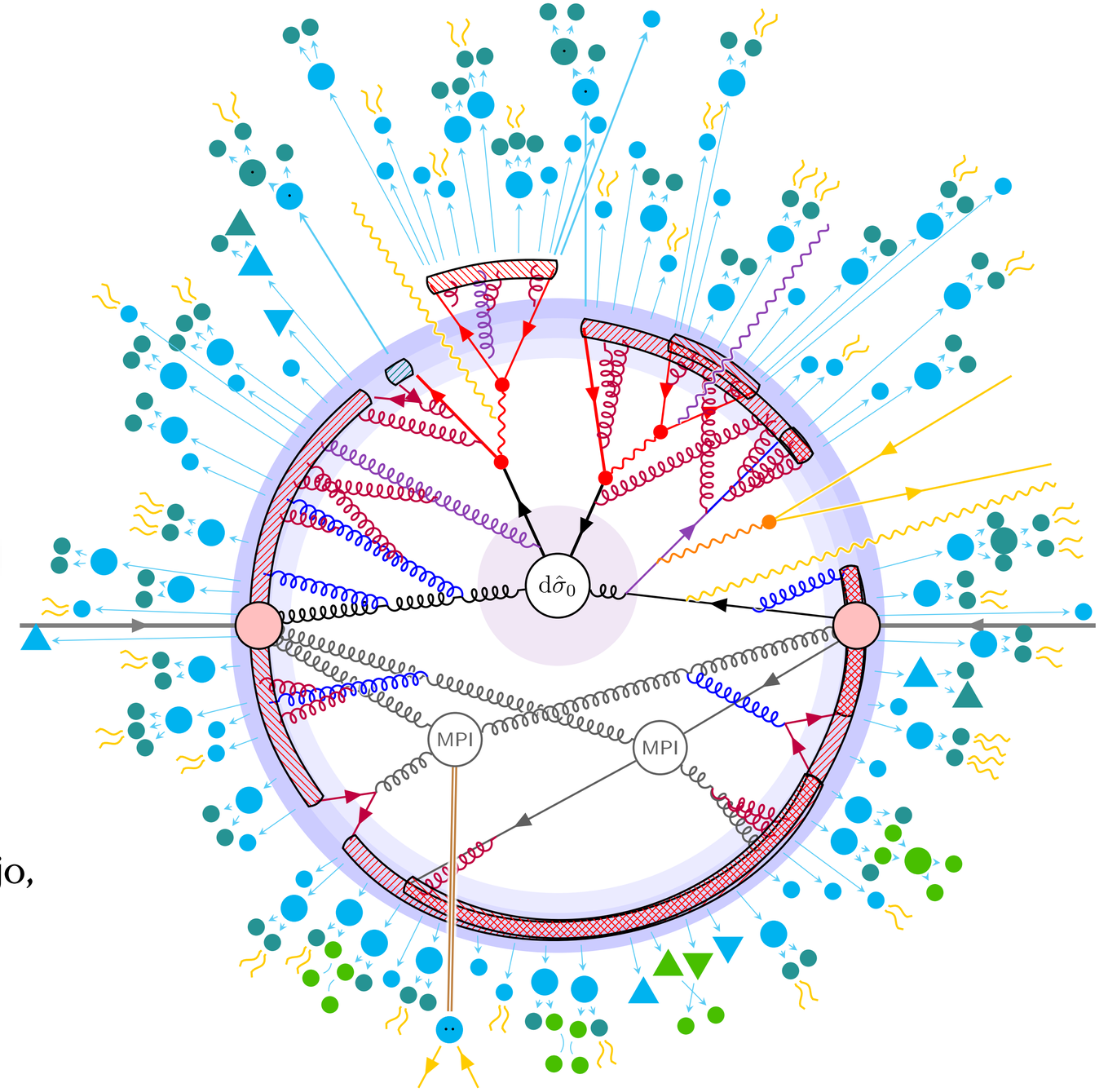
Updates towards Hyperoptimizing on PDF Distributions



