Algorithm 1 CAPO: Context-Aware Prompt Optimization

Require: Dataset $\mathcal{D} = \{(X_i, y_i)\}_{i=1}^{n}$, Meta-LLM $\Phi(x)$, Downstream LLM $\phi(x)$, Cost function $\ell(y, \hat{y})$, Initial prompts ν, Population size p, Block size b, Number of iterations n

Divide dataset $\mathcal{D}$ into blocks $\mathcal{B} = \{B_1, ..., B_k\}$ where $|B_i| = b$
Create $d \in \mathbb{R}^x$ for fewshot examples
for $i = 1$ to n **do**
    $P \leftarrow \text{random\_selection}(\nu)$
    $O \leftarrow \text{cross\_over}(P)$
    $\nu \leftarrow \nu.\text{update}(O)$
    $\nu \leftarrow \text{mutate}(\nu)$
    prepend few-shot examples to ν
    $\nu \leftarrow \text{do\_racing}(\nu, \text{top\_k} = p)$
end for
best_prompt $\leftarrow \text{do\_racing}(\nu, \text{top\_k} = 1)$
return best_prompt

Algorithm 2 do_racing

Require: Prompts ν, Top-k p, cost function $\ell(y, \hat{y})$, blocks $\mathcal{B}$, Downstream LLM $\phi(x)$

survivors $\leftarrow \nu$
$i \leftarrow 0$
scores $\leftarrow [0] * \text{len}(\nu)$
shuffle $\mathcal{B}$
while len(survivors) > top_k $\wedge$ $i <$ len($\mathcal{B}$) **do**
    $i \leftarrow i + 1$
    scores $\leftarrow \frac{1}{i}$ (evaluate(ν, B_i) + $(i-1) *$ scores)
    survivors $\leftarrow$ racing_elimination(scores, $i * b, \alpha, k$)
end while
return survivors

Algorithm 3 Confidence-based Racing Elimination

Require: Prompts P, scores S, number of evaluations a, confidence level α, total blocks B, top k

$z_\alpha \leftarrow \Phi^{-1}(1 - \alpha/2)$ ▷ Critical value
survivors $\leftarrow P$
for $p_i \in P$ **do**
    if $\sum_{j \neq i} \mathbf{1}_{\left((s_j - s_i)\sqrt{a} > z_\alpha\right)} \geq k$ **then** ▷ # prompts better than p_i exceeds k
        survivors $\leftarrow$ survivors $\setminus \{p_i\}$ ▷ Eliminate p_i
    end if
end for
return survivors