Adjusting dropout scores to reflect partial support

In the neural GLM we are performing model selection and model evaluation by fitting the full model, containing many different regressors, then fitting a series of reduced models with each regressor selectively removed "dropped-out", or all but that regressor removed "single-drop out". This approach allows us to evaluate the relative value of different regressors in describing the df/f signal. However, the current implementation is biased by the fraction of the time series in which each regressor has support. That is, if a rare event has high explanatory value but only happens rarely, there is a limitation to how large its dropout score can be. Here I develop two adjustment procedures that remove this bias. I propose these adjustments be used alongside the original score to evaluate model selection.

Non-overlapping rare and common events

Lets consider a basic model:

$$y = Xw, \tag{1}$$

where Y is a timeseries of length N, w is a weight vector of length 3, and X is a design matrix (N x 3). The first column of X is for the intercept regressor, and is all ones. The second column is for a common event (event "A"), that alternates ±1 except for one timestep where it takes value 0. The third column is for a rare event (event "B") that is zero everywhere, except for one timestep where it takes value 1. Figure 1 shows the timeseries in question. Figure 2 shows the result of dropping out these regressors.

The naive dropout assigns a large explanatory value to A, because removing it decreases the variance explained by about 90%, and a low explanatory value to B, because removing it decreases the variance explained by only about 5%. We arrive at this conclusion regardless of whether we dropout A or B, or perform the single-dropout for A or B (labeled "JustA" and "JustB" in Figure 2).

One small note on computing the dropout scores. When removing a single regressor I compute the dropout score d by:

$$d_i = -(1 - \frac{V_i}{V_F}). \tag{2}$$

Where V_i is the variance explained for model i, and V_F is the full model variance explained. However when looking at the dropout score for the single-dropout models we need to compute the dropout as :

$$d_i = -\frac{V_i}{V_F} \tag{3}$$

to reflect that our intepretation of the single-dropout is reversed given that the model contains ONLY that regressor. In the plots below "dropout" refers to not taking this correction into account, while "alternative dropout" does take this correction into account.

To account for the partial support of the rare event B, we can compute the variance explained of the reduced model only on the timesteps where B has support. We can think of this as the "explainable variance" of B. This adjusted variance explained is given by:

$$\hat{V_i} = \frac{var_i(y, mask_i) - var_i(y_i, mask_i)}{var_i(y_F, mask_i)}, \tag{4}$$

where $var_i(y_i, mask_i)$ is defined as:

$$var_i(y, mask_i) = E\left[(y[mask_i] - \mu)^2)\right]. \tag{5}$$

Here y_i is the timeseries for model i. Model i can be either the dropout for regressor i, or the single-dropout for model i. y_F is the full model timeseries. $mask_i$ is the mask of where regressor i has support. Importantly μ is the mean of y_F on all timesteps, not limited to the timesteps where i has support.

We can see the effect of this adjustment in Figure 2. Under this adjustment "dropA" and "dropB" are almost equivalent. The small difference is due to the intercept regressor. This is more clearly seen by comparing "justA" and "justB", where A and B both explain all of the explainable variance on their respective timesteps as the full model.

