GÖTEBORGS
UNIVERSITET

CHALMERS

GÖTEBORGS
UNIVERSITET

CHALMERS

A Generic Model of Motivation in Artificial Animals Based on Reinforcement Learning

Authors:
Pietro Ferrari,
Birger Kleve
Supervisor:
Claes Strannegård
Examiner:
Devdatt Dubhashi

Tuesday, 15 June, 2021

Introduction

- Build a model of motivation
  - Inspired by biology
  - To create reward signal
- Simulate Artificial Animals in Ecosystem
  - Animats
  - Simulate six interesting behaviours
  - Simulate Copepod
- Part of Ecosystem research at Chalmers [1]
  - Ecotwin.se

Motivation

- Simulating ecosystems is useful to understand dynamics
    - Impact of overfishing
    - Reintroduction of species
    - Impact of climate changes or regulation
- Traditionally analyzed using analytical models
    - Lotka-Volterra systems of differential equations
- Simulating with Reinforcement Learning
    - Higher detail
    - More complex simulations/information

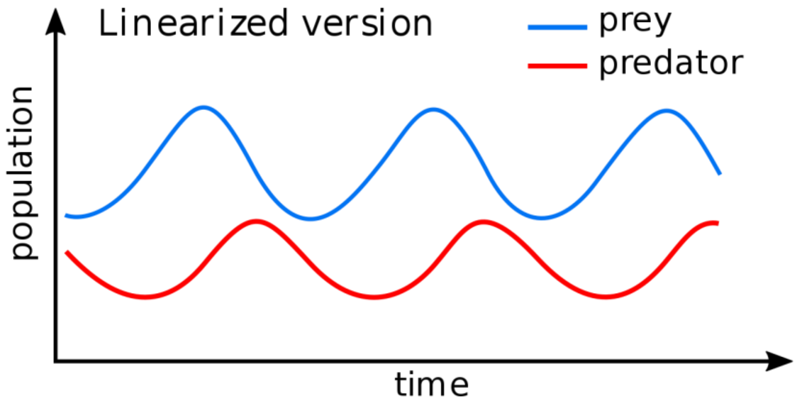


Fig: Lotka-Volterra prey-predator system.

Goal

- *Homeostatic regulation* as means of motivation
    - Regulating physiological conditions (and sensory stimuli)
    - Strive to maintain *homeostasis*
- Elicit behaviours simply by striving for homeostasis
- Generate animats reward by their homeostatic state
- Implement, and compare to previous theoretical work by Keramati et al.

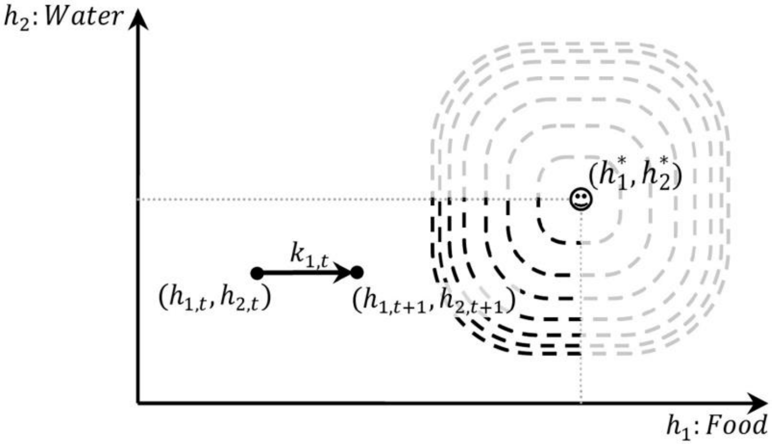


Fig: Homeostatic space showing homeostasis points for food and water.

Mehdi Keramati and Boris S Gutkin. "A reinforcement learning theory for homeostatic regulation". In: *Advances in neural information processing systems* 24 (2011), pp. 82–90.

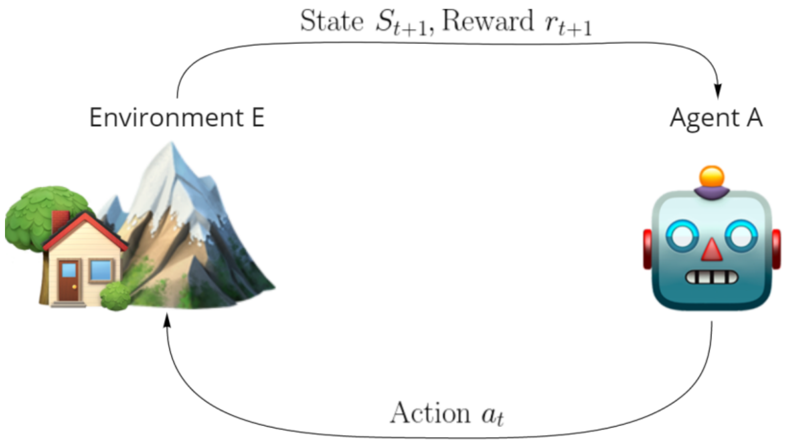
Research Questions

1. Is homeostatic regulation a feasible generic model of motivation in artificial animals based on reinforcement learning?
2. Can such a model be used for replicating basic behaviors observed in some copepod species?

Fig. Panda motivated by maintaining homeostasis (having a full belly)

Theory

- Interaction loop
  - Agent observe (partial) state
  - Performs an action on the environment
  - Environment update state and emits a *reward*



`Fig: Agents interaction loop with its environment.`

Theory

- Agents samples action from a policy
  - The policy captures the agents behaviour
- Wants to maximize its cumulative reward (called the return)
- Central goal is to find policy which maximize expected return

$$a_t \sim \pi(\cdot | s_t)$$

$$R(\tau) = \sum_{t=0}^{T} \gamma^t r_t$$

$$\pi^* = \underset{\pi}{\operatorname{argmax}}\, J(\pi)$$

$$J(\pi) = E_\pi[R_{\gamma=1}(\tau)]$$

Theory - Types of Methods

- Model-Based
  - Have a model of its environment
  - Can plan
  - Unfeasible
- Model-free
  - Have to explore its environment
  - On-policy - Explore with the same policy that is being learned
  - Off-policy - Explore using a second policy different from the policy that is being learned

Theory - Policy Gradient

- Parameterize policy directly $\pi(a|s;\theta)$
- Find parameters that maximize expected return using Gradient Ascent:

$$J(\theta) = E_\theta[R(\tau)]$$

$$\theta_{k+1} = \theta_k + \alpha \nabla_\theta J(\theta_k) = \theta_k + \alpha \nabla_\theta E_{\theta_k}[R(\tau)]$$

Theory - Policy Gradient

- Parameterize policy directly $\pi(a|s;\theta)$
- Find parameters that maximize expected return using $J(\theta) = E_\theta[R(\tau)]$

  Gradient Ascent:

$$\theta_{k+1} = \theta_k + \alpha \nabla_\theta J(\theta_k) = \theta_k + \alpha \nabla_\theta E_{\theta_k}[R(\tau)]$$

- Update not connected to optimal policy
  - Prone to get stuck in local optima

Update only relevant for current policy

Theory - Proximal Policy Optimization

- Builds on Policy Gradient
- Key Idea: Only do sensible updates

Theory - PPO

- Builds on Policy Gradient
- Key Idea: Only do sensible updates

Advantage of action over default behaviour

$$L(\theta_k) = E\left[\frac{\pi_{\theta_k}(a_t|s_t)}{\pi_{\theta_{k-1}}(a_t|s_t)} A(s_t, a_t)\right]$$

Probability ratio of action using current vs previous parameters

Theory - PPO

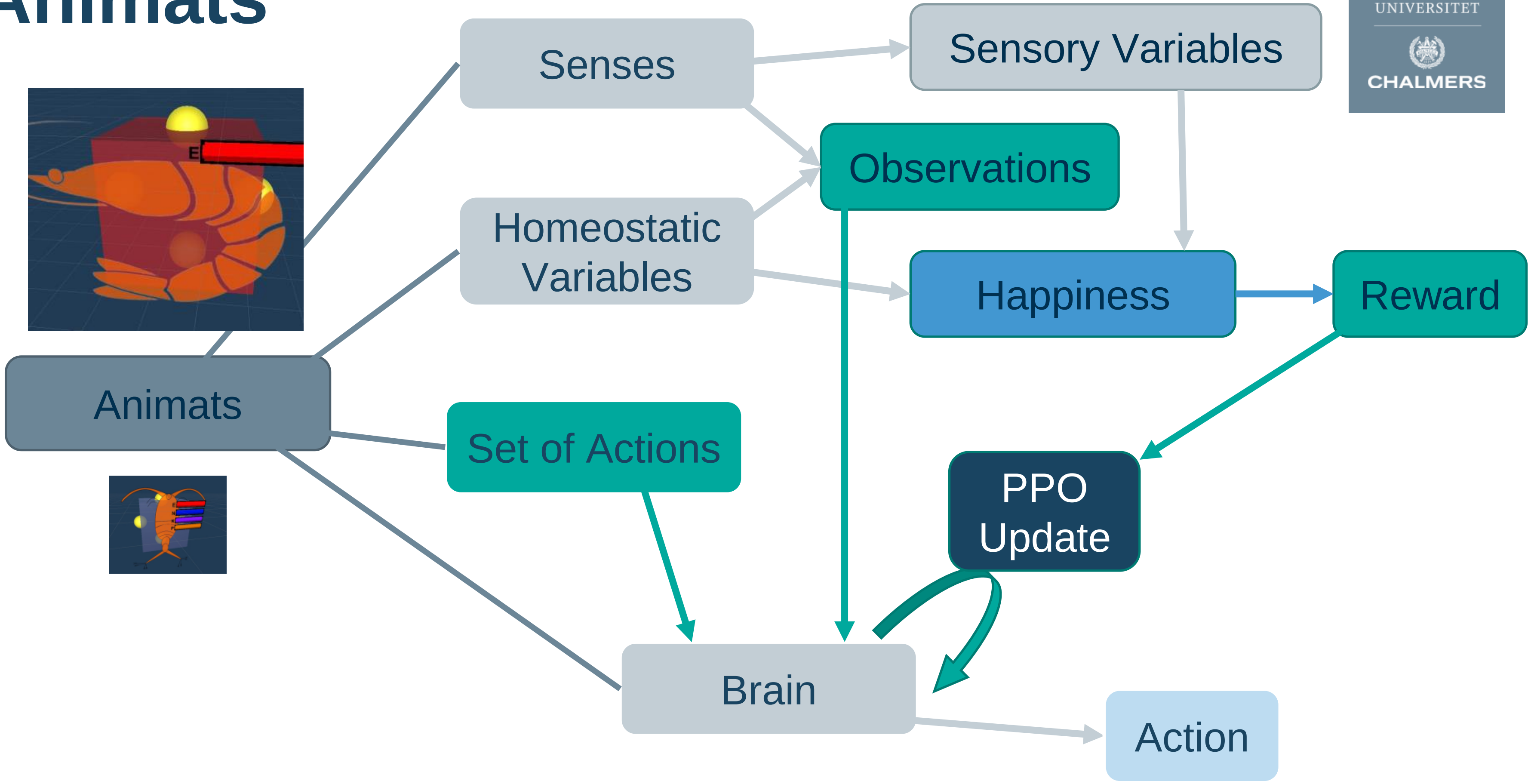
- Builds on Policy Gradient
- Key Idea: Only do sensible updates

