Appendix: Optimal Continual Learning has Perfect Memory and is NP-HARD

Jeremias Knoblauch [1] **Hisham Husain** [2] **Tom Diethe** [3]

A. Proofs: Optimal CL is NP-HARD

The main results relating to the computational hardness of CL which are not proven in the main paper are carefully derived in this supplement.

A.1. Relationship with other problems

Notice that at each step, the functions $\mathcal{A}_{\boldsymbol{\theta}}$ and $\mathcal{A}_{\boldsymbol{I}}$ of optimal Idealized CL define an *optimization problem*, as we are tasked with a parameter value $\boldsymbol{\theta}_t$ at each iteration satisfying the specified criterion on all previous tasks.

Definition 1. *Given a fixed hypothesis class $\mathcal{F}_{\boldsymbol{\Theta}}$, criterion $\mathcal{C}$, a set $A \in \text{SAT}_{\mathcal{Q}}$ and $B \in \text{SAT}_{\cap}$ for*

$$\text{SAT}_{\cap} = \{\cap_{i=1}^{t} A_i : A_i \in \text{SAT}_{\mathcal{Q}}, \\ 1 \leq i \leq t \text{ and } 1 \leq t \leq T,\ \ T \in \mathbb{N}\},$$

*the **optimal Idealized** CL **optimization problem** is to find a $\boldsymbol{\theta} \in A \cap B$. Accordingly, the **optimal Idealized** CL **decision problem** is to decide if a solution exists, i.e. if $\boldsymbol{\theta} \in A \cap B \neq \emptyset$.*

We first show that the optimal Idealized CL optimization problem is at least as hard as its corresponding decision problem.

Proposition 1. *If one can solve the optimal Idealized* CL *optimization problem, one can solve the optimal Idealized* CL *decision problem.*

Proof. if the optimal Idealized CL optimization problem can be solved, then there exists some function $f : \text{SAT}_{\mathcal{Q}} \times \text{SAT}_{\cap} \to \boldsymbol{\Theta}$ such that

$$f(A, B) = \boldsymbol{\theta}$$

such that $\boldsymbol{\theta} \in A \cap B$, for any A and B as in Definition 1. But then, one can construct the indicator function

$$1(A, B) = \begin{cases} 1 & \text{if } f(A, B) \notin \emptyset \\ 0 & \text{otherwise,} \end{cases}$$

which clearly solves the decision problem. □

The interpretation of this result is clear: Computationally, the optimal Idealized CL optimization problem is at least as hard as the optimal Idealized CL decision problem. This insight is useful mainly because of the next proposition, which shows that any optimal CL algorithm can solve the optimal Idealized CL optimization problem.

Proposition 2. *If a* CL *algorithm is optimal, then it can solve the optimal Idealized* CL *optimization problem.*

Proof. By Lemma 1, it suffices to show this for an optimal Idealized CL algorithm. Suppose that $\mathcal{A}_{\boldsymbol{\theta}}$, $\mathcal{A}_{\text{I}}$ define an optimal Idealized CL algorithm. For a given problem instance of the optimal Idealized CL optimization problem with $A \in \text{SAT}_{\mathcal{Q}}$ and $\cap_{t=1}^{T-1} B_t = B \in \text{SAT}_{\cap}$ such that $B_t \in \text{SAT}_{\mathcal{Q}}$ for all $1 \leq t \leq (T-1)$, we can use the optimal Idealized CL algorithm to solve the problem. To see this, construct the problem instance as $\text{SAT}_i = B_i$ for $1 \leq i \leq (T-1)$ and $\text{SAT}_T = A$. Clearly, the value $\boldsymbol{\theta}_T$ generated by the optimal Idealized CL algorithm satisfies that $\boldsymbol{\theta}_T \in A \cap B$. □

Again, this has a clear interpretation: Computationally, the optimal Idealized CL algorithm is at least as hard as the optimal Idealized CL optimization problem.

A.2. Proof of Lemma 2

A straightforward Corollary follows. Rewriting this result, one obtains Lemma 2.

