Algorithm 2: FairPFN Synthetic Data Generation

Input:
- Number of exogenous causes U
- Number of endogenous variables $U \times H$
- Number of features and samples $M \times N$

begin
- Define MLP ϕ with depth H and width U
- Initialize random weights $W : (U \times U \times H)$
- Sample sparsity masks P with same dimensionality as weights
- Sample H per-layer non-linearities $z_i \sim \{Identity, ReLU, Tanh\}$
- Initialize output matrix $X : (U \times H)$
- Sample location k of protected attribute in X_0
- Sample locations of features X_{biased} in $X_{1:H-1}$, and outcome y_{bias} in X_H
- Sample protected attribute threshold a_t and binary values $\{a_0, a_1\}$
- Sample output threshold y_t

for $n = 1$ **to** N samples **do**
- Sample values of exogenous causes $X_0 : (U \times 1)$
- Sample values of additive noise terms $\epsilon : (U \times H)$

for $i = 1$ **to** H layers **do**
- Pass intermediate representation through hidden layer $X_{i+1} = z_i(P_i \cdot W_i^T X_i + \epsilon_i)$

end for
- Select prot. attr. A, features X_{bias} and outcome y_{bias} from X_0, $X_{1:H-1}$, and X_H
- Binarize $A \in \{a_0, a_1\}$ and $y_{bias} \in \{0, 1\}$ over threshold a_t and y_t
- Set input weights in row k of W_0 to 0

for $i = 1$ **to** L layers **do**
- Pass intermediate representation through hidden layer $X_{i+1} = z_i(P_i \cdot W_i^T X_i + \epsilon_i)$

end for
- Select the *fair* outcome y_{fair} from X_H
- Binarize $y_{fair} \in \{0, 1\}$ over threshold y_t

end for

Output: $D_{bias} = (A, X_{bias}, y_{bias})$ and y_{fair}