

Appendix: Mathematical Proofs for Convergence Analysis

A PROOF OF CONVERGENCE FOR MCTS ACTION AND EXPANSION NODES

A.1 Mathematical Framework and Notation

A.1.1 Problem Definition.

DEFINITION 1 (MCTS POLICY TREE STRUCTURE). Let \mathcal{T} be a policy tree containing:

- **Action nodes** \hat{v} : Make binary decisions $a \in \{0, 1\}$ where $a = 0$ means terminate, $a = 1$ means continue.
- **Expansion nodes** \bar{v} : Determine allocation ratios $\mathbf{r} = (r_1, \dots, r_k) \in \Delta^k$ where $\Delta^k = \{x \in [0, 1]^k : \sum_{i=1}^k x_i = 1\}$.
- **Data nodes** v' : Terminal nodes containing trajectory sets.

A.1.2 Reward Structure.

DEFINITION 2 (REWARD PROCESS). At time step t , node v receives reward $R_v(t) \in [0, R_{\max}]$ where $R_{\max} > 0$ is a known upper bound. The empirical average reward after n visits is:

$$\hat{\mu}_v(n) = \frac{1}{n} \sum_{t=1}^n R_v(t) \quad (1)$$

The true expected reward is $\mu_v^* = \mathbb{E}[R_v(t)]$.

A.2 Fundamental Concentration Results

LEMMA 1 (HOEFFDING'S INEQUALITY FOR BOUNDED REWARDS). Let R_1, R_2, \dots, R_n be independent random variables with $R_i \in [a, b]$. Let $\bar{R}_n = \frac{1}{n} \sum_{i=1}^n R_i$ and $\mu = \mathbb{E}[R_1]$. Then for any $\epsilon > 0$:

$$\mathbb{P}(|\bar{R}_n - \mu| \geq \epsilon) \leq 2 \exp\left(-\frac{2n\epsilon^2}{(b-a)^2}\right) \quad (2)$$

PROOF. This follows directly from Hoeffding (1963)[2]. The proof uses the method of bounded differences and exponential moment generating functions. \square

A.3 Action Node Convergence Analysis

A.3.1 Action Selection Mechanism.

DEFINITION 3 (ACTION NODE DECISION RULE). At action node \hat{v} with children expansion node \bar{v} (action $a = 1$) and data node v' (action $a = 0$), the selection rule chooses:

$$a_{\text{selected}} = \operatorname{argmax}_{a \in \{0,1\}} \{\hat{\mu}_a(n_a)\} \quad (3)$$

where n_a is the visit count for action a .

DEFINITION 4 (OPTIMAL ACTION). The theoretically optimal action at node \hat{v} is:

$$a^* = \operatorname{argmax}_{a \in \{0,1\}} \{\mu_a^*\} \quad (4)$$

where μ_a^* is the true expected reward for taking action a .

A.3.2 Step-by-Step Convergence Proof for Action Nodes.

THEOREM 1 (ACTION NODE CONVERGENCE). Let \hat{v} be an action node and suppose action 1 is strictly better, i.e.,

$$\mu_1^* > \mu_0^* + \Delta$$

for some $\Delta > 0$. After n total visits to node \hat{v} , let n_0, n_1 be the visit counts for actions 0 and 1 respectively ($n_0 + n_1 = n$). If for each action $a \in \{0, 1\}$ we have

$$n_a \geq \frac{8R_{\max}^2 \ln(4/\delta)}{\Delta^2}, \quad (5)$$

then with probability at least $1 - \delta$, the empirically best action is the correct one, i.e. $a^* = 1$.

