Amateur agent cat

Zenodo	:	zenodo.org/records/13626537
Github (early depictions!)	:	https://github.com/djuwidjaandrew/concat/tree/main/prop

Hi , im andrew

im working on this agent cat paper

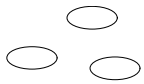
with mr Daniel and in this framework

we try to define the cat by more of surface level movements

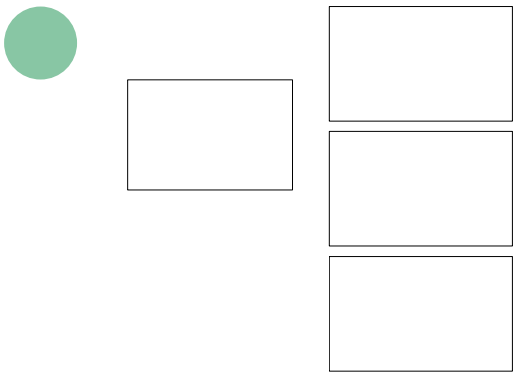
that we observe : the timings the reaction

and try to model it onto a framework

overall it sortof looked like this



cat . object



vision to transpose

screen for (linger)

module for (output)

chem. for (timing)

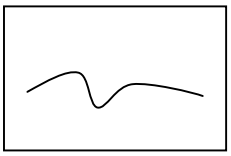
(ontop of other features such as queue and pause etc)

a 3 part type of process

screens to represent what it conscious about

modules to produce specific output

chempool to define timings and inputs



Free Energy Implementation

so thats our cat framework

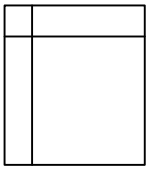
and we were thinking about how to apply

the machine learning part onto this

and read mr andrewpashea's (colleague) tutorial

about mblr and hierarchal and kalman filtering

which each is about:



Link

https://reactivebayes.github.io/RxInfer.jl/stable/examples/basic_examples/Bayesian%20Linear%20Regression%20Tutorial/#Part-2.-Hierarchical-Bayesian-Linear-Regression

>

MULTIVARIATE BAYESIAN LINEAR REGRESSION

>refining predictions over a criteria

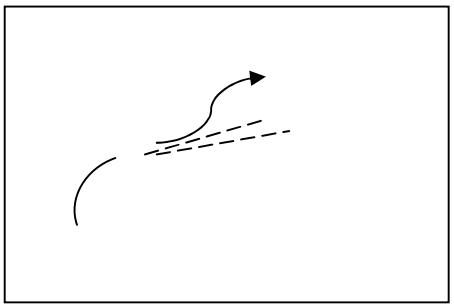
and trying to get the guesswork to

fit more onto the actual (observed)

HIERARCHAL (similar implementation but with Laplace Mean for parameters of different scale)

KALMAN FILTERING

(fitting predicted trajectories and missing input datas and refining free energy aswell)



Minimizing free energy whilst continuing

Nearest timestep trajectory prediction

>or in case of kalman

its based on trajectories and what trajectory

could be; and what we guess (before then accounting the speed)

(despite missing blips of information)

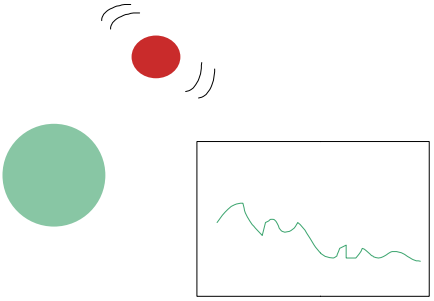
so then we realize that

upon trying to guess the parameters of the object (rat) that the cat observes

this would be such a fit 

initially (simplified sample):

!!



speed matching with recorded obj concept of “rat” / nearest

cat trying to get his free energy (prediction discrepancy with observation) to stabilize

Because

As a hedging (functions and formulas that are hardcoded to epigenetic)

Initial guesswork on the maximum threat of speed;

Along with the panic^^^ token (a representation of alarming object appearing)