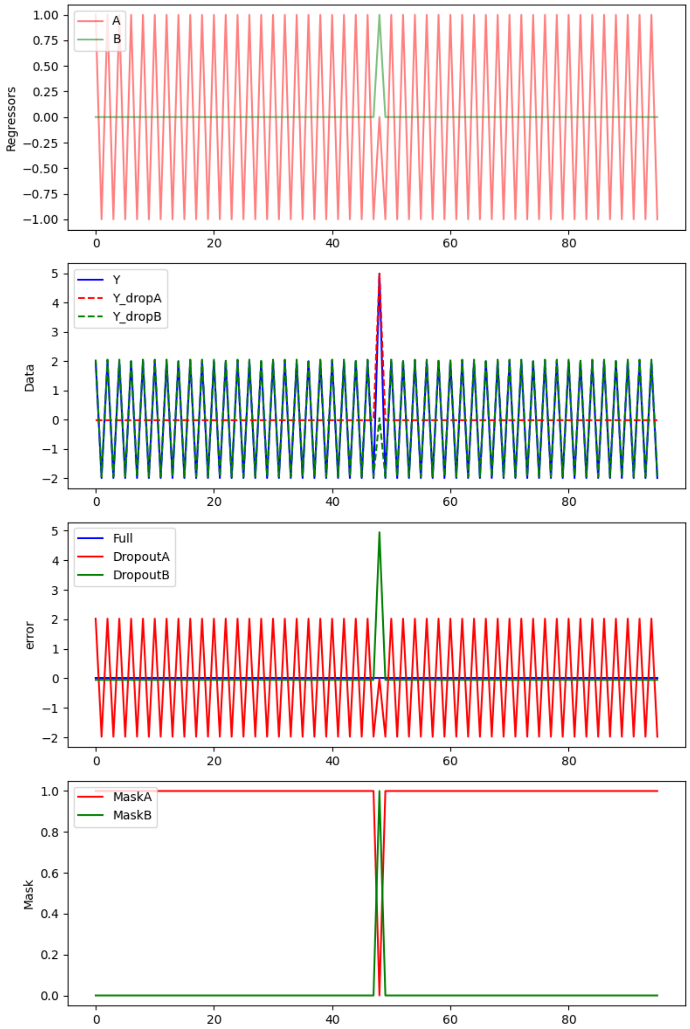

Figure 1: (Top) The second and third columns of the design matrix showing the common events A and the rare event B. (2nd from top) The target timeseries Y can be perfectly described by the full model. However the reduced models "dropA" and "dropB" cannot perfectly fit the model. (3rd from top) The error for the three models. (Bottom) The masks that show which timesteps the regressors have support.

