A. SLAM-net components

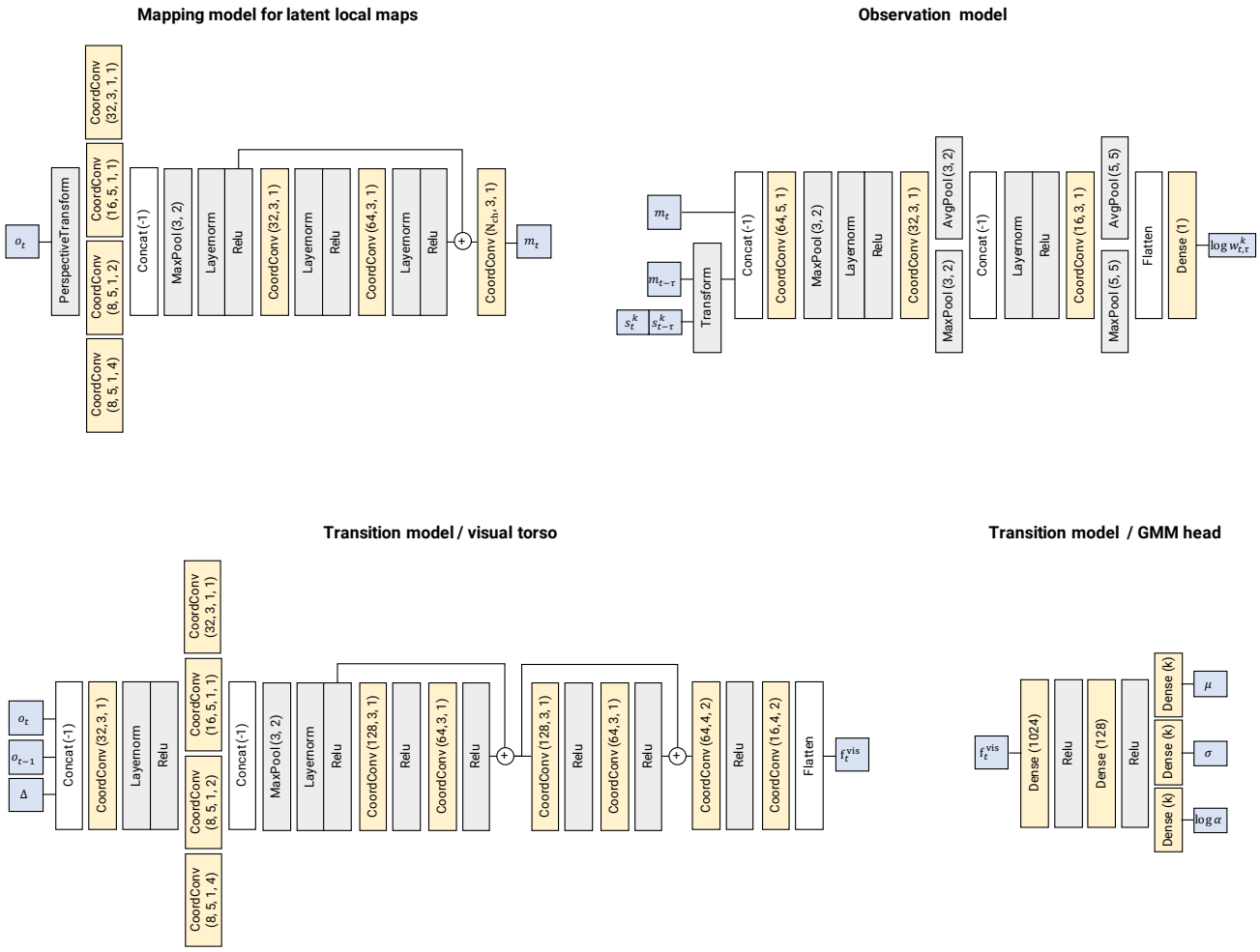

Figure 4: **Network architectures of SLAM-net components.** The notation is as follows. **Conv**(f, k, s, d): convolutional layer with f filters, k×k kernel, s×s strides, d×d dilation. **CoordConv**: convolutional layer that takes in extra channels that encode constant pixel coordinates. **MaxPool**(k, s): max-pooling layer with k×k kernel and s×s strides. **AvgPool**(k, s): average-pooling layer with k×k kernel and s×s strides. **Dense**(h) dense fully-connected layer with h output units. **Concat**(-1): concatenate input images along their channel axis. **Transform**: spatial image transformation using a spatial transformer network [29]. Given a relative pose and orientation it applies translational and rotational image transformations. **PerspectiveTransform**: geometric perspective transformation to a top-down view, given a known camara matrix and a target resolution. **Inputs and outputs**. o_t: input image at time t, either RGB or depth. m_t: local map at time t. s_t^k: state s along particle trajectory for particle k and time t. $\log w_{t,\tau}^k$: estimated compatibility of map-pose pairs at time t and $t-\tau$ for particle k. Δ: difference of input image pair $o_t - o_{t-1}$. $\mathrm{f}^{\mathrm{vis}}{}_t$: intermediate visual features in the transition model. μ: mean predictions for a Gaussian mixture model (GMM) with $k=3$ components. σ: predicted standard deviations. $\log\alpha$: mixture log-probabilities.

