Hierarchical Synchronization

Local Training, Global State, and the Composition of Updates

Flyxion
Independent Researcher

Abstract

Geographically distributed training trades communication cost against optimization quality. Frequent synchronization limits divergence but depends on bandwidth and uniform participation, while infrequent synchronization lets sites work independently at the cost of reconciliation. This paper asks under what conditions independently produced changes can be composed into an authorized transition of a shared state. Records of what each observer has seen are kept apart from the numerical combination of shifts and from the dynamics of repeated transitions. Submission views form a sheaf under equality on overlaps, and a flat information order joins consistent but unequal views. Receipt, rejection and timeout are observer-indexed events. Numerical summaries cannot be glued: a plain weighted sum double counts a retransmission, and a merge that requires disjoint contributor sets certifies counting once only for a tree evaluated from validated leaves, not for what a remote relay reports. Null Convention Logic separates a received zero shift from an absent contribution and full receipt from policy closure, and the commit certificate records which claims rest on which assumption about relays. A two-site quadratic example shows that exact, fully participated reduction converges to a point that depends on the local horizon and, for unequal curvatures, differs from the minimizer. An idealized control-variate correction with the exact global gradient removes the bias, and the mean local shift at a reference solution diagnoses it. A two-worker simulation with modeled costs reaches its target at 500 local steps on a mildly heterogeneous task, a modeled saving of about 5.5 over 50 steps that follows from the cost model once the target is reached. On strongly heterogeneous shards the same runs miss the target at long intervals and the mean local shift at the reference increases the validation loss, with the same pattern under memoryless SGD. A prototype with a relay hierarchy and quorum closure is specified and has not been built.

1 The Problem of a Shared Training State

Consider a collection of training sites, each with enough local capacity to hold a model and execute an optimizer. A site may be a desktop with a single GPU or a data centre whose accelerators synchronize internally. The sites differ in network capacity, access to data, and the amount of work they can complete within a given interval. They periodically receive a common checkpoint, train from it, and return a description of the change they produced. Intermediate nodes can combine contributions and pass summaries upstream. The root decides when the available contributions justify closing the round and publishing a successor checkpoint.

The assumption of a global state is compatible with local ignorance. During an open round, different sites may know different facts about the current computation. A relay can know

that its own subtree has completed without knowing whether a distant site has received the checkpoint. A worker can know the parameters it has reached without knowing whether its contribution will enter the next commit. The global state specifies the common object of the computation. The distribution of knowledge specifies which claims each participant can make about it.

This distinction matters because three questions arise at a synchronization boundary. The first is whether the local records agree wherever they describe the same event. The second is how to combine the distinct updates described by those records. The third is whether applying the combined update advances the intended objective. An affirmative answer to any one of these questions leaves the other two open. A hierarchy can preserve provenance perfectly and still aggregate poorly. An optimizer can produce a useful step from a partial set of workers while the record remains insufficient to verify that the configured participation rule was followed.

The central proposal is to give these questions separate mathematical objects. Records glue by agreement. Based shifts reduce by an explicit operation. Checkpoints advance by an authorized transition. The optimization analysis concerns the dynamics produced by repeated transitions. A fourth object, introduced in the course of the development, is the commit certificate, which travels with a successor checkpoint and records which of the preceding guarantees the transition relies on and under what assumptions about the participants. The category-theoretic language organizes the construction, with histories as a category and endpoint displacement as a functor, and it carries no result beyond telescoping. The load-bearing results are the propositions and the routing theorem of the sections that follow.

1.1 Relation to Existing Work

The training pattern studied here has a long history under other names. Federated averaging combines local stochastic gradient steps with periodic averaging of model parameters [20], and the analysis of local SGD established that infrequent averaging can preserve convergence rates while reducing communication [25]. More recent work on low-communication training of language models adopts the same structure with an adaptive outer optimizer applied to the averaged parameter shift [5], and the use of server-side optimizers on aggregated updates is analyzed in the federated setting by [22]. Hierarchical variants with separate edge and cloud aggregation intervals are the closest precedent for the distinction in Section 4 between pure relays and intermediate synchronization [15]. The gap between local and global objectives under heterogeneity is documented as client drift, with control-variate corrections proposed to remove it [10], as an objective inconsistency that arises when sites perform different amounts of local work [26], and as a proximal regularization problem [14]. The quadratic example of Section 6 is an exactly solvable instance of this phenomenon. Its role in the present paper is to locate the phenomenon relative to the protocol guarantees, since it persists when every protocol-level hypothesis holds.

The treatment of records draws on two further literatures. The information order on partial maps is the flat domain construction of Scott [23], and the resulting join of consistent observations is a join-semilattice of the kind used for state-based conflict-free replicated data types [24]. The contrast developed in Section 3 is that the record of a round has this idempotent structure, while the numerical aggregate computed from it does not. The treatment of faults

in the relay layer draws on the standard distinction between crash and Byzantine behavior [13, 4], and the observation in Section 5 that a sum-only interface limits what an aggregator can be asked to prove is related to the design constraints of secure aggregation [3]. The categorical and sheaf-theoretic vocabulary follows [17, 18, 2], the dependent-type vocabulary follows [19, 9], and Null Convention Logic is used in the sense of [6, 7], with the circuit-level account of input completeness and threshold gates taken from [11].

2 Checkpoints and Based Shifts

Let M denote the manifold of admissible parameter states. The prototype takes M to be an affine parameter space for a fixed model schema. The manifold formulation allows later chapters to consider parameter constraints, changes of chart, or equivalent representations of a model. Those extensions require explicit geometric choices. A general manifold supplies neither a canonical difference between two points nor a canonical average of several endpoints.