In the chempool triggerspace

-

To gradually become panic^^ and panic^ to panic~ (downward)

we implemented this onto the

modWatch and modLook

(and modTude for object feature determination later)

and other budgeting /

Prediction purposes

(we realized this also is a useful tool (perfect fit mayhaps))

For the purposes of budgeting resources for realtime updating of these guessworks

(danger level; speed; trajectory; etc; half hardcoded)

The toolkit (RxInfer & Free Energy determination)

Was such a good fit for

Biologically inspired builds

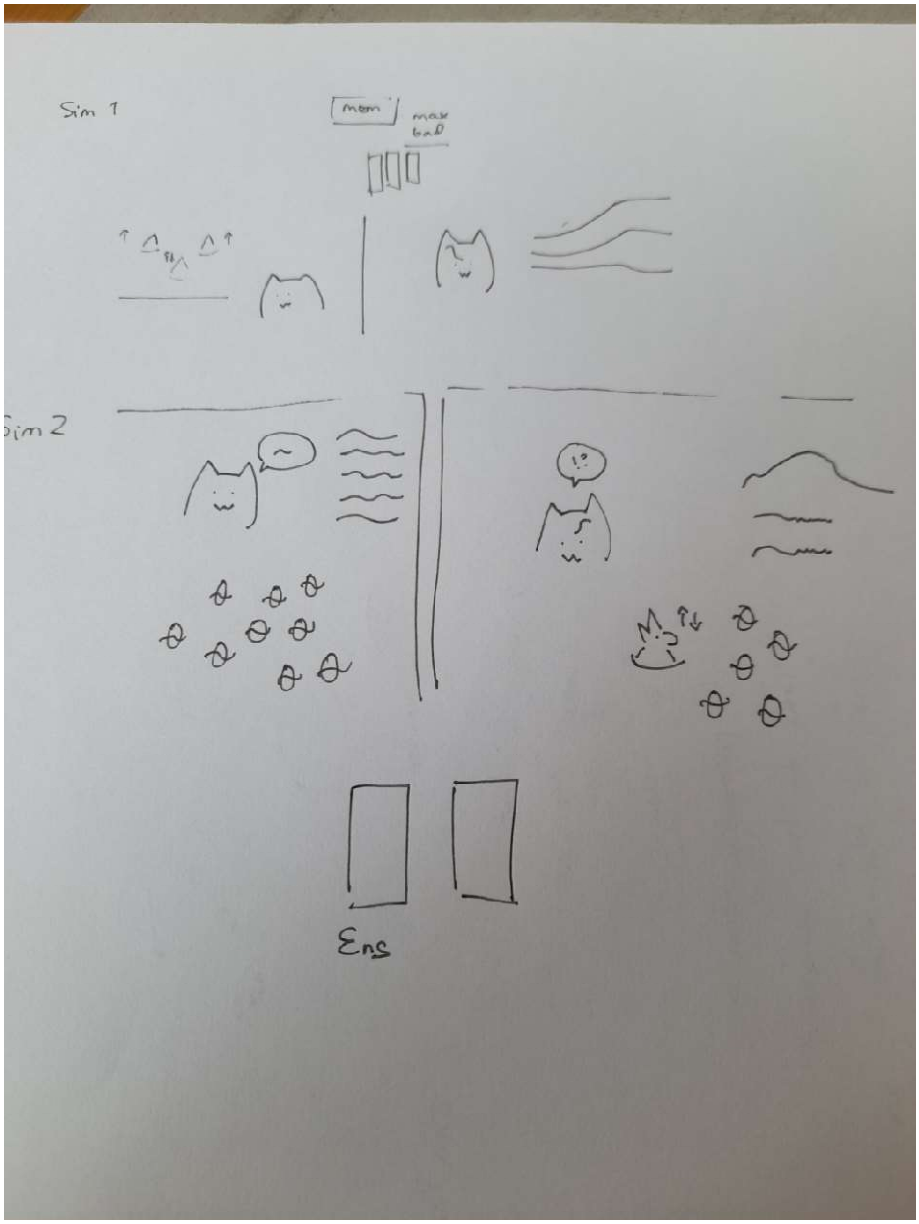
(hoorah concept)