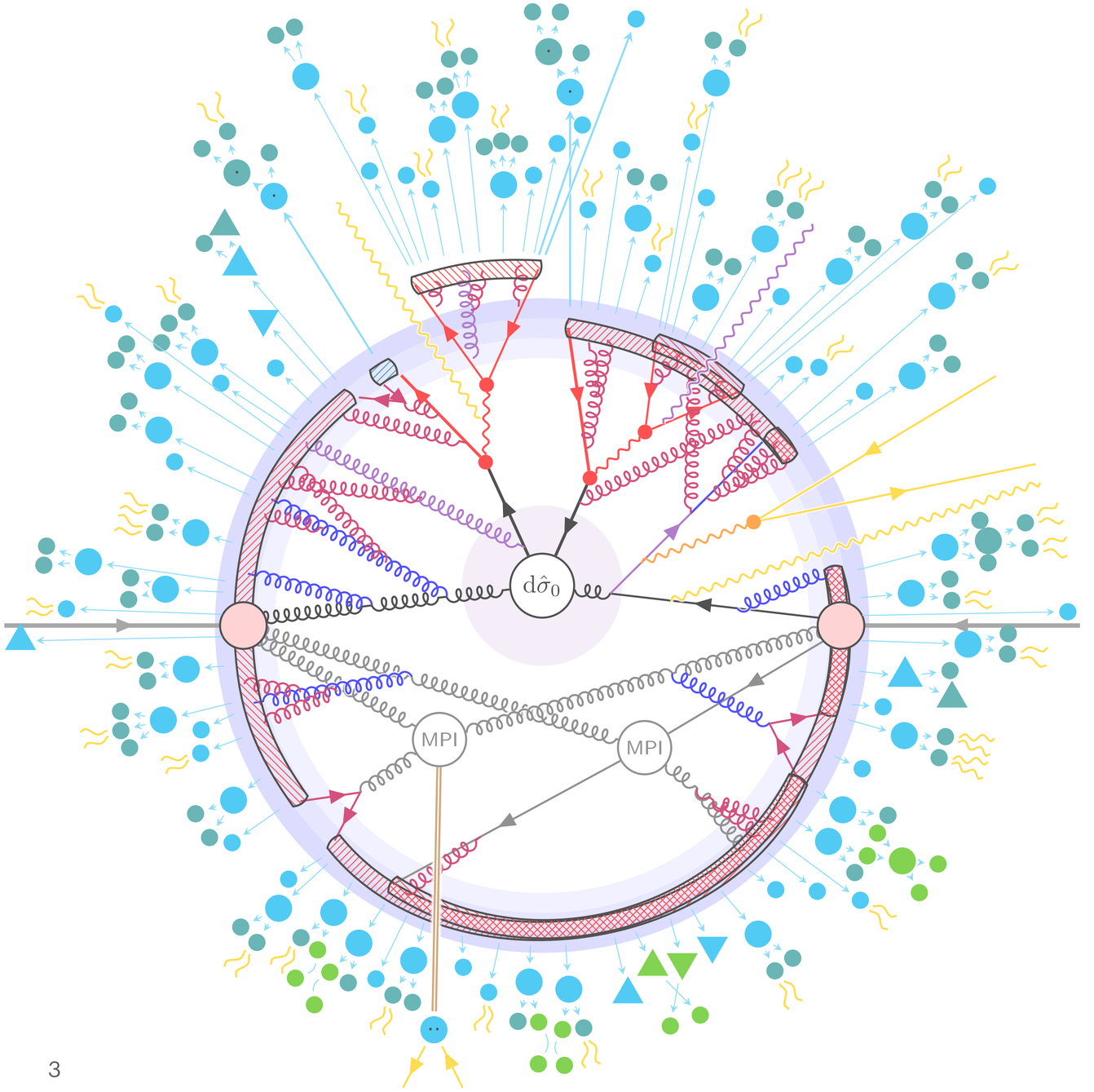
Tanjona R. Rabemananjara w/ J. M. Cruz-Martinez, J. Rojo, R. Stegeman, and ASDI eScience Engineer

Hyperoptimizing PDF Distributions:

1. **On the χ^2 of the mean**
2. **On the "Roy-ness"**
3. **On the bias/variance ratio in CT**
4. **On $|\chi^2 - 1|$ for non-fitted folds**
5. **Combinations of the above**
6. **More ideas ... ?**

SHORT OUTLINE OF THE TALK:

1. Technical Developments
2. The Curse of Hyperoptimizing on Distributions
3. Taming the Issues
4. Preliminary Results
5. Short-term Plan & Outlook



Multiple Refactoring (Several PRs Merged Already)

Refactoring model creation code #1734

Merged **APJansen** merged 74 commits into `master` from `model_refactor` on Jul 14

Conversation 186 · Commits 74 · Checks 4 · Files changed 15

APJansen commented on May 17 • edited — Collaborator

Aim

I am refactoring the code that creates the PDF model (and perhaps the full one as well). The main objective is to make the code and the resulting architecture more readable, without changing anything in the behavior.
If anything I expect the changes to make it slightly faster and smaller in memory, but either way this will likely be negligible.