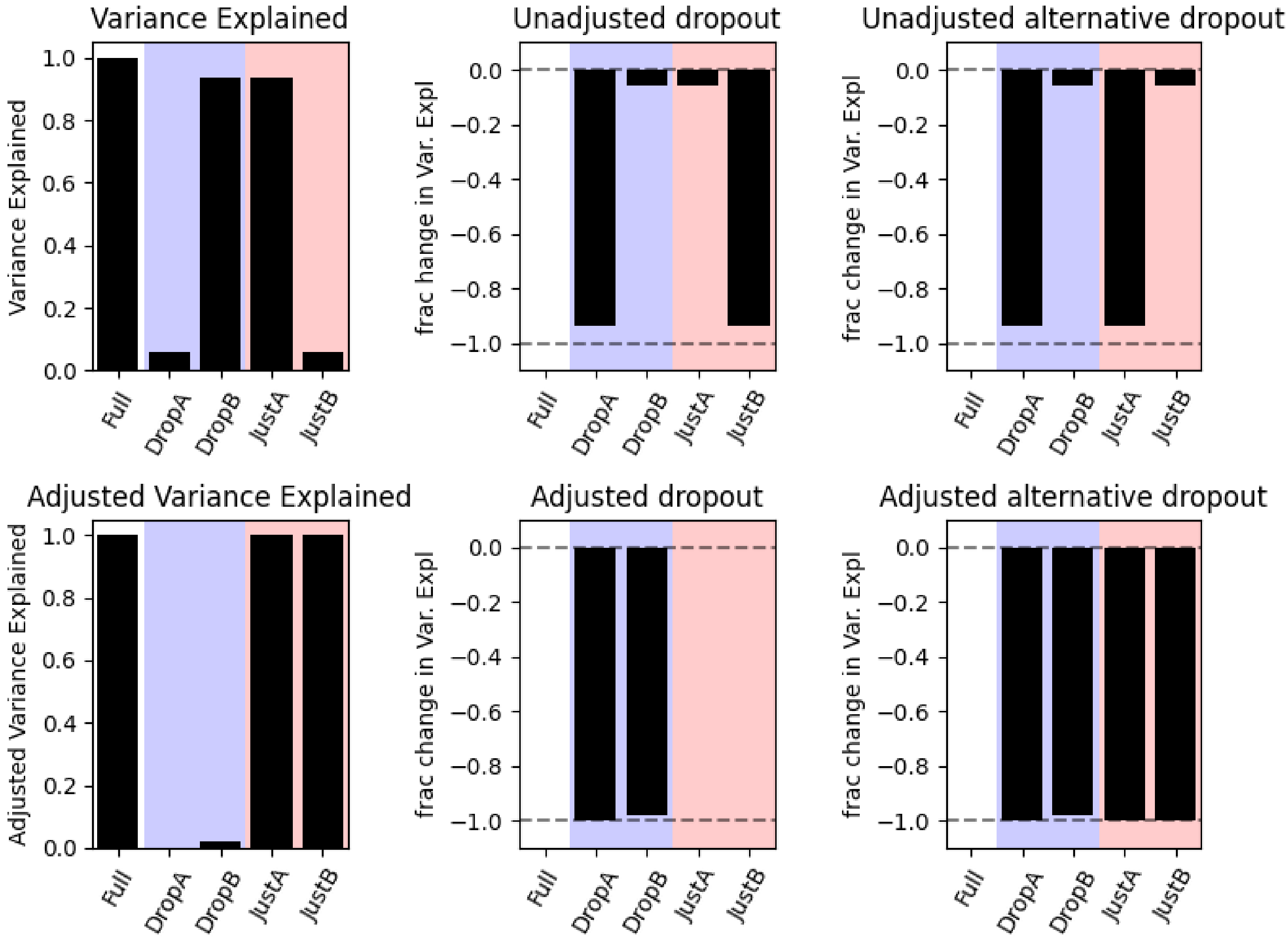

Figure 2: (Top) Variance explained and dropout metrics without considering partial support. (Bottom) Variance explained and dropout metrics adjusted for partial support of regressors. (Left column) Variance explained by full model and each of the dropout models. (Middle Column) dropout metric that does not flip the sign for the single-dropouts. (Right column) Dropout metric that flips sign of the single-dropouts for easier interpretation.

Overlapping rare and common events

For a clear exploration of dropout scores and the adjustments provided here I repeat the same analysis but make the regressor for A alternate ±1 including on the timestep where the rare event B occurs. This lets A have slightly larger explantory power of