$$L^{clip}(\theta_k) = E_t\left[\min\left(r_t(\theta_k)A(s_t,a_t), \underbrace{\text{clip}(r_t(\theta_k), 1-\epsilon, 1+\epsilon)}_{\text{Clip policy updates}}A(s_t,a_t)\right)\right]$$

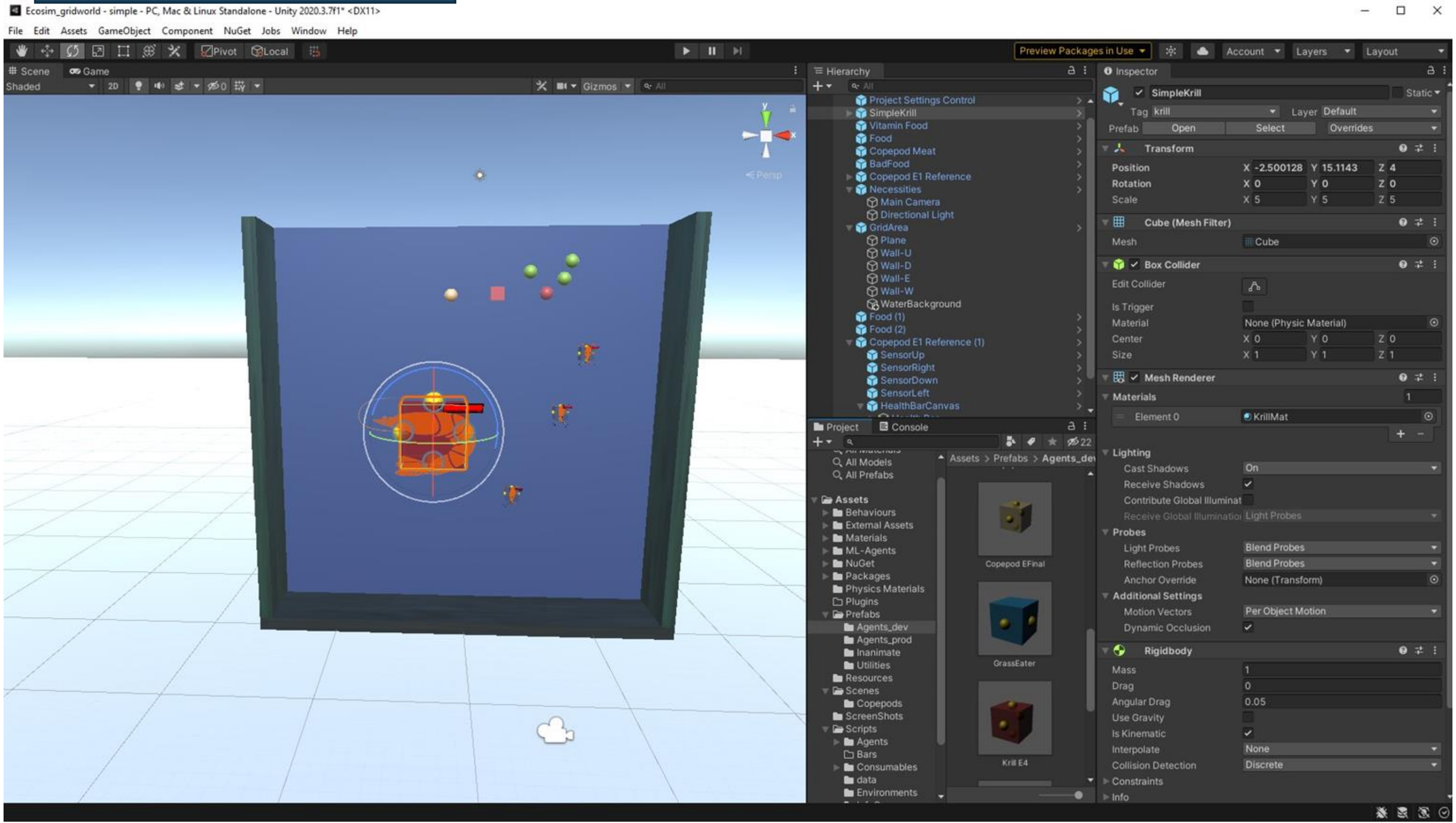
Theory - PPO

- Additionally improvements
  - Break sample correlation using several agents
  - Use efficient (w.r.t. bias-variance tradeoff) estimator of the advantage
  - Entropy regularization
    - Add weighted policy entropy to loss function
    - Facilitates exploration
- Great (State-of-the-art) but not perfect

Animats


Senses
Sensory Variables
Observations
Homeostatic Variables
Happiness
Reward
Animats
Set of Actions
PPO Update
Brain
Action

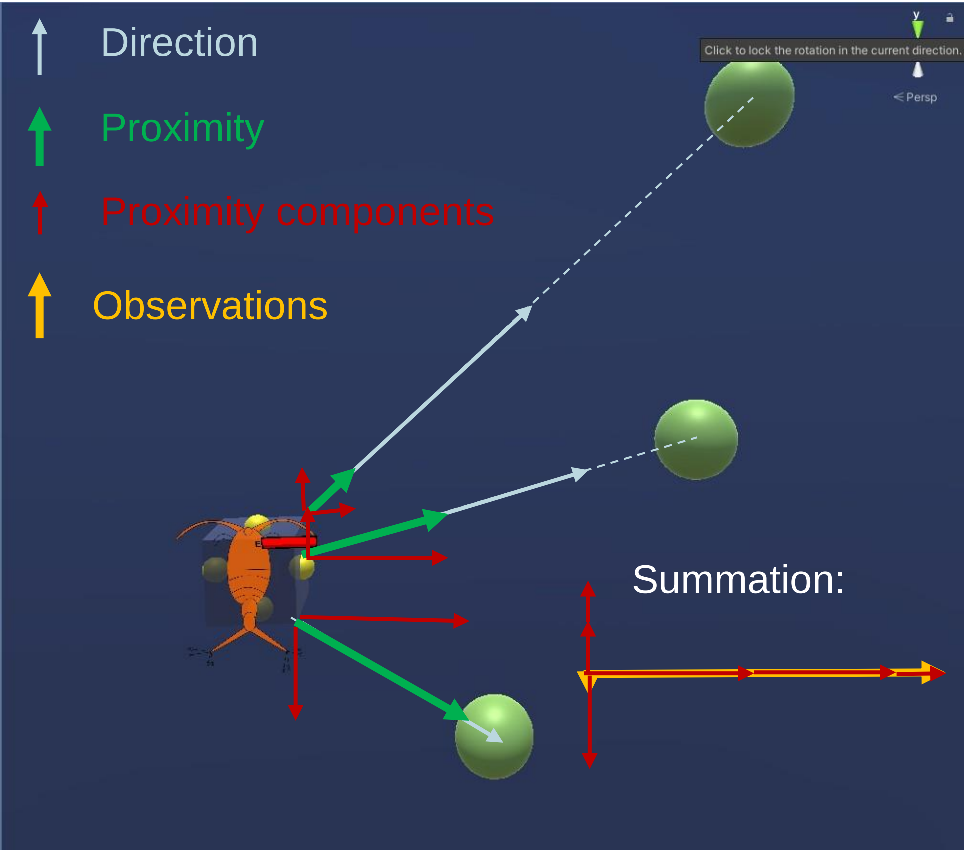
Unity game engine



Animats

Senses

Proto-Vision



$$S^u_{pv} = \sum_{obj \in T} \frac{obj.pos - agent.pos}{|obj.pos - agent.pos|^3} \cdot u;$$

