# Supplementary Material

## 1 Fly Model

The Fly-vs-Fly dataset we use (Eyjolfsdottir et al., 2014) contains annotated tracks of fruit flies interacting with each other. In order to expose this to our general model, we are interested in the most basic representation or encoding of perceptual input and behavioral actions. At each timestep, the fly has a field of view available to it, which can contain solid surfaces (walls of the petri dish) and any number of other flies. In keeping with Eyjolfsdottir et al. (2016), the agent's visual field is divided into 72 individual slices, and the first index encodes the inverse distance to an object starting at the slice directly behind the fly's orientation, (i.e. 180 degrees). This procedure is repeated for each slice going clockwise, until slice 72 is again at 180 degrees. This provides two generic visual field encodings for the fly, one denoting walls and the other denoting flies (see Figure **??**). Finally, this encoding scheme takes care of realistic conditions such as occlusion, multiple other flies, and new environments as well.

The action space is treated as follows: for each fly, its permissible actions are forward and backward motion, changing its wing angle, changing its wing length, extending and contracting its body (thereby producing a change in the visual field of other agents around it), and finally yawing or turning in place. At each timestep, these actions are encoded by a delta to the previous position. That is, the fly knows where it currently is and chooses for each of the 9 discrete actions, some delta away from its position in the corresponding unit of measurement. If a fly wishes to walk towards an object at its 3 o'clock, it will produce a 90 degree turn, followed by a movement forward some number of units. Another example is during a mating ceremony, male flies often encircle the female and vigorously flap its wings, which is represented by a series of sharp and quick wing deltas and changing of angles.

**Spatial Localization.** Let:

$$v_{t,f} = \{o_{t,f}, m_{t,f}^{fwd}, m_{t,f}^{lat}, w_{t,f}^{ll}, w_{t,f}^{la}, w_{t,f}^{rl}, w_{t,f}^{la}, b_{t,f}^{maj}, b_{t,f}^{min}\}$$

For simplicity, all of the following are understood as the deltas, or the change in the specified variable at a given timestep:

- Let $o_{t,f}$ denote the orientation of the fly.
- Let $m_{t,f}^{fwd}$ and $m_{t,f}^{lat}$ denote the motion parallel to and orthogonal to the fly's orientation, respectively (i.e. forward and lateral movement)
- Let $\{w_{t,f}^{ll}, w_{t,f}^{la}, w_{t,f}^{rl}, w_{t,f}^{la}\}$ denote the left wing length, left wing angle, right wing length, and right wing angle, respectively. Wing angles are measured with respect to the axis given by the fly's current orientation.
- Let $b_{t,f}^{maj}$ and $b_{t,f}^{min}$ denote the fly body major and minor axis length. While the flies do not actually change their body size, they might reorient themselves in the third dimension, for example by climbing the walls of the dish, which in 2D view results in changing their body size.

For clarity, at each timestep, the observed motions $m_{t,f}^{fwd}$ and $m_{t,f}^{lat}$ are measured with respect to the fly's new orientation, after it makes a rotation in place according to $o_{t,f}$. For these actions, each can be thought of as a velocity of sorts, with the basis vector being the fly's own body axis. Cueva and Wei (2018) found that modeling movement using velocities leads to the emergence of neurological grid cells resemblance in the RNN parametrization, which provides a rationale for this encoding.

**Sensory Encoding.** In the fly model this consists of the fly's visual input and the relative positions of its body parts

- Let $s_{t.f}^{wall}$ denote 72-dimensional visual input of surrounding walls. Each slice contains the inverse Eu-

clidean distance to an object in the field of view, with 0 denoting no object present.

- Let $s_{t,f}^{fly}$ denote the 72-dimensional visual input of other agents/flies present, with the same formula as above.
- Let $\{\hat{o}_{t,f}, \hat{w}_{t,f}^{ll}, \hat{w}_{t,f}^{la}, \hat{w}_{t,f}^{rl}, \hat{w}_{t,f}^{la}, \hat{b}_{t,f}^{maj}, \hat{b}_{t,f}^{min}\}$ encode the flies current physical state, which are body and wing configurations. Note that unlike the actions, these are specified as absolute values and not deltas. We include knowledge of the fly's global orientation, since flies are known to have internal compasses (Clandinin and Giocomo, 2015).