A checkpoint contains more than its parameter point. Write C_v for a committed checkpoint with parameter projection $\theta_v \in M$. The checkpoint identifies the run, its position in the commit history, the model schema, and the root optimizer state S_v required to continue. Site-local optimizer state can be represented separately, with a declared rule for whether it persists, resets, or changes upon rejoining. Two checkpoints can agree in their parameter projection while admitting different continuations because their optimizer memory or histories differ. Two distinct histories can also reach the same parameter point and optimizer state at the same round. For this reason the identity of a checkpoint is taken to be a digest of the full tuple $(\mathrm{run}, v, \sigma, \theta_v, S_v, \mathrm{id}(C_{v-1}), \mathrm{id}(\tau_{v-1}))$, where $\mathrm{id}(C_{v-1})$ is the digest of the parent checkpoint and τ_{v-1} is the transition record of Section 5 that authorized the step into C_v. The genesis checkpoint has a null parent. Identity then includes the authorized history, and equality of parameter projections is never used as a substitute for equality of checkpoints.

Let $T_{i,v}$ be the local training transformation for site i in round v. Once the local data stream, random state, and optimizer state are fixed, it maps a local initial state to a local final state. In an affine chart, the site reports the parameter shift

$$\Delta_{i,v} = \theta_{i,v}^{\mathrm{end}} - \theta_v.$$

The report identifies C_v as its origin. This origin is part of the meaning of the shift. The same numerical vector can occur in different rounds without carrying the same warrant for application. A shift from an earlier checkpoint can enter a later round only through an operation with an explicit interpretation of how it was rebased and what that operation preserves. Staleness-tolerant protocols that apply old shifts to newer checkpoints are therefore different protocols, each with its own account of what the rebased shift represents, and the framework treats them as extensions that supply a rebasing rule rather than as instances of the basic round.

The shift-theoretic distinction is clearest when one compares successive and parallel changes. If one transition takes θ_0 to θ_1 and a second takes θ_1 to θ_2, their affine endpoint differences add by telescoping. The second transition is defined on the result of the first. Two sites that both start at θ_0, however, produce parallel changes. Their endpoint differences can enter a

weighted reduction, but that reduction does not represent either site's transformation applied after the other. The local transformations may depend on different data and different optimizer memories, and their order can affect their composite.

On a curved M, a contribution could be represented by a tangent vector at θ_v obtained through a specified local inverse to a retraction [1]. The root could reduce those vectors in $T_{\theta_v}M$ and map the result back to M using the retraction. This makes the dependence on the common origin explicit. A rule for comparing or transporting shifts between different origins introduces further structure. Any path dependence should be attributed to the particular transport or training rule that produces it. In particular, the claim that an average of endpoints is meaningful depends on a chart in which the average is taken, and two charts related by a nonlinear change of coordinates give different averages of the same endpoints. The affine case avoids this by fixing the chart, and the statements of later sections are made in that case unless otherwise noted.

3 Local Records, Information Order, and Descent

Fix a round v and let E be the roster of contribution identifiers that the round policy declares eligible. The first object is the submission. For $i \in E$ let S_i be the set of possible immutable submissions for i, where a submission consists of its base checkpoint, its model schema, its declared weight, and the digest that identifies its payload. A submission does not change when it is received, rejected, delayed or lost. What varies among participants is what each has seen, and this is the second object, the view of an observer. For a first construction, take observation domains to be subsets $U \subseteq E$, ordered by inclusion, with ordinary set covers. The domain U_j of an observer j is the set of identifiers whose status that observer could in principle see. A section over U is an element of

$$\mathcal{F}(U) = \prod_{i \in U} (S_i \sqcup \{\bot\}),$$

an assignment to each identifier of U of either a submission or the symbol $\bot$, which the protocol vocabulary calls NULL. Restriction to $V \subseteq U$ is coordinate projection. The domain of definition $\operatorname{dom} s = \{i \in U : s(i) \neq \bot\}$ is the set of identifiers the observer can attest, meaning those for which it holds a validated submission. A NULL entry means that the observer contributes no assertion about the identifier, and in particular makes no claim about whether a submission exists.

Two structures on these sections must be kept apart. The first is the sheaf condition. With ordinary covers and equality on overlaps, $\mathcal{F}$ is a sheaf, since sections over overlapping domains that are equal on the overlap, with $\bot$ equal only to $\bot$, glue uniquely. This is exact reconstruction. It does not apply to observers with different delivery histories. If one observer holds the submission for i and another does not, their restrictions to $\{i\}$ are unequal, although neither contradicts the other. The second structure is the information order. Order each $S_i \sqcup \{\bot\}$ flatly, with $\bot \leq r$ for every $r \in S_i$ and distinct submissions incomparable, and order $\mathcal{F}(U)$ pointwise, so that $s \leq t$ when for every i either $s(i) = \bot$ or $s(i) = t(i)$. Two sections are consistent when they agree wherever both are non-NULL, which is equivalent to having an upper bound. The join of consistent observations accumulates incomplete evidence. It extends

each observation, and its restriction to the domain of a less informed observer can strictly extend that observer's section, which is why the join is not a gluing in the sheaf sense. The two constructions are both retained. The sheaf describes exact agreement on overlaps, and the order describes the accumulation of evidence from observers who have seen different parts of a round.

Two observers who have both seen the submission with identifier i must agree on its immutable fields, namely its base checkpoint, schema, declared weight and payload digest. This is what consistency requires on a shared identifier, and an observer who has seen nothing for i is consistent with every other observer. Distinct workers can submit different shifts while their views remain compatible, because their submissions have different identifiers. Their disagreement in update direction is a fact represented by the joined record and does not obstruct the join.