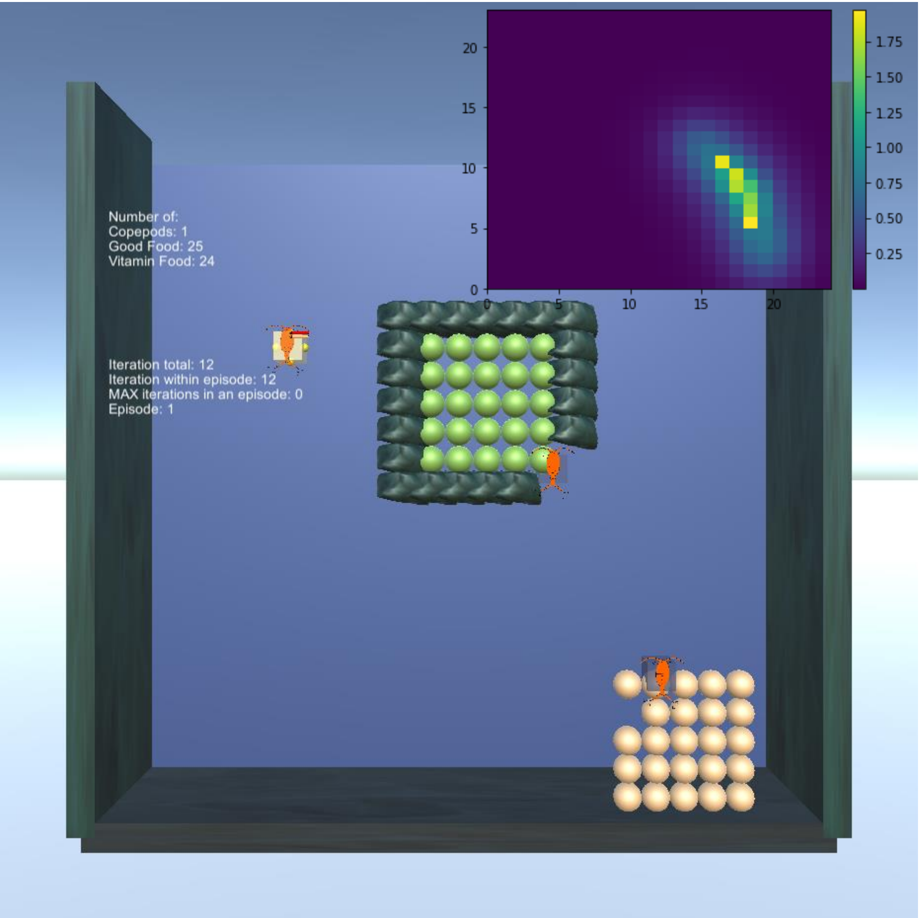
$$S^w_{pv} = \sum_{obj \in T} \frac{obj.pos - agent.pos}{|obj.pos - agent.pos|^3} \cdot w.$$

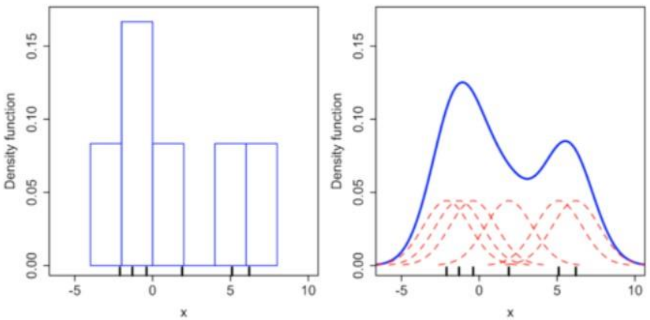
- Yields 2 observations per object type observed
- Some animats have light-sensitive proto-vision

Animats

Senses

Smell





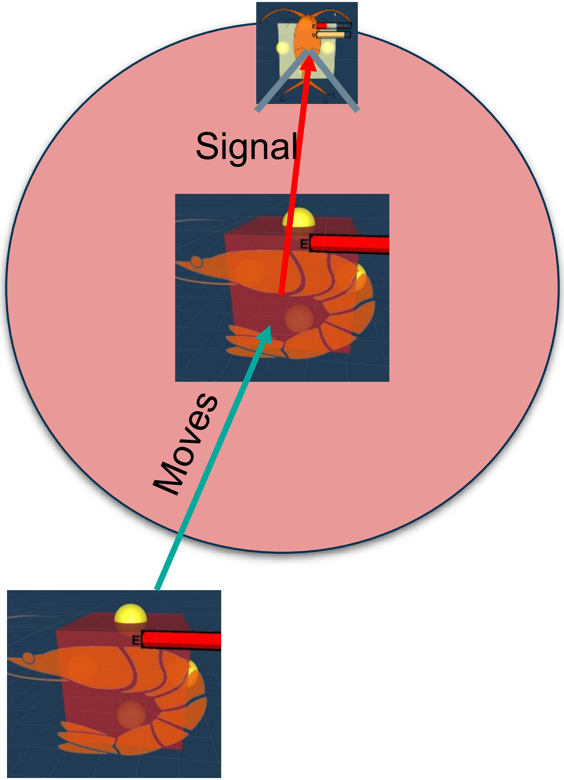
Inspired by Kernel Density Estimation, KDE, but generalized to 2D

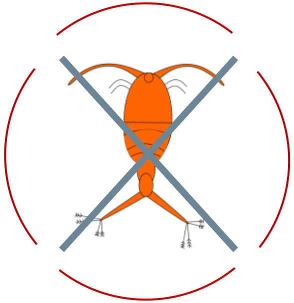
- Smell is released in the cell where the animat is
- Smell is dissipated with a Gaussian Kernel K_g , convoluted over the smell matrix M_s :

$$(K_g * M_s)[m, n]$$

- Smell dampening by a factor of 0.95.

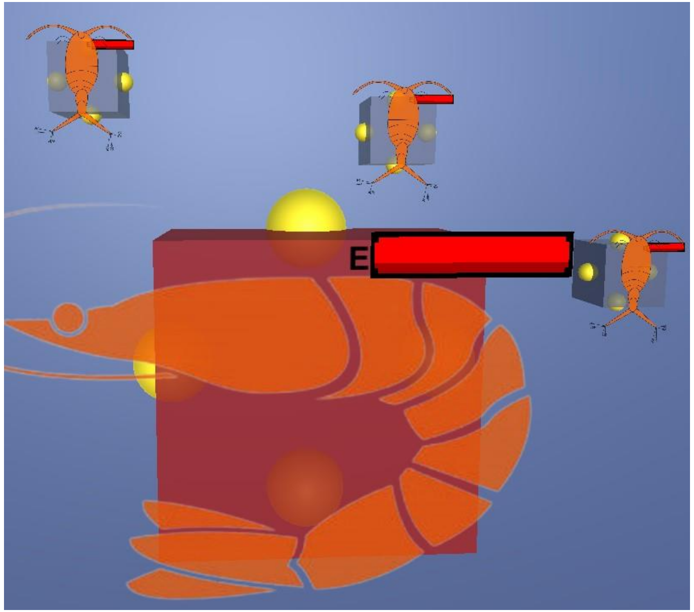
Senses - Fluid Deformation

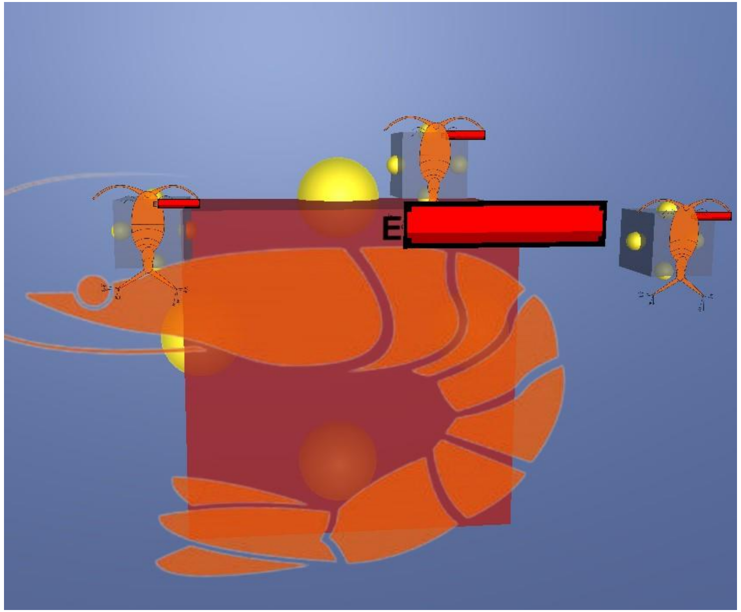




- 1 observation per each 90° range
- Uses Exponential Moving Average

Senses - Touch





Homeostatic & Sensory Variables

$$H = (h_1, \ldots, h_{N_h})$$

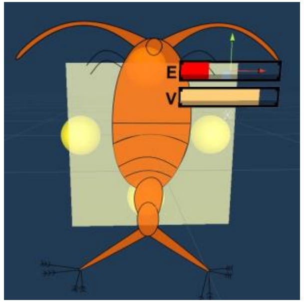
Examples: energy, vitamins, potassium, perceived temperature, libido, …

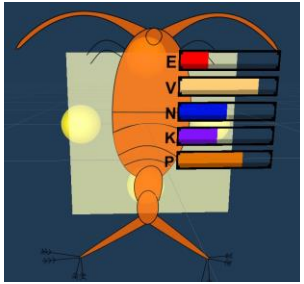
$$S = (s_1, \ldots, s_{N_s})$$

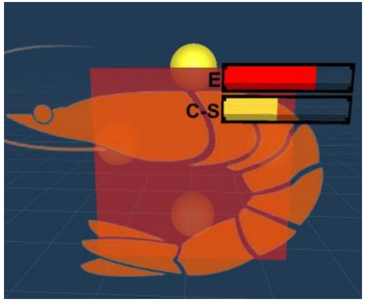
Examples: smell of food, vision of flowers, light intensity, …