PROOF. **Step 1: Concentration inequality.** By Hoeffding's inequality for bounded rewards in $[0, R_{\max}]$, for any $a \in \{0, 1\}$:

$$\mathbb{P}(|\hat{\mu}_a(n_a) - \mu_a^*| \geq \epsilon) \leq 2 \exp\left(-\frac{2n_a\epsilon^2}{R_{\max}^2}\right). \quad (6)$$

Step 2: Choice of ϵ . Set $\epsilon = \Delta/4$. Then

$$\mathbb{P}\left(|\hat{\mu}_a(n_a) - \mu_a^*| \geq \frac{\Delta}{4}\right) \leq 2 \exp\left(-\frac{n_a\Delta^2}{8R_{\max}^2}\right). \quad (7)$$

Step 3: Confidence level. Require the Right-Hand Side $\leq \delta/2$, i.e.

$$2 \exp\left(-\frac{n_a\Delta^2}{8R_{\max}^2}\right) \leq \frac{\delta}{2}. \quad (8)$$

Taking logarithms gives

$$n_a \geq \frac{8R_{\max}^2 \ln(4/\delta)}{\Delta^2}. \quad (9)$$

Step 4: Union bound. With probability at least $1 - \delta$, both inequalities

$$|\hat{\mu}_0 - \mu_0^*| \leq \frac{\Delta}{4}, \quad |\hat{\mu}_1 - \mu_1^*| \leq \frac{\Delta}{4}$$

hold simultaneously.

Step 5: Correct action selection. Under this high-probability event,

$$\hat{\mu}_1 - \hat{\mu}_0 \geq (\mu_1^* - \frac{\Delta}{4}) - (\mu_0^* + \frac{\Delta}{4}) \quad (10)$$

$$= (\mu_1^* - \mu_0^*) - \frac{\Delta}{2} \quad (11)$$

$$\geq \Delta - \frac{\Delta}{2} = \frac{\Delta}{2} > 0, \quad (12)$$

so action 1 is strictly preferred. This completes the proof. \square

A.4 Expansion Node Convergence Analysis

THEOREM 2 (ALLOCATION RATIO CONVERGENCE). Consider an expansion node \bar{v} with k children. Let $\hat{\mu}_i(n_i)$ denote the empirical mean reward for child i after n_i visits, and let μ_i^* denote the true expected reward. Define the allocation ratios as:

$$r_i(n) = \frac{\hat{\mu}_i(n_i)}{\sum_{j=1}^k \hat{\mu}_j(n_j)}, \quad r_i^* = \frac{\mu_i^*}{\sum_{j=1}^k \mu_j^*}$$

Assume:

(1) Rewards are bounded: $R_i \in [0, R_{\max}]$ for all i .

(2) $\mu_{\min}^* := \min_{i=1,\dots,k} \mu_i^* > 0$.

$$(3) \mu_{\text{sum}}^* := \sum_{j=1}^k \mu_j^* > 0.$$

For any $\epsilon \in (0, 1)$ and $\delta \in (0, 1)$, if each child is visited at least $n_{\min} = \frac{8k^2 R_{\max}^2 \ln(2k/\delta)}{\epsilon^2 (\mu_{\text{sum}}^*)^2}$ times, then: $\mathbb{P}(\max_{i=1,\dots,k} |r_i(n) - r_i^*| \leq \epsilon) \geq 1 - \delta$.

PROOF. Step 1: Establish Concentration for Individual Empirical Means

Similar to Lemma 1, we begin by applying Hoeffding's inequality to bound deviations of empirical means.

Application to our setting: Since rewards are bounded in $[0, R_{\max}]$, we have $b_i - a_i = R_{\max}$ for all observations. Therefore, for child i with n_i visits:

$$\mathbb{P}(|\hat{\mu}_i(n_i) - \mu_i^*| \geq \epsilon_0) \leq 2 \exp\left(-\frac{2n_i \epsilon_0^2}{R_{\max}^2}\right)$$

Step 2: Apply Union Bound for Uniform Concentration

We need all empirical means to be close to their true values simultaneously.

Define the event $\mathcal{E}: \mathcal{E} = \{\max_{i=1,\dots,k} |\hat{\mu}_i(n_i) - \mu_i^*| \leq \epsilon_0\}$

By the union bound: $\mathbb{P}(\mathcal{E}^c) = \mathbb{P}\left(\bigcup_{i=1}^k \{|\hat{\mu}_i(n_i) - \mu_i^*| > \epsilon_0\}\right) \leq \sum_{i=1}^k \mathbb{P}(|\hat{\mu}_i(n_i) - \mu_i^*| > \epsilon_0)$