Corollary 1. *If a* CL *algorithm is optimal, then it can solve the optimal Idealized* CL *decision problem.*

Proof. Combine Propositions 1 and 2. □

A.3. A further refinement

Indeed, the connection between an optimal CL algorithm and the optimal Idealized CL optimization and decision problems can be made much tighter. The below alternative Lemma could be used in the proof of Theorem 2 to show that optimal CL does not only solve an NP-HARD problem, it solves an NP-HARD problem at *each iteration*. We do

[1]Department of Statistics, Warwick University & The Alan Turing Institute, London [2]Australian National University, Canberra & Data61, Sydney [3]Amazon. Correspondence to: Jeremias Knoblauch <j.knoblauchwarwick.ac.uk>.

Proceedings of the 37^{th} International Conference on Machine Learning, Vienna, Austria, PMLR 119, 2020.

not discuss this in the main paper as the consequences remain the same, though at the expense of complicating the presentation.

Lemma 1. *A* CL *algorithm is optimal if and only if it solves* T *optimal Idealized* CL *optimization problem instances given by* $\{(A_t, B_t)\}_{t=1}^T$ *with* $A_t = \text{SAT}_t = \text{SAT}(\widehat{\mathbb{P}}_t)$ *and* $B_t = \text{SAT}_{1:(t-1)} = \cap_{i=1}^{t-1}\text{SAT}_i$. *Similarly, a* CL *algorithm is optimal only if it can be used to solve the collection of* T *optimal Idealized* CL *decision problems corresponding to* $\{(A_t, B_t)\}_{t=1}^T$.

Proof. Using Lemma 1 once again, it suffices to prove this for an optimal Idealized CL algorithm. The first part of the proposition then follows by definition, as $\boldsymbol{\theta}_t \in \text{SAT}_{1:t} = \text{SAT}_t \cap \left(\cap_{i=1}^{t-1}\text{SAT}_i\right)$. Setting $B = \cap_{i=1}^{t-1}\text{SAT}_i$ and $A = \text{SAT}_t$ reveals this to be an optimal Idealized CL optimization problem. The second part follows by combining the first part with the same arguments used in the proof of Proposition 1. □

A.4. Proof of Theorem 1

With Lemma 2 in place, the proof of Theorem 1 follows by relatively simple arguments that we summarize in two separate propositions.

Proposition 3. *If* $\mathcal{Q}$ *and* $\mathcal{C}$ *are such that* $\text{SAT}_{\mathcal{Q}} \supseteq S$ *or* $\text{SAT}_{\cap} \supseteq S$ *so that* S *is the set of tropical hypersurfaces or the set of polytopes on* $\boldsymbol{\Theta}$*, then the optimal Idealized* CL *decision problem is* NP-COMPLETE.

Proof. This is a simple application of the results in Theobald (2006) for the case where S is the set of tropical hypersurfaces. For the case where S is the set of polytopes, it is a simple application of the results in Tiwary (2008b) and Tiwary (2008a). □

Proposition 4. *The optimal idealized* CL *optimization problem is* NP-HARD.

Proof. By proposition 1, the idealized CL optimization problem is at least as hard as the idealized CL decision problem. □

With this in hand, the proof of Theorem 1 is readily obtained by combining Lemma 2 with Propositions 2, 3 and 4.

Proof. First, use Lemma 2: Optimal CL can correctly decide if $A \cap B = \emptyset$, for all $A \in \text{SAT}_{\cap}$ and $B \in \text{SAT}_{\mathcal{Q}}$. Second, use Proposition 3 to conclude that this decision problem is NP-COMPLETE. Third, it follows by Proposition 4 that the optimization problem corresponding to an NP-COMPLETE decision problem is NP-HARD. Thus, by Proposition 2, the result follows. □

A.5. Proof of Corollary 1

It is straightforward to generalize the results of Theorem 1 for all collections $\text{SAT}_{\mathcal{Q}}$ whose intersections are as hard to compute as polytopes.

Proof. Re-use the proof of Theorem 1 and note that if the decision problem is at least as hard as for polytopes, then the computational complexity of derived as a result of Theorem 1 provides a lower bound. □

B. Proofs: Optimal CL has Perfect Memory

Next, we show give detailed derivations for the perfect memory result in the main paper.

B.1. Proof of Lemma 3

For convenience, we compile the results in Lemma 3 into two separate Propositions.