$$V = H \cup S$$

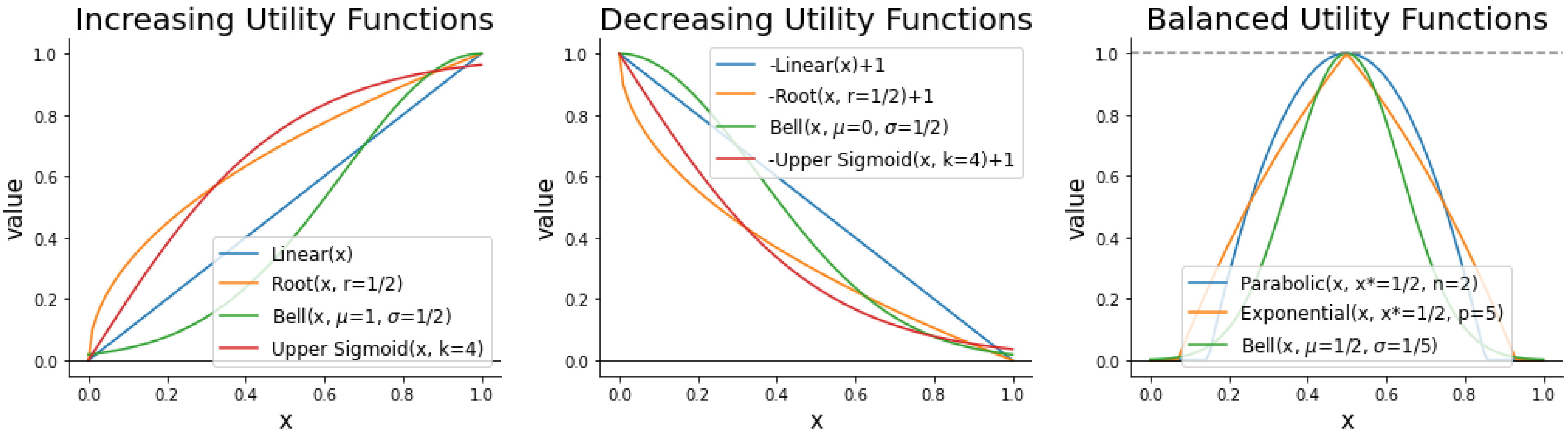
Happiness Variables







Happiness - Utility Functions



Happiness Functions, Reward

GÖTEBORGS UNIVERSITET

CHALMERS

$$f_1(V_t) = happiness_t(V_t) = \sum_{i=1}^{N_v} w_{v_i} u_{v_i}(v_{i,t})$$

$$f_3(V_t) = 1 - \sqrt[m]{\sum_{i=1}^{N_v} w_{v_i} |v_i^* - v_{i,t}|^n}$$

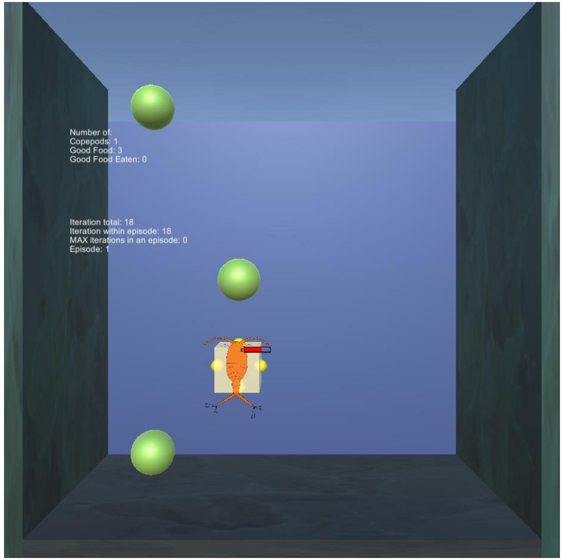
$$happiness_t(V_t) = \prod_{i=1}^{N_v} (a_{v_i} + w_{v_i} u_{v_i}(v_{i,t}))$$

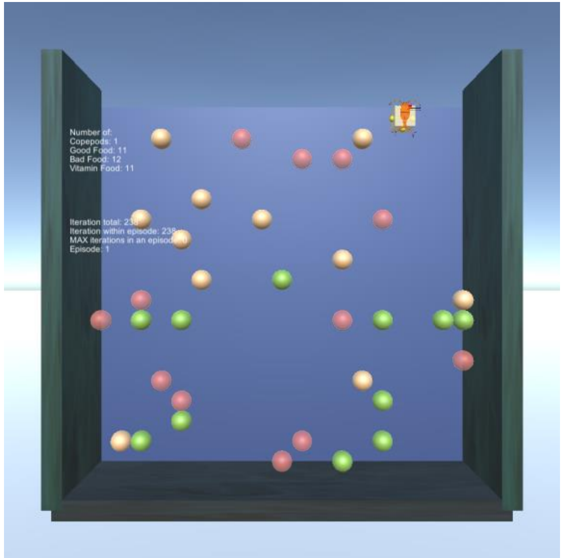
$$f_4(V_t) = 1 - \sqrt[m]{\sum_{i=1}^{N_v} |v_i^* - v_{i,t}|^n}$$

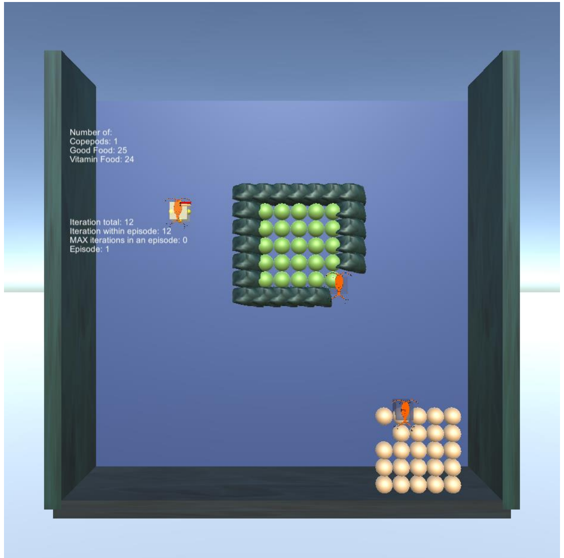
$$f_2(V_t) = happiness_t(V_t) = \prod_{i=1}^{N_h} u_{h_i}(h_{i,t}) * \prod_{i=1}^{N_s} (1 + w_{s_i} u_{s_i}(s_{i,t}))$$

$$\text{reward}_t = happiness_t - happiness_{t-1}$$

Environments & Behaviours





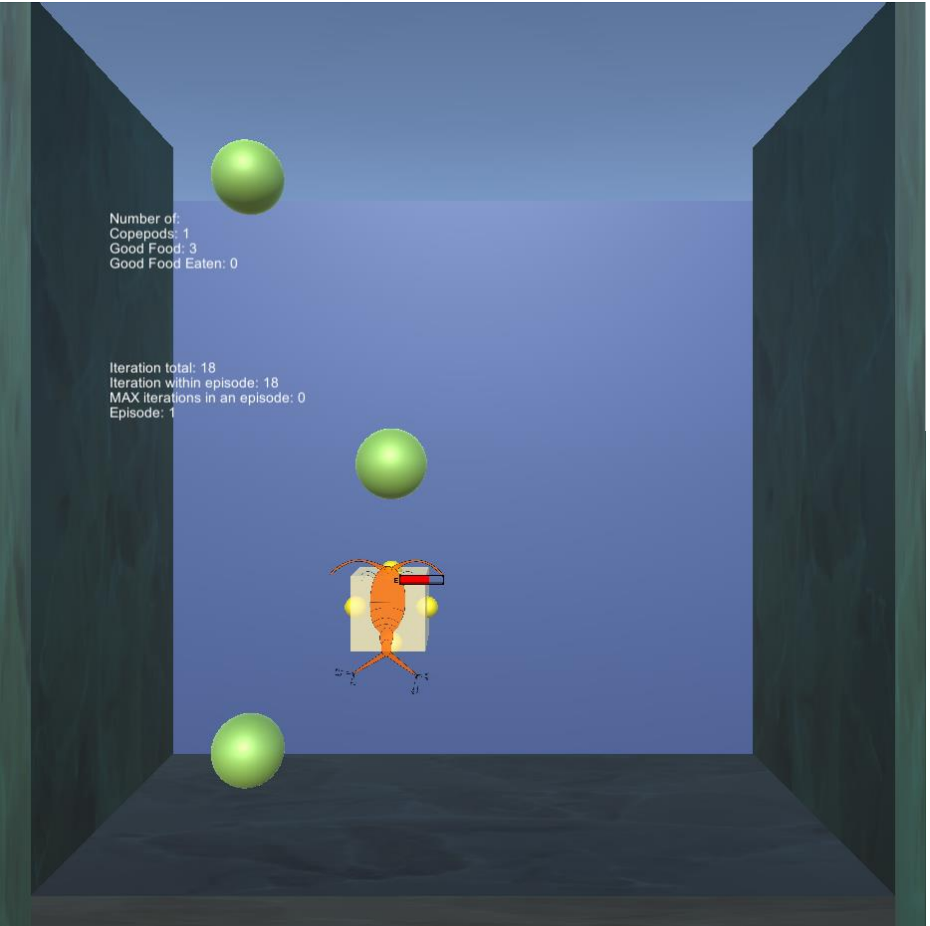


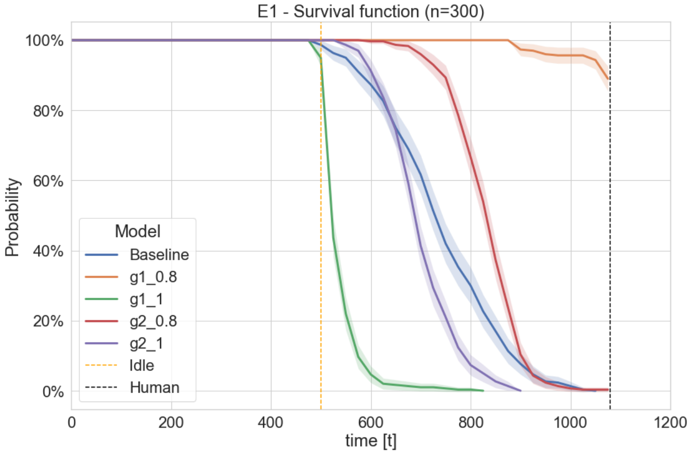
- **Behaviour B1**:
Regulation of hunger when food is scarce

