Branch-and-Bound with Predictions for Variable Selection

Yatharth Dubey

(University of Illinois at Urbana-Champaign)
Certifying bounds in pure binary ILP:

\[
\max \left\{ cx : x \in P \cap \{0,1\}^n \right\} \leq v^*
\]

where \( P = \{x \in [0,1]^n : Ax \leq b\}, A \in \mathbb{Z}^{m \times n}, b \in \mathbb{Z}^m \)
Certifying bounds in pure binary ILP:

\[
\max \left\{ cx : x \in P \cap \{0,1\}^n \right\} \leq v^* \]

where \( P = \{ x \in [0,1]^n : Ax \leq b \}, A \in \mathbb{Z}^{m \times n}, b \in \mathbb{Z}^m \)

Constructing a BB tree that certifies a bound is completely determined by the variable selection rule.
Certifying bounds in pure binary ILP:

$$\max \left\{ cx : x \in P \cap \{0,1\}^n \right\} \leq v^*$$

where $P = \{x \in [0,1]^n : Ax \leq b\}, A \in \mathbb{Z}^{m \times n}, b \in \mathbb{Z}^m$

Constructing a BB tree that certifies a bound is completely determined by the variable selection rule

**Importance:** total time = time per node × # of nodes
Certifying bounds in pure binary ILP:

\[
\max \left\{ cx : x \in P \cap \{0,1\}^n \right\} \leq v^*
\]

where \( P = \{x \in [0,1]^n : Ax \leq b\}, A \in \mathbb{Z}^{m \times n}, b \in \mathbb{Z}^m \)

Constructing a BB tree that certifies a bound is completely determined by the variable selection rule.

**Importance**: total time = time per node × # of nodes

Most infeasible: fast, too uninformed

Strong branching: informed, too costly per node
Certifying bounds in pure binary ILP:

\[
\max \{ cx : x \in P \cap \{0,1\}^n \} \leq \nu^* \\
\text{where } P = \{ x \in [0,1]^n : Ax \leq b \}, A \in \mathbb{Z}^{m \times n}, b \in \mathbb{Z}^m
\]

Constructing a BB tree that certifies a bound is completely determined by the \text{variable selection rule}

\text{Importance: total time} = \text{time per node} \times \# \text{ of nodes}

Most infeasible: fast, too uninformed

Strong branching: informed, too costly per node

\text{How can we do time-efficient and informed branching at each node?}
How can we do time-efficient and informed branching at each node?

A compromise—Reliability Branching: do strong branching early in the tree, until we have “enough information” about how each variable affects the objective; use this information through the rest of the tree.
How can we do time-efficient and informed branching at each node?

A compromise—**Reliability Branching**: do strong branching early in the tree, until we have “enough information” about how each variable affects the objective; use this information through the rest of the tree

A recent insight—**Machine learning**!
How can we do time-efficient and informed branching at each node?

A compromise—Reliability Branching: do strong branching early in the tree, until we have “enough information” about how each variable affects the objective; use this information through the rest of the tree.

A recent insight—Machine learning!

Existing literature on ML to branch: how to use strong branching as an expert to imitate.
How can we do time-efficient and informed branching at each node?

A compromise—**Reliability Branching**: do strong branching early in the tree, until we have “enough information” about how each variable affects the objective; use this information through the rest of the tree.

A recent insight—**Machine learning**!

Existing literature on ML to branch: **how to use strong branching as an expert to imitate** (The idea: obtain a fast approximation of strong branching scores/relative strong branching ranking).
How can we do time-efficient and informed branching at each node?

A compromise—Reliability Branching: do strong branching early in the tree, until we have “enough information” about how each variable affects the objective; use this information through the rest of the tree.

A recent insight—Machine learning!

