Table of Contents

[Can our models differentiate the data generating process? 2](#_Toc161755283)

[How well are the models recovering their original parameters? 4](#_Toc161755284)

[Parameter Recovery: aDDM 5](#_Toc161755285)

[Parameter Recovery: AddDDM 10](#_Toc161755286)

[Parameter Recovery: RaDDM 15](#_Toc161755287)

[How should we report this information in the paper? 20](#_Toc161755288)

# **Can our models differentiate the data generating process?**

What are we doing?

We are considering 3 potential models: the aDDM, AddDDM, and RaDDM. The aDDM has 4 free parameters: drift, noise, multiplicative attentional bias, and starting point bias. The AddDDM replaces multiplicative attentional bias with an additive attentional bias. The RaDDM uses multiplicative attentional bias, but transforms raw value signals into their distance from the minimum possible value in a given context (here, “context” means “an experimental condition”).

For each model and condition, we generate 20 simulated datasets using stimuli and fixation data from the numeric study (study 2). We then fit each simulated dataset using a separate grid of parameters for all three models. Sometimes the grids overlap for certain variables. For instance, the grid for drift is always , regardless of which model we are trying to fit. After fitting, we ask which model has the highest posterior model probability (PMP)? Note that the PMP implicitly assumes that the data generating model is contained in the set of models that we are fitting, which is true for our parameter recovery exercise, but is surely wrong with real-world data.

How to read the figure?

In the figure below, columns indicate which model was used to generate the simulated data (call this “GEN” from now on) and rows indicate which condition the data was generated in. We then plot the PMP for each model fit to that specific simulated data (a combination of GEN-condition). Dots denote the PMP for one simulation, and box plots show the (25, 50, 75) percentiles. Note that because we are working with PMP, for every dot, there exists 2 dots in the other columns within the subplot that, taken all together, sum to 1. I chose not to draw lines between corresponding dots for the sake of visual clarity.

Main takeaway?

In most cases, the data generating model was also the most probable fit. This is especially true in the loss condition, where the reference point in the RaDDM is much further away from the implicit 0 reference point in the aDDM (i.e. 1 and 0 are close; -6 and 0 are far), thus making the two models more discernable than in the gain condition.

****

# **How well are the models recovering their original parameters?**

What are we doing?

Now that we know the data generating model is also the most likely model during recovery on average, let’s check how well each model is recovering it’s own parameters. For each GEN-condition, we run 20 simulations by drawing a set of parameter values from the grid. Below, I’m going to compare the (discrete-approximated) marginal posterior distribution for each parameter with the original value, separately for all 20 simulations. It’s A LOT of plots.

Important Notes

* I was unable to recover collapsing bounds with the toolbox aDDM.jl. Collapsing bounds always reflected as either larger drift rates, larger noise, or both. This is true not just for me, but it is also what Zeynep found on this page for the toolbox: <https://addm-toolbox.github.io/ADDM.jl/dev/tutorials/04_custom_model/>. (Note to my future self: my tests also correct for some issues with adjusting the speed of collapse to the timestep in the simulation script and start the collapse after the start of the trial, no consideration of NDT.) I think finding the problem and determining a solution will take a significantly long time considering everything else on Zeynep’s plate, so ***in order to get the paper out asap, I think we should remove consideration of collapsing boundaries from our analysis and paper.***

Legend

Black dashed lines: True, data generating parameter values.

Bars: Posterior probability of that parameter value. Green is gain condition. Red is loss condition.

Column titles indicate the parameter of interest in that column.

# **Parameter Recovery: aDDM**

Results

* Overall, the aDDM is doing a decent job of recovering the original parameters.
* When bias is large (), the model sometimes struggles to fit the other parameters. The largest error tends to be in the noise parameter. The converse is also true: when noise is large, the model sometimes struggles to fit bias. That begin said, I think large biases of are rare, and so are noise levels of up to 0.075, so I’m not too worried about these errors.
* I think the final two simulations are more representative of values we’ll see in the data, and there we’re doing a decent job of recovering the original parameters.



# **Parameter Recovery: AddDDM**

Results

* The AddDDM does a decent job of recovering its paramteres.
* When the bias is large, it sometimes has a hard time fitting the noise. I’m not too worried since I don’t think bias is often this large in real-world data.



# **Parameter Recovery: RaDDM**

Results

* The RaDDM does a good job of recovering it’s parameters. In the cases where drift is smaller (left two columns), noise is smaller (left two columns), and bias is not too extreme (between -1 and 1), then we are fitting decently. These are the range of parameter values I’d expect to see in real data.
* Sometimes it finds a bias when none exist. It seems that is happens when drift is very large. There’s probably some sort of connection between first fixation location, first fixation duration, large drift rates, and thinking there’s a bias towards one of the options depending on the probability of first fixating left. Regardless, I think a drift rate of .0075 is pretty unrealistic, so I’m not too worried about these errors.



# **How should we report this information in the paper?**

I think we should include the first figure with the model recovery to show that the right class of model will likely be returned regardless of whether people are doing some form of aDDM, AddDDM, or RaDDM. That’s 1 supplementary figure.

Then, for each data generating process, I’ll generate a vector of 4 randomly drawn integers between 1 and 20. We’ll use these to present the parameter recovery results for 4 simulations per data generating process. One supplementary figure consisting of 4 simulations per data generating process makes 3 supplementary figures.