- **Behaviour B2**:
Selective eating by differentiating different types of food

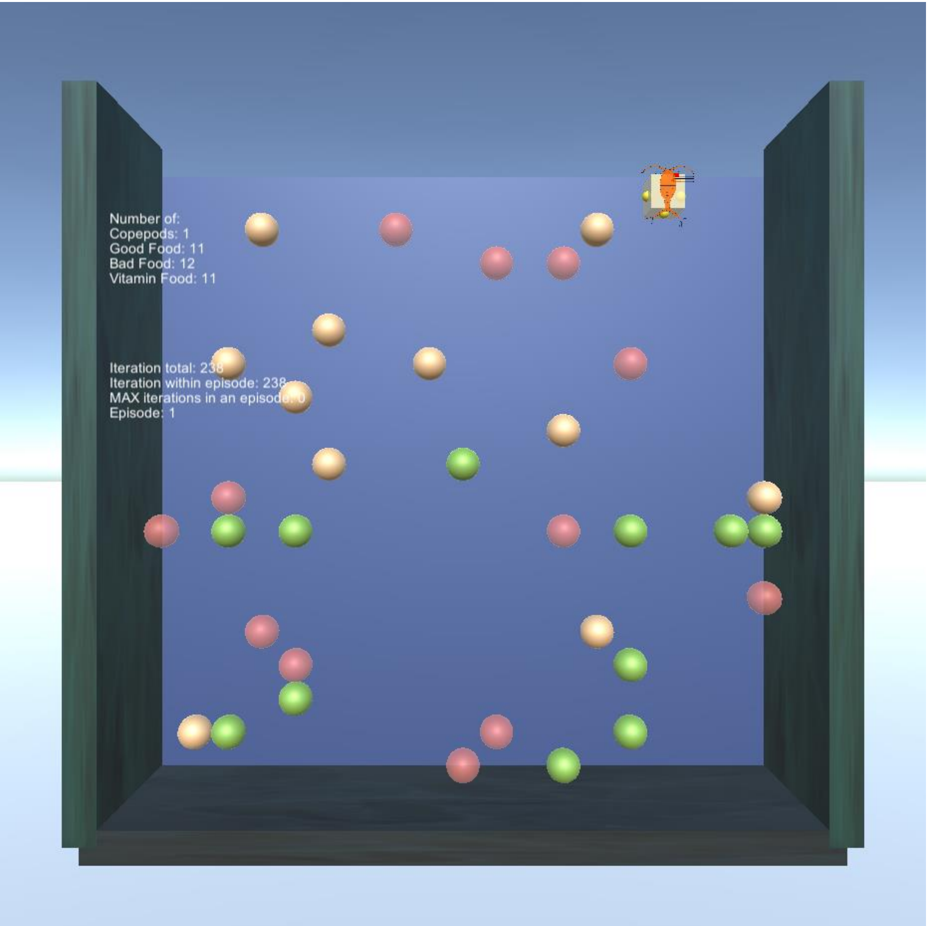
- **Behaviour B3**:
Chemotaxis, by following scent trails

