**Single causal graph**

**1. Measure how strongly a feature is associated with the chosen outcome**

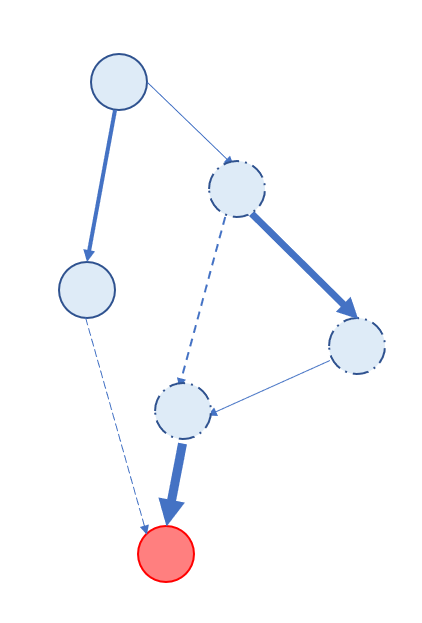
* Both numerical: Pearson Correlation Coefficient
* Categorical and numerical: multiple regression R2
* Both categorical: Cramer’s V

Select the top n highly relative features and show the distributions of the selected features.

**2. Causal discovery algorithm (PC or F-GES) to draw a graph**

**Map:**

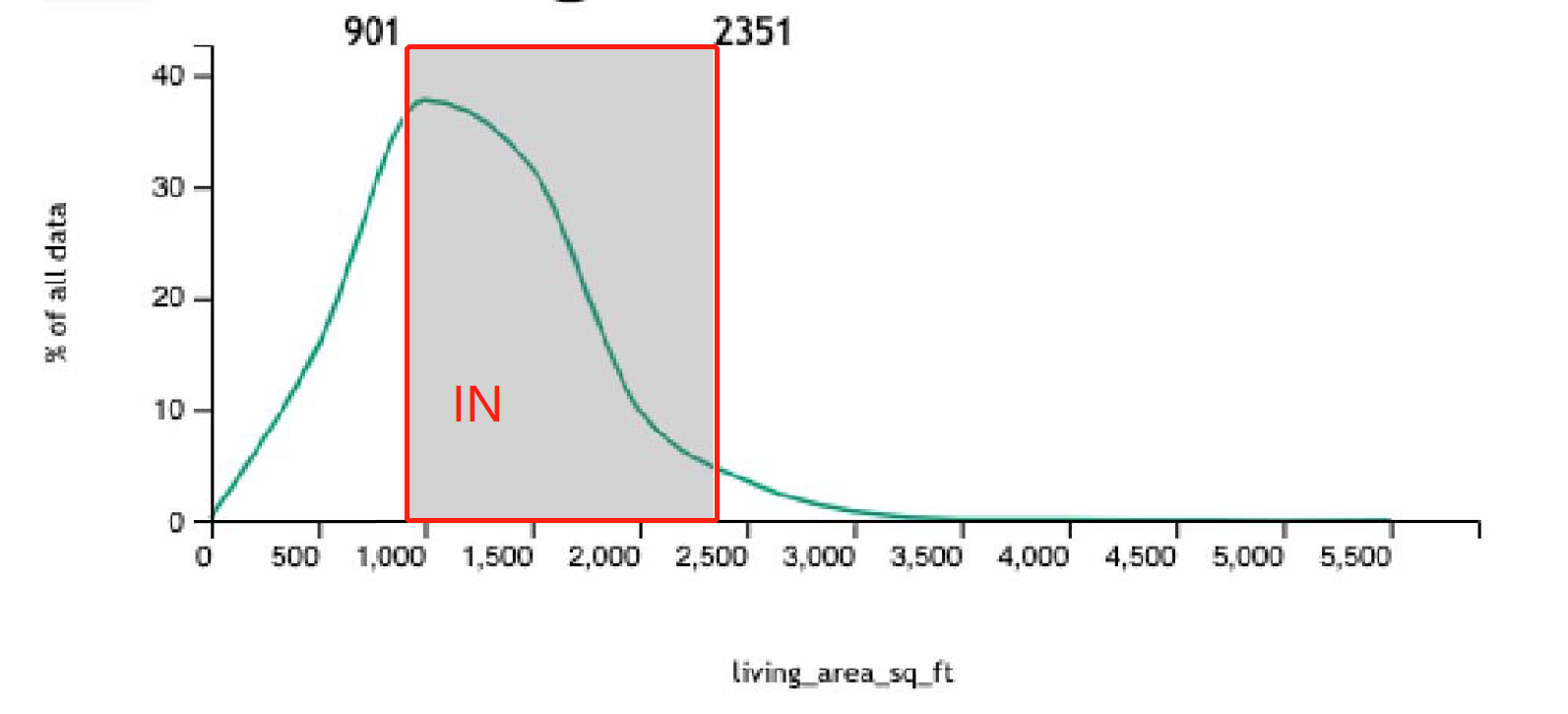
1. variable + label: node
   1. variable type: solid node (numerical), dashed node (categorical)
2. causal direction: direction of edge arrow
3. causal effect size（**only counts those with edges**）: edge weight
   1. positive effect size: solid edge
   2. negative effect size: dashed edge
4. Outcome is highlighted in red