Existing literature on ML to branch: how to use strong branching as an expert to imitate (The idea: obtain a fast approximation of strong branching scores/relative strong branching ranking)

But... is strong branching the expert we should be imitating?
Next, we give a framework through which we can think about this question.
**Strong Branching:** (below $v(S)$ is the LP optimal value of subproblem $S$)

At subproblem $S$ branch on $j^* = \arg \max_{j \in C \subset n} (v(S) - v(S_{j=0})) + (v(S) - v(S_{j=1}))$

\[
\Delta_j^- \quad \Delta_j^+
\]
Strong Branching: (below $v(S)$ is the LP optimal value of subproblem $S$)

At subproblem $S$ branch on $j^* = \arg\max_{j \in C \subset \mathbb{n}} (v(S) - v(S_{j=0})) + (v(S) - v(S_{j=1}))$

Equivalently, branch on $j^* = \arg\min_{j \in C \subset \mathbb{n}} v(S_{j=0}) + v(S_{j=1})$
Strong Branching: (below $v(S)$ is the LP optimal value of subproblem $S$)

At subproblem $S$ branch on $j^* = \arg\max_{j \in C \subset n} (v(S) - v(S_{j=0})) + (v(S) - v(S_{j=1}))$

Equivalently, branch on $j^* = \arg\min_{j \in C \subset n} v(S_{j=0}) + v(S_{j=1})$

i.e., it chooses the pair of subproblems greedily in terms of the objective
**Strong Branching:** (below $\nu(S)$ is the LP optimal value of subproblem $S$)

At subproblem $S$ branch on $j^* = \arg \max_{j \in C \subset n} (\nu(S) - \nu(S_{j=0})) + (\nu(S) - \nu(S_{j=1}))$

Equivalently, branch on $j^* = \arg \min_{j \in C \subset n} \nu(S_{j=0}) + \nu(S_{j=1})$

i.e., it chooses the pair of subproblems *greedily* in terms of the objective

But branch-and-bound actually admits an *optimal recurrence relation*:

$$\theta(S, \nu^*) = \min_{j \in [n]} \theta(S_{j=0}, \nu^*) + \theta(S_{j=1}, \nu^*)$$

where $\theta(S, \nu^*)$ is the size of the smallest BB tree certifying bound $\nu^*$ for subproblem $S$
**Strong Branching:** (below $v(S)$ is the LP optimal value of subproblem $S$)

At subproblem $S$ branch on $j^* = \arg \max_{j \in C \subseteq n} (v(S) - v(S_{j=0})) + (v(S) - v(S_{j=1}))$

Equivalently, branch on $j^* = \arg \min_{j \in C \subseteq n} v(S_{j=0}) + v(S_{j=1})$

i.e., it chooses the pair of subproblems *greedily* in terms of the objective

But branch-and-bound actually admits an *optimal recurrence relation*:

$$\theta(S, v^*) = \min_{j \in [n]} \theta(S_{j=0}, v^*) + \theta(S_{j=1}, v^*)$$

where $\theta(S, v^*)$ is the size of the smallest BB tree certifying bound $v^*$ for subproblem $S$

This suggests branching on $j^* = \arg \min_{j \in [n]} \theta(S_{j=0}, v^*) + \theta(S_{j=1}, v^*)$

(which would *obtain a BB tree of minimum size*)
**ESTIMATING THE OPTIMAL RULE**

**Optimal rule:** at subproblem $S$, branch on $j^* = \arg \min_{j \in [n]} \theta(S_{j=0}, v^*) + \theta(S_{j=1}, v^*)$

where $\theta(S, v^*)$ is the size of the smallest BB tree certifying bound $v^*$ for subproblem $S$
**Optimal rule:** at subproblem $S$, branch on $j^* = \arg \min_{j \in [n]} \theta(S_{j=0}, v^*) + \theta(S_{j=1}, v^*)$

where $\theta(S, v^*)$ is the size of the smallest BB tree certifying bound $v^*$ for subproblem $S$

**Challenge:** computing all necessary values of $\theta(S, v^*)$ is at least as hard as solving IP!
**Optimal rule:** at subproblem $S$, branch on $j^* = \arg \min_{j \in [n]} \theta(S_{j=0}, v^*) + \theta(S_{j=1}, v^*)$

where $\theta(S, v^*)$ is the size of the smallest BB tree certifying bound $v^*$ for subproblem $S$

**Challenge:** computing all necessary values of $\theta(S, v^*)$ is at least as hard as solving IP!

This motivates the need for an estimate $\hat{\theta}(S, v^*)$ of $\theta(S, v^*)$

Then, we can branch according to $\hat{\theta}(S, v^*)$: $j^* = \arg \min_{j \in [n]} \hat{\theta}(S_{j=0}, v^*) + \hat{\theta}(S_{j=1}, v^*)$
Optimal rule: at subproblem $S$, branch on $j^* = \arg\min_{j \in [n]} \theta(S_{j=0}, v^*) + \theta(S_{j=1}, v^*)$

where $\theta(S, v^*)$ is the size of the smallest BB tree certifying bound $v^*$ for subproblem $S$

Challenge: computing all necessary values of $\theta(S, v^*)$ is at least as hard as solving IP!