Together these values constitute the fly's perceptual input. Note that the fly does not have direct perception of its position in space, but can infer that information from the distances to walls in different directions.

$$v_{t,f} = \{s_{t.f}^{wall}, s_{t,f}^{fly}, \hat{o}_{t,f}, \hat{w}_{t,f}^{ll}, \hat{w}_{t,f}^{la}, \hat{w}_{t,f}^{rl}, \hat{w}_{t,f}^{la}, \hat{b}_{t,f}^{maj}, \hat{b}_{t,f}^{min}\}$$

### 1.1 Generation and Inference

We introduce one additional piece of notation for convenience. In this 2D world, let $(c_f^{x_0}, c_f^{y_0})$ denote the initial cartesian coordinates of the fly, relative to some arbitrary reference point. For consistency, we consider an environment bounded by some area, $A$ with the top-left coordinate as (0,0). Similarly, let $\hat{i}_f = \{\hat{o}_{0,f}\hat{w}_{0,f}^{ll}, \hat{w}_{0,f}^{la}, \hat{w}_{0,f}^{rl}, \hat{w}_{0,f}^{la}, \hat{b}_{0,f}^{maj}, \hat{b}_{0,f}^{min}\}$, which denotes the starting physical configuration of the fly.

We will show how to run the forward model and then to perform inference in the model. Although the model is factorized in flies, each fly interacts in the non-stationary, multi-agent setting by incorporating the perceptual inputs of other agents in a generalized way. We construct the factored generative model of behavior as follows:

- For $t = 0$, for $f$ in $\{1, \ldots, F\}$:
  - Initialize RNNs
  - $(c_f^{x_0}, c_f^{y_0}) \sim \text{Uniform}(A)$
  - $\hat{i}_f \sim \hat{p}(\cdot|\ldots)$

Subsequently, we will outline model specifics and give the intuition for their design.

This model uses $p_{t,f}$ to paramterize the timestep-wise VAE, by allowing the sensory information to be given to the latent RNN *before* generating $z_{t,f}$, (sometimes $y_{t,f}$) and $x_{t,f}$. In all VRNN models, we directly use the sensory data immediately after an action is taken and before the next action is produced. For clarity, we refer to all latents, which may include a discrete $y$ as $z_{t,f}$ below:

- For $0 < t < T$ , for $f$ in $\{1, \ldots, F\}$:

$$\begin{aligned} v_{t,f} &= \zeta(\{x_{i,k}\}_{i=1,k=f}^{t-1}, (c_f^{x_0}, v_f^{y_0}), \hat{i}_f, \{v_{t,k}\}_{k=1}^{F \setminus f}) \\ h^{\langle t-1,f\rangle} &= \gamma_\psi\big(h^{\langle t-2,f\rangle}, z_{t-1,f}, v_{t,f}, x_{t-1,f}\big) \\ z_{t,f} &\sim p_{\theta_1}(\cdot|h^{\langle t-1,f\rangle}) \\ x_{t,f} &\sim p_{\theta_2}(\cdot|h^{\langle t-1,f\rangle}, z_{t,f}) \end{aligned}$$

The joint probability of the above model factorizes as:

$$p(z_{1:T,1:F}) = \prod_{f=1}^{F}\prod_{t=1}^{T} p_{\theta_1}(z_{t,f}|h^{\langle t-1,f\rangle}) p_{\theta_2}(x_{t,f}|h^{\langle t-1,f\rangle}, z_{t,f}) \tag{1}$$

The proposal distribution is as follows:

$$q_\phi(z_{1:T,1:F}) = \prod_{f=1}^{F}\prod_{t=1}^{T} q_{\phi_1}(z_{t,f}|h^{\langle t-1,f\rangle}, v_{t,f}) \tag{2}$$

To be precise, $\zeta(\cdot)$ is a function that returns the sensory encodings, $v_{t,f}$ of a fly, given its past trajectories to a point, its own initial conditions, and the position of other flies at the time. In practice, it can be implemented in a recursive manner. Given the past coordinates and the most recent action, update $v_{t,f}$ for every fly based on individual actions, and memoize the new coordinates.