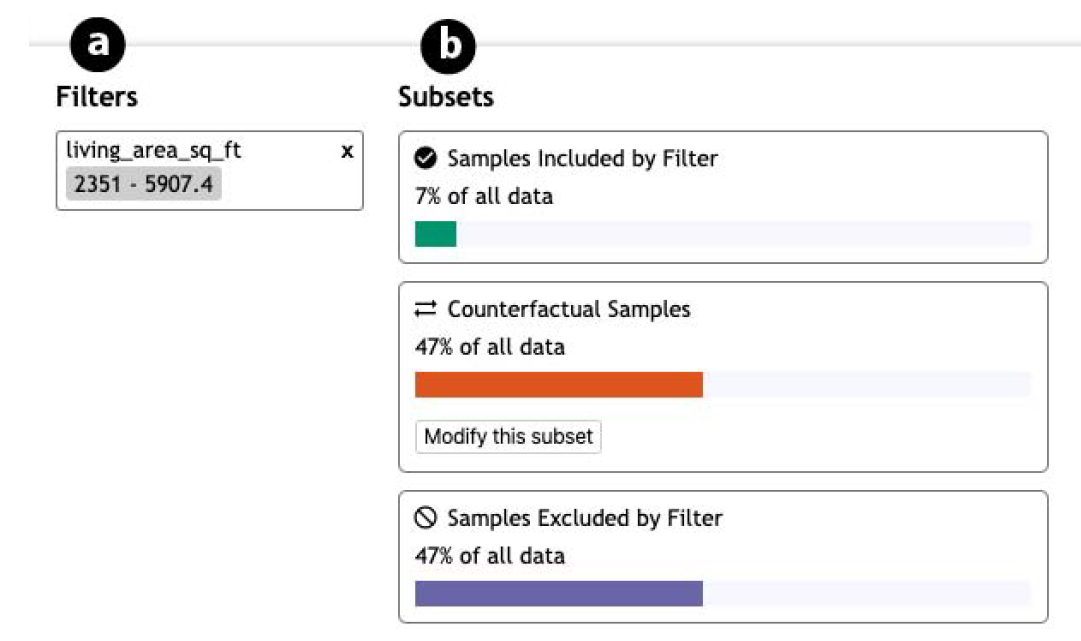


**3. counterfactual approach to identify potential confounding**

***[ S. Kaul, D. Borland, N. Cao and D. Gotz, "Improving Visualization Interpretation Using Counterfactuals," in*IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 1, pp. 998-1008, Jan. 2022, doi: 10.1109/TVCG.2021.3114779.]***