This motivates the need for an estimate $\hat{\theta}(S, v^*)$ of $\theta(S, v^*)$

Then, we can branch according to $\hat{\theta}(S, v^*)$: $j^* = \arg\min_{j \in [n]} \hat{\theta}(S_{j=0}, v^*) + \hat{\theta}(S_{j=1}, v^*)$

e.g., strong branching branches according to an estimate $\hat{\theta}_{gap}(S, v^*) = f(v(S) - v^*)$
ESTIMATING THE OPTIMAL RULE

Optimal rule: at subproblem $S$, branch on $j^* = \arg\min_{j \in [n]} \theta(S_{j=0}, v^*) + \theta(S_{j=1}, v^*)$

where $\theta(S, v^*)$ is the size of the smallest BB tree certifying bound $v^*$ for subproblem $S$

Challenge: computing all necessary values of $\theta(S, v^*)$ is at least as hard as solving IP!

This motivates the need for an estimate $\hat{\theta}(S, v^*)$ of $\theta(S, v^*)$

Then, we can branch according to $\hat{\theta}(S, v^*)$: $j^* = \arg\min_{j \in [n]} \hat{\theta}(S_{j=0}, v^*) + \hat{\theta}(S_{j=1}, v^*)$

e.g., strong branching branches according to an estimate $\hat{\theta}_{gap}(S, v^*) = f(v(S) - v^*)$

Question 1: How does the quality of the estimate affect the size of the resulting tree? If $\hat{\theta}(S, v^*) \approx \theta(S, v^*)$ will we get a near-minimum-size tree?
Optimal rule: at subproblem $S$, branch on $j^* = \arg\min_{j \in [n]} \theta(S_{j=0}, v^*) + \theta(S_{j=1}, v^*)$

where $\theta(S, v^*)$ is the size of the smallest BB tree certifying bound $v^*$ for subproblem $S$

Challenge: computing all necessary values of $\theta(S, v^*)$ is at least as hard as solving IP!

This motivates the need for an estimate $\hat{\theta}(S, v^*)$ of $\theta(S, v^*)$

Then, we can branch according to $\hat{\theta}(S, v^*)$: $j^* = \arg\min_{j \in [n]} \hat{\theta}(S_{j=0}, v^*) + \hat{\theta}(S_{j=1}, v^*)$

e.g., strong branching branches according to an estimate $\hat{\theta}_{gap}(S, v^*) = f(v(S) - v^*)$

**Question 1:** How does the quality of the estimate affect the size of the resulting tree? If $\hat{\theta}(S, v^*) \approx \theta(S, v^*)$ will we get a near-minimum-size tree?

**Question 2:** How can we get a good estimate $\hat{\theta}$? Not clear since obtaining samples with true supervised labels $\theta(S, v^*)$ is not computationally viable
We assume \( \theta(S, v^*) = \hat{\theta}(S, v^*) + r_{\hat{\theta}}(S, v^*) \) where \( r_{\hat{\theta}}(S, v^*) \sim N(0, \sigma^2) \).
We assume \( \theta(S, v^*) = \hat{\theta}(S, v^*) + r_{\hat{\theta}}(S, v^*) \) where \( r_{\hat{\theta}}(S, v^*) \sim N(0, \sigma^2) \)

Consider the following definition capturing the error of an estimate \( \hat{\theta} \)

\[
\epsilon_{\hat{\theta}}(S, v^*) = \theta(S_{j' = 0}) + \theta(S_{j' = 1}) - \min_{j \in [n]} \left[ \theta(S_{j = 0}) + \theta(S_{j = 1}) \right]
\]

where \( j' = \arg\min_{j \in [n]} \hat{\theta}(S_{j = 0}) + \hat{\theta}(S_{j = 1}) \)
We assume \( \theta(S, v^*) = \hat{\theta}(S, v^*) + r_{\hat{\theta}}(S, v^*) \) where \( r_{\hat{\theta}}(S, v^*) \sim N(0, \sigma^2) \)