Applying Hoeffding's inequality to each term:

$$\mathbb{P}(\mathcal{E}^c) \leq \sum_{i=1}^k 2 \exp\left(-\frac{2n_i \epsilon_0^2}{R_{\max}^2}\right) \leq 2k \exp\left(-\frac{2n_{\min} \epsilon_0^2}{R_{\max}^2}\right)$$

Step 3: Choose Parameters to Achieve Target Probability

To ensure $\mathbb{P}(\mathcal{E}) \geq 1 - \delta$, we require: $2k \exp\left(-\frac{2n_{\min} \epsilon_0^2}{R_{\max}^2}\right) \leq \delta$

Taking logarithms: $-\frac{2n_{\min} \epsilon_0^2}{R_{\max}^2} \leq \ln\left(\frac{\delta}{2k}\right) = -\ln\left(\frac{2k}{\delta}\right)$

Therefore: $n_{\min} \geq \frac{R_{\max}^2 \ln(2k/\delta)}{2\epsilon_0^2}$

Step 4: Analyze Allocation Ratios Under Concentration Event

Assume event \mathcal{E} holds. Then for all $i \in \{1, \dots, k\}$: $\mu_i^* - \epsilon_0 \leq \hat{\mu}_i(n_i) \leq \mu_i^* + \epsilon_0$

Step 4a: Bound the numerator $|\hat{\mu}_i(n_i) - \mu_i^*| \leq \epsilon_0$

Step 4b: Bound the denominator from below $\sum_{j=1}^k \hat{\mu}_j(n_j) \geq \sum_{j=1}^k (\mu_j^* - \epsilon_0) = \sum_{j=1}^k \mu_j^* - k\epsilon_0 = \mu_{\text{sum}}^* - k\epsilon_0$

Step 4c: Bound the denominator from above $\sum_{j=1}^k \hat{\mu}_j(n_j) \leq \sum_{j=1}^k (\mu_j^* + \epsilon_0) = \mu_{\text{sum}}^* + k\epsilon_0$

Step 5: Derive Error Bound for Allocation Ratios

For any child i , we need to bound: $|r_i(n) - r_i^*| = \left| \frac{\hat{\mu}_i(n_i)}{\sum_{j=1}^k \hat{\mu}_j(n_j)} - \frac{\mu_i^*}{\mu_{\text{sum}}^*} \right|$

Step 5a: Common denominator manipulation $r_i(n) - r_i^* =$

$$\frac{\hat{\mu}_i(n_i)\mu_{\text{sum}}^* - \mu_i^* \sum_{j=1}^k \hat{\mu}_j(n_j)}{\mu_{\text{sum}}^* \sum_{j=1}^k \hat{\mu}_j(n_j)}$$

Step 5b: Bound the numerator

$$\begin{aligned} & \left| \hat{\mu}_i(n_i)\mu_{\text{sum}}^* - \mu_i^* \sum_{j=1}^k \hat{\mu}_j(n_j) \right| \\ &= \left| \hat{\mu}_i(n_i)\mu_{\text{sum}}^* - \mu_i^* \mu_{\text{sum}}^* + \mu_i^* \mu_{\text{sum}}^* - \mu_i^* \sum_{j=1}^k \hat{\mu}_j(n_j) \right| \\ &= \left| (\hat{\mu}_i(n_i) - \mu_i^*)\mu_{\text{sum}}^* + \mu_i^* \left(\mu_{\text{sum}}^* - \sum_{j=1}^k \hat{\mu}_j(n_j) \right) \right| \end{aligned}$$

By triangle inequality:

$$\begin{aligned} & \leq |\hat{\mu}_i(n_i) - \mu_i^*| \cdot \mu_{\text{sum}}^* + \mu_i^* \left| \mu_{\text{sum}}^* - \sum_{j=1}^k \hat{\mu}_j(n_j) \right| \\ &\leq \epsilon_0 \mu_{\text{sum}}^* + \mu_i^* \left| \sum_{j=1}^k (\mu_j^* - \hat{\mu}_j(n_j)) \right| \\ &\leq \epsilon_0 \mu_{\text{sum}}^* + \mu_i^* \sum_{j=1}^k |\mu_j^* - \hat{\mu}_j(n_j)| \\ &\leq \epsilon_0 \mu_{\text{sum}}^* + \mu_i^* k \epsilon_0 \\ &= \epsilon_0 (\mu_{\text{sum}}^* + \mu_i^* k) \end{aligned}$$