Receipt, rejection and timeout are not fields of a submission and do not belong to a section. Each is an event (j, v, i, κ) indexed by the observer j that issued it, the round v and the identifier i, with a kind κ that is one of received, rejected with a stated reason, or timed out at a named cutoff of j. An event is a claim about the observer that issued it, and it carries the position in that observer's local order at which it was issued. A receipt is compared with a named cutoff, and it is eligible for that cutoff when the submission validates and arrives before it. A timeout by j at cutoff t asserts only that j held no eligible submission for i at t, so it is compatible with receipt of the same submission by another observer, and it adds nothing to $\operatorname{dom} s_j$ because it attests nothing about the content of any submission. The same observer may issue a timeout at t and later a receipt for the same identifier, when the receipt arrived after the cutoff or failed validation. The invariant is that no observer's log contains both an eligible receipt by t and a timeout at t for the same identifier. Events from distinct observers never conflict, and the log of a round is a set of such events whose union is again idempotent. A rejection refers to a received submission by its digest and records that j declined to accept it. The policy that decides whether a round may close consults both objects, the joined submissions and the log of events.

Proposition 1 (Descent of submission views)**.** *Let $\{s_j \in \mathcal{F}(U_j)\}_{j \in J}$ be a finite family of views whose domains cover U. (a) If the views agree on overlaps, so that s_j and s_k have equal restrictions to $U_j \cap U_k$ for all j, k, there is exactly one $s \in \mathcal{F}(U)$ whose restriction to each U_j is s_j. (b) Extend each view to U by $\perp$ outside U_j. The extended family has an upper bound in the information order if and only if it is pairwise consistent, and in that case the pointwise join $s = \bigcup_j s_j$ is its unique least upper bound. The join is commutative, associative and idempotent. When the views are consistent but unequal on overlaps, the restriction of s to U_j extends s_j and can do so strictly.*

Proof. For (a), each identifier of U lies in some U_j, and agreement on overlaps makes the value $s_j(i)$ independent of the choice of j, which defines s and gives the restrictions. Uniqueness holds because s is determined coordinatewise. For (b), at each coordinate the non-NULL values of the family must coincide for an upper bound to exist, which is pairwise consistency. Any upper bound then agrees with the common value wherever one is present, so the pointwise join is least and unique. Commutativity, associativity and idempotence are those of set union on graphs of partial maps. If $s_j(i) = \perp$ while another view holds a submission at $i \in U_j$, then $s(i) \neq s_j(i)$. □

This elementary model separates the existence of a global record from the sufficiency of that record for a commit. Compatible partial observations can join even when a worker is absent, and the joined record then has that worker's identifier in NULL. A commit requires a separate decision about whether the joined record satisfies the round policy. The decision is taken on the joined record and is not a property of the gluing.

Idempotence is the property that separates records from numerical summaries. Two observers who both hold the record of a worker's contribution give a joint record containing that contribution once. Two relays whose summaries both include the worker's shift give a numerical aggregate containing it twice, as the next subsection shows. The order-theoretic content of the construction is that the observation of a contribution is a fact that can be repeated without changing the record. The arithmetic content is that the use of a contribution in a sum is an event that cannot. In the vocabulary of replicated data, the record is a state-based object with a semilattice join, while the aggregate is an operation-based object whose operations are not idempotent and must be delivered exactly once [24].

The record of a closed round has a further element. A closure verdict c is itself an event of the root, issued at a cutoff. It converts selected NULL entries of the root's view into an entry ABSENT(c), which asserts that the root held no accepted submission for the identifier at that cutoff and that the policy excluded it. The entry is a claim about the root's view and the policy, and it says nothing about whether a submission for the identifier exists elsewhere. It is consistent with NULL and inconsistent with the root's acceptance of the same submission in the same phase. A submission that reaches the root after the cutoff is a record of a later phase and does not restrict backward into the closed round. The passage from an open record to a closed record is a transition in the protocol, and its treatment must preserve the order in which evidence became available. An observer at an earlier stage cannot restrict a later finalized verdict by treating it as though it were already known, so the open and closed phases are separate presheaves connected by the closure transition and by no restriction map. The accepted set of a committed round is therefore fixed by the closure record, while the historical record of what later arrived can continue to grow. An authorized closure attests what the root accepted. That the root included every submission it was obliged to accept is a further claim, and establishing it requires evidence about the completeness of the root's observations, for example attestations from other observers whose events show receipt of submissions that the root omitted.

The routing hierarchy and the cover serve different purposes. A routing tree describes the path by which a record or summary reaches its destination. A cover describes which observation domains jointly support a claim. Two different trees can convey the same information. One tree can also carry reports whose observation domains overlap, as when a worker retransmits through a second relay. A theory of synchronization must therefore account for identity and overlap independently of network topology. The powerset construction gives a case in which gluing can be proved directly, and a later development can enrich the domains with checkpoint histories and causal observations [12], which requires restriction maps with precise semantics.

3.1 Summaries Do Not Restrict

The submission views of a round form a sheaf, and in particular a presheaf, since a view over U restricts to every $V \subseteq U$. The numerical summaries used in reduction do not. For an accepted set A with prescribed weights w_i and shifts Δ_i fixed by identity, the summary

$$R(A) = \Big(\sum_{i \in A} w_i \Delta_i,\ \sum_{i \in A} w_i\Big)$$

determines nothing about $R(B)$ for $B \subset A$. The summary behaves covariantly: it extends by addition over disjoint unions. The target monoid is not idempotent, so R does not factor through the join of Proposition 1(b) when the covering observations overlap.

Proposition 2 (Inclusion and exclusion)**.** *For accepted sets $A, B \subseteq E$,*

$$R(A \cup B) = R(A) + R(B) - R(A \cap B),$$

and with positive weights, $R(A) + R(B) = R(A \cup B)$ holds if and only if $A \cap B = \emptyset$.