A.1. Observation model

In the particle filter the observation model estimates the log-likelihood $\log w_t^k$, the probability of the current observation o_t given a particle trajectory $s_{1:t}^k$ and past observations $o_{1:t-1}$. The particle weight is multiplied with the estimated log-likelihood.

In SLAM-net we decompose this function. A mapping model first predicts a local map m_t from o_t. The observation model

than estimates the compatibility of m_t with $m_{1:t-1}$ and $s^k_{1:t}$, by summing pair-wise compatibility estimates.

$$\log w^k_t \approx \sum_{\tau \in T} \log w^k_{t,\tau} \tag{1}$$

$$\log w^k_{t,\tau} = f^{\text{obs}}_\theta(m_t, s^k_t, m_{t-\tau}, s^k_{t-\tau}) \tag{2}$$

The network architecture of f^{obs}_θ is shown in Fig. 4. Weights are shared across particles and time steps. The important component of this model is the image transformation (Transform), that applies translational and rotations image transformations to $m_{t-\tau}$ given the relative poses defined by s^k_t and $s^k_{t-\tau}$. We use spatial transformer networks [29] that implement these transforms in a differentiable manner.

A.2. Mapping model

The mapping model component takes in the o_t observation (160×90 RGB or depth image) and outputs a local map m_t. Local maps are $40\times40\times N_{\text{ch}}$ grids, which can be understood as images with N_{ch} channels. The local maps either encode latent features ($N_{\text{ch}} = 16$) or they are trained to predict the occupied and visible area. Each cell in the local map corresponds to a 12×12cm area in front of the robot, *i.e.*, the local map covers a 4.8×4.8m area.

The network architecture (configured for latent local maps) is shown in Fig. 4. The same network architecture is used for RGB and depth input. We apply a fixed perspective transform to the first-person input images, transforming them into 160×160 top-down views that cover a 4.8×4.8m area. The perspective transformation assumes that the camera pitch and roll, as well as the camera matrix are known.

When SLAM-net is configured with local maps that predict occupancy we use a similar architecture but with separately learned weights. In case of RGB input our network architecture is the same as the mapping component in ANS [10], using the same ResNet-18 conv-dense-deconv architecture. We freeze the first three convolutional layer of the ResNet to reduce overfitting. In case of depth input our network architecture is similar to the one in Fig. 4, but it combines top-down image features with first-person image features. When SLAM-net is configured with both occupancy and latent local maps we use separate network components and concatenate the local map predictions along their last (channel) dimension.

For the KITTI experiments we use 40×40 local maps where cells are 70×70cm, i.e., a local map captures a 28×28m area. We use an equivalent network architecture that is adapted to the wider input images.

A.3. Transition model

The transition model takes in the last two consecutive observations (o_t, o_{t-1}) and it outputs a distribution over the relative motion components of the robot. It can also take in the last robot action a_{t-1} when it is available.

Our transition model parameterizes a Gaussian mixture model (GMM) with $k=3$ mixture components, separately for each coordinate (x, y, yaw) and each discrete robot action a. The network architecture is shown in Fig. 4. The model first extracts visual features $\text{f}^{\text{vis}}{}_t$ from a concatenation of o_t, o_{t-1} and their difference $\Delta = o_t - o_{t-1}$. The visual features are then fed to GMM heads for each combination of robot action a and x, y, yaw motion coordinates. Each GMM head uses the same architecture with independent weights.

B. Additional figures

Figure 5: **SLAM-net with RGBD sensor.** Trajectory sample from the Gibson data traj_exp_rand test set. In this configuration SLAM-net local maps predict occupancy (third row). The enlarged view (bottom row) shows a 2×2m window of the predicted global map. The true trajectory is in red; particle trajectories are in blue. The shade of blue indicate particle weights (darker color means larger particle weight). Notice that particle trajectories maintain multi-modal trajectories. Low weight particles are dropped after resampling. The figure is best viewed using zoom.