Making a Sim Sample

letssay in this case we make a

simple sim of a cat watching a whackamouse

what we try



sim1

stationary agent cat

spots whacakamouse

and gets surprised (!!!)

has to process the

panic^^^ token

and stabilize free energy

(discrepancy of speed

prediction

with actual observed)

sim2

same thing but

start from at-ease

(modwatch budgets 5 at-ease stabilized MBLR)

To then Surprised!! On which 3 of the 5 budgeted

Becomes taken over by the (MBLR as with sim1)

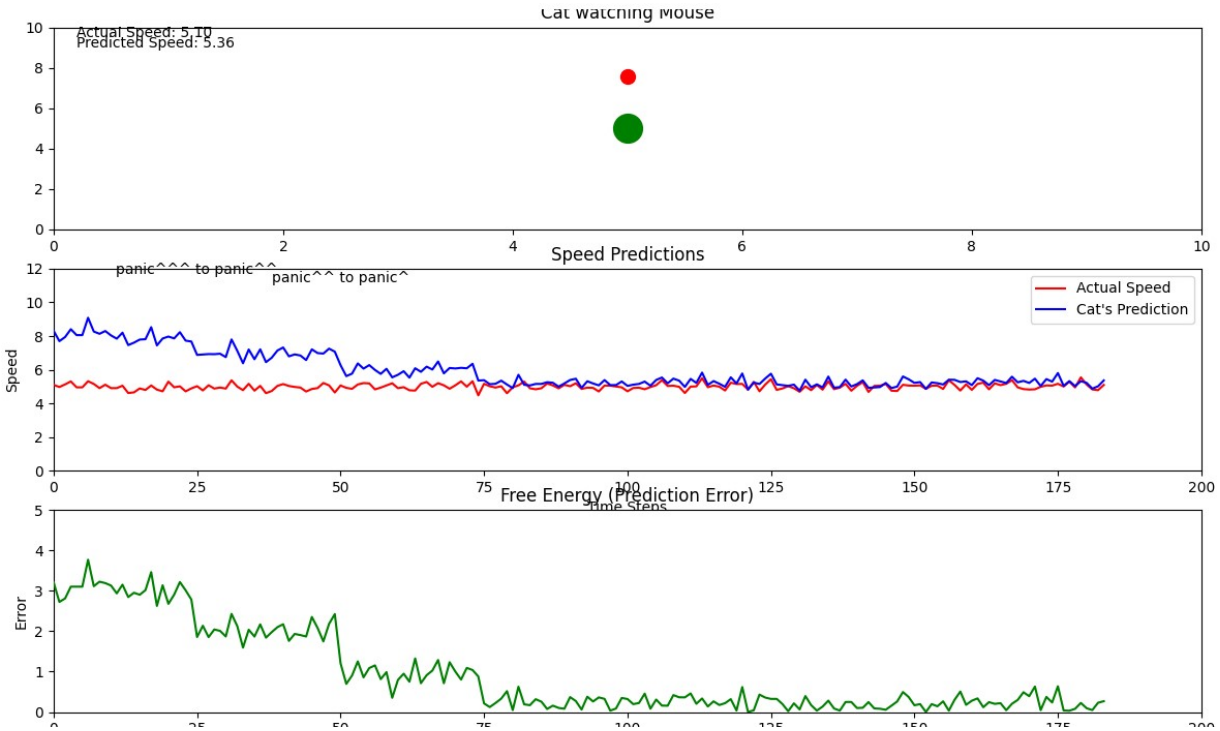
Onto finally stabilizing that one too; back onto the (relative) homeostasis at ease

(back to relative 5 at-ease stabilized object kept-track of)