Proof. Both sides are sums over identifiers, and an identifier in $A \cap B$ is counted twice in $R(A) + R(B)$ and once in $R(A \cup B)$. The weight component of $R(A \cap B)$ is positive exactly when $A \cap B$ is nonempty. □

A small case shows the failure. Let three workers have weights $1, 1, 2$ and scalar shifts $2, -1, 3$, and let one relay aggregate the first two and a second relay aggregate the last two, so that the second worker retransmitted through both. The summaries are $R(A) = (1, 2)$ and $R(B) = (5, 3)$, and their sum is $(6, 5)$, with normalized shift 1.2. The true summary of the three accepted workers is $R(A \cup B) = (7, 4)$, with normalized shift 1.75, and the overlap summary is $R(A \cap B) = (-1, 1)$. The identity of Proposition 2 recovers $(7, 4)$ from the two relay summaries and the overlap. The records seen by the two relays agree exactly on the second worker's payload, so the sheaf condition is satisfied, and the arithmetic is nonetheless wrong.

The proposition shows that the weight component of a summary detects overlap only for a party who independently knows $R(A \cup B)$. A relay or a root that receives $R(A)$ and $R(B)$ alone has no such knowledge. The reuse of a contribution identity, which makes duplicates detectable at a single receiving point, does not help unless the identity is available at the merge, and the pair of weighted sum and total weight discards it.

A relay interface that supports the routing-invariance result therefore needs one of three additions, and the first is the one the specification adopts. Each summary carries the set of accepted contribution identifiers beneath it, so that a summary is a pair $(A, R(A))$, and a merge is defined only when the two sets are disjoint. The parent checks disjointness of the reported sets, and whether each remote payload was computed from exactly its reported set depends on the relay's evidence, as Section 5 develops. The set can be encoded as an explicit list or as a bitmap over the roster. A bare hash of the set would not serve, since disjointness of two sets cannot be tested from their hashes. If the sets overlap, the parent cannot deduplicate the numerical sum from the two summaries alone, because Proposition 2 requires the overlap summary $R(A \cap B)$. The parent must therefore recover the leaf contributions below the overlap, or ask a child for a revised summary that excludes the shared identifiers, and until it does

so the merge is undefined and the round does not close on that subtree. The second addition is a structural routing rule under which disjointness holds by construction: each relay accepts contributions only from the identifiers assigned to its subtree, and a retransmission through another relay is rejected on arrival. This rule gives up failover through alternative paths. The third is to forward the summary of the overlap and apply Proposition 2, which requires a party to hold $R(A \cap B)$ and is available mainly where overlap is deliberate, for example when two relays redundantly aggregate the same leaves so that their results can be compared.

Remark 1. The tag costs one bit per roster entry, while the weighted sum is a vector with one entry per model parameter. When the roster is small relative to the parameter dimension, the tagged interface preserves essentially all of the communication saving of hierarchical reduction.

4 Functional Composition Through the Hierarchy

The category of authorized histories has checkpoints as objects and authorized transitions as generating arrows [2]. Identity arrows and finite composition then give histories of continuation. A committed transition has a source and a target, so two transitions compose when the target of the first is the source of the second. Parallel worker shifts belong to a different construction. They are contributions indexed by a shared base and are combined by a reduction within that index.

The two constructions meet in the affine case through a functor. Let **Hist** be the category of authorized histories and let $B\mathbb{R}^d$ be the one-object category whose arrows are the vectors of $\mathbb{R}^d$ under addition. Sending each committed transition from C_v to C_{v+1} to the vector $\theta_{v+1} - \theta_v$ defines a functor, and functoriality is the telescoping identity of Section 2. The functor forgets everything except the parameter displacement, and in particular it forgets the certificate, the optimizer state, and the accepted set. The composition of certificates along a history is a separate structure that records, for each step, the evidence under which it was authorized, and the paper's later sections concern that structure.

In the affine case, the summary of an accepted set A is $R(A)$ as in Section 3. For disjoint accepted sets A and B,

$$R(A \sqcup B) = R(A) + R(B),$$

where addition acts componentwise. The root normalizes the first component by the second when the total weight is positive. This expresses the relevant functional composition. A relay can reduce its own leaves and forward the result because the summary preserves the information needed by the next reduction. A site average by itself generally loses that information. If one site holds two workers of weight one with shifts p and q and another holds one worker of weight one with shift r, the flat weighted average is $(p+q+r)/3$ while the average of the two site averages is $\big((p+q)/2 + r\big)/2$, which gives the lone worker weight one half in place of one third. Averaging site averages gives different effective worker weights whenever sites contribute unequal total weight, and the forwarded total weight is what allows a parent to correct for it.

Theorem 1 (Routing invariance)**.** *Given the same uniquely counted accepted contributions, the same base and coordinate representation, and the same prescribed weights, every reduction tree computes the same summary in exact arithmetic. With tagged summaries the uniqueness hypothesis is checked at each merge: every tree in which each merge is defined produces a*

summary tagged with A, the union of the leaves below the root, and its numerical component is $R(A)$ provided that each reported summary equals the sum of the leaves in its reported set.