Consider the following definition capturing the error of an estimate \( \hat{\theta} \)

\[
\epsilon_{\hat{\theta}}(S, v^*) = \theta(S_{j' = 0}) + \theta(S_{j' = 1}) - \min_{j \in [n]} \left[ \theta(S_{j = 0}) + \theta(S_{j = 1}) \right]
\]

where \( j' = \arg \min_{j \in [n]} \hat{\theta}(S_{j = 0}) + \hat{\theta}(S_{j = 1}) \)
We assume $\theta(S, v^*) = \hat{\theta}(S, v^*) + r_\theta(S, v^*)$ where $r_\theta(S, v^*) \sim N(0, \sigma^2)$

Consider the following definition capturing the error of an estimate $\hat{\theta}$

$$
\epsilon_\hat{\theta}(S, v^*) = \theta(S_{j' = 0}) + \theta(S_{j' = 1}) - \min_{j \in [n]} \left[ \theta(S_{j = 0}) + \theta(S_{j = 1}) \right]
$$

where $j' = \arg\min_{j \in [n]} \hat{\theta}(S_{j = 0}) + \hat{\theta}(S_{j = 1})$
We assume $\theta(S, v^*) = \hat{\theta}(S, v^*) + r_{\hat{\theta}}(S, v^*)$ where $r_{\hat{\theta}}(S, v^*) \sim N(0, \sigma^2)$

Consider the following definition capturing the error of an estimate $\hat{\theta}$

$$
\epsilon_{\hat{\theta}}(S, v^*) = \theta(S_{j^*=0}) + \theta(S_{j^*=1}) - \min_{j \in [n]} \left[ \theta(S_{j=0}) + \theta(S_{j=1}) \right]
$$

where $j^* = \arg \min_{j \in [n]} \hat{\theta}(S_{j=0}) + \hat{\theta}(S_{j=1})$
BETTER ESTIMATES MEAN SMALLER TREES
Proposition (D.)

Let \( \hat{\theta}_1, \hat{\theta}_2 \) be estimates such that \( r_{\hat{\theta}_1}(S, v^*) \sim N(0, \sigma_1^2) \) and \( r_{\hat{\theta}_2}(S, v^*) \sim N(0, \sigma_2^2) \) with \( \sigma_2 > \sigma_1 \). Then, \( \mathbb{E} \left[ \epsilon_{\hat{\theta}_2}(S, v^*) \right] > \mathbb{E} \left[ \epsilon_{\hat{\theta}_1}(S, v^*) \right] \).
Proposition (D.)
Let $\hat{\theta}_1, \hat{\theta}_2$ be estimates such that $r_{\hat{\theta}_1}(S, v^*) \sim N(0, \sigma_1^2)$ and $r_{\hat{\theta}_2}(S, v^*) \sim N(0, \sigma_2^2)$ with $\sigma_2 > \sigma_1$. Then, $\mathbb{E}[\epsilon_{\hat{\theta}_2}(S, v^*)] > \mathbb{E}[\epsilon_{\hat{\theta}_1}(S, v^*)]$.

Theorem (D.)
Let $\hat{\theta}$ be such that $\mathbb{E}[\epsilon_{\hat{\theta}}(S, v^*)] = \alpha \theta(S, v^*)$, where $\alpha \in [0,1]$, for all possible subproblems $S$, and let $\mathcal{T}_{\hat{\theta}}(P, v^*)$ be the BB tree certifying bound $v^*$ for the integer program $P$ that branches according to $\hat{\theta}$. Then, $\mathbb{E}[|\mathcal{T}_{\hat{\theta}}(P, v^*)|] = (1 + \alpha)^n \theta(P, v^*)$. 
We settle for an estimate of a more easily computable, accurate signal $\hat{\theta} \approx \theta_{\text{signal}} \approx \theta$.
We settle for an estimate of a more easily computable, accurate signal $\hat{\theta} \approx \theta_{signal} \approx \theta$

Recall that strong branching can be interpreted as branching according to a signal

$$\theta_{gap}(S, v^*) = f(v(S) - v^*)$$ where $f$ is any positive monotone function.
We settle for an estimate of a more easily computable, accurate signal \( \hat{\theta} \approx \theta_{\text{signal}} \approx \theta \).

Recall that strong branching can be interpreted as branching according to a signal

\[
\theta_{\text{gap}}(S, v^*) = f(v(S) - v^*) \quad \text{where} \ f \ \text{is any positive monotone function}
\]

\[
r_{\theta_{\text{gap}}}(S, v^*) \sim N(0, 0.4018^2)
\]
We settle for an estimate of a more easily computable, accurate signal \( \hat{\theta} \approx \theta_{\text{signal}} \approx \theta \).