the rare event B. The unadjusted dropout scores are not impacted by this modification. However the adjusted dropout scores are highly sensitive to this difference when looking at the “dropB” model. The slight explanatory power of the intercept is magnified, and underlies the importance in this case of looking at the “justB“ model.

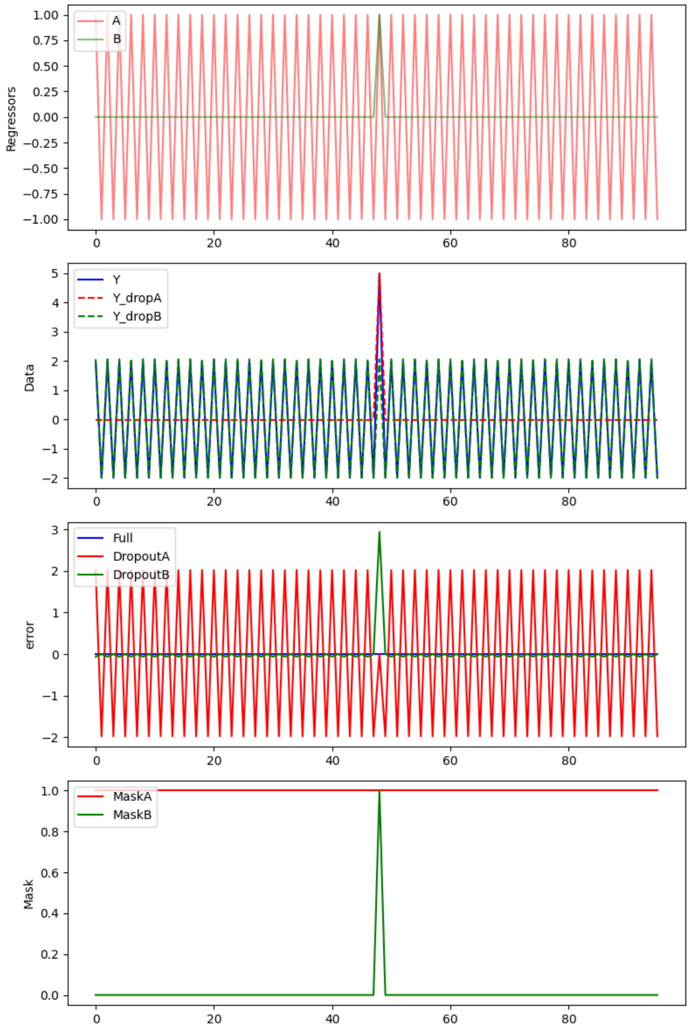

Figure 3: (Top) The second and third columns of the design matrix showing the common events A and the rare event B. (2nd from top) The target timeseries Y can be perfectly described by the full model. However the reduced models “dropA” and “dropB” cannot perfectly fit the model. (3rd from top) The error for the three models. (Bottom) The masks that show which timesteps the regressors have support.