Refactor xintegrator #1779

Merged **scarlehoff** merged 4 commits into `master` from `refactor_xintegrator` on Jul 28

Conversation 7 · Commits 4 · Checks 5 · Files changed 3

APJansen commented on Jul 18 — Collaborator

Adds a regression test for `xIntegrator` and refactors it to use a tensor product on the x-axis rather than replicating the grid tensor.

I expected it to be faster but the difference is negligible (few % faster).

Refactor msr #1781

Merged **APJansen** merged 23 commits into `master` from `refactor_msr` on Aug 18

Conversation 38 · Commits 23 · Checks 4 · Files changed 5

APJansen commented on Jul 18 — Collabo

Adds a regression test for the `MSR_Normalization` layer.
Mainly improving readability by rewriting the indices.
Also makes it compatible with higher rank pdfs by changing a `flatten(y)` to `y[0]` and allowing for an `out_shape` than only a 1d `output_dim`.

Awaiting further tests as per discussion here.

Refactor rotations #1780

Merged **APJansen** merged 4 commits into `master` from `refactor_rotations` on Aug 8

Conversation 11 · Commits 4 · Checks 5 · Files changed 3

APJansen commented on Jul 18 — Collaborator

This is a small refactor of the rotation layer, now requiring a `rotation_axis` argument that will be used to take a tensor product with that axis from the input and the 0th axis of the rotation matrix. (the resulting axis is by default put at the end, so this is transposed back afterward).
It's more intuitive than the previous `axes` which specified the amount of axes to contract (starting from the end of the input and the start of the rotation matrix).
More importantly, it now requires no change when axes are added to the pdf.

Many more Improvements still Ongoing (crucial for next steps!)

Refactor stopping #1792

Open **APJansen** wants to merge 20 commits into `master` from `refactor_stopping`

Conversation 12 · Commits 20 · Checks 5 · Files changed 3

APJansen commented on Aug 15 • edited — Collaborator

This simplifies the `stopping.py` module, reducing
`WriterWrapper`.

I started looking at this because #1782 requires c
do exactly the same.

Go from one step being an epoch to one step being a batch #1802

Open **APJansen** wants to merge 6 commits into `master` from `epochs_to_batches`

Conversation 11 · Commits 6 · Checks 4 · Files changed 2

APJansen commented 3 weeks ago • edited — Collaborator

As part of the changes mentioned in issue #1803, this is a very simple change that results in a factor 2 speedup.

It simply copies the input x grids by the number of epochs, and calls one step a batch rather than an epoch. This avoids
ake up nearly 50% of the total training time.

tream as it's crashing (perhaps just undoing

Parallel replicas with varying tr-vl masks #1788

Open **goord** wants to merge 38 commits into `master` from `trvl-mask-layers`

Conversation 8 · Commits 38 · Checks 4 · Files changed 17

goord commented on Aug 9 — Collaborator

This is a continuation of the pull request #1661 that implements the parallel replicas with `same-trvl-seed=false` support.
The branch has been migrated to this original repo and the latest master was merged.

Multi Replica PDF #1782

Draft **APJansen** wants to merge 1 commit into `master` from `multi_replica_pdf`

Conversation 10 · Commits 1 · Checks 6 · Files changed 3

APJansen commented on Jul 24 — Collaborator

Question

This will be some work, so before continuing past this I'd like to confirm that you agree that once finished this will be a beneficial change.

Hyperopt loss #1726

Draft **APJansen** wants to merge 49 commits into `master` from `hyperopt_loss`

Conversation 1 · Commits 49 · Checks 4 · Files changed 22

APJansen commented on May 4 — Collaborator

Improving hyperoptimization, experimenting with different hyperoptimization loss functions