Proof. The proof is by induction over the tree. Each leaf carries the weighted shift and weight of one contribution. If each child summary equals the sum of its descendant leaves, the parent summary equals the sum of all leaves below it. Disjointness at each merge, which the tagged interface requires as a precondition, ensures that no term is counted twice. Associativity and commutativity then give the same root value for every partition and ordering of the accepted leaves. □

Uniqueness is checkable from explicit tags in the tagged interface, and it can also follow from the structural routing rule. In either case the check concerns the reported sets. That a remote relay computed its numerical payload from exactly its reported set is the calculation claim of Section 5, and a parent that checks tags does not thereby establish it. A weighted sum and total weight alone determine the arithmetic result once the accepted leaves are fixed, and they cannot show that two child summaries have disjoint contributors. This theorem is conditional on a fixed accepted set. Rerouting may affect arrival times and thereby change which contributions a deadline admits. The theorem also concerns the mathematical sum. Bitwise reproducibility in floating-point arithmetic, where addition is not associative [8], requires a specified reduction order, such as a fixed tree shape with a fixed order of operands, or an accumulation method with a suitable reproducibility guarantee. Topology can change which contributions reach the root before a deadline under either interface. A structural routing rule that rejects out-of-subtree contributions gives up alternative paths, so a link failure can exclude a subtree. Multiple permitted paths with tags can improve delivery under failure, but arrival times still depend on the route taken, and so does the accepted set that a deadline admits. These qualifications locate precisely where network topology can affect the result.

An intermediate training site does more than route a sum when it synchronizes internally and then continues training, as in hierarchical schemes with separate edge and cloud intervals [15]. Its subsequent optimizer steps depend on the state produced by the internal synchronization. Its final contribution is a new shift from the global checkpoint to the endpoint of that nested process. Consequently, reorganizing pure relays preserves reduction under the theorem's hypotheses, while moving an actual training boundary can change the optimization dynamics. This distinction lets a global hierarchy accommodate several synchronization frequencies without implying that every hierarchy implements the same algorithm. The nested case also compounds the obstruction of Section 6, since each internal synchronization boundary applies its own saturation of local curvature before the next level sees the shift.

5 Dependent Types, Round Completion, and the Scope of a Certificate

The ordinary aggregation interface can express its requirements through indexed types [19, 9]. A contribution has a type $\mathrm{Shift}(C_v, \sigma)$, where C_v is the base checkpoint and σ is the model schema. A tagged summary has a type $\mathrm{Summary}(C_v, \sigma, A)$ indexed additionally by its contributor set $A \subseteq E$. A leaf constructor takes a validated contribution with identity

i and returns an element of $\mathrm{Summary}(C_v, \sigma, \{i\})$. The merge operation takes elements of $\mathrm{Summary}(C_v, \sigma, A)$ and $\mathrm{Summary}(C_v, \sigma, B)$ together with a proof of $A \cap B = \emptyset$ and returns an element of $\mathrm{Summary}(C_v, \sigma, A \sqcup B)$. The proof is discharged by a decidable test on the tags where the tagged interface is used, and by an argument from the routing rule where the structural interface is used. The commit constructor takes a summary, a closure record for the round, and evidence that the closure satisfies the declared policy, and returns a successor checkpoint together with its certificate. The commit constructor requires that the contributor set indexing the summary equal the accepted set named in the closure record.

A common base checkpoint and the disjointness of contributor sets license different operations. Sharing the index C_v licenses comparison of two shifts, since both are expressed against the same origin. It does not license adding two summaries, because the sum counts a contribution once only if no contribution lies below both. Evidence of disjointness licenses addition, and the merge constructor demands that evidence as an argument. The records of Section 3 and the typed merge then fit together in a single result.

Proposition 3 (Certified counting)**.** *Let $\{s_j\}$ be pairwise consistent views with joined view s, as in Proposition 1(b). Let T be a tree evaluated by one party, whose leaves are leaf constructors applied to the validated submissions $s(i)$ for $i \in L$ and whose internal nodes are typed merges. Then every identifier of L labels exactly one leaf, the root has type* $\mathrm{Summary}(C_v, \sigma, L)$*, and its numerical component is $(\sum_{i \in L} w_i \Delta_i, \sum_{i \in L} w_i)$, where w_i and Δ_i are read from $s(i)$. The value does not depend on which observer supplied the entry $s_j(i)$, nor on the shape of T.*

Proof. Pairwise consistency makes the immutable fields of the joined entry $s(i)$ single-valued, so the weight and payload of each leaf are the same whichever observer's copy was used. A merge is defined only for disjoint contributor sets, so by induction on the tree the contributor set of every node is the disjoint union of the identifiers at its leaves, and no identifier occurs at two leaves. The numerical component of a leaf is the weighted shift and weight of its submission, and a merge adds components. The root is therefore the sum over its leaves, each counted once, and associativity and commutativity of addition remove any dependence on the shape of T. □

The proposition separates what the record layer supplies from what the arithmetic layer supplies. Compatible observations reconstruct each submission record, and the typed merge certifies that a summary evaluated from validated leaves counts each accepted submission once. It does not extend to a summary received from a remote relay. A parent can verify that two reported contributor sets are disjoint, whereas the claim that each reported numerical payload was calculated from exactly its reported set rests on the relay's evidence, as the next subsection describes. Neither layer establishes that the aggregate advances the declared objective, which is the subject of Sections 6 and 7.

There is a difference between a type index and the evidence that establishes it at a network boundary. An incoming packet can claim any checkpoint identifier or any contributor set. Decoding and validation must establish that the payload has the declared representation and satisfies the integrity checks before the program introduces it as a typed contribution. The type system then preserves the relationship between a validated object and its origin throughout the rest of the computation. It does not establish that a validated object is truthful, and the distinction recurs below.

Null Convention Logic supplies a discipline for distinguishing the completion state of an input from its numerical content [6, 7], and the information order of Section 3 gives it a precise form. An identifier of the roster is NULL in a section when the observer holds no record for it. A received tensor containing only zeros is DATA. It is a complete contribution and, if accepted, participates with its prescribed weight. A complete contribution subsequently rejected by policy has a recorded disposition. After closure, an identifier for which the root held no accepted submission and which policy excluded has the entry ABSENT(c). A timeout is an event of a named observer, as in Section 3, and it does not by itself change the NULL status of the identifier for any other observer. The algebraic identity $(0, 0)$ is useful when computing a reduction, but it cannot serve as the observation that a worker replied. The return of a branch to NULL between waves, which Null Convention Logic uses for handshaking, corresponds here to the boundary between rounds, and sections are therefore indexed by round.