Recall that strong branching can be interpreted as branching according to a signal

\[
\theta_{\text{gap}}(S, v^*) = f(v(S) - v^*) \quad \text{where} \quad f \text{ is any positive monotone function}
\]

We propose two other signals:

\[
\theta_{\text{mostinf}}(S, v^*) = f( | \mathcal{T}_{\text{mostinf}}(S, v^*) | )
\]

\[
\theta_{sb}(S, v^*) = f( | \mathcal{T}_{sb}(S, v^*) | )
\]

\[r_{\theta_{\text{gap}}}(S, v^*) \sim N(0, 0.4018^2)\]
We settle for an estimate of a more easily computable, accurate signal \( \hat{\theta} \approx \theta_{signal} \approx \theta \).

Recall that strong branching can be interpreted as branching according to a signal
\[
\theta_{gap}(S, v^*) = f(v(S) - v^*) \quad \text{where } f \text{ is any positive monotone function}
\]

\[
r_{\theta_{gap}}(S, v^*) \sim N(0, 0.4018^2)
\]

We propose two other signals:
\[
\theta_{mostinf}(S, v^*) = f(|\mathcal{T}_{mostinf}(S, v^*)|)
\]
\[
\theta_{sb}(S, v^*) = f(|\mathcal{T}_{sb}(S, v^*)|)
\]

Disclaimer: This data comes from random multi-dimensional knapsack problems, where strong branching is known to struggle.
IMPERFECT, BUT GOOD SIGNALS

\[ \log \theta(S, v^*) \sim N(0, 0.4018^2) \]

\[ r_{\theta_{\text{gap}}}(S, v^*) \sim N(0, 0.1842^2) \]

\[ r_{\theta_{\text{mostinf}}}(S, v^*) \sim N(0, 0.1286^2) \]

\[ r_{\theta_{\text{sb}}}(S, v^*) \sim N(0, 0.1286^2) \]
The theory tells us BB trees branching according to these stronger signals should produce smaller trees than those produced by strong branching.
The theory tells us BB trees branching according to these stronger signals should produce smaller trees than those produced by strong branching.

E.g., branching according to $\theta_{mostinf}$:

At subproblem $S$, branch on the variable $\arg\min_{j \in [n]} | \mathcal{T}_{mostinf}(S_j=0, v^*) | + | \mathcal{T}_{mostinf}(S_j=1, v^*) |$
Indeed, we see that the BB trees branching according to these stronger signals are significantly smaller than those produced by strong branching.

<table>
<thead>
<tr>
<th></th>
<th>geomean +/- geostd</th>
</tr>
</thead>
<tbody>
<tr>
<td>(</td>
<td>\mathcal{F}_{\text{strong}}</td>
</tr>
<tr>
<td>(</td>
<td>\mathcal{F}<em>{\theta</em>{\text{mostinf}}}</td>
</tr>
<tr>
<td>(</td>
<td>\mathcal{F}<em>{\theta</em>{sb}}</td>
</tr>
</tbody>
</table>

**RELATIVE RANKING — FREQUENCY**

![Graphs showing relative ranking — frequency for different sets of trees.](image)
BB trees branching according to these stronger signals are significantly smaller than those produced by strong branching even when strong branching is excellent.

**RELATIVE RANKING — FREQUENCY**

**Data:**
Randomly generated max stable set problems on Albert-Barabasi graphs (100 nodes, affinity=8)
Clique cover relaxation

<table>
<thead>
<tr>
<th></th>
<th>Geomean +/- Geostd</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>\mathcal{T}_{\text{mostinf}}</td>
</tr>
<tr>
<td>$</td>
<td>\mathcal{T}_{\text{strong}}</td>
</tr>
<tr>
<td>$</td>
<td>\mathcal{T}<em>{\theta</em>{\text{mostinf}}}</td>
</tr>
<tr>
<td>$</td>
<td>\mathcal{T}<em>{\theta</em>{\text{reliability}}}</td>
</tr>
</tbody>
</table>
RECAP, SO FAR

Most of the successful ML for BB research aims to approximate (with ML) the signal \( \theta_{gap} \) with some learned estimate \( \hat{\theta}_{gap} \).
Most of the successful ML for BB research aims to approximate (with ML) the signal $\theta_{\text{gap}}$ with some learned estimate $\hat{\theta}_{\text{gap}}$.