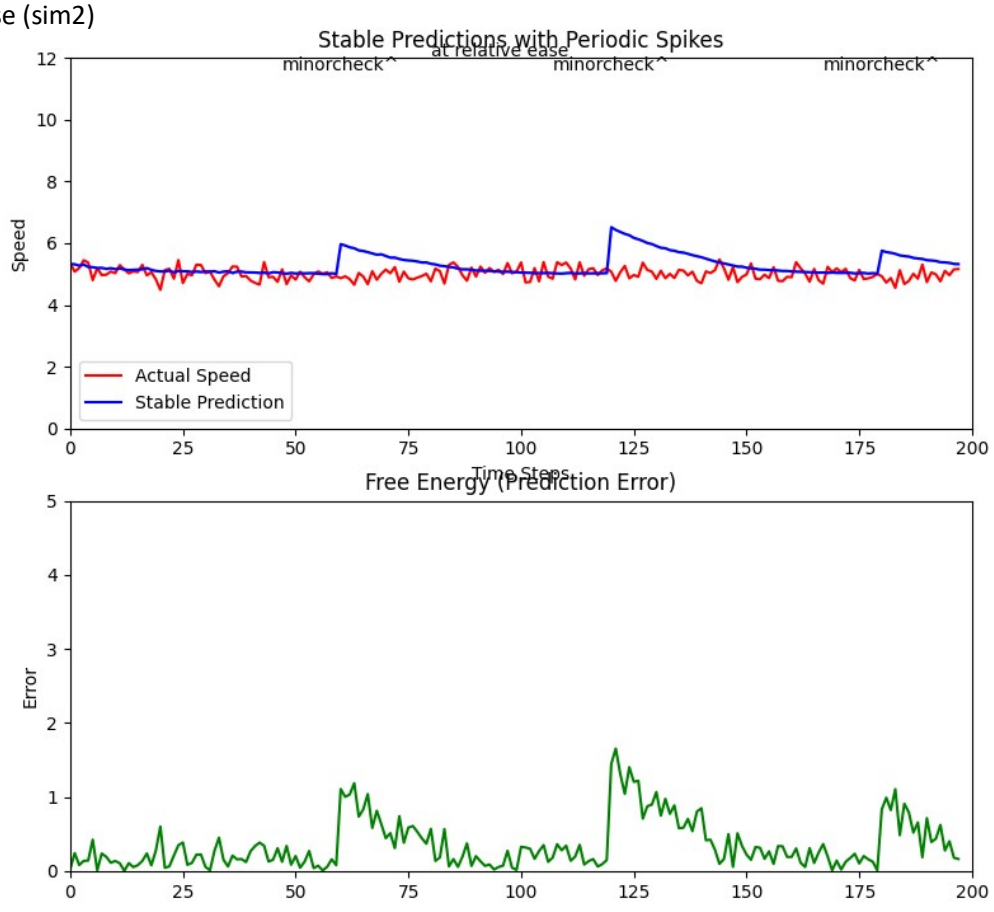
We also try to add a (Backbone framework for the overall agent)

Governed by Ensemble (as with the Zenodo Paper)