Step 5c: Bound the denominator from below

Under event $\mathcal{E}: \sum_{j=1}^k \hat{\mu}_j(n_j) \geq \mu_{\text{sum}}^* - k\epsilon_0$

Step 6: Combine Bounds to Get Final Error Estimate

$$|r_i(n) - r_i^*| \leq \frac{\epsilon_0 (\mu_{\text{sum}}^* + \mu_i^* k)}{\mu_{\text{sum}}^* (\mu_{\text{sum}}^* - k\epsilon_0)}$$

Since $\mu_i^* \leq \mu_{\text{sum}}^*$: $|r_i(n) - r_i^*| \leq \frac{\epsilon_0 (\mu_{\text{sum}}^* + \mu_{\text{sum}}^* k)}{\mu_{\text{sum}}^* (\mu_{\text{sum}}^* - k\epsilon_0)} = \frac{\epsilon_0 (1+k)}{\mu_{\text{sum}}^* - k\epsilon_0}$

Step 7: Choose ϵ_0 to Achieve Target Accuracy

To ensure $|r_i(n) - r_i^*| \leq \epsilon$, we need: $\frac{\epsilon_0 (1+k)}{\mu_{\text{sum}}^* - k\epsilon_0} \leq \epsilon$

Step 7a: Solve for ϵ_0 $\epsilon_0 (1+k) \leq \epsilon (\mu_{\text{sum}}^* - k\epsilon_0) \epsilon_0 (1+k) \leq \epsilon \mu_{\text{sum}}^* - \epsilon k \epsilon_0 \epsilon_0 (1+k+\epsilon k) \leq \epsilon \mu_{\text{sum}}^* \epsilon_0 \leq \frac{\epsilon \mu_{\text{sum}}^*}{1+k+\epsilon k}$

Step 7b: Conservative choice For $\epsilon \leq 1$ and $k \geq 1$, we have $1+k+\epsilon k \leq k+k+k = 3k$. Therefore, it suffices to choose: $\epsilon_0 = \frac{\epsilon \mu_{\text{sum}}^*}{4k}$

Step 8: Final Sample Complexity

Substituting our choice of ϵ_0 into the sample complexity bound from Step 3:

$$\begin{aligned} n_{\min} &\geq \frac{R_{\max}^2 \ln(2k/\delta)}{2\epsilon_0^2} = \frac{R_{\max}^2 \ln(2k/\delta)}{2 \cdot \left(\frac{\epsilon \mu_{\text{sum}}^*}{4k}\right)^2} \\ &= \frac{R_{\max}^2 \ln(2k/\delta)}{2 \cdot \frac{\epsilon^2 (\mu_{\text{sum}}^*)^2}{16k^2}} = \frac{16k^2 R_{\max}^2 \ln(2k/\delta)}{2\epsilon^2 (\mu_{\text{sum}}^*)^2} \\ &= \frac{8k^2 R_{\max}^2 \ln(2k/\delta)}{\epsilon^2 (\mu_{\text{sum}}^*)^2} \end{aligned}$$

Step 9: Conclusion

We have shown that under event \mathcal{E} , which occurs with probability at least $1 - \delta$ when n_{\min} satisfies our bound, we have: $\max_{i=1,\dots,k} |r_i(n) - r_i^*| \leq \epsilon$

This completes the proof. \square

A.5 Sample Complexity Results

COROLLARY 1 (ACTION NODE SAMPLE COMPLEXITY). For action node convergence with gap $\Delta > 0$ and confidence $1 - \delta$, the required sample complexity is:

$$n = O\left(\frac{R_{\max}^2 \ln(1/\delta)}{\Delta^2}\right) \quad (13)$$

COROLLARY 2 (EXPANSION NODE SAMPLE COMPLEXITY). For expansion node allocation ratio convergence to within ϵ with confidence $1 - \delta$, the required sample complexity is:

$$n_{\min} = O\left(\frac{k^2 R_{\max}^2 \ln(k/\delta)}{\epsilon^2 (\mu_{\text{sum}}^*)^2}\right) \quad (14)$$

B PROOF OF CONVERGENCE FOR CACHE-ENHANCED WEIGHTED LEARNING

B.1 Introduction and Problem Formulation

We are concerned with the bilevel optimization problem defined as:

$$w^* = \arg \min_{w \in \Delta} F(w) \triangleq \mathcal{L}_{\text{val}}(\theta^*(w)) \quad (15)$$

$$\text{s.t. } \theta^*(w) = \arg \min_{\theta} \mathcal{L}_{\text{train}}(\theta, w) \quad (16)$$

Here, w represents a vector of trajectory-level weights, and the inner problem finds the optimal model parameters $\theta^*(w)$ for a given set of weights. The outer problem then seeks to find the weights w^* that yield the best performance on a validation set.

The algorithm under consideration approximates this bilevel problem by avoiding the full inner-loop optimization at each step. Instead, it performs an online procedure where, at each iteration t :

- (1) A single "virtual" gradient descent step is performed on the inner objective to get $\tilde{\theta}_t$.
- (2) The weights w_t are updated based on the performance of $\tilde{\theta}_t$ on the validation set.
- (3) The actual model parameters θ_t are updated using the new weights w_{t+1} .

Our goal is to rigorously prove that this computationally efficient approximation scheme is theoretically sound and converges to a stationary point of the true outer objective $F(w)$.

B.2 Convergence Analysis

Our proof strategy is to show that the gradient used by the algorithm is a sufficiently good approximation of the true, intractable gradient. This allows us to prove that the algorithm makes consistent progress in descending the outer-level loss surface.

B.2.1 Assumptions: The Foundation of the Proof. Our analysis rests on the following standard assumptions from optimization theory.

ASSUMPTION 1 (SMOOTHNESS AND BOUNDEDNESS). The loss functions $\mathcal{L}_{\text{train}}(\theta, w)$ and $\mathcal{L}_{\text{val}}(\theta)$ are twice continuously differentiable. Their gradients and Hessians are Lipschitz continuous and uniformly bounded. That is, there exist constants $L, C > 0$ such that the norms of all relevant gradients and Hessians are bounded by C , and their Lipschitz constants are bounded by L .

Intuition: This is a "no sharp turns or explosions" rule. It ensures the loss landscape is well-behaved. The boundedness prevents values from going to infinity, while Lipschitz continuity guarantees that the gradient doesn't change too erratically for a small change in parameters, making gradient-based updates predictable.

ASSUMPTION 2 (STRONG CONVEXITY OF THE INNER OBJECTIVE). For any fixed weights $w \in \Delta$, the training loss $\mathcal{L}_{\text{train}}(\theta, w)$ is μ -strongly convex with respect to the model parameters θ .

Intuition: This assumption means that for any choice of weights, the inner optimization landscape has a clear, unique "bottom of the bowl." This guarantees the existence of a single optimal model $\theta^*(w)$. While deep learning models are generally non-convex, this assumption often holds in a local region around an optimum or can be encouraged with techniques like L2 regularization.

B.2.2 Characterizing the True and Approximate Gradients. The core of the analysis lies in comparing the gradient the algorithm computes with the ideal one.

The True Gradient, $\nabla F(w)$. To find the gradient of the outer objective $F(w) = \mathcal{L}_{\text{val}}(\theta^*(w))$, we apply the chain rule:

$$\nabla_w F(w) = \frac{\partial \theta^*(w)}{\partial w}^\top \nabla_\theta \mathcal{L}_{\text{val}}(\theta^*(w))$$

The term $\frac{\partial \theta^*(w)}{\partial w}$, a Jacobian matrix, is unknown. To find it, we use the optimality condition of the inner problem: at the minimum $\theta^*(w)$, the gradient must be zero.

$$\nabla_\theta \mathcal{L}_{\text{train}}(\theta^*(w), w) = 0$$

We can now differentiate this entire equation with respect to w using the total derivative:

$$\nabla_w^2 \mathcal{L}_{\text{train}}(\theta^*, w) + \frac{\partial \theta^*(w)}{\partial w}^\top \nabla_{\theta\theta}^2 \mathcal{L}_{\text{train}}(\theta^*, w) = 0$$