Preliminary experiments show that $\theta_{\text{gap}}$ can actually be a fairly noisy signal (i.e., with significant variance from $\theta$).
Most of the successful ML for BB research aims to approximate (with ML) the signal $\theta_{gap}$ with some learned estimate $\hat{\theta}_{gap}$.

Preliminary experiments show that $\theta_{gap}$ can actually be a fairly noisy signal (i.e., with significant variance from $\theta$).

We propose the estimation of a signal that better approximates $\theta$, e.g., we can get realizations of the signals $\theta_{reliability}$ from previous solves using reliability branching.
Most of the successful ML for BB research aims to approximate (with ML) the signal \( \theta_{\text{gap}} \) with some learned estimate \( \hat{\theta}_{\text{gap}} \).

Preliminary experiments show that \( \theta_{\text{gap}} \) can actually be a fairly noisy signal (i.e., with significant variance from \( \theta \)).

We propose the estimation of a signal that better approximates \( \theta \), e.g., we can get realizations of the signals \( \theta_{\text{reliability}} \) from previous solves using reliability branching.
Most of the successful ML for BB research aims to approximate (with ML) the signal \( \theta_{\text{gap}} \) with some learned estimate \( \hat{\theta}_{\text{gap}} \).

Preliminary experiments show that \( \theta_{\text{gap}} \) can actually be a fairly noisy signal (i.e., with significant variance from \( \theta \)).

We propose the estimation of a signal that better approximates \( \theta \), e.g., we can get realizations of the signals \( \theta_{\text{reliability}} \) from previous solves using reliability branching.

\[
(\Phi(S_1, v_1^*) - \theta_{\text{rule}}(S_1, v_1^*)) \quad (\Phi(S_2, v_2^*) - \theta_{\text{rule}}(S_2, v_2^*))
\]

\[
(\Phi(S_N, v_N^*) - \theta_{\text{rule}}(S_N, v_N^*))
\]
\[ \hat{\theta}(S, v^*) = \beta_{\text{gap}}(f(v(S) - v*)) \]

+ \( \beta_{\text{frac}} \) (fractionality of optimal LP solution)

+ \( \beta_{\text{dual}} \) (dual information)
## BRANCHING ACCORDING TO ESTIMATES

<table>
<thead>
<tr>
<th>$\mathcal{F}_{\text{strong}}$</th>
<th>geomean +/- geostd</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>\mathcal{F}_{\text{strong}}</td>
</tr>
<tr>
<td>$</td>
<td>\mathcal{F}_{\hat{\theta}}</td>
</tr>
<tr>
<td>% won</td>
<td></td>
</tr>
<tr>
<td>$</td>
<td>\mathcal{F}_{\text{strong}}</td>
</tr>
<tr>
<td>$</td>
<td>\mathcal{F}_{\hat{\theta}}</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$\mathcal{F}_{\text{strong}}$</th>
<th>geomean +/- geostd</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>\mathcal{F}_{\text{strong}}</td>
</tr>
<tr>
<td>$</td>
<td>\mathcal{F}_{\hat{\theta}}</td>
</tr>
<tr>
<td>% won</td>
<td></td>
</tr>
<tr>
<td>$</td>
<td>\mathcal{F}_{\text{strong}}</td>
</tr>
<tr>
<td>$</td>
<td>\mathcal{F}_{\hat{\theta}}</td>
</tr>
<tr>
<td>(</td>
<td>\mathcal{T}_{\text{strong}}</td>
</tr>
<tr>
<td>(</td>
<td>\mathcal{T}_{\hat{\theta}}</td>
</tr>
<tr>
<td>(</td>
<td>\mathcal{T}_{\text{strong}}</td>
</tr>
<tr>
<td>(</td>
<td>\mathcal{T}_{\hat{\theta}}</td>
</tr>
</tbody>
</table>

|\(|\mathcal{T}_{\text{strong}}|\) | \(18.44 \pm 2.08\) |
|\(|\mathcal{T}_{\hat{\theta}}|\) | \(17.82 \pm 2.21\) |
|\(|\mathcal{T}_{\text{strong}}|\) | \(32\%\) |
|\(|\mathcal{T}_{\hat{\theta}}|\) | \(51\%\) |