GWM: Towards Scalable Gaussian World Models for Robotic Manipulation

Supplementary Material

A. Datasets and Benchmarks

Robocasa. The dataset consists of robot manipulation data extracted from the MuJoCo simulation environment using a Franka Emika Panda robot, with a focus on kitchen scenarios. For our experiments, we used the Human-50 (H-50) and Generated-3000 (G-3000) datasets provided by RoboCasa, which are automatically generated using MimicGen [51] based on human demonstrations. The benchmark includes 24 atomic tasks, as detailed in Table 2.

Metaworld. MetaWorld is a commonly used benchmark for meta-reinforcement learning and multi-task learning. It consists of 50 distinct robotic manipulation tasks involving a Sawyer robot arm in simulation. The observation is an RGB image of size 64 × 64, and the action is a 4-dimensional continuous vector.

Table A1. Hyper-parameters of the model-based RL experiments.

Model-based RL	Hyper-parameter	Value
Rollout Phase	Init rollout batch size	640
	Interval	200 steps
	Batch size	32
	Horizon	10
Training phase	Init training steps	1000
	world model training interval	10 steps
	Batch size	16
	Sequence length	12
	Context frames	2
	Prediction horizon per inference	1
	Learning rate	1×10^{-4}
	Optimizer	AdamW

B. Implementation Details

B.1. EDM Preconditioning

As mentioned in Section 3.2, we list the preconditioners here that are designed to improve network training [32]:

$$c_{in}^{\tau} = \frac{1}{\sqrt{\sigma(\tau)^2 + \sigma_{data}^2}} \tag{A1}$$

$$c_{out}^{\tau} = \frac{\sigma(\tau)\sigma_{data}}{\sqrt{\sigma(\tau)^2 + \sigma_{data}^2}} \tag{A2}$$

$$c_{noise}^{\tau} = \frac{1}{4}\log(\sigma(\tau)) \tag{A3}$$

$$c_{skip}^{\tau} = \frac{\sigma_{data}^2}{\sigma_{data}^2 + \sigma^2(\tau)}, \tag{A4}$$

where $\sigma_{data} = 0.5$. The noise parameter $\sigma(\tau)$ is sampled to maximize the effectiveness of training as follows:

$$\log(\sigma(\tau)) \sim \mathcal{N}(P_{mean}, P_{std}^2), \tag{A5}$$

where $P_{mean} = -0.4, P_{std} = 1.2$.

Table A2. Hyper-parameters of the Imitation Learning experiments.

Hyper-parameter	Value
Policy Embedding dimension	512
Number of transformer layers	6
Number of attention heads	8
Context length	10
Activation	GELU
Algorithm	Behavioral Cloning
Batch size	16
Learning rate	1e-4
Optimizer	AdamW
L2 regularization	0.01
Number of atomic tasks	24
Training data	50 demos per task
Frame stack	10

B.2. Architectural Design

The variational autoencoder employs a transformer-based architecture with point embedding for encoding point cloud inputs. It uses farthest point sampling to downsample the original point cloud ($N = 2048$) to a manageable number of latent points ($M = 512$), followed by a series of self-attention and cross-attention blocks. For the probabilistic variant, the encoder outputs mean and logvar parameters to sample latent vectors through the reparameterization trick, with an optional KL divergence regularization term. The diffusion model $\mathcal{D}_\theta$ is structured as a Vision Transformer (DiT), processing pointmap patches through multiple transformer blocks with adaptive layer normalization (adaLN) for conditioning on timesteps and actions. The input consists of stacked current, noisy next observations, time embedding, and the current action embedding. The model predicts the denoised next state according to EDM formulation. The reward model R_ψ combines convolutional encoding with sequential modeling, consisting of ResBlocks with optional attention layers followed by an LSTM. The encoder processes pairs of observations (current and next states) while conditioning on embedded actions, and the

LSTM captures temporal dependencies before final reward prediction through an MLP head. Before inference, the LSTM hidden states are initialized through a burn-in procedure with conditioning frames.

B.3. Hyper-parameters

The hyper-parameters of the Robocasa and Metaworld experiments are listed in Table A2 and Table A1, respectively.