Proposition 5. $\boldsymbol{\theta}' \in \text{E}(\boldsymbol{\theta}) \iff \text{E}(\boldsymbol{\theta}) = \text{E}(\boldsymbol{\theta}')$. *Further, whenever* $\boldsymbol{\theta}' \notin \text{E}(\boldsymbol{\theta})$*, it hold that* $\text{E}(\boldsymbol{\theta}) \cap \text{E}(\boldsymbol{\theta}') = \emptyset$.

Proof. Suppose that $\boldsymbol{\theta}' \in \text{E}(\boldsymbol{\theta})$. From the definition of $\text{E}(\boldsymbol{\theta})$, this immediately implies that any $A \in S(\boldsymbol{\theta})$ contains $\boldsymbol{\theta}'$. In other words, $\boldsymbol{\theta}' \in A \iff \boldsymbol{\theta} \in A$ for all $A \in \text{SAT}_{\mathcal{Q}}$. From this, it immediately follows that $S(\boldsymbol{\theta}) = S(\boldsymbol{\theta}')$ so that

$$\text{E}(\boldsymbol{\theta}) = \bigcap_{A \in S(\boldsymbol{\theta})} A = \bigcap_{A \in S(\boldsymbol{\theta}')} A = \text{E}(\boldsymbol{\theta}'),$$

which proves the first claim. The second claim then follows by contradiction: Suppose there was a point $\widetilde{\boldsymbol{\theta}}$ such that $\widetilde{\boldsymbol{\theta}} \in \text{E}(\boldsymbol{\theta}) \cap \text{E}(\boldsymbol{\theta}')$. But then, $\widetilde{\boldsymbol{\theta}} \in \text{E}(\boldsymbol{\theta})$, which by the first claim would imply that $S(\boldsymbol{\theta}) = S(\widetilde{\boldsymbol{\theta}}) = S(\boldsymbol{\theta}')$ so that

$$\text{E}(\boldsymbol{\theta}) = \bigcap_{A \in S(\boldsymbol{\theta})} A = \bigcap_{A \in S(\widetilde{\boldsymbol{\theta}})} A = \bigcap_{A \in S(\boldsymbol{\theta}')} A = \text{E}(\boldsymbol{\theta}').$$

But since $\boldsymbol{\theta}' \notin \text{E}(\boldsymbol{\theta})$ and $\boldsymbol{\theta}' \in \text{E}(\boldsymbol{\theta}')$ it holds that $\text{E}(\boldsymbol{\theta}) \neq \text{E}(\boldsymbol{\theta}')$, which yields the desired contradiction. □

Proposition 6. *For all* $A \in \text{SAT}_{\mathcal{Q}}$ *and all* $\boldsymbol{\theta} \in \boldsymbol{\Theta}$*, either* $\text{E}(\boldsymbol{\theta}) \subseteq A$ *or* $\text{E}(\boldsymbol{\theta}) \cap A = \emptyset$.

Proof. This follows by definition of $\text{E}(\boldsymbol{\theta})$: Either $A \in S(\boldsymbol{\theta})$, in which case it must follow that $\text{E}(\boldsymbol{\theta}) \subseteq A$. Alternatively, if $A \notin S(\boldsymbol{\theta})$, then by Proposition 5 one has that $A \notin S(\boldsymbol{\theta}')$ for any $\boldsymbol{\theta}' \in \text{E}(\boldsymbol{\theta})$, which means that $\text{E}(\boldsymbol{\theta}) \cap A = \emptyset$. □

B.2. Proof of Lemma 4

We first define the notion of a Decision Problem Oracle set. Note that Lemma 4 in the main paper revolves around such a Decision Problem Oracle set (albeit without using this name).

Definition 2. *Given a set* $\text{SAT}_{1:t} \in \text{SAT}_{\cap}$*, a set* C *is a* ***Decision Problem Oracle set*** *for* $\text{SAT}_{1:t}$ *if*

$$C \cap A = \emptyset \iff \text{SAT}_{1:t} \cap A = \emptyset,$$

for any $A \in \text{SAT}_{\mathcal{Q}}$.

Proposition 7. *If a* CL *algorithm is optimal, there exists a function* $h : \boldsymbol{\Theta} \times \mathcal{I} \to 2^{\boldsymbol{\Theta}}$ *such that for* $\boldsymbol{\theta}_t, \text{I}_t$ *as in Definition 6,* $C_t = h(\boldsymbol{\theta}_t, \text{I}_t)$ *is a Decision Problem Oracle set for* $\text{SAT}_{1:t}$*, and this holds for all* $1 \leq t \leq T$.