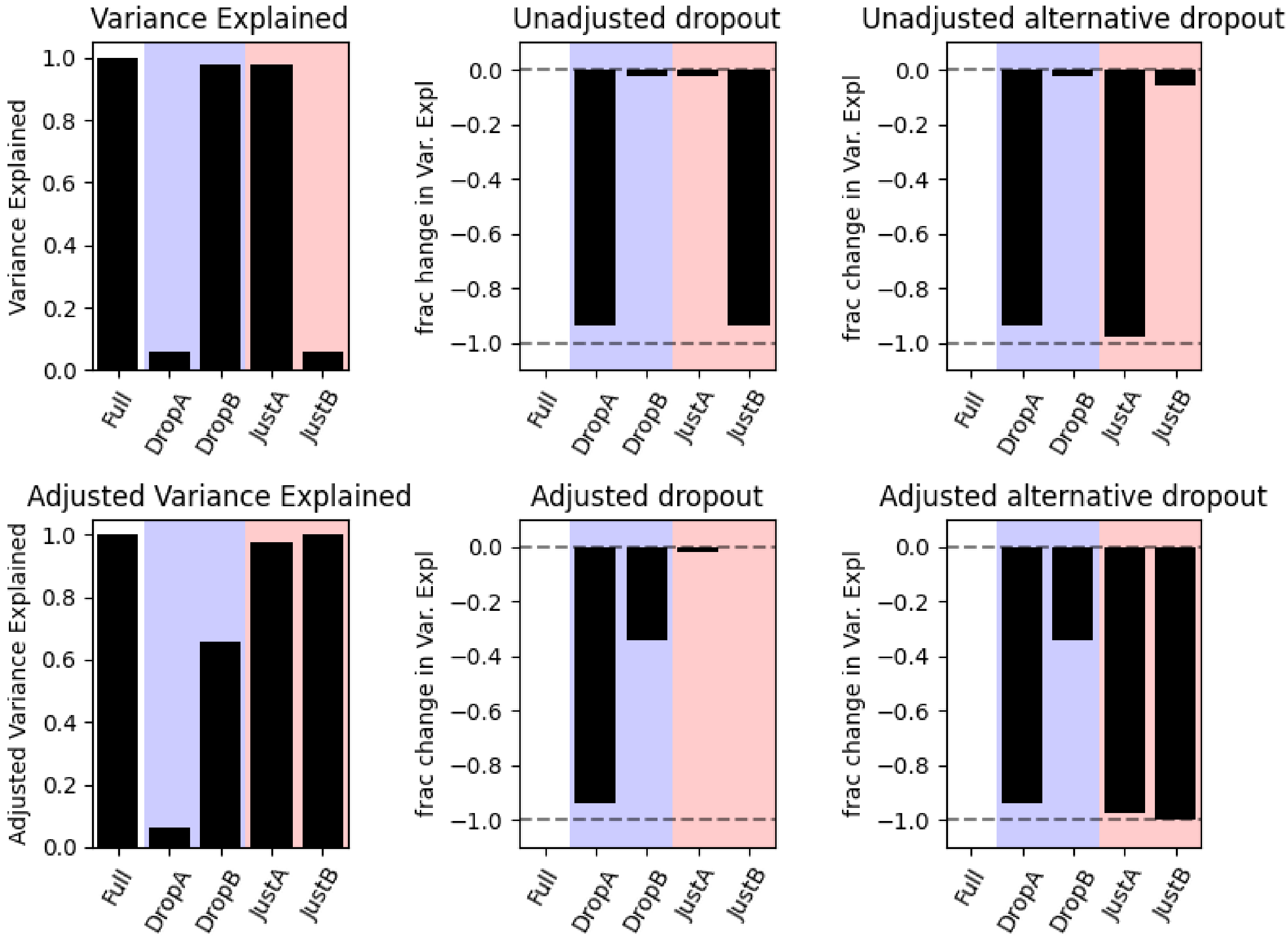

Figure 4: (Top) Variance explained and dropout metrics without considering partial support. (Bottom) Variance explained and dropout metrics adjusted for partial support of regressors. (Left column) Variance explained by full model and each of the dropout models. (Middle Column) dropout metric that does not flip the sign for the single-dropouts. (Right column) Dropout metric that flips sign of the single-dropouts for easier interpretation.

The need to compute the full model comparison on the mask as well In the above examples, the full model has 100% accuracy, so it obscures a small detail in the computation of the dropout adjustment scores. When computing the dropout for each kernel on its mask you need to compare against the full model's variance explained on the same mask. Without this step, we will get misleading and noisey dropout scores. To illustrate I can add some noise the the timeseries so the full model cannot have perfect accuracy, then look at the dropouts without taking this into account "Adjusted dropout", and taking the full model mask into account "Full Corrected Adjusted".

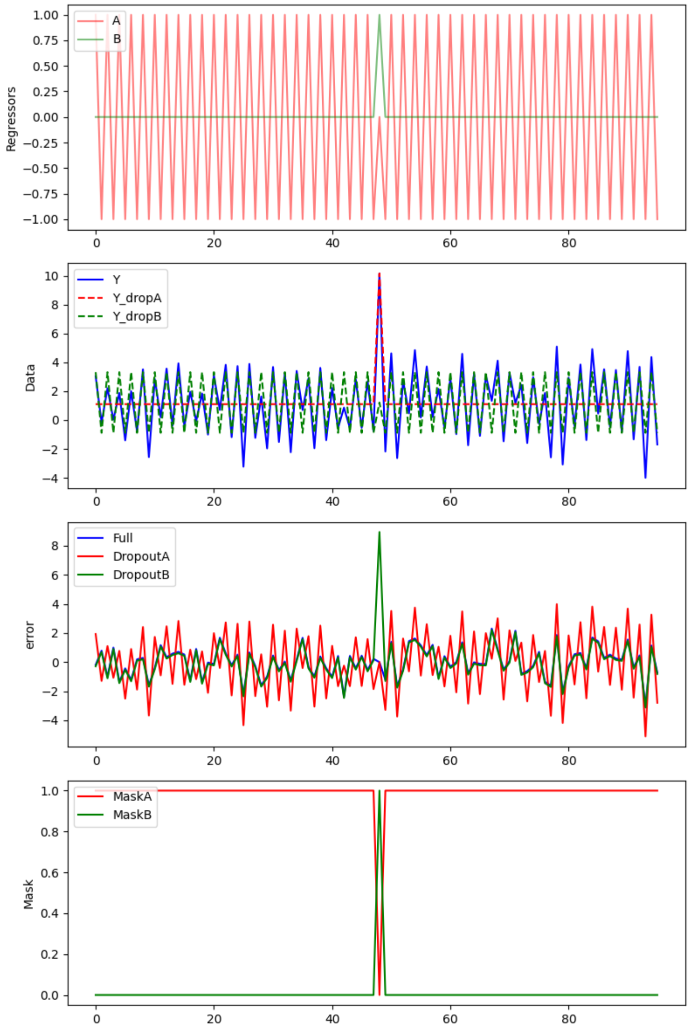

Figure 5: (Top two rows) Same as above, but with noise added to the data. (Bottom Row). The adjusted and alternative adjusted dropouts both compare the dropout's variance explained with the full-model's variance explained on the same mask. This resolves non-perfect full-model performance, and gives us interpretable dropout values.