Input completeness and policy closure. The circuit-level account of Null Convention Logic gives this discipline a sharper form [11, Sec. 2.1, p. 12]. A stage is input complete when its output enters DATA only after all of its inputs have entered DATA, and returns to NULL only after all of its inputs have returned to NULL, with a NULL wave front separating successive DATA wave fronts. The corresponding rule for a relay with a fixed roster is that it can attest full receipt once it holds a complete contribution from every expected input. A relay or root that closes with a quorum makes a different claim. It attests that the contributions available at the cutoff satisfy the declared closure policy, and its closure record identifies the accepted set and the disposition of the remaining inputs. Two events that the word completion would otherwise merge are therefore kept apart: full receipt, after which no input is outstanding, and policy closure, which authorizes proceeding with inputs outstanding. The closure record states which of the two issued the verdict, so that a later reader of a certificate can tell whether every expected input arrived or the policy authorized proceeding without some of them.

The basic element in Kim's account is the M-of-N threshold gate with hysteresis. Its output enters DATA when the number of DATA inputs reaches the threshold and returns to NULL only when all inputs are NULL [11, Sec. 2.2, p. 22], and weighted variants let inputs count with different weights [11, Sec. 2.2, p. 25]. Full receipt over n children corresponds to threshold n, the case in which the gate waits for every input. A quorum over total accepted weight resembles a weighted threshold gate in its firing condition. Persistence is implemented by the round protocol. Once the root records closure, the accepted set is fixed. Late messages can extend the historical record for that round, and the next round evaluates contributions under a fresh identifier. The correspondence is structural. The timing assumption on gate feedback paths and the orphan-path constraint that Kim identifies are properties of circuits [11, Sec. 2.1, p. 12], and the paper claims neither for the network protocol, whose deadline is an external policy and not part of the gate.

The distinction also determines how a branch is reused. In a circuit, a NULL wave separates successive DATA waves. In the distributed protocol, the round identifier separates successive uses of a branch. Once round v has closed, a message tagged with v can extend the historical record for that round, while only a message tagged with $v + 1$ can complete the branch's input for the successor round.

Deadline and quorum rules belong to the distributed protocol. They determine when the

root may close a round while some branches remain NULL. The closure record is the join of the root's observations over the roster, together with a verdict that a stated policy predicate holds of that joined record and that assigns ABSENT(c) to the identifiers excluded at the cutoff. The quorum threshold is a statement about total accepted weight. When weights are prescribed as a function of identity, that total is determined by the contributor set and requires no further trust. When weights depend on completed work, the weight of each leaf is a claim by that leaf about the training it performed, and the quorum check inherits the status of that claim. Late delivery can extend the historical record of what happened, while the accepted set of the committed round stays fixed. Retransmission reuses a contribution identity so that acknowledgment loss does not give a worker a second vote in the reduction.

5.1 What a Certificate Can Establish

A commit certificate contains the transition record τ_v, which comprises the digest of the base checkpoint, the accepted set A, the identifier of the policy, the closure record, which states whether the verdict was issued under full receipt or by quorum, the digest of the aggregate, and a digest of the outer update rule together with its configuration. The certificate also carries the digest of the successor checkpoint, which commits in turn to τ_v and to the parent digest, so that no circularity arises. Base, accepted set and aggregate alone identify the inputs and an intermediate object of the step. Binding the successor and the update rule identifies the particular transition that the recipient is asked to trust. The claims the certificate can support are of different kinds, and the evidence for each differs.

Origin, uniqueness, eligibility and closure are claims about the record. Origin follows from validation at the boundary. Uniqueness of the reported sets follows from the disjointness test at each merge. Eligibility follows from the policy predicate applied to the joined record, and closure follows from the closure verdict, which attests what the root accepted and does not by itself show that the root accepted everything it was obliged to accept. The commit condition that ties these together is that the contributor set indexing the summary equal the accepted set named in the closure record. Each of these is established by the mechanisms of the preceding sections, under the assumption that the evidence at the boundary is itself authentic.

The claim that the aggregate equals $R(A)$ is a claim about computation and depends on a fault model that the protocol must state [13, 4]. Three regimes are available. Under the first, relays are trusted to report the sum of the contributions they received. The certificate then supports the calculation claim relative to that trust, and a recipient who does not share the assumption cannot infer it. Under the second, leaves sign their contributions and the root recomputes the aggregate from the signed leaves. The calculation claim is then established at the root, but the root receives every contribution, and the communication benefit of hierarchical reduction disappears for any claim that requires recomputation. Redundant aggregation by several relays over the same leaves, with comparison at the root, gives an intermediate regime in which the overlap of Proposition 2 is deliberate and serves as a check. Under the third, relays attach proofs that their summaries are correct sums of the signed leaves below them. An aggregate digest identifies the object to which the claim refers and does not itself establish that the object is the sum. Where the certificate is meant to establish arithmetic correctness independently, the recipient needs access to the corresponding evidence, whether an authenticated relay attestation, a recomputable set of leaves, or a proof. The existence

and cost of such proofs is outside the scope of this paper, and the interaction of such proofs with privacy-preserving aggregation, where relays by design see only masked contributions, is a further constraint [3].