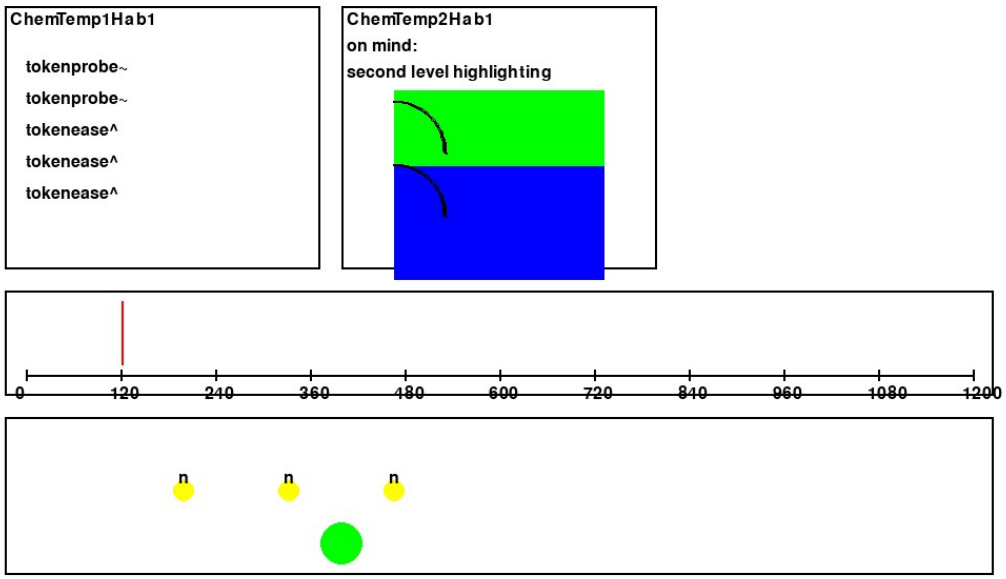
Surprised (sim1)



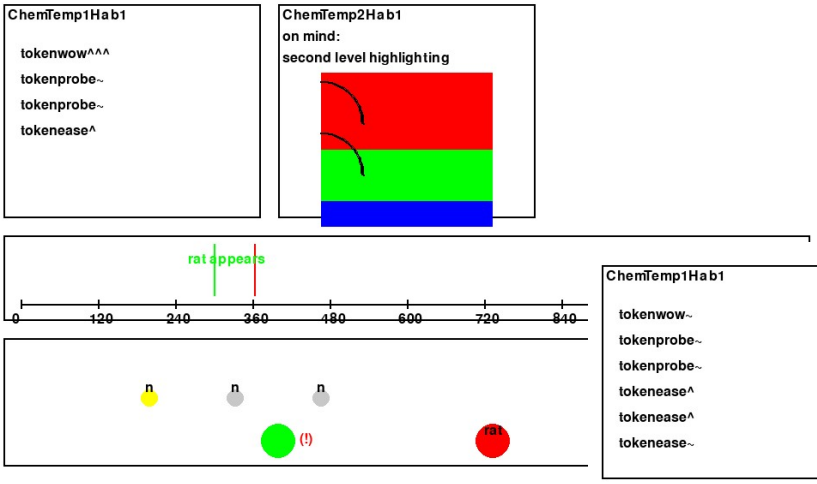
at ease (sim2)



(sim 2 : atease -> panic -> atease (again))

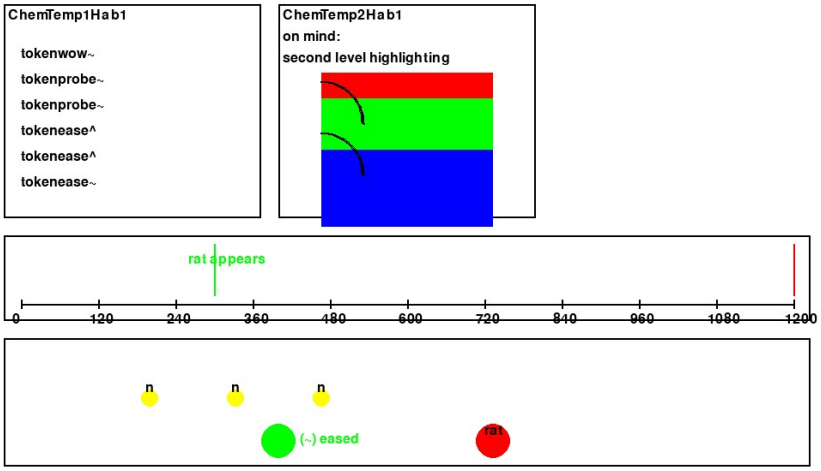


at ease (initial)



rat appears (t)step 340

red(rat)



(t)step 1300

Tho; this comes with major simplification

(as in it hasn't been plugged onto a vision-framework)

(we plan to try ROS + OpenCV)