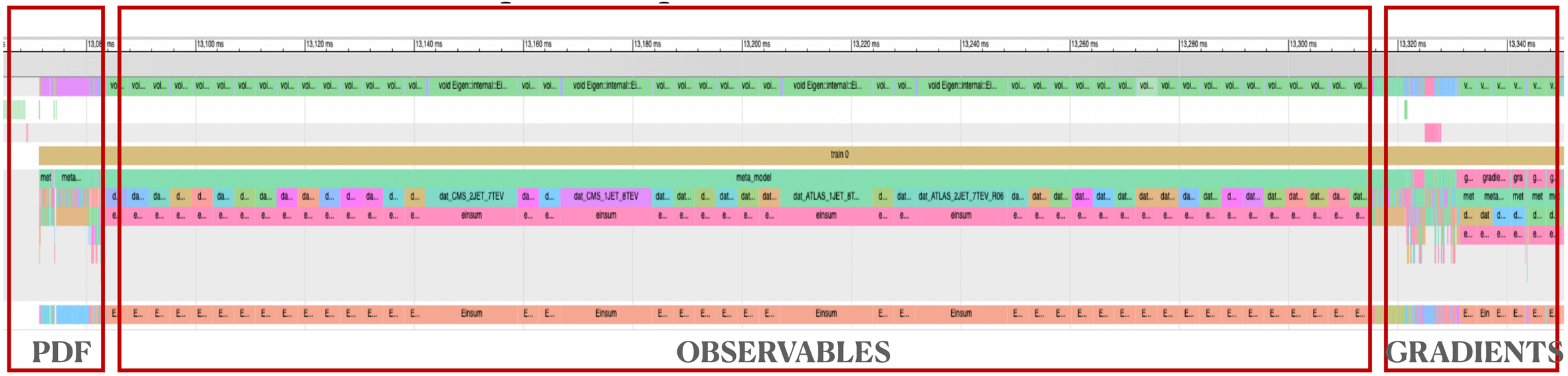
The Curse of Hyperoptimizing Distributions

$$\tilde{L} = \frac{1}{n_{\text{fold}}} \frac{1}{N_{\text{rep}}} \sum_{k=1}^{n_{\text{fold}}} \sum_{r=1}^{N_{\text{rep}}} \chi^2_{k,r}$$

Hyperopt with full distribution in a Global Fit:

Nb. Replicas	Nb. GPUs	Nb. Trials	Time (s)	SUB/Trial	Projected Full Hyperopt (s)
50	1	2	693	2.5k	360k
100	1	2	965	3k	680k

Performance of 100 replicas setup with Tensorboard:



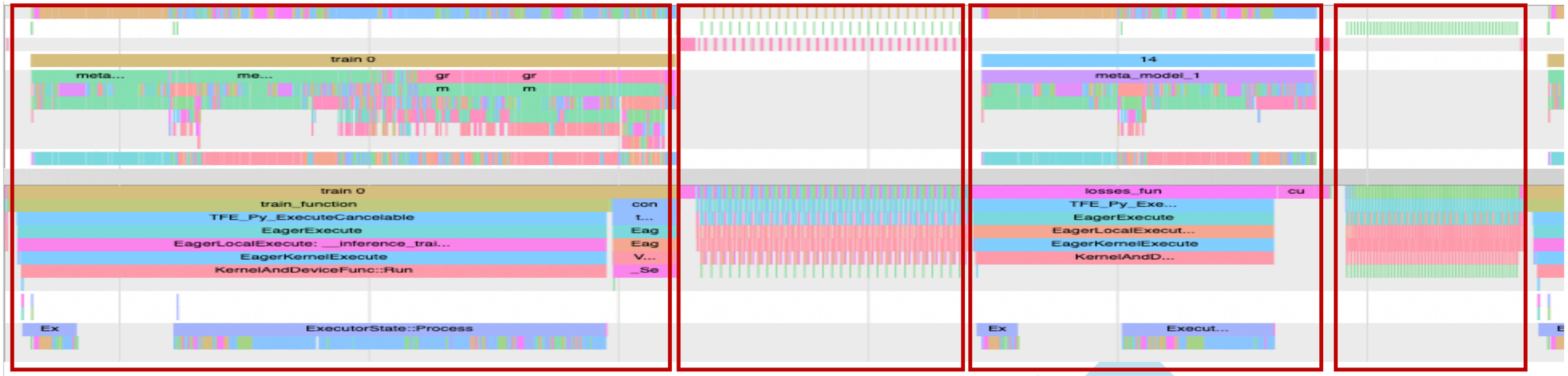
The Curse of Hyperoptimizing Distributions

$$\tilde{L} = \frac{1}{n_{\text{fold}}} \frac{1}{N_{\text{rep}}} \sum_{k=1}^{n_{\text{fold}}} \sum_{r=1}^{N_{\text{rep}}} \chi^2_{k,r}$$

Hyperopt with full distribution in a Global Fit:

Nb. Replicas	Nb. GPUs	Nb. Trials	Time (s)	SUB/Trial	Projected Full Hyperopt (s)
50	1	2	693	2.5k	360k
100	1	2	965	3k	680k

Performance of 100 replicas setup with Tensorboard:



