Solution 1:

Goal A/B/C. (A) train a final deployable model $\hat{f}$, (B) tune the graph learner (select $\hat{\boldsymbol{\lambda}} \in \tilde{\Lambda}$), (C) estimate generalization error for the tuned procedure.

1) **Resampling need + usable data fraction (per goal).**

   A) Train deployable final model $\hat{f}$

   (a) *No resampling* (just fit once with the chosen $\hat{\boldsymbol{\lambda}}$).

   (b) Use 100% of $\mathcal{D}$ for the final fit.

   B) Tune the graph learner (HPO / find $\hat{\boldsymbol{\lambda}} \in \tilde{\Lambda}$)

   (a) *Resampling* (e.g. CV) to estimate $c(\boldsymbol{\lambda}) = \widehat{GE}(I, J, \rho, \boldsymbol{\lambda})$ for each hyperparameter configuration (HPC) $\boldsymbol{\lambda}$.

   (b) With $K = 4$: train on 3/4, validate on 1/4 (across all folds, every observation is used).

   C) Estimate the generalization error (unbiased for the tuned procedure)

   (a) *Nested resampling* (outer loop for evaluation, inner loop for tuning) to satisfy the "untouched test set" principle and avoid overtuning bias.

   (b) With 3-fold outer CV, each outer iteration trains the tuned learner on 2/3 and tests on 1/3. Inside that, the 4-fold inner CV uses 3/4 of the outer-train part for training, hence $\frac{2}{3} \cdot \frac{3}{4} = \frac{1}{2}$ of $\mathcal{D}$ per inner-fold model fit.

2) **Order of goals.**

   A leakage-safe order is

   $$\boxed{\textbf{C (with B inside)} \;\rightarrow\; \textbf{B on full } \mathcal{D} \;\rightarrow\; \textbf{A}.}$$

   Rationale: performance estimation treats outer test data as "untouched"; hence tuning must be nested inside the outer evaluation. After obtaining an (approximately) unbiased $\widehat{GE}$, tune once on all data to get a single $\hat{\boldsymbol{\lambda}}$, then fit the final deployable $\hat{f}$.

3) **Pseudo-algorithm (nested CV + final tuning + final fit).**

   Subroutine: Tuning (4-fold CV grid search).

Algorithm 1 Tuning($\mathcal{D}'; I, \rho, \tilde{\Lambda}$)

Input: data $\mathcal{D}' = (X', y')$, inducer $I : \mathcal{D} \times \Lambda \to H$, performance measure ρ, search space $\tilde{\Lambda}$
Split $\mathcal{D}'$ into $(\mathcal{D}'^{(k)}_{\text{train}}, \mathcal{D}'^{(k)}_{\text{valid}})$, $k = 1, \ldots, 4$ (4-fold CV on $\mathcal{D}'$)
for each $\boldsymbol{\lambda} \in \tilde{\Lambda}$ **do**
  for $k \in \{1, 2, 3, 4\}$ **do**
    Train: $\hat{f}_k = I(\mathcal{D}'^{(k)}_{\text{train}}, \boldsymbol{\lambda})$
    Validate: $e_k \leftarrow \rho\Big(y'^{(k)}_{\text{valid}},\ \hat{f}_k(X'^{(k)}_{\text{valid}})\Big)$
  end for
  $c(\boldsymbol{\lambda}) \leftarrow \frac{1}{4} \sum_{k=1}^{4} e_k$
end for
Output: $\hat{\boldsymbol{\lambda}} \in \arg\min_{\boldsymbol{\lambda} \in \tilde{\Lambda}} c(\boldsymbol{\lambda})$

Main procedure: Nested CV for GE estimation, then one tuning on full data, then final fit.

Algorithm 2 Nested CV for GE estimation + final tuning + final fit

Input: data $\mathcal{D} = (X, y)$, inducer I, performance measure ρ, search space $\tilde{\Lambda}$
Outer resampling (3-fold CV): split $\mathcal{D}$ into $(D^{(b)}_{\text{train}}, D^{(b)}_{\text{test}})$, $b = 1, \dots, 3$
for $b \in \{1, 2, 3\}$ **do**
    Tune: $\hat{\boldsymbol{\lambda}}^{(b)} \leftarrow \text{Tuning}(D^{(b)}_{\text{train}}; I, \rho, \tilde{\Lambda})$ ▷ tuning subroutine for outer fold b
    Refit on all $D^{(b)}_{\text{train}}$: $\hat{f}_b = I(D^{(b)}_{\text{train}}, \hat{\boldsymbol{\lambda}}^{(b)})$
    Outer test evaluation: $e_b = \rho\Big(y^{(b)}_{\text{test}}, \hat{f}_b(X^{(b)}_{\text{test}})\Big)$
end for
Generalization error estimate (goal C with B inside): $\widehat{GE} = \frac{1}{3}\sum_{b=1}^{3} e_b$
Final tuning on full data (goal B): $\hat{\boldsymbol{\lambda}} \leftarrow \text{Tuning}(\mathcal{D}; I, \rho, \tilde{\Lambda})$ ▷ same tuning subroutine, applied to $\mathcal{D}$
Final fit on full data (goal A): $\hat{f} = I(\mathcal{D}, \hat{\boldsymbol{\lambda}})$

4) **Total number of model trainings (hierarchical "model choice").**

The intended configuration space is hierarchical:

$$\tilde{\Lambda} = \underbrace{\tilde{\Lambda}_{\text{NN}}}_{5 \text{ choices}} \cup \underbrace{\tilde{\Lambda}_{\text{RF}}}_{16 \text{ choices}}, \qquad |\tilde{\Lambda}| = 5 + 16 = 21,$$

because an HPC corresponds to *either* NN with chosen #layers *or* RF with chosen $(\text{ntree}, \text{depth})$.

Nested part (outer 3-fold, inner 4-fold). Per outer fold:

$$|\tilde{\Lambda}| \cdot 4 \; + \; 1 \; = \; 21 \cdot 4 + 1 \; = \; 85.$$

Across all 3 outer folds:

$$3 \cdot 85 = 255.$$

Final tuning on full $\mathcal{D}$ (4-fold) + final fit.

$$21 \cdot 4 \; + \; 1 = 84 + 1 = 85.$$

Total trainings:

$$\boxed{255 + 84 + 1 \; = \; 340 \text{ model trainings.}}$$

(Each training fits exactly one arm, since the configuration selects either NN or RF.)