On which a lot of MBLR would have to be deployed

By then (also for budgeting; etc)

panic^^ to panic^ to ease

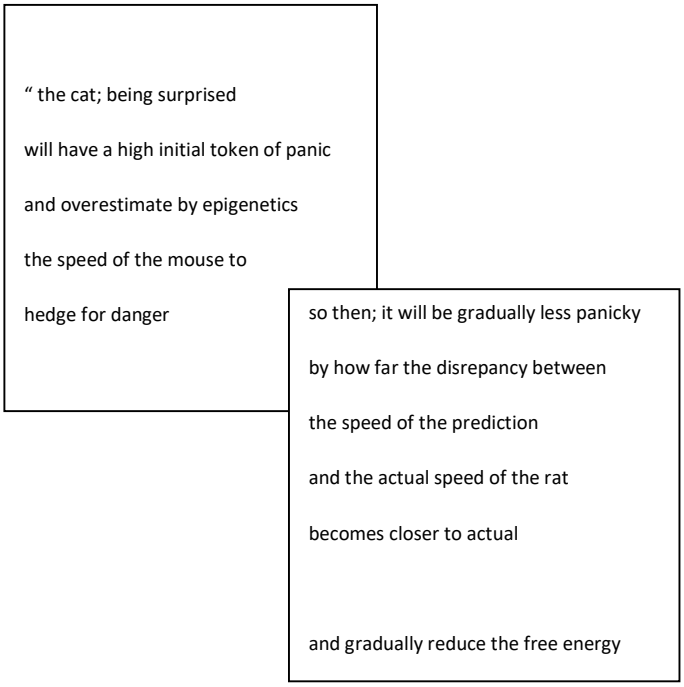
we also made the version where

the cat starts from at ease

to surprised

to at ease again

(sim2)



Python depictions

;;

here is the link if you'd like to see more

(github pictograms)

:

: prop\s1 .py s2.py n stabilized sample.py

: prop\build2\chemsql.py (needs the data\.db to operate)

For this depiction; we lagged; thus we used a pictogram method

(to replicate Multivariate Bayesian Linear Regression as tutorial)

But it was based upon same theory and formulas

To then apply (Hierarchal BLR and Kalman Filtering)

For other parameters

;; meanwhile
on the potential safety application

the idea is that eventually we might be able to

ponder safety for these animal/agent type models

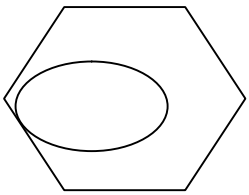
by inquiring each of their modules

and the range that it can influence

the behavior and

thus having an idea on the span

Pic Agent + Span- by modules (conveniently designed to be output-oriented)



Whilst

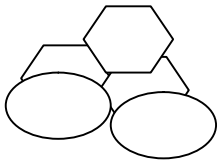
For Multiagent AI Safety

We figure;

If we polish this and this might fit a proper Agent builds in the future

Perhaps using the same

Free Energy overall perturbations (to represent Allostasis / Homeostasis disruption)



Many clashes!

In the agents; if they are Active Inference Based / Free Energy Based

-

We might be able to come up with some sort of discerning parameter

To predict collateral failure / combinatorial failure between circumstances

Mean sustained Free Energy? (translated onto chemtokens; representing Stress in Society)

That narrows the agent's typical safety protocol or just promote certain unwanted behavior

(This is for later upon some level of our build)

(thus opens the avenue to test for direct-society-to-agent parameters such as Stress)

(by proxy of Free Energy)

;;

currently we are trying to plug this onto

ROS and openCV for the robotics and vision part

and seems to be abit limited due to

the LAG and heavy load due to uhhh

primitive laptop;

>>

overall RxInfer and the tutorial

(free energy minimization approach)

becomes the main way that we approach

this

Using RxInfer & or Free Energy Minimalization Approach to

(Multivariate Bayesian Linear Regression)

(Hierarchal Bayesian Linear Regression)

(Kalman Filtering)