* Applying a filter constraint (f) based on one of the features selected in step 1 and create three subsets----the included subset (IN)、counterfactual subset (CF)、excluded subset (EX)

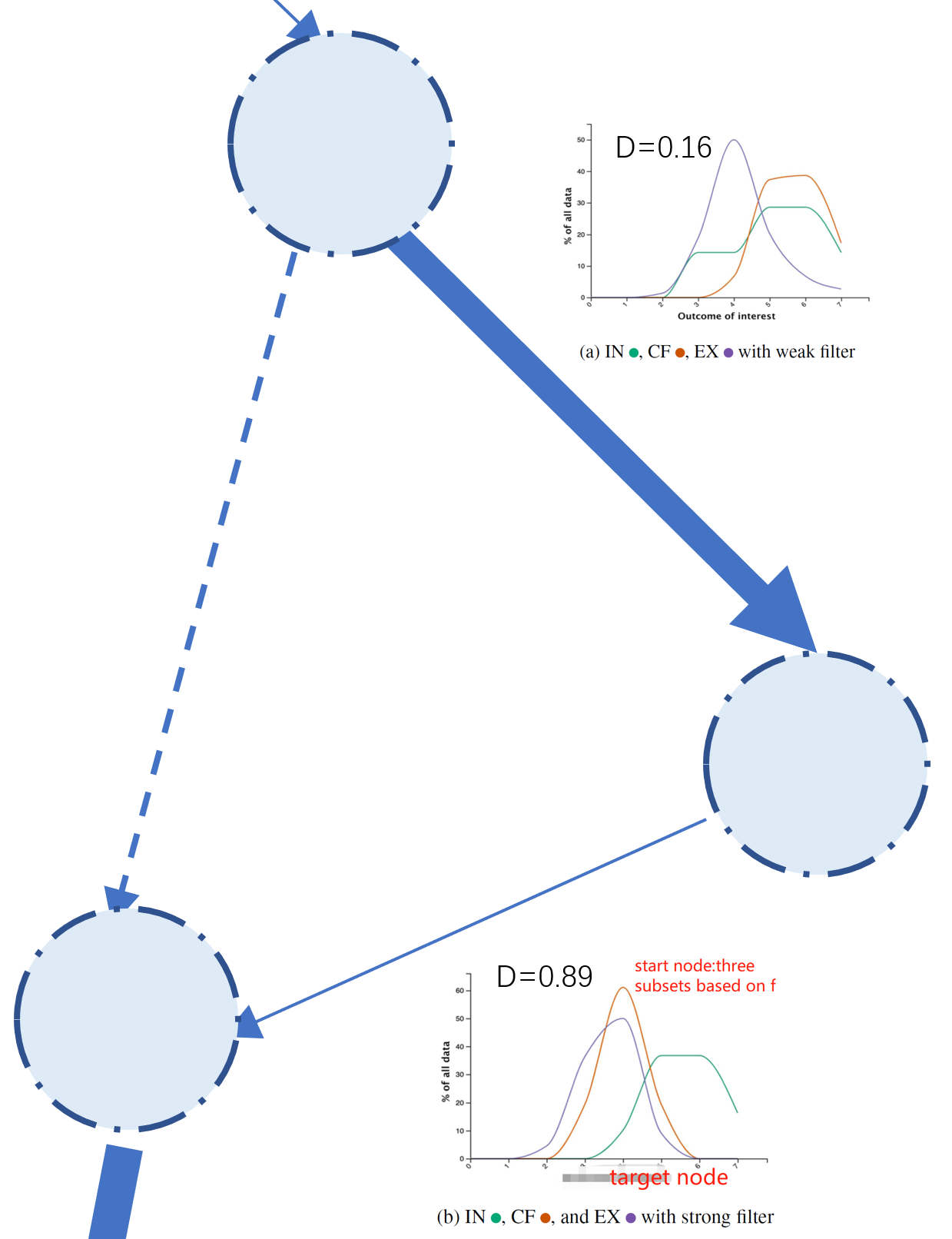




* Observe impact on the **target node** distributions of the IN, CF, and EX subsets of the **start node** ---- OIN, OCF, OEX

Compute the value of OIN/OCF (d) to measure the difference between OIN and OCF,

D ranks between 0-1, **0 corresponds to no impact by f (i.e., OIN = OCF) [likely to be a confounding variable]** and 1 corresponds to a large impact (i.e., a maximal difference between OIN and OCF).