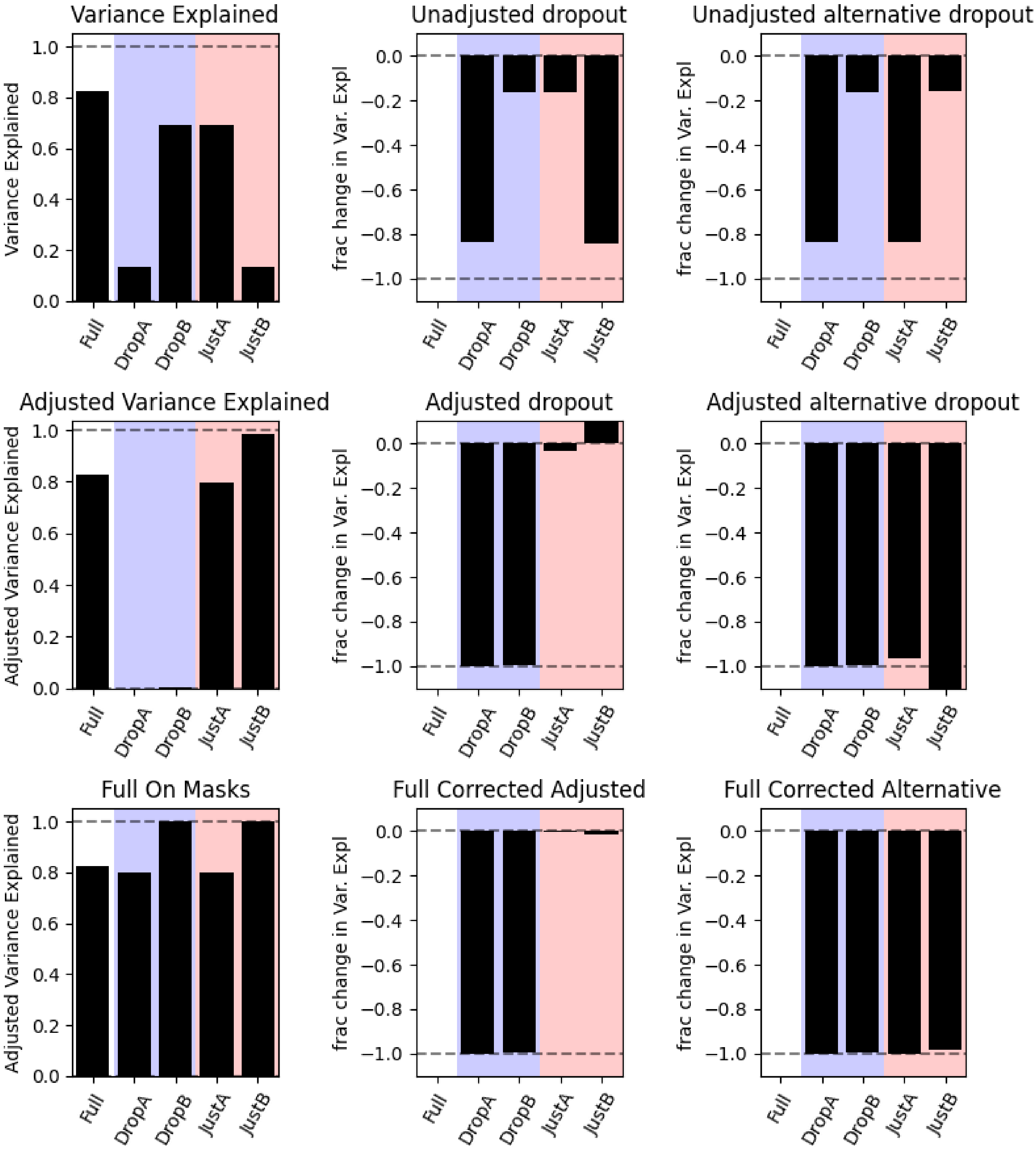
Variance Explained
Variance Explained
1.0
0.8
0.6
0.4
0.2
0.0
Full
DropA
DropB
JustA
JustB
Unadjusted dropout
frac hange in Var. Expl
0.0
−0.2
−0.4
−0.6
−0.8
−1.0
Unadjusted alternative dropout
frac change in Var. Expl
Adjusted Variance Explained
Adjusted Variance Explained
Adjusted dropout
frac change in Var. Expl
Adjusted alternative dropout
frac change in Var. Expl
Full On Masks
Adjusted Variance Explained
Full Corrected Adjusted
frac change in Var. Expl
Full Corrected Alternative
frac change in Var. Expl

Figure 6: