±0.20	0.73 ±0.11 (C-FP)	0.69 ±0.18 (C-TP)	0.70 ±0.01	0.27 ±0.05 (D-TP)	0.09 ±0.01 (D-FP)	0.19 ±0.09	0.80 ±0.11 (D-TP)	0.77 ±0.08 (C-FP)	0.79 ±0.01
OpenDoor	0.76 ±0.30	0.66 ±0.10 (C-TP&D-FP)	0.61 ±0.05 (C-FP)	0.65 ±0.02	0.40 ±0.08 (C-TP)	0.11 ±0.27 (D-FP)	0.28 ±0.13	0.81 ±0.23 (D-FP)	0.73 ±0.08 (C-TP)	0.77 ±0.04

Table 6: *Robustness towards modality dropout during deployment* (Easy). We test MMWM against others *by removing one of* {RGB/Depth images}. We report the mean, min, max success rates averaged over 3 seeds. The sensor is of the format (RGB Color (C)/Depth (D) - Third Person (TP)/First Person (FP)) as marked below the numbers.

Results: We show the results for the Easy scenario in Table 6. Not surprisingly, we observe that removing only one modality during evaluation leads to better performance across all baselines. However, there still remains a considerable gap between Informed TD-MPC2 and RME. Additionally, for the case of RME w/o Modality Dropout, we observe that even removal of a single modality can have huge impacts with a relative **357%** drop in performance compared to RME. For the Easy scenario, the RME has a relative increase of 18.1% compared to Informed TD-MPC2 baseline.

8.5 Wall-clock time

In Table 7, we present the wall-clock running time on a single NVIDIA Quadro RTX 8000 GPU. The increase in relative wall-clock time because of addition of RME is 27% when we don't include the Walker tasks. Adding the walker tasks skews the results to a relative 72% increase in the wall-clock time. However, the Walker tasks take the least amount of time to learn (3 hours), so the impact of such an increase is not significant in practice.

Task	TD-MPC2 ↓	RME ↓
Walker	1	3.1
Cassie	70	70.1
PickCube	13	20
StackCube	25	32

Table 7: **Wall-clock training time in (hrs).**

8.6 Ablation: Stability of Transformer as opposed to Masking MLP for low-dimensional proprioceptive inputs

As mentioned in Section 4, we observed that, to accommodate low-dimensional sensors such as proprioception, treating each low-dim sensor as an independent input after encoding via MLP led to instability in training. We show one such result in Table 6 on Walker Walk, where only 1 seed for a transformer with 16 heads in the multi-head attention led to a successful policy. Remaining 4 seeds as well as *all* seeds for 8 and 4-head transformer layer led to no successful training. TThis could be potentially due to the difficulty of converting a small-dimensional vector, such as the one in proprioceptive input, into a meaningful 512 dimensional modality encoding vector via MLP. This also can add redundant dimensions in the latent space, slowing down the transformer's training

8.7 Compounding errors in imputation

We additionally analyze how the error of the imputation network behaves as a function of time steps. Specifically, for the imputation baseline in Cassie simulation, we plot the mean-squared error between the predicted missing sensor values and the ground truth value. The imputation network is

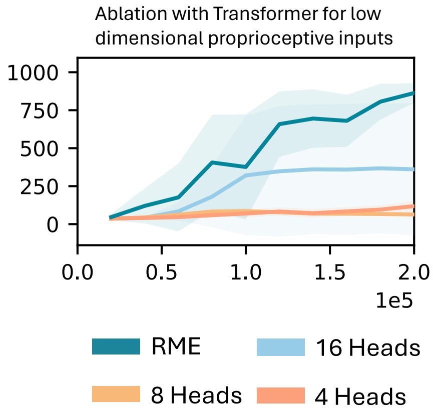

Figure 6: *Ablation study on instability of using low-dimensional sensors as direct input to transformer*

trained to predict the current timestep's value based on a history of the previous 10 timesteps.
Observation: We observe that the initial 5-10 timestep predictions of the imputation network are accurate. However, the prediction quickly worsens around t=15-20. Visually, this corresponds to Cassie finding it hard to balance its legs on the ground and after this point, it tries to maintain its balance before falling onto the ground in about 250 timesteps. We hypothesize that states that correspond to this peak in the plot are potentially underexplored in the replay buffer on which the imputation network was trained – hence causing a compounding effect in the imputation prediction values, and thus causing the robot to fail.

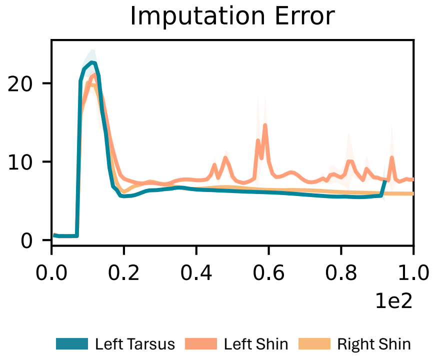

Figure 7: *Imputation error as a function of time steps (on horizontal axis).*