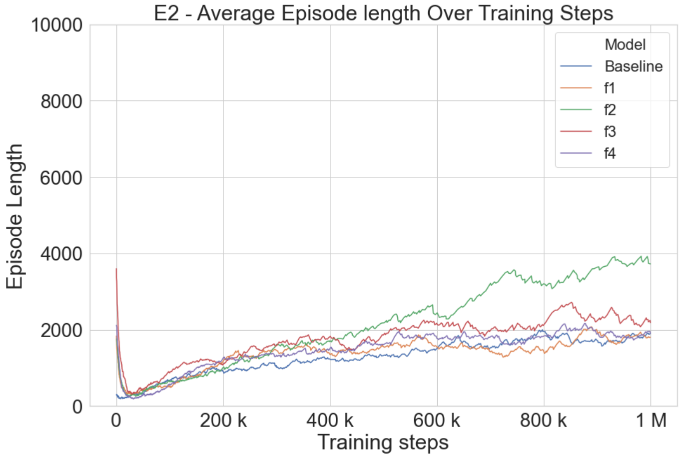
Experiments - E1

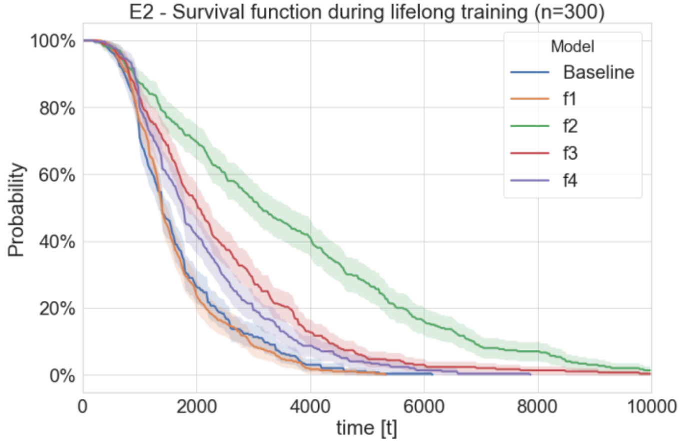




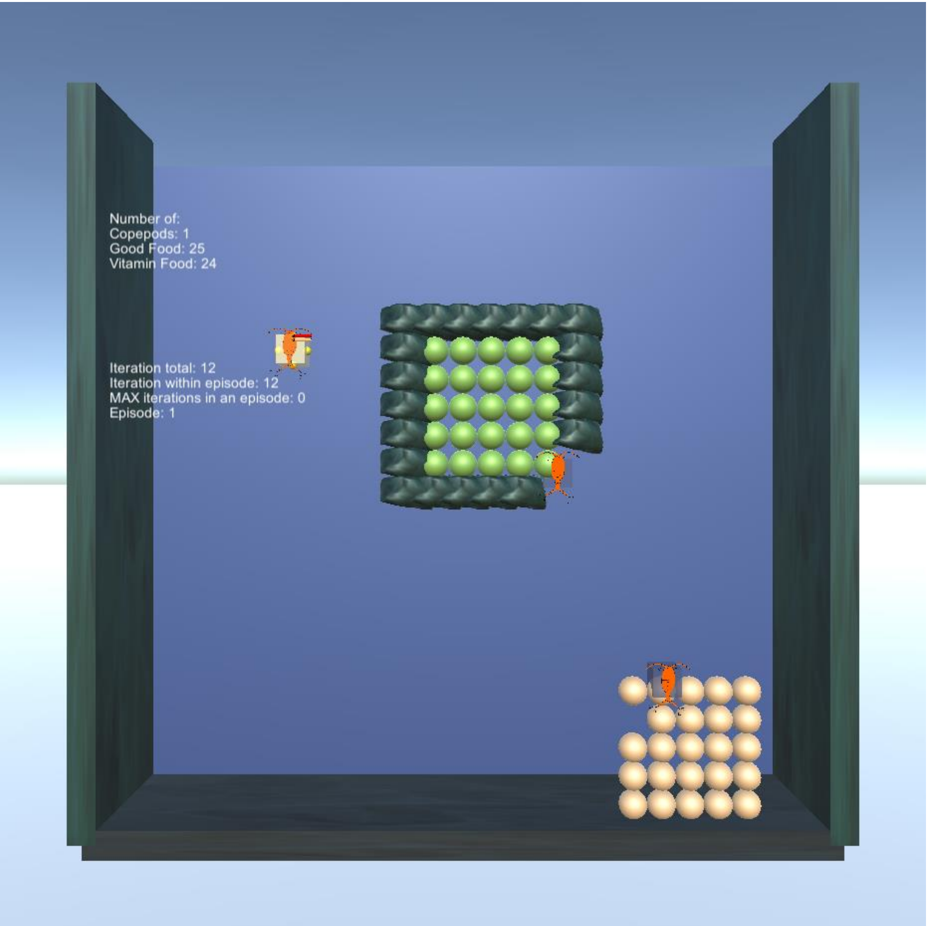
Experiments - E2

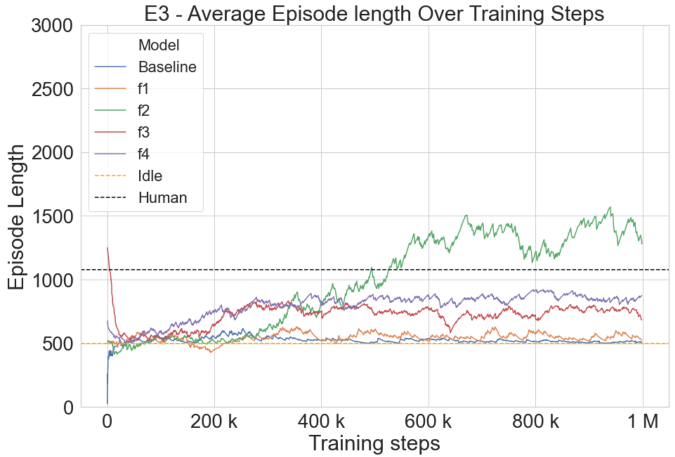


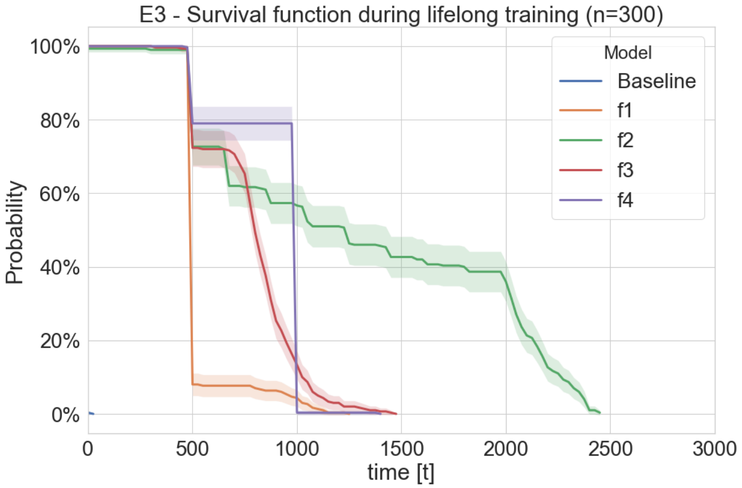




Experiments - E3



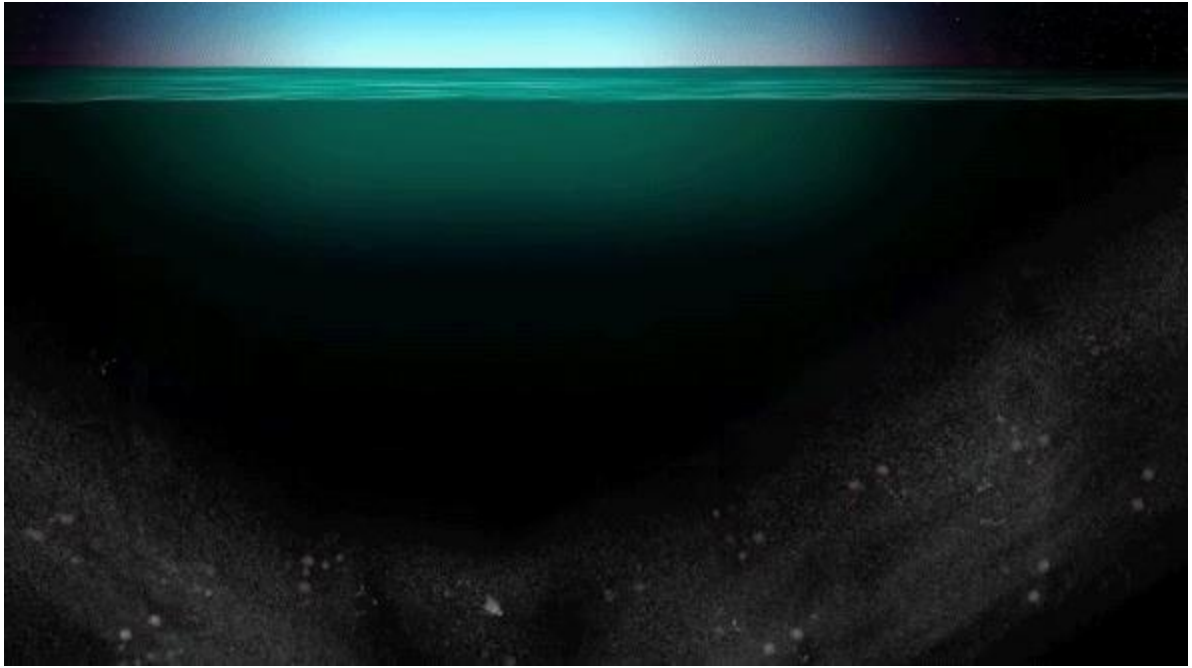




Diel Vertical Migration

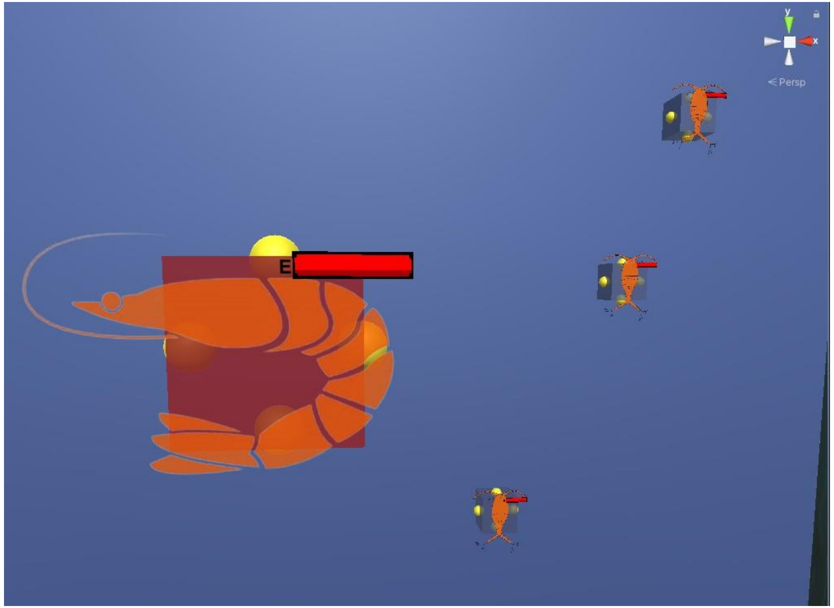
Picture of a copepod

Picture of a krill

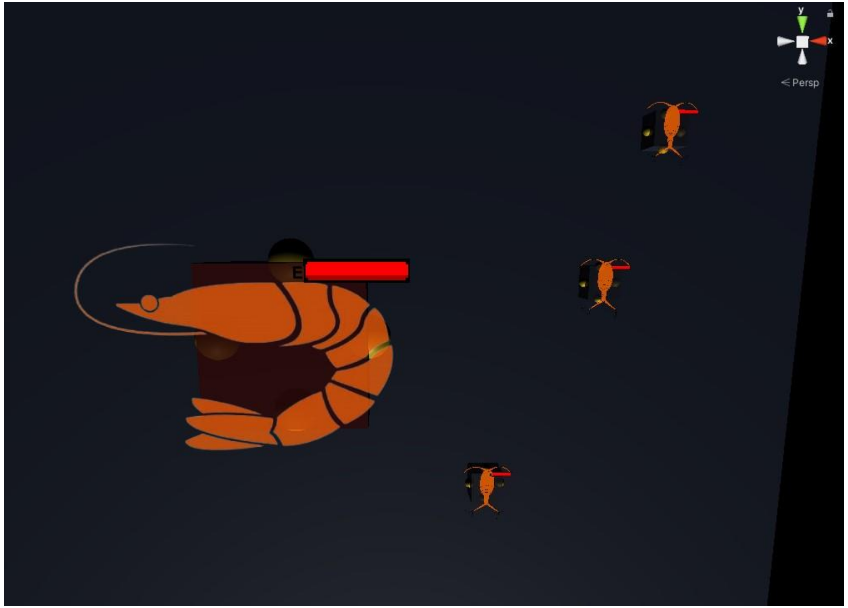


Animation from “This twilight zone is dark, watery, and yes, also full of intrigue”, NASA Blog, 21 August 2018.

Krill predators & Diel Light

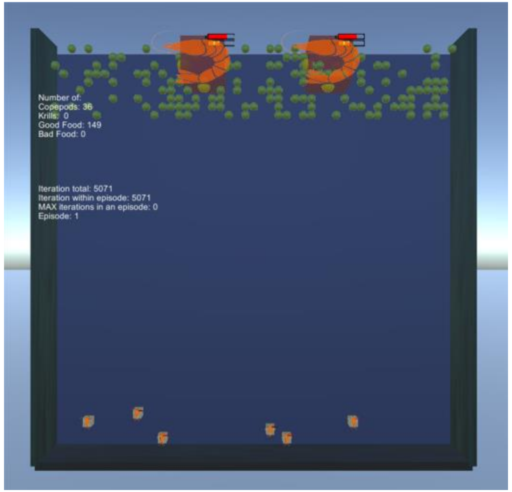


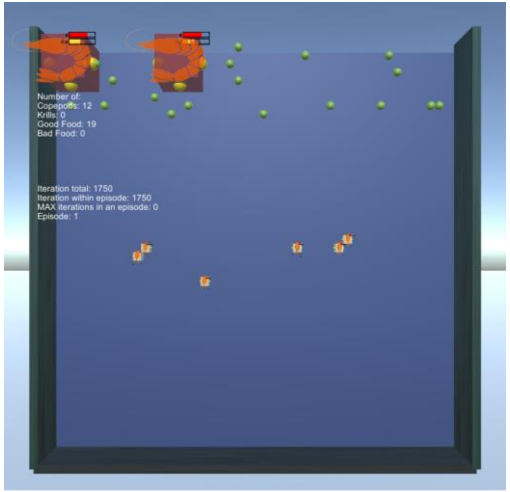
Daylight: krills can see copepods



Dark: krills cannot see copepods

Copepod Environments & Behaviours





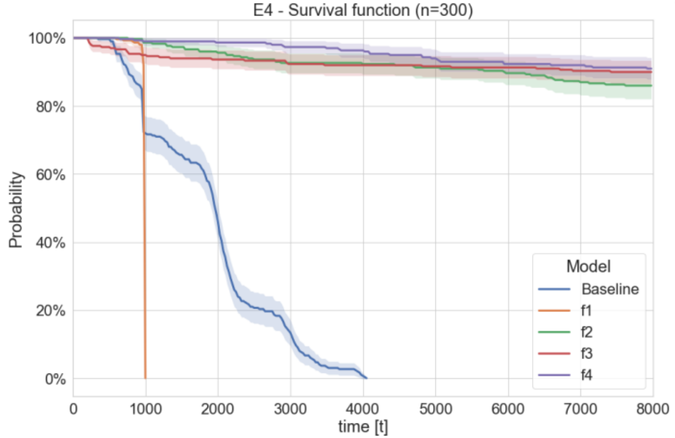
- **Behaviour B4**:
Diel Vertical Migrations (DVM)

- **Behaviour B5**:
Quick escape reactions

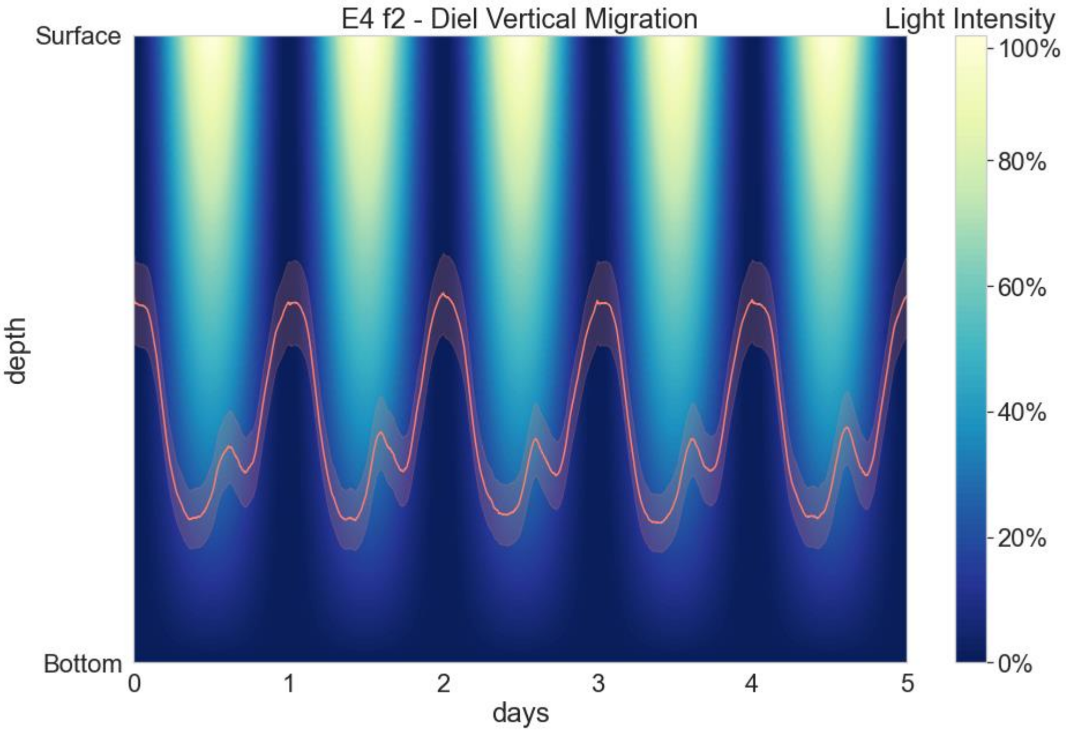
- **Behaviour B6**:
Chemotaxis, escape predators sensed by scent