### 1.2 Data cleaning

For the most part, the data is exhaustive, but a small percentage of tracking data is missing, which we fill in with a linear interpolation between the previous and next known frames. For example, if there is data missing for the flies position for 2 frames, we assume it walked in a straight line from its previous known location to the next known location. When the missing data is rotational, we interpolate with the assumption that the fly rotated along the shorter of the two possible arcs to the known orientation.

Exact measurements are given for $y_{t,f}$ and $\{\hat{o}_{t,f}\hat{w}_{t,f}^{ll}, \hat{w}_{t,f}^{la}, \hat{w}_{t,f}^{rl}, \hat{w}_{t,f}^{la}, \hat{b}_{t,f}^{maj}, \hat{b}_{t,f}^{min}\}$, but $s_{t,f}^{fly}$ and $s_{t,f}^{fly}$ are manually calculated. Sensory data for the other fly is approximated with the opposite fly being an exact circle with radius 12.95. In reality, these flies are ellipses.

**Training** Each VRNN model is trained with discrete and continuous latents. The wake loss of $\theta$ (update of $\theta$ using IWAE) is annealed to a constant multiplier of 1.0 on the regularization term (KL between $q(z|y)$ and $p(z)$) over 10000 steps. We use Adam with learning rate of $0.00002$ with 25 particles for training all VRNNs, using CWS or not. For the RNN baseline, we use RMSprop with 0.5 weight decay and train to 1500 iterations with a learning rate of $0.00002$. All models are trained with a batch size of 32 (or 16 per fly).

The dataset we use for training and testing consists of 200 length sequences that comprise at least 25 percent well defined behavior. That is, our dataset is labeled with actions consisting of lunges, wing threats, charges, holds, and tussles. Erratic behavior is also labeled as well as idle behavior. We define interesting actions as not idle nor erratic. the training dataset consists of 3292 sequences pairs or 6584 total sequences of length 200 behavior. Of the 1,316,800 frames, 717188 are idle, 3567 are erratic, 83990 are tussles, 7946 are holds, 337 are charges, 485335 are wing threats, and finally, 24924 are lunges. (the labeled behavior sums to 1,323,287 because there are overlapping frames, i.e. flies may charge or hold as part of tussling)

**Model architecture** The neural network architecture for the RNN uses two parallel GRU cells with hidden dimension of 150. Input to the first is 160 dimensional. The output from the second GRU is used as input to a 2-layer MLP with ReLU activations, to produce the final action output which is 9-dimensional.

For VRNN models, link functions in the VRNN models are parametrized by 3-layer MLPs with 100 hidden dimensions. The inference network link functions work similarly, and take as input variables and hidden outputs from the generative model. When multiple link functions enter a single node, we concatenate all the incoming vectors. For the recurrence, we use a 2-layer GRU for both inference and the generative model.

## References


Thomas R Clandinin and Lisa M Giocomo. Neuroscience: Internal compass puts flies in their place. *Nature*, 521(7551):165, 2015.

Christopher J Cueva and Xue-Xin Wei. Emergence of grid-like representations by training recurrent neural networks to perform spatial localization. *arXiv preprint arXiv:1803.07770*, 2018.

Eyrun Eyjolfsdottir, Steve Branson, Xavier P Burgos-Artizzu, Eric D Hoopfer, Jonathan Schor, David J Anderson, and Pietro Perona. Detecting social actions of fruit flies. In *European Conference on Computer Vision*, pages 772–787. Springer, 2014.

Eyrun Eyjolfsdottir, Kristin Branson, Yisong Yue, and Pietro Perona. Learning recurrent representations for hierarchical behavior modeling. In *International Conference on Learning Representations*, 2016.