Proof. If the CL algorithm is optimal, it can solve the optimal Idealized CL decision problem given by $A = \text{SAT}_{t+1}$ and $B = \text{SAT}_{1:t}$ at the $(t+1)$-th task. (This follows by combining Lemma 1 with Proposition 1) Specifically, because

$$\mathcal{A}(\boldsymbol{\theta}_t, \text{I}_t, A) \notin \emptyset \iff \text{SAT}_{1:t} \cap A \neq \emptyset,$$

it is clear that there must exist a function $g : \boldsymbol{\Theta} \times \mathcal{I} \times \text{SAT}_{\mathcal{Q}} \to 2^{\boldsymbol{\Theta}}$ for which $C_t = g(\boldsymbol{\theta}_t, \text{I}_t, A)$ is such that

$$C_t \cap A \neq \emptyset \iff \text{SAT}_{1:t} \cap A \neq \emptyset.$$

Furthermore, it is clear that g will be constant in A (since $\text{SAT}_{1:t}$ is), so that one can write $C_t = h(\boldsymbol{\theta}_t, \boldsymbol{I}_t)$ for some $h : \boldsymbol{\Theta} \times \mathcal{I} \to 2^{\boldsymbol{\Theta}}$ instead. □

B.3. Assumptions for the Decision Problem Oracle set

We use the observation of the last subsection to investigate the memory requirements of optimal CL algorithms. Before doing so, we first make some assumptions that are useful for proving Theorem 2 and Corollary 2.

B.3.1. ASSUMPTIONS FOR THEOREM 2

Assumption 1 (Storage efficiency)**.** $C_t \subseteq \text{SAT}_{1:t}$

Assumption 2 (Information efficiency)**.** $C_t \subseteq C_{t-1}$

Assumption 3 (Finite identifiability)**.** *There exists a finite sequence of sets* $\{A_t\}_{t=1}^{T}$ *in* $\text{SAT}_{\mathcal{Q}}$ *such that* $\cap_{t=1}^{T} A_t = \text{E}(\boldsymbol{\theta})$*, for all* $\boldsymbol{\theta} \in \boldsymbol{\Theta}$.

Remark 1. *Assumption 1 ensures that* C_t *takes up as little space in memory as possible. To illustrate this, suppose that* $\widetilde{C}_t \cap \text{SAT}_{t+1} = \emptyset \iff \text{SAT}_{1:t} \cap \text{SAT}_{t+1} = \emptyset$*, but that* $\widetilde{C}_t \setminus \text{SAT}_{1:t} \neq \emptyset$*. In this case, it clearly holds for* $C_t = \widetilde{C}_t \cap \text{SAT}_{1:t} \subset \widetilde{C}_t$ *that* $C_t \cap \text{SAT}_{t+1} = \emptyset \iff \text{SAT}_{1:t} \cap \text{SAT}_{t+1} = \emptyset$*, too. In other words, one can construct an alternative and stritly smaller Decision Problem Oracle set* C_t *from* $\widetilde{C}_t$ *by removing all points that are not also in* $\text{SAT}_{1:t}$.

Remark 2. *Assumption 2 ensures that the algorithm learns monotonically. Specifically, it ensures that each additional task will shrink the set* $\text{SAT}_{1:t}$ *of parameter values that satisfy the criterion* C *on all task* $1, 2, \ldots t$*. This is intuitively appealing since it means that the algorithm never incorrectly discards a parameter only to add it back in at a later task.*

Remark 3. *Assumption 3 says that equivalence sets are reachable with finitely many tasks. In other words, there exist collections of tasks which satisfy the algorithm's optimality criterion* C *only if the parameter that is learnt lies in a single equivalence set.*

B.3.2. ASSUMPTIONS FOR COROLLARY 2

As we shall see shortly, if we strengthen Assumption 3, we can drastically simplify the proof of Theorem 2 and drop the other two assumptions required for the result.

Assumption 4 (Identifiability)**.** $\text{E}(\boldsymbol{\theta}) \in \text{SAT}_{\mathcal{Q}}$*, for all* $\boldsymbol{\theta} \in \boldsymbol{\Theta}$.