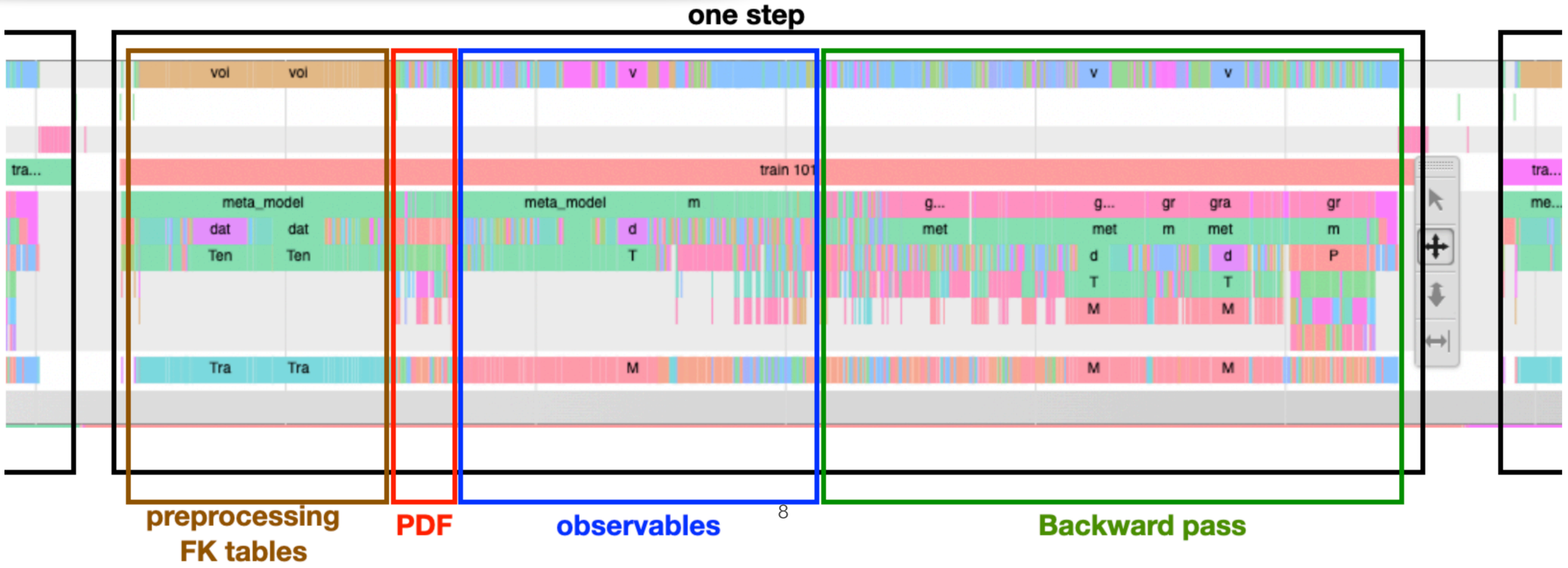
TRAINING GAPS? VALIDATION GAPS?

Clever Solutions

1. **Re-order indexes** for Tensor Multiplications
2. **Fixed Shapes of Masks** (Crucial for DY FK tables)
3. Using a **Single PDF Model** for all the replicas (similar to what is currently done for flavours)
4. Improve computation of the **Validation**

Full HyperOpt w/ 100 replicas & 5 Folds

	Time (mn)	Hrs/Trial	SBU/Trial
No. Improv.	28	47	6k
$1 \oplus 2$	11	19	2k
$\oplus 3$	2.5	4	530
$\oplus 4$	1.3	1.6	220