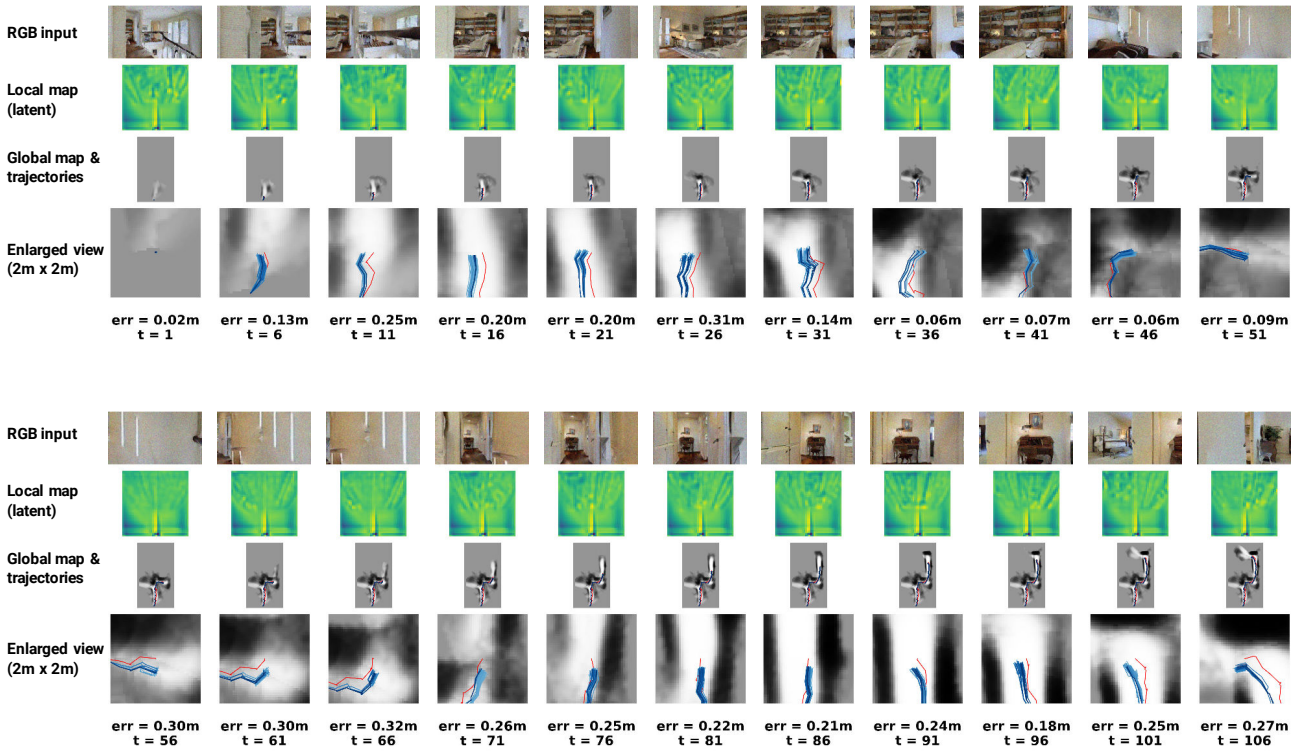

Figure 6: **SLAM-net with RGB only sensor.** Trajectory sample from the Gibson data traj_exp test set. In this configuration SLAM-net local maps have 16 latent feature channels. We visualize normalized features of a single channel (second row). The enlarged view (bottom row) shows a 2×2m window of the predicted global map. The true trajectory is in red; particle trajectories are in blue. The shade of blue indicate particle weights (darker color means larger particle weight). Notice that particle trajectories maintain multi-modal trajectories. Low weight particles are dropped after resampling. The figure is best viewed using zoom.

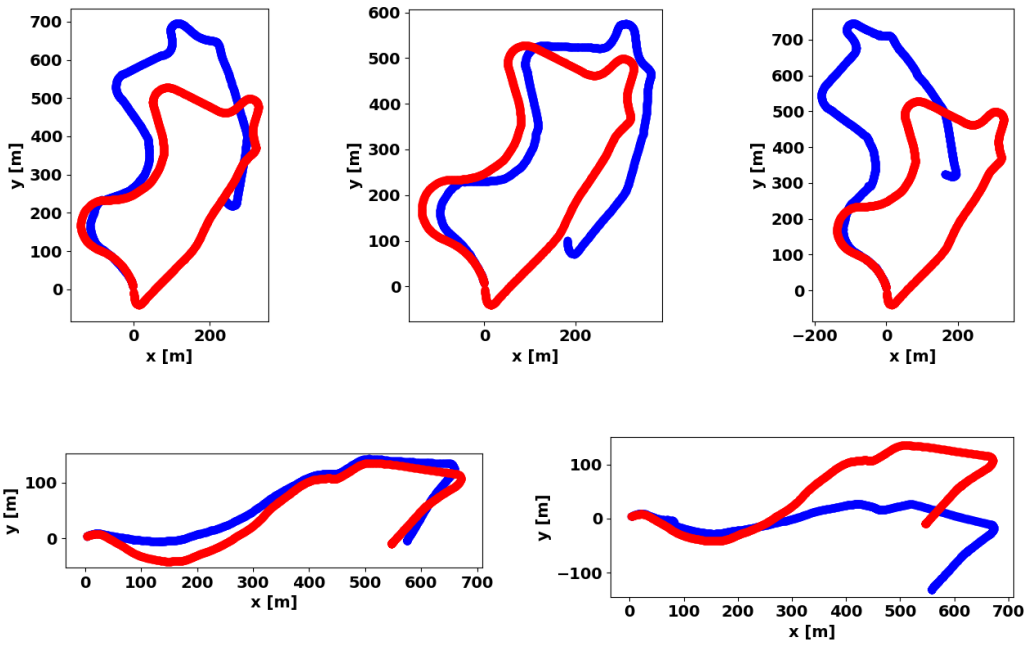

Figure 7: **KITTI trajectories.** The figure shows SLAM-net predictions (before alignment) with different random seeds for the Kitti-09 trajectory (top row) and the Kitti-10 trajectory (bottom row). The predicted trajectory is in blue, the true trajectory is in red.