Remark 4. *Simply put, this means that each equivalence set can be "reached" with a single task. In other words, each equivalence set is identifiable with a single task.*

B.4. Proof of Theorem 2

Notice that proving Theorem 2 is equivalent to proving the proposition below.

Proposition 8. *Under Assumptions 1, 2 and 3, any optimal* CL *algorithm has perfect memory.*

Proof. We show this by proving that for some arbitrary minimal representation $\{f(i)\}_{i \in I}$ and $F = \cup_{i \in I}\{f(i)\}$, $C_t \supseteq F \cap \text{SAT}_{1:t}$.

First, we show that $\widetilde{C}_t = F \cap \text{SAT}_{1:t}$ is a Decision Problem Oracle set. In other words, we show that $\widetilde{C}_t \cap A = \emptyset \iff \text{SAT}_{1:t} \cap A = \emptyset$, for all $A \in \text{SAT}_{\mathcal{Q}}$ and any $\text{SAT}_{1:t} \in \text{SAT}_{\cap}$. We do so by contradiction: Suppose that $\exists A \in \text{SAT}_{\mathcal{Q}}$ so that $\widetilde{C}_t \cap A = \emptyset$, but $\text{SAT}_{1:t} \cap A \neq \emptyset$. But then, $A \cap \text{SAT}_{1:t}$ contains at least one point, say $\boldsymbol{\theta}$. By construction of $\widetilde{C}_t$, it also follows that $F \cap A = \emptyset$. This yields the desired contradiction, since by virtue of $A \supseteq E(\boldsymbol{\theta})$ it also implies that $F \cap E(\boldsymbol{\theta}) = \emptyset$, even though F contains exactly one point for each equivalence set by definition, including the equivalence set $E(\boldsymbol{\theta})$. In other words, it is *sufficient* for the optimal CL algorithm to be able to reconstruct the Decision Problem Oracle set $\widetilde{C}_t = F \cap \text{SAT}_{1:t}$ at task $(t+1)$.

Second, we demonstrate that this is also *necessary*: Suppose that there exists some $\boldsymbol{\theta} \in \widetilde{C}_t$ such that $\widetilde{C}_t \setminus \{\boldsymbol{\theta}\}$ is also a Decision Problem Oracle set, for all $1 \leq t \leq T$. By virtue of Assumption 3, we can construct a finite sequence of sets $\{A_i\}_{i=t+1}^{T}$ such that $A_i \in \text{SAT}_{\mathcal{Q}}$ and $\cap_{i=t+1}^{T} A_i = E(\boldsymbol{\theta})$. By construction, $\text{SAT}_{1:t} \cap \left(\cap_{i=t+1}^{T} A_i\right) \neq \emptyset$, but $\left(\widetilde{C}_t \setminus \{\boldsymbol{\theta}\}\right) \cap \left(\cap_{i=t+1}^{T} A_i\right) = \emptyset$. Since it also holds that $C_t \subseteq C_{t-1}$, it follows that $\left(\widetilde{C}_{T-1} \setminus \{\boldsymbol{\theta}\}\right) \cap A_T = \emptyset$, which completes the proof. □

B.5. Proof of Corollary 2

Alternatively, one could drop the first two assumptions and strengthen the third to draw the same conclusion.

Proposition 9. *Under Assumption 4, the optimal* CL *algorithm has perfect memory.*

Proof. The proof of sufficiency is exactly equal to the one in Proposition 8. The proof of necessity follows along the same lines as before but is even easier: Since one can always select $A = \mathrm{E}(\boldsymbol{\theta})$, $\textsc{Sat}_{1:t} \supseteq C_t \supseteq \textsc{Sat}_{1:t} \cap F$ readily follows. (Indeed, it follows that $C_t = \textsc{Sat}_{1:t}$ because $\textsc{Sat}_{1:t} \cap F$ for any $F = \cup_{i \in I}\{f(i)\}$ generated through a Minimal Representation $\{f(i)\}_{i \in I}$.) $\square$

References

Theobald, T. On the frontiers of polynomial computations in tropical geometry. *Journal of Symbolic Computation*, 41(12):1360–1375, 2006.

Tiwary, H. R. On the hardness of computing intersection, union and Minkowski sum of polytopes. *Discrete & Computational Geometry*, 40(3):469–479, 2008a.

Tiwary, H. R. *Complexity of some polyhedral enumeration problems*. PhD thesis, Saarländische Universität, 2008b.