Two further claims lie beyond any of these regimes. That a remote worker performed the training it reported requires evidence about the training itself, and the signature on a leaf establishes only who submitted it. That the successor model improved on the objective requires a statement about the objective and the dynamics, which is the subject of Sections 6 and 7. The certificate should therefore carry a statement of the fault-model regime under which it was issued, since that statement fixes what a later recipient can independently infer from it. This is the connection to portable warrant: the certificate travels with a successor checkpoint, and the regime recorded in it determines the scope of what the checkpoint warrants.

6 An Exact Obstruction to Long Local Rounds

The algebra of reduction leaves open whether the resulting global step is useful. A pair of scalar quadratic objectives is enough to expose the problem. Let

$$f_1(x) = \frac{a}{2}(x-u)^2, \qquad f_2(x) = \frac{b}{2}(x-v)^2, \qquad a, b > 0.$$

With equal site weights, the global objective is $F = (f_1 + f_2)/2$, whose minimizer is

$$x^\star = \frac{au + bv}{a+b}.$$

Starting from the same x, let each site take H gradient steps with step size η, chosen so that $0 < \eta < 1/\max(a, b)$. The endpoints are

$$x_1(H) = u + (1-\eta a)^H (x-u), \qquad x_2(H) = v + (1-\eta b)^H (x-v).$$

Define $d_a(H) = 1-(1-\eta a)^H$ and $d_b(H) = 1-(1-\eta b)^H$, both in $(0,1)$. Averaging the endpoint shifts yields the round map

$$G_H(x) = x - \frac{1}{2}\big[d_a(H)(x-u) + d_b(H)(x-v)\big].$$

Proposition 4 (Fixed point of the round map)**.** *For each $H \geq 1$ the map G_H is a contraction with contraction factor $1 - \frac{1}{2}(d_a + d_b)$ and fixed point*

$$x_H = \frac{d_a(H)u + d_b(H)v}{d_a(H) + d_b(H)}.$$

The displacement from the minimizer of F is

$$x_H - x^\star = \frac{(u-v)\,(b\,d_a(H) - a\,d_b(H))}{(a+b)\,(d_a(H) + d_b(H))}.$$

At $H = 1$ the coefficients are ηa and ηb, and $x_1 = x^\star$. As $H \to \infty$ both coefficients approach one, and x_H approaches $(u+v)/2$, with limiting displacement $(b-a)(u-v)/(2(a+b))$. This limit differs from $x^\star$ exactly when $a \neq b$ and $u \neq v$.

Proof. G_H is affine with slope $1 - \frac{1}{2}(d_a + d_b) \in (0, 1)$, so it is a contraction. Setting $G_H(x) = x$ gives $d_a(x-u)+d_b(x-v) = 0$, which yields x_H. Writing $x_H = v+(u-v)d_a/(d_a+d_b)$ and $x^\star = v + (u - v)a/(a + b)$ and subtracting gives the displacement. The displacement vanishes when $d_a/a = d_b/b$, which holds at $H = 1$ because both ratios equal η. For the limit, $(1 - \eta a)^H \to 0$ since $0 < \eta a < 1$, and likewise for b. Finally $(u + v)/2 - x^\star = (b - a)(u - v)/\big(2(a + b)\big)$. □

The dynamics can thus remain stable and converge systematically to a point that is wrong for the declared global objective. The example requires no stale checkpoints, missing workers, message corruption, or approximation error in the reduction. A numerical evaluation with $a = 1$, $b = 4$, $u = 0$, $v = 1$ and $\eta = 0.2$, for which $x^\star = 0.8$, gives the values in Table 1, and an iteration of the round map from an arbitrary start converges to x_H in each case.

H	x_H	$x_H - x^\star$
1	0.8000	0.0000
2	0.7273	−0.0727
5	0.5979	−0.2021
20	0.5029	−0.2971
100	0.5000	−0.3000

Table 1: Fixed point of the averaged round map for $a = 1$, $b = 4$, $u = 0$, $v = 1$, $\eta = 0.2$. The minimizer of the global objective is $x^\star = 0.8$ and the limit as $H \to \infty$ is 0.5.

The same argument gives the general form. For n sites with weights p_i, curvatures a_i and minimizers u_i, the fixed point is $\sum_i p_i d_i(H) u_i / \sum_i p_i d_i(H)$ with $d_i(H) = 1 - (1 - \eta a_i)^H$. The effective weight of site i is $p_i d_i(H)$. For small $\eta a_i H$ this is approximately proportional to $p_i a_i$, which is the weighting that the global objective assigns. For large H it saturates at p_i, and the curvature information is lost. This gives a sharper meaning to local drift: longer trajectories change how each local objective enters the effective global dynamics, replacing the original curvature weights by coefficients that saturate with the local horizon. The result is an exactly solvable case of the objective inconsistency identified for heterogeneous local work [26] and of client drift [10].

Scaling the averaged shift by a fixed positive outer learning rate leaves the zeros of that shift unchanged. It can alter stability and speed, but it cannot by itself remove the fixed-point bias in this example. The same holds for outer optimizers whose stationary points are exactly the zeros of the aggregate shift, which includes heavy-ball and Nesterov momentum applied to the negated shift [22]: momentum parameters can change the transient and the stability region, and they cannot move the fixed point.

A correction to the local direction changes the shifts entering the reduction and can therefore change the fixed point. The control-variate construction of [10] replaces the local gradient $\nabla f_i(y)$ by $\nabla f_i(y) - \nabla f_i(x) + \nabla F(x)$, where x is the round's base point. For this example, and with the exact global gradient at the base point, the correction is exact and it recovers the intended objective. The construction is idealized in that respect, since SCAFFOLD maintains control variates that estimate the gradient and does not assume access to the exact one at no cost.