Solving for the Jacobian term, we get:

$$\frac{\partial \theta^*(w)}{\partial w}^\top = -\nabla_w^2 \mathcal{L}_{\text{train}}(\theta^*, w) [\nabla_{\theta\theta}^2 \mathcal{L}_{\text{train}}(\theta^*, w)]^{-1}$$

Substituting this back gives the expression for the **true gradient**:

$$\nabla F(w) = -\nabla_w^2 \mathcal{L}_{\text{train}}(\theta^*, w) [\nabla_{\theta\theta}^2 \mathcal{L}_{\text{train}}(\theta^*, w)]^{-1} \nabla_\theta \mathcal{L}_{\text{val}}(\theta^*) \quad (17)$$

Why this is intractable: This equation requires two things we don't have: the true inner optimum $\theta^*(w)$, and the inverse of a potentially massive Hessian matrix, $\nabla_{\theta\theta}^2 \mathcal{L}_{\text{train}}$. Computing this at every step is prohibitively expensive.

The Algorithm's Gradient, g_t . The algorithm approximates this by first taking a single virtual step: $\tilde{\theta}_t = \theta_t - \alpha \nabla_{\theta} \mathcal{L}_{\text{train}}(\theta_t, w_t)$. It then computes its gradient g_t by differentiating $\mathcal{L}_{\text{val}}(\tilde{\theta}_t)$ with respect to w_t .

$$g_t \triangleq \nabla_w \mathcal{L}_{\text{val}}(\tilde{\theta}_t) = \frac{\partial \tilde{\theta}_t}{\partial w_t}^\top \nabla_{\theta} \mathcal{L}_{\text{val}}(\tilde{\theta}_t)$$

The Jacobian $\frac{\partial \tilde{\theta}_t}{\partial w_t}$ is now easy to compute from the virtual update equation:

$$\frac{\partial \tilde{\theta}_t}{\partial w_t} = \frac{\partial}{\partial w_t} (\theta_t - \alpha \nabla_{\theta} \mathcal{L}_{\text{train}}(\theta_t, w_t)) = -\alpha \nabla_{w\theta}^2 \mathcal{L}_{\text{train}}(\theta_t, w_t)$$

This gives the expression for the algorithm's gradient:

$$g_t = -\alpha \nabla_{w\theta}^2 \mathcal{L}_{\text{train}}(\theta_t, w_t)^\top \nabla_{\theta} \mathcal{L}_{\text{val}}(\tilde{\theta}_t) \quad (18)$$

The Approximation: Comparing Eq. 17 and 18, we see the algorithm makes two key simplifications: (1) It uses the current, suboptimal parameters θ_t and $\tilde{\theta}_t$ instead of the true optimum θ^* . (2) It replaces the expensive inverse Hessian term with a simple identity matrix scaled by the learning rate, αI .

B.2.3 Bounding the Gradient Approximation Error. This lemma is the heart of the entire proof. It formally shows that the algorithm's gradient is a "good enough" proxy for the true gradient.

LEMMA 2 (GRADIENT APPROXIMATION ERROR). Under Assumptions 1 and 2, the error between the true and approximate gradients is bounded as:

$$\|\nabla F(w_t) - g_t\| \leq L_A \|\theta_t - \theta^*(w_t)\| + O(\alpha) \quad (19)$$

for some constant $L_A > 0$. The error is composed of a suboptimality error (proportional to how far θ_t is from its optimum) and an approximation error (proportional to the inner learning rate α).