Results - E4

E4 - Average Episode length Over Training Steps
Baseline
f1
f2
f3
f4
Episode Length
Training steps
0
2000
4000
6000
8000
10000
0
200 k
400 k
600 k
800 k
1 M

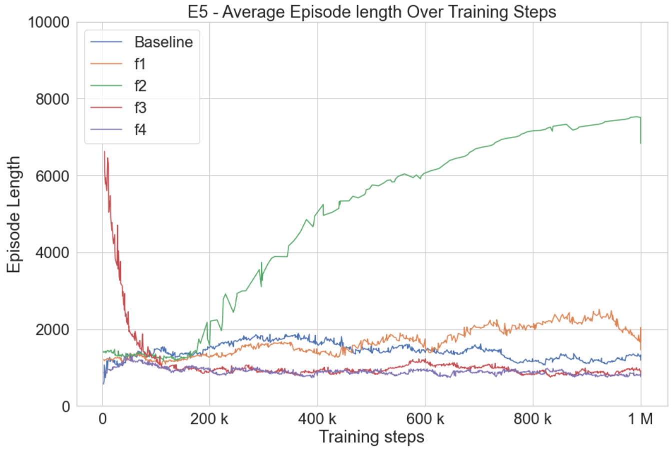


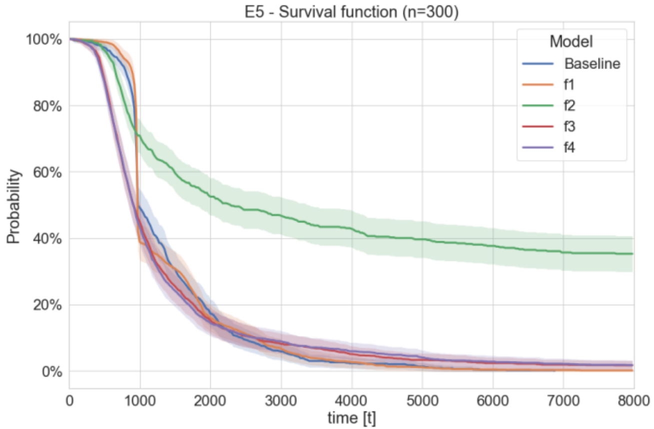
E4 - Survival function (n=300)
Probability
time [t]
100%
80%
60%
40%
20%
0%
0
1000
2000
3000
4000
5000
6000
7000
8000
Model
Baseline
f1
f2
f3
f4

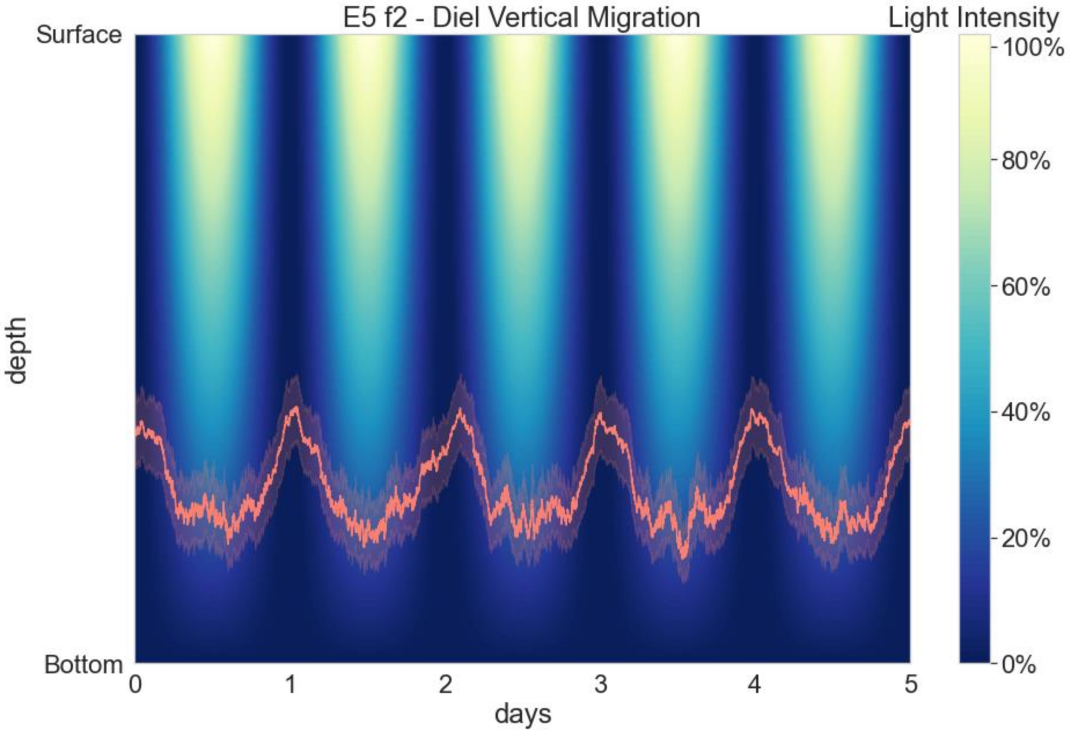


E4 f2 - Diel Vertical Migration
Light Intensity
Surface
Bottom
depth
days
0
1
2
3
4
5
100%
80%
60%
40%
20%
0%

