We would like to thank the reviewers for their insightful comments and suggestions. Due to space limitation, we only address their main concerns below, but all the other (more minor) issues and suggestions will be properly addressed in the final paper (ie big picture description without causal discovery jargon, intuitive sepset (in)consistency examples, refs to SAT-solver, application to other PC-related methods, change of title to distinguish from asymptotic consistency).

• **Reply to Reviewer 1** (***"discuss computational complexity"***):
The reviewer is right in pointing out that the complexity for ensuring sepset consistency needs to be clarified. As alluded to in the original manuscript, the complexities including *versus* excluding edge orientation needed to be addressed separately, although both approaches lead to similar results, in terms of performance, Figs. 2&3 and new Figure below.

The skeleton-consistency can be done at worst with a linear complexity increment, $\mathcal{O}(|\boldsymbol{V}| + |\boldsymbol{E}|)$, relative to PC algorithm*, by using the biconnected component analysis based on block cut tree decomposition detailed in Suppl. Material. [*PC algorithm runs in exponential time in the worst case but usually in polynomial time on sparse DAGs].

By contrast, the orientation-consistency introduced in the original submission involved the search of collider-free paths, entailing in the worst case an exponential complexity, which could be largely mitigated, in practice, by first applying the very efficient skeleton-consistent algorithm, as discussed in the original manuscript. However, the situation is actually better, as we recently realized that the orientation-consistency could be implemented with the same linear complexity as the skeleton-consistency, by exploiting PC conditional independence search restricted to $\mathrm{N}(\{X,Y\})$, the neighborhood of X and Y, with a more appropriate definition of orientation-consistency, ie $\mathrm{Consist}(X,Y|\mathcal{G}) = \{Z \in \mathrm{N}(\{X,Y\}) \mid Z \text{ is on a path, } \gamma^{Z}_{XY}, \text{ from } X \text{ to } Y \text{ and } Z \text{ is not a child of } X \text{ or } Y\}$. The proof of sepset consistency (Theorem 4) using this definition is essentially the same and the results are still very similar to those obtained with the (unmodified) skeleton-consistency approach, as seen in the new Figure below. The new complexity is thus at worst linear in all cases.

• **Reply to Reviewer 2** (***"provide additional experimental evaluation on standard benchmarks"***):
Following suggestions by Reviewers 2 & 3, we performed additional evaluations on standard benchmarks from the Bayesian Network repository, which display a clearer performance improvement over standard PC, see Figure below.

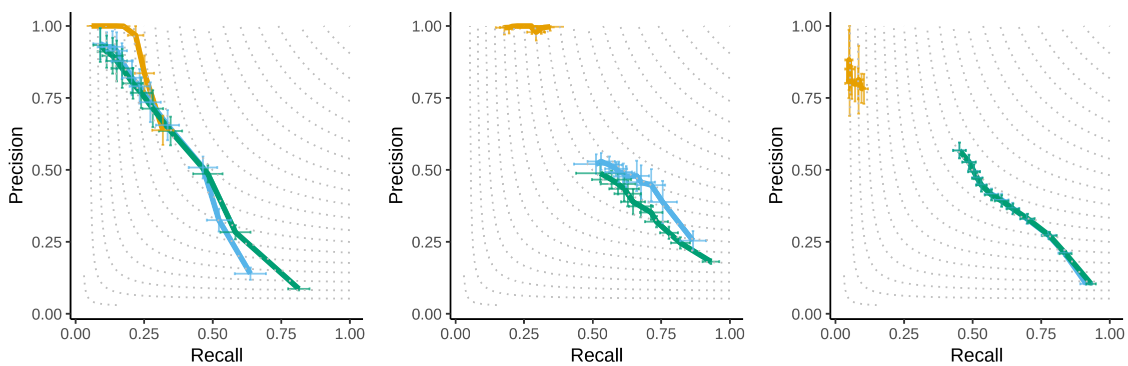

Figure 1: Precision-recall curves (1e-25 $< \alpha <$ 0.5) for the original (yellow), skeleton-consistent (green) and (new) orientation-consistent (blue) PC-stable. Hepar2 (left), Insurance (middle) and Barley (right) benchmarks from `www.bnlearn.com` ($N = 1000$). Original PC: Rec $<$ 0.15-0.35 and Prec $>$ 0.65 at max Fscore, see iso-Fscore dotted lines ($\alpha_{\rm opt} =$ 0.5 / 0.5 / 6e-5, respectively); Sepset-consistent PC-stable: Rec $\simeq$ 0.5 and Prec $\simeq$ 0.5-0.6 at max Fscore $\simeq$ 0.5-0.55 ($\alpha_{\rm opt} =$ 5e-3 / 8e-7(1e-17) / 1e-20, respectively).

• **Reply to Reviewer 3** (***"show a more clear improvement over the standard PC algorithm"***):
As underlined in the last paragraph of the manuscript (which will also be emphasized earlier in the final paper), the main improvement of our method over standard PC concerns its guarantee on the consistency of the separating sets (Theorem 4), not its performance in terms of precision-recall plot (Fig. 2). In addition, we show that ensuring consistency of separating sets improves their validity in terms of actual d-separation (Fig. 3).

Yet, we agree with this reviewer that our choice of empirical evaluation was not optimal... as it showcased settings for which the original PC-stable algorithm performs already very well (ie the maximum Fscore reached for an optimal α is already close to the top right corner in the precision-recall plot). However, as is well known in the field, PC-related algorithms typically exhibit poor recalls on standard benchmarks, as shown in the examples of the new Figure above, whereas enforcing sepset consistency leads to a better balance between precision and recall at maximum Fscore.

While the "high-end of precision" can be seen as important in order to trust the discovered edges, it also corresponds to the "low-end of recall", which usually implies that many true edges are actually dismissed. For this reason, improving the low recall of PC-related algorithms has been a long standing yet unachieved goal in the field (see eg section 5.2 of Colombo et al, JMLR, 15 (2014) 3741-3782). It is especially important for the interpretability of real-life applications where one would like to discover, beyond the obvious edges, as many non-obvious edges as possible or else the conditional independences with consistent separating sets to explain their removal. This is the motivation for our paper.