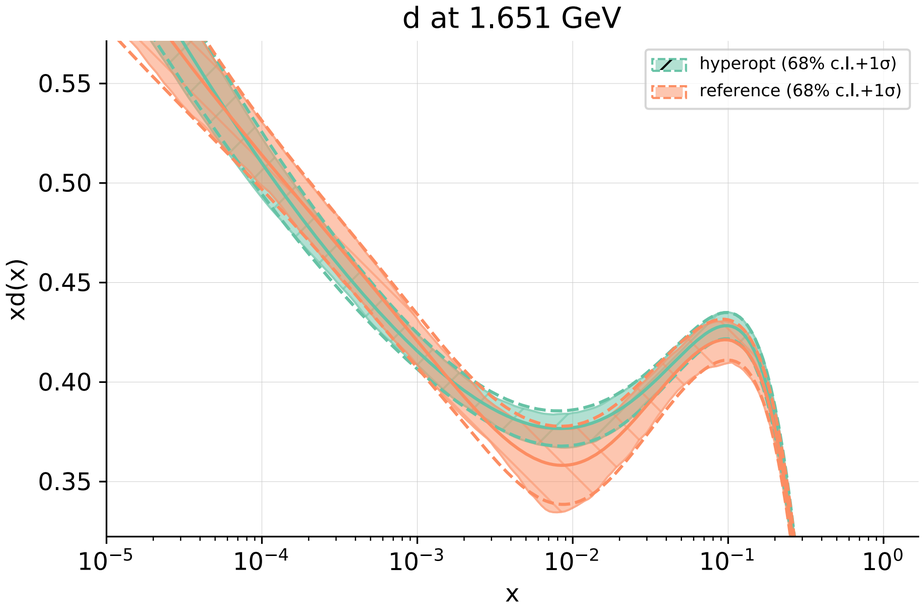


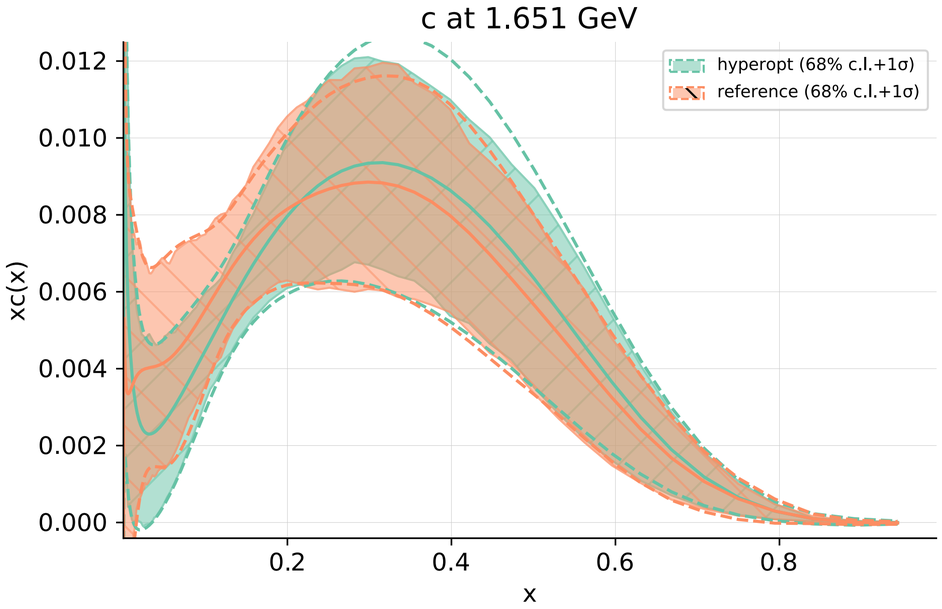
New Hyperparameters Configurations

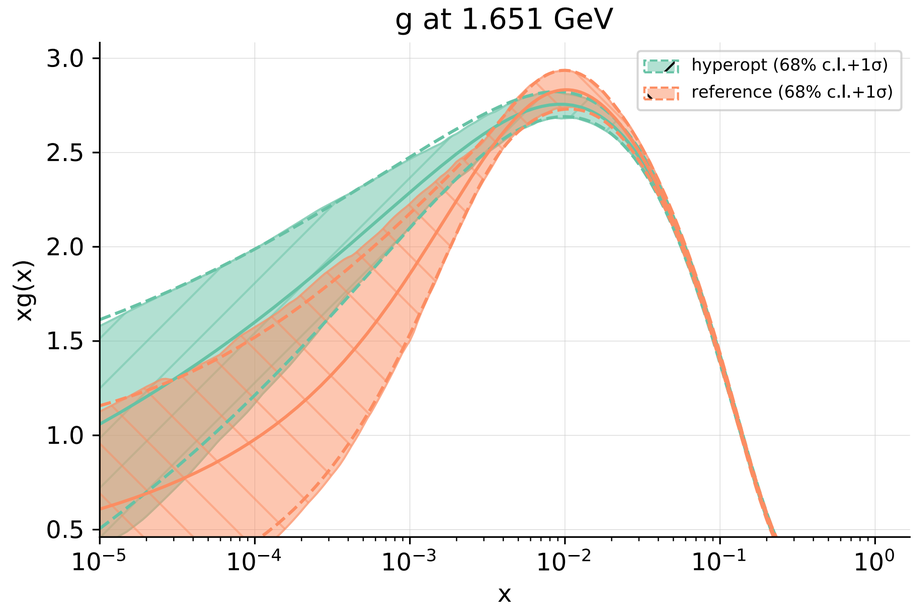
Parameter		NNPDF4.0
Architecture	2-14-42-8	2-25-20-8
Activation function		hyperbolic tangent
Initializer		`glorot_normal`
Optimizer	Adam	`Nadam`
Clipnorm	2.8×10^{-6}	6.0×10^{-6}
Learning rate	1.2×10^{-3}	2.6×10^{-3}
Maximum # epochs		17×10^{3}
Stopping patience		10% of max epochs
Initial positivity $\Lambda^{(\mathrm{pos})}$		185
Initial integrability $\Lambda^{(\mathrm{int})}$		10

PDF Fit Results (Full Report)

	hyperopt	reference
χ^2	1.19543	1.17051
$< E_{trn} >$	2.281±0.061	2.256±0.059
$< E_{val} >$	2.37±0.10	2.36±0.10
$< TL >$	15500±2300	13100±2500
$< \chi^2 >$	1.212±0.013	1.194±0.015
ϕ	0.1289±0.0051	0.1549±0.0048

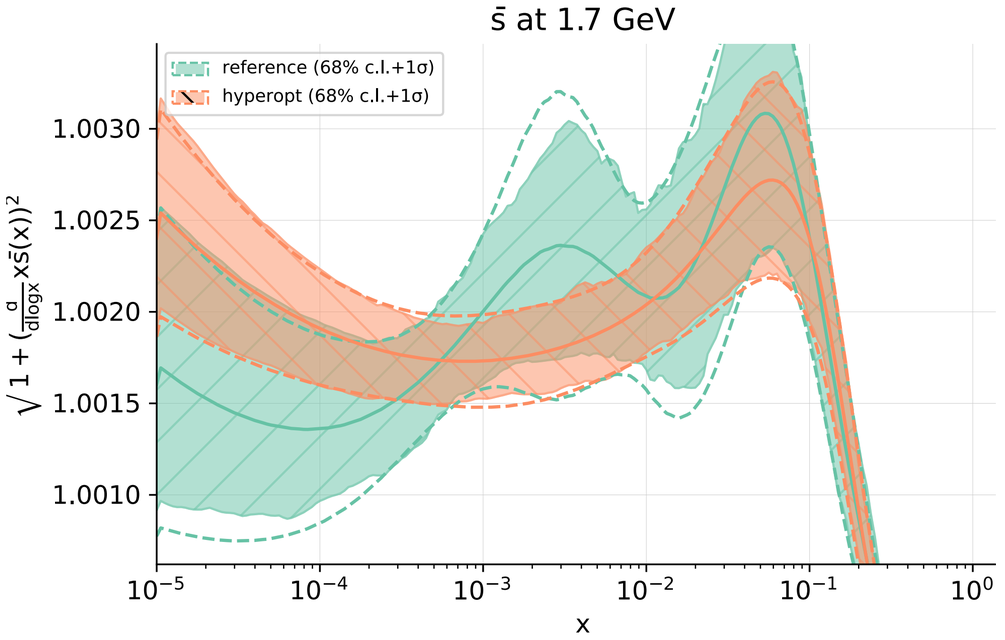


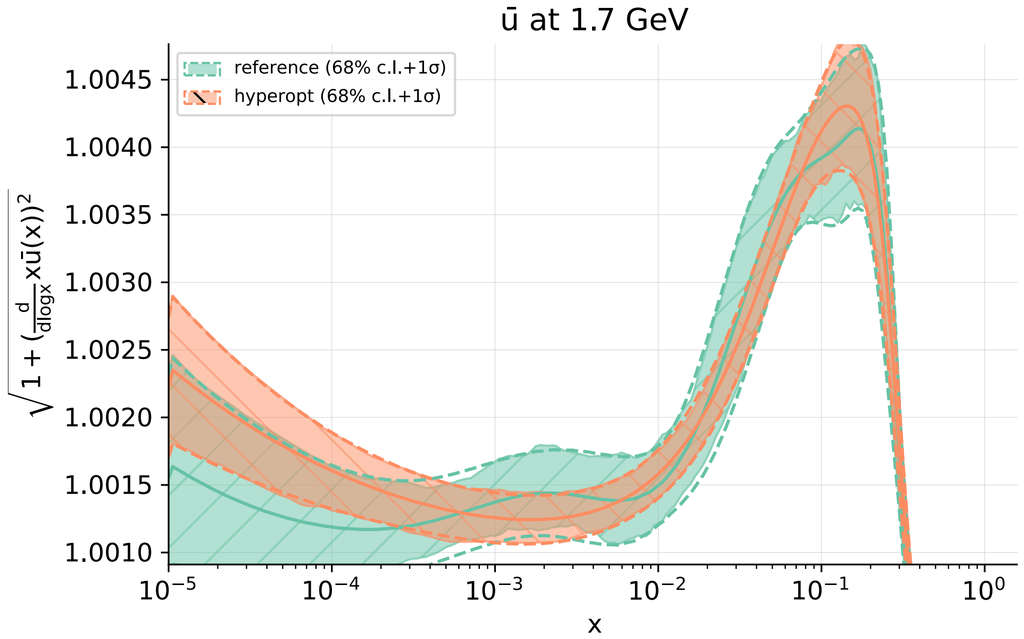


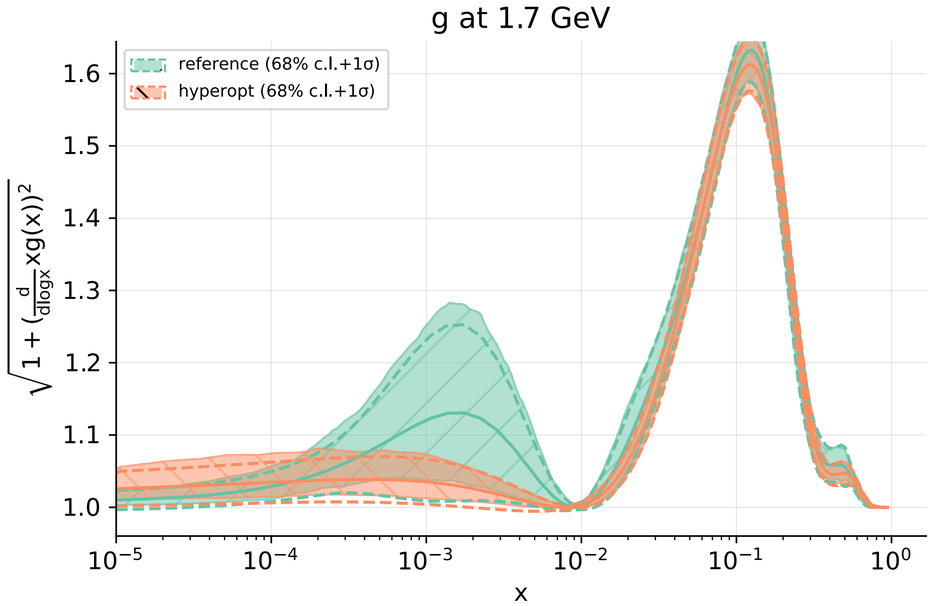


Kinetic Energy (Full Report)

	hyperopt	reference
χ^2	1.19543	1.17051
$< E_{trn} >$	2.281±0.061	2.256±0.059
$< E_{val} >$	2.37±0.10	2.36±0.10
$< TL >$	15500±2300	13100±2500
$< \chi^2 >$	1.212±0.013	1.194±0.015
ϕ	0.1289±0.0051	0.1549±0.0048







Short-Term Plan & Outlook

- Finish implementation of all technical improvements
- Closure Tests / Future Tests / etc.
- New ideas/propositions?

Steps remaining

Unfortunately I'll have very little time in the next month to work on this (holidays and other projects). Below I'll list the steps necessary to integrate it, and **where help could be useful**.

1. Get the trvl-mask-layers branch to pass the tests.
   - Some help here would be useful, **@Radonirinaunimi** would take a look
2. The changes in point 1 above are in a branch gpu-go-brrr (;P), off of trvl-mask-layers. This can be merged into it.
3. Once merged they should be tested and reviewed.
   - What would be super useful here is to have a list of runcards to test, and also the actual results from master. Since this is already much faster, it's a lot easier if we can only run them on this branch, and have something to compare against already.
4. The rewriting from epochs to batches is independent of all the other changes. If anyone wants to pick that up that'd be great. I started it in **Go from one step being an epoch to one step being a batch** #1802
5. The rewriting of the 3 "models" into one and instead 3 losses (By which btw I don't necessarily mean that we put that part in an actual `keras.Loss` or something, not sure if that's efficient or not, just to not repeat the computations), I think this is also relatively independent of the rest. If anyone wants to do this that'd be great, it doesn't have the highest payout/effort ratio of all these, so I can also do it myself after the last point.
6. The multi replica PDF, this is the most work, and the most specialized, so I think it's best if I focus on this. It's not very far I think, I need to look at **Refactor stopping** #1792 again, then I hope write another PR that concatenates the PDFs as soon as possible and makes the remaining logistics around it compatible but hopefully remain numerically identical, and finally drop in the actual joint model. This then needs to be tested again, but since I'll be able to reuse the results from master at step 3 it should be quick.

The Judgement of Paris by Jacques-Clément Wagrez