Results - E5



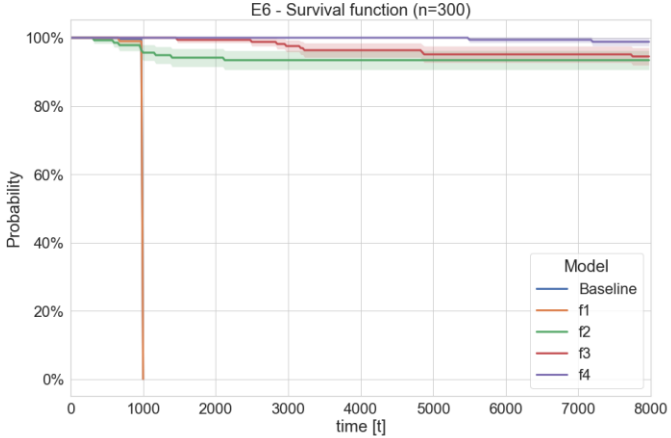


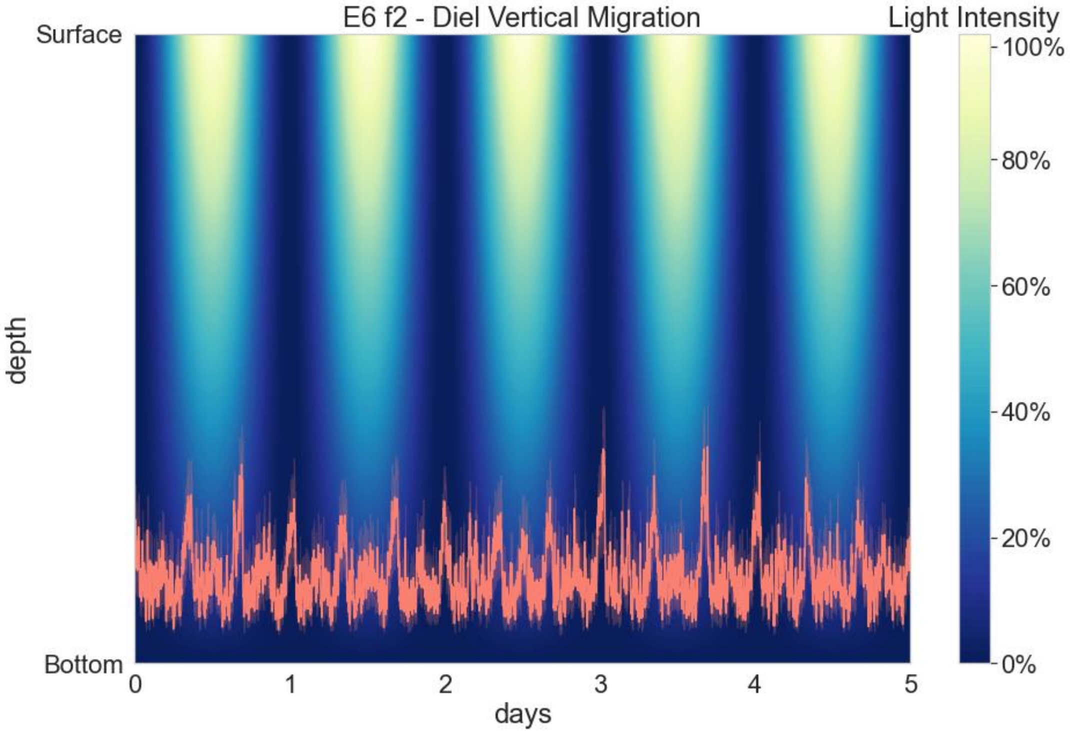


GÖTEBORGS UNIVERSITET
CHALMERS

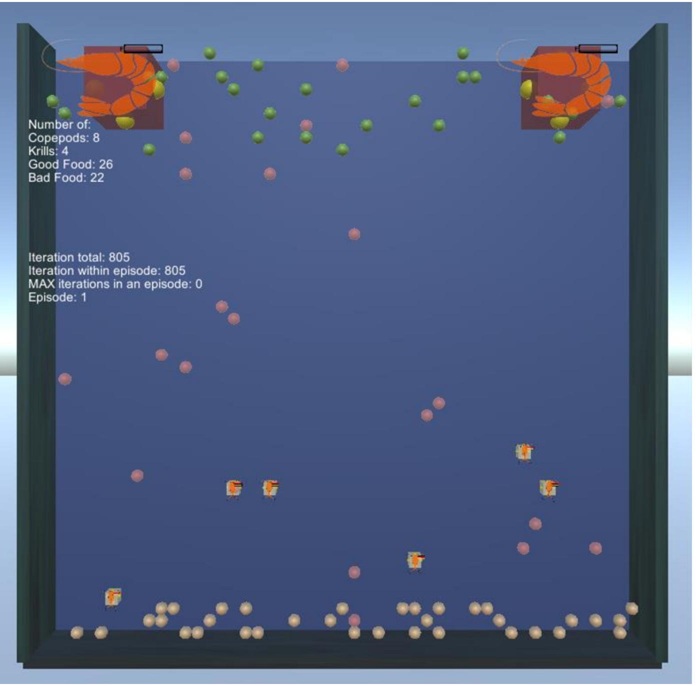
Results - E6


E6 - Average Episode length Over Training Steps
Episode Length
Training steps
10000
8000
6000
4000
2000
0
0
200 k
400 k
600 k
800 k
1 M
Baseline
f1
f2
f3
f4

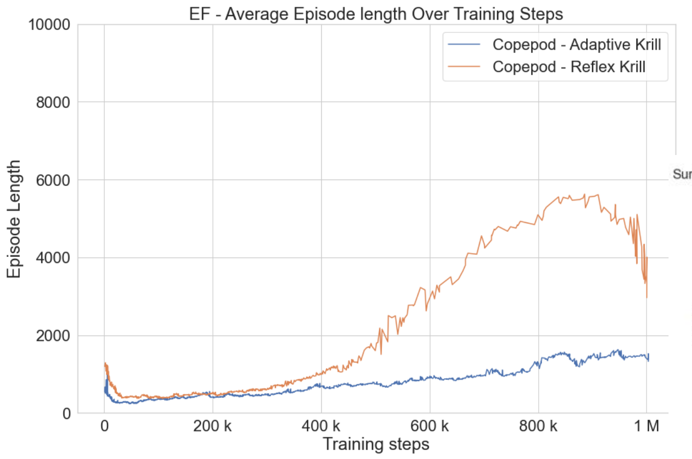

E6 - Survival function (n=300)
Probability
time [t]
100%
80%
60%
40%
20%
0%
0
1000
2000
3000
4000
5000
6000
7000
8000
Model
Baseline
f1
f2
f3
f4


E6 f2 - Diel Vertical Migration
Light Intensity
Surface
depth
Bottom
days
0
1
2
3
4
5
100%
80%
60%
40%
20%
0%

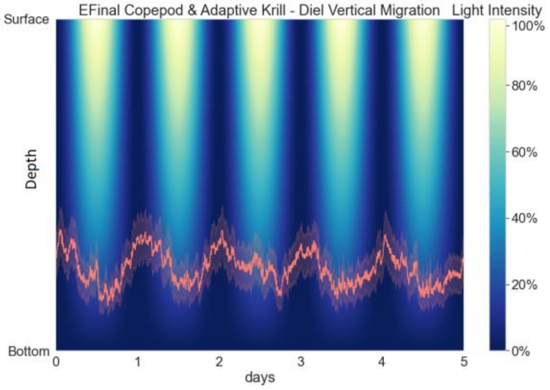
Results - EF



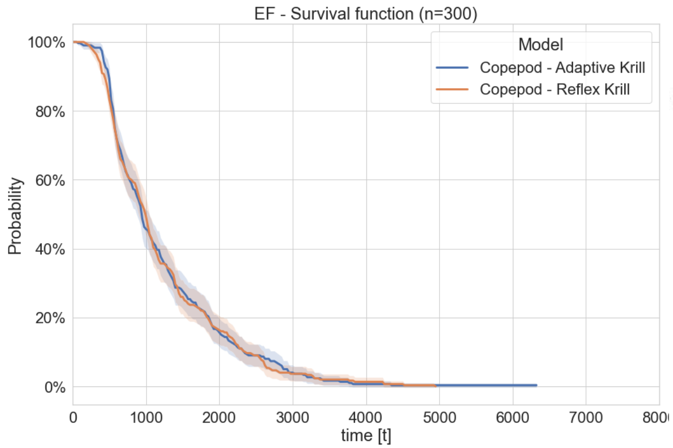
Number of:
Copepods: 8
Krills: 4
Good Food: 26
Bad Food: 22
Iteration total: 805
Iteration within episode: 805
MAX iterations in an episode: 0
Episode: 1



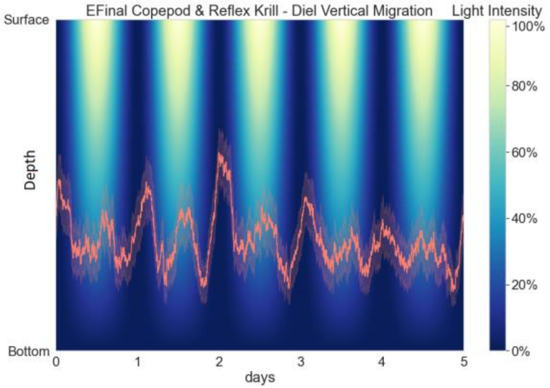
EF - Average Episode length Over Training Steps
Copepod - Adaptive Krill
Copepod - Reflex Krill
Episode Length
Training steps
0
200 k
400 k
600 k
800 k
1 M
10000
8000
6000
4000
2000



EFinal Copepod & Adaptive Krill - Diel Vertical Migration
Light Intensity
Surface
Bottom
Depth
days
100%
80%
60%
40%
20%
0%



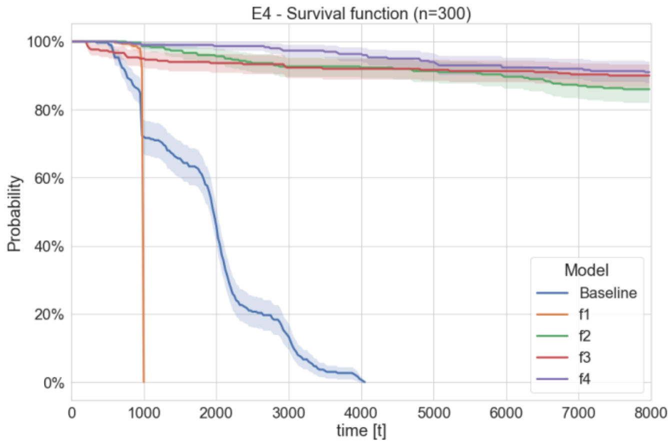
EF - Survival function (n=300)
Model
Copepod - Adaptive Krill
Copepod - Reflex Krill
Probability
time [t]
100%
80%
60%
40%
20%
0%
0
1000
2000
3000
4000
5000
6000
7000



EFinal Copepod & Reflex Krill - Diel Vertical Migration
Light Intensity
Surface
Bottom
Depth
days
100%
80%
60%
40%
20%
0%

Discussion

- Main success criteria is survival time
- Baseline model is a sensible choice
  - Reward directly connected to success criteria
- Performs ok-ish
- Requires exploration
  - Increases vulnerability for collapsing into local optimas



GÖTEBORGS UNIVERSITET
CHALMERS

Discussion

- Keramati et al. model suffer zero cumulative reward on closed loop
  - Return is constant and negative if the animat dies
  - Return is not tied to survival
  - Can break this by surviving (as in E4-E6 & EF)

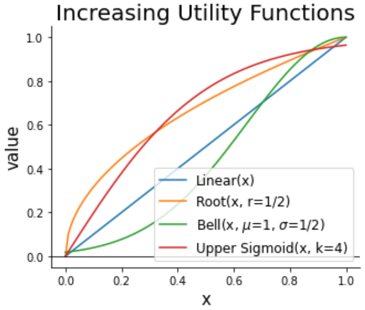
Discussion

- Keramati et al. model can not ignore *irrelevant* drives
- They claim that their model has this property
  - Instead it can ignore drives with low distance to its setpoint
  - In their model all variables are equally important have limited expressability in their interactions
- Our proposed model do have this property
  - Vital variables scale non-vital variables impact

Discussion

- Keramati et al. allow for:
  - Defining optimal state
  - selecting m, n
- This makes it quite hard to model homeostatic regulation
- Our framework instead allows for modelling each marginal utility at a time
- And model the interactions among variables

$$d(H_t) = \sqrt[m]{\sum_{i=1}^{N} |h_i^* - h_{i,t}|^n}$$



GÖTEBORGS UNIVERSITET
CHALMERS

Discussion

- On-policy exploring without sample reuse requires policy and value function approximators to capture all experience
    - This means that previous exploring can be forgotten
- Using our framework it is possible to trade bias towards strong reward signals to mitigate need for exploration
    - This is useful in ecosystems where the biases are known and desired (instincts)

Conclusion

- Our framework allows for
  - flexible interaction among variables
  - use of critical and non-critical variables
  - conditionally ignore non-vital variables
  - higher application of intuition as marginal utilities and interactions can be modelled separately
- Simply trying to maintain homeostasis is sufficient to build a general model of motivation, elicit each behaviour and in particular build a model of copepods

GÖTEBORGS UNIVERSITET

CHALMERS