PROOF. To prove this, we introduce an intermediate term, \hat{g}_t , which represents the true gradient's structure but evaluated at the current suboptimal point θ_t :

$$\hat{g}_t \triangleq -\nabla_{w\theta}^2 \mathcal{L}_{\text{train}}(\theta_t, w_t) [\nabla_{\theta\theta}^2 \mathcal{L}_{\text{train}}(\theta_t, w_t)]^{-1} \nabla_{\theta} \mathcal{L}_{\text{val}}(\theta_t)$$

Using the triangle inequality, we can split the total error into two manageable parts:

$$\|\nabla F(w_t) - g_t\| \leq \underbrace{\|\nabla F(w_t) - \hat{g}_t\|}_{\text{Term A: Suboptimality Error}} + \underbrace{\|\hat{g}_t - g_t\|}_{\text{Term B: Approximation Error}}$$

Part 1: Bounding Term A (Suboptimality Error). This term arises from using θ_t instead of $\theta^*(w_t)$. Every component inside the expressions for $\nabla F(w_t)$ and \hat{g}_t (the gradients and Hessians) is a Lipschitz continuous function of θ by Assumption 1. Therefore, the difference between evaluating these functions at θ_t and θ^* is bounded by their distance. For example:

$$\|\nabla_{\theta} \mathcal{L}_{\text{val}}(\theta^*) - \nabla_{\theta} \mathcal{L}_{\text{val}}(\theta_t)\| \leq L \|\theta_t - \theta^*(w_t)\|$$

By applying this property to all components and using the fact that all terms are bounded, we can conclude that the total difference is proportional to this distance:

$$\|\nabla F(w_t) - \hat{g}_t\| \leq L_A \|\theta_t - \theta^*(w_t)\|$$

Part 2: Bounding Term B (Approximation Error). This term captures the error from approximating the inverse Hessian with αI .

$$\begin{aligned} \|\hat{g}_t - g_t\| &= \|-\nabla_{w\theta}^2 \mathcal{L}_{\text{train}} [\nabla_{\theta\theta}^2 \mathcal{L}_{\text{train}}]^{-1} \nabla_{\theta} \mathcal{L}_{\text{val}}(\theta_t) - (-\alpha \nabla_{w\theta}^2 \mathcal{L}_{\text{train}} \nabla_{\theta} \mathcal{L}_{\text{val}}(\tilde{\theta}_t))\| \\ &\leq \|\nabla_{w\theta}^2 \mathcal{L}_{\text{train}}\| \left\| \alpha \nabla_{\theta} \mathcal{L}_{\text{val}}(\tilde{\theta}_t) - [\nabla_{\theta\theta}^2 \mathcal{L}_{\text{train}}]^{-1} \nabla_{\theta} \mathcal{L}_{\text{val}}(\theta_t) \right\| \end{aligned}$$

First, since $\tilde{\theta}_t = \theta_t - \alpha \nabla_{\theta} \mathcal{L}_{\text{train}}$, and by smoothness, we know that $\nabla_{\theta} \mathcal{L}_{\text{val}}(\tilde{\theta}_t) = \nabla_{\theta} \mathcal{L}_{\text{val}}(\theta_t) + O(\alpha)$. Substituting this gives:

$$\|\hat{g}_t - g_t\| \leq C \|(\alpha I - [\nabla_{\theta\theta}^2 \mathcal{L}_{\text{train}}]^{-1}) \nabla_{\theta} \mathcal{L}_{\text{val}}(\theta_t) + O(\alpha^2)\|$$

As shown in the literature on bilevel optimization and meta-learning, a single gradient step is equivalent to the first-order truncation of the Neumann series for the matrix inverse. The error of this approximation is proportional to α . Therefore, $\|(\alpha I - [\nabla_{\theta\theta}^2 \mathcal{L}_{\text{train}}]^{-1})\| = O(\alpha)$. This leads to:

$$\|\hat{g}_t - g_t\| \leq O(\alpha)$$

Combining the bounds for Term A and Term B completes the proof of the lemma. \square

B.2.4 Main Convergence Theorem. With the gradient error bounded, we can now prove the main convergence result.