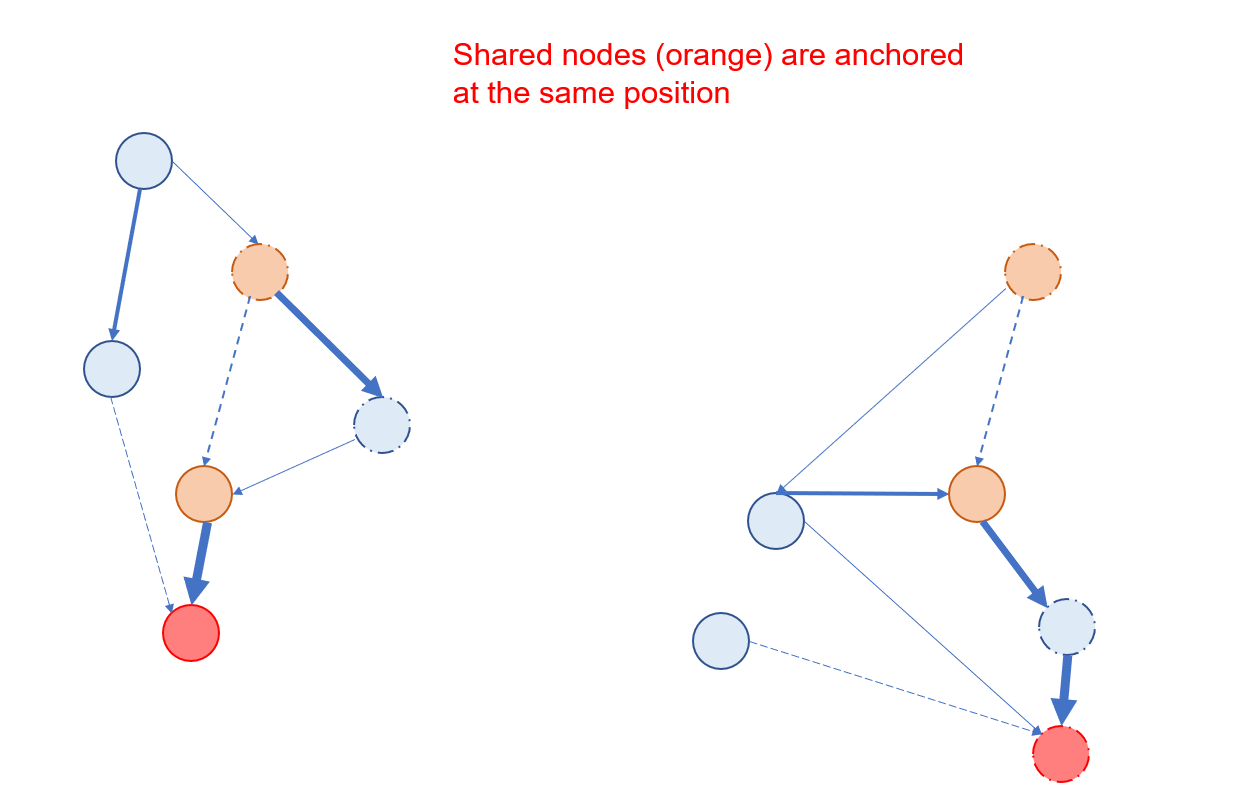
**4. manually delete edges based on domain knowledge and potential confounding detected in step 3**

supporting real-time update → save specific single graph to history

**multiple causal graphs**

**method 1:**

juxtaposition ---- anchor and highlight the same nodes in different groups.



**method 2:** merge same nodes