THEOREM 3 (CONVERGENCE TO A STATIONARY POINT). Under Assumptions 1 and 2, if the learning rates α and β are sufficiently small, the algorithm converges to a stationary point of $F(w)$. Formally:

$$\lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\|\nabla F(w_t)\|^2] = O(\alpha^2) + O(\beta)$$

PROOF. By the smoothness of $F(w)$, we have the standard descent lemma:

$$F(w_{t+1}) \leq F(w_t) + \langle \nabla F(w_t), w_{t+1} - w_t \rangle + \frac{L_F}{2} \|w_{t+1} - w_t\|^2$$

Using the update rule $w_{t+1} - w_t \approx -\beta g_t$ (ignoring the projection, which only helps descent):

$$F(w_{t+1}) \leq F(w_t) - \beta \langle \nabla F(w_t), g_t \rangle + \frac{L_F \beta^2}{2} \|g_t\|^2$$

The key is to analyze the inner product. We can rewrite it as:

$$\langle \nabla F(w_t), g_t \rangle = \|\nabla F(w_t)\|^2 - \langle \nabla F(w_t), \nabla F(w_t) - g_t \rangle$$

Using Young's inequality ($\langle a, b \rangle \leq \frac{1}{2} \|a\|^2 + \frac{1}{2} \|b\|^2$) on the second term:

$$\langle \nabla F(w_t), g_t \rangle \geq \|\nabla F(w_t)\|^2 - \left(\frac{1}{2} \|\nabla F(w_t)\|^2 + \frac{1}{2} \|\nabla F(w_t) - g_t\|^2 \right) = \frac{1}{2} \|\nabla F(w_t) - g_t\|^2$$

Substitute this back into the descent inequality and rearrange to isolate the term we care about:

$$\frac{\beta}{2} \|\nabla F(w_t)\|^2 \leq (F(w_t) - F(w_{t+1})) + \frac{\beta}{2} \|\nabla F(w_t) - g_t\|^2 + \frac{L_F \beta^2}{2} \|g_t\|^2$$

Now, we sum this expression from $t = 0$ to $T - 1$ and take the expectation:

$$\frac{\beta}{2} \sum_{t=0}^{T-1} \mathbb{E}[\|\nabla F(w_t)\|^2] \leq \mathbb{E}[F(w_0) - F(w_T)] + \frac{\beta}{2} \sum_{t=0}^{T-1} \mathbb{E}[\|\nabla F(w_t) - g_t\|^2] + O(T\beta^2)$$

The first term on the right is a telescoping sum, bounded by $F(w_0) - F_{\min}$. For the second term, we use our result from Lemma 2 (and a supporting lemma stating that $\mathbb{E}[\|\theta_t - \theta^*\|^2]$ is bounded).

$$\sum_{t=0}^{T-1} \mathbb{E}[\|\nabla F(w_t) - g_t\|^2] \leq \sum \mathbb{E}[(L_A \|\theta_t - \theta^*\| + O(\alpha))^2] \leq T \cdot O(\alpha^2)$$

Plugging this in and dividing the entire inequality by $T\beta/2$:

$$\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\|\nabla F(w_t)\|^2] \leq \frac{2(F(w_0) - F_{\min})}{T\beta} + O(\alpha^2) + O(\beta)$$

As we let the number of iterations $T \rightarrow \infty$, the first term vanishes, leaving us with:

$$\lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\|\nabla F(w_t)\|^2] \leq O(\alpha^2) + O(\beta)$$

Final Result: This result states that the average squared norm of the true gradient converges to a small "error ball" whose size is controlled by the learning rates. By choosing sufficiently small α and β , we can guarantee that the algorithm finds a point arbitrarily close to a stationary point where the true gradient is zero.

B.3 Conclusion

Through a step-by-step analytical process, we have rigorously demonstrated the convergence properties of the Cache-Enhanced Weighted Learning algorithm. The detailed derivation shows that its core mechanism—a single-step, online approximation of the bilevel optimization gradient—is theoretically sound. The proof hinges on Lemma 2, which carefully bounds the error of this approximation, showing that it is controlled by the inner-loop learning rate and the suboptimality of the model parameters. This analysis confirms that each weight update, while imperfect, systematically pushes the true validation objective towards a minimum, guaranteeing that the algorithm successfully converges to a stationary point and learns effective trajectory weights. \square

REFERENCES

- [1] Luca Franceschi, Paolo Frasconi, Saverio Salzo, Riccardo Zecchina, and Massimiliano Pontil. Bilevel programming for hyperparameter optimization and meta-learning. In Proceedings of the 35th International Conference on Machine Learning, ICML 2018, pages 1563–1572. PMLR, 2018.
- [2] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. Journal of the American Statistical Association, 58(301):13–30, 1963.