Proposition 5 (Idealized control-variate correction)**.** *With the corrected local direction built from the exact global gradient $F'(x)$ at the base point, and outer step $\lambda > 0$, the endpoint shift of site 1 is $-\frac{d_a(H)}{a}F'(x)$ and that of site 2 is $-\frac{d_b(H)}{b}F'(x)$. The round map is*

$$\tilde{G}_H(x) = x - \frac{\lambda}{2}\Big(\frac{d_a(H)}{a} + \frac{d_b(H)}{b}\Big)F'(x),$$

whose unique fixed point is $x^\star$ for every H and every $\lambda > 0$, and which is a contraction when $\lambda(a+b)\big(d_a/a + d_b/b\big) < 8$.

Proof. For a quadratic, $\nabla f_1(y) - \nabla f_1(x) = a(y-x)$, so the corrected step is $y \leftarrow y - \eta\big(a(y-x) + g\big)$ with $g = F'(x)$. Writing $z = y - x$, the recursion $z \leftarrow (1-\eta a)z - \eta g$ has solution after H steps $z_H = -\frac{d_a(H)}{a}g$. Averaging the two shifts and applying the outer step gives $\tilde{G}_H$. The coefficient of $F'(x)$ is positive, so the fixed points are the zeros of F', which is $x^\star$. Since $F'' = (a+b)/2$, the slope of $\tilde{G}_H$ is $1 - \frac{\lambda}{4}(a+b)(d_a/a + d_b/b)$, which lies in $(-1, 1)$ under the stated condition. □

For $H = 1$ the contraction condition reduces to $\lambda\eta(a+b) < 4$, the familiar step-size bound for a function with second derivative $(a+b)/2$. In the numerical example above the corrected iteration converges to 0.8 for every H in Table 1. The proposition establishes the claim for two quadratic sites. If the root must obtain a fresh exact global gradient in every round, that exchange belongs in any later comparison of communication costs. Whether the same correction recovers the intended objective for non-quadratic local objectives, for optimizer state that persists across rounds, or for partial participation, is a further proposition to establish. The correction also changes the protocol: the root must distribute $\nabla F(x)$ or an estimate of it, each site must hold a local gradient at the base point, and the reported quantity remains a shift but one whose composition with the reduction has been altered. A local anchoring term has a different effect, since it changes the local objective and with it the point to which the local trajectory is attracted, which may recover the intended objective only under conditions on the anchoring coefficient [14].

7 Stability and Participation in the General System

A useful extension is to write the received aggregate as a reference step plus discrepancies with separate causes. Let $\Delta_{\mathrm{ref}}(\theta)$ be the update associated with the declared global objective under a specified reference method. Let $\Delta_{\mathrm{full}}(\theta)$ be the aggregate that the full eligible population would produce under the declared local procedure, and let $\Delta_{\mathrm{recv}}(\theta)$ be the aggregate actually received after closure. Then

$$\Delta_{\mathrm{recv}} - \Delta_{\mathrm{ref}} = (\Delta_{\mathrm{recv}} - \Delta_{\mathrm{full}}) + (\Delta_{\mathrm{full}} - \Delta_{\mathrm{ref}}).$$

The second difference captures the effect of local trajectories, and Section 6 computes it exactly in the quadratic case. The first difference captures the effect of participation and weighting. The two differences are not independent. The participation term is evaluated on a subset population whose local trajectories are those analyzed in the second term, so the decomposition

states obligations and does not by itself support separate bounds. The outer optimizer then maps the received aggregate and its stored state to the actual parameter transition.

This decomposition identifies several different theorem obligations. For smooth local objectives, one can ask how far the endpoint reduction departs from a reference descent direction as a function of local horizon, step size, and heterogeneity. For a deadline policy, one can ask whether the accepted population preserves the intended objective in expectation. For an outer optimizer with memory, one can study the response to sequences of biased or intermittent aggregate directions. Each analysis has different hypotheses, and a result about one layer does not automatically extend to the others.

Participation deserves particular attention because a successful protocol can systematically change the training objective. If slower sites hold a distinctive portion of the data and frequently miss the deadline, the root may repeatedly commit updates drawn from an unrepresentative subset. The extreme case is exact in the scalar example.

Proposition 6 (Systematic exclusion)**.** *If site* 2 *is excluded from every accepted set and the weights are renormalized over the accepted set, the fixed point of the round map is* u*, the minimizer of* f_1 *alone, for every* $H \geq 1$*.*

Proof. With only site 1 accepted, the round map is $x \mapsto x - d_a(H)(x - u)$, whose fixed point is u. □

The protocol in this case satisfies every guarantee of the record layer: each commit has a valid closure, a unique accepted set, and a correct aggregate. It nonetheless optimizes a different objective than the declared one. Correcting for participation requires knowledge of the participation mechanism. If site i is accepted independently of its shift with known probability q_i, then weighting an accepted shift by w_i/q_i makes the unnormalized aggregate an unbiased estimate of the full weighted aggregate, at the cost of added variance. The independence hypothesis fails exactly in the situation that matters, where lateness correlates with data or with the amount of local work, and in that situation the correction requires a model of the dependence. Weighting by completed work can reinforce the effect if the intended objective gives sites or examples a different relative importance. The correct weights follow from the objective and the sampling design. Throughput describes how much work arrived, while the objective determines what that work should represent.

These questions are amenable to symbolic analysis even when the complete deployment cannot be tested. One can construct cases where a deadline preserves progress but induces bias, cases where relay invariance holds but floating-point replay differs, and cases where the optimizer converges despite persistent disagreement between site updates. Such examples delimit the claims that a specification may make before an implementation exists.

8 The Running Experiment and the Prototype

The preceding sections separate what a protocol can guarantee from what an optimization analysis must establish. This section reports a small simulation that exercises both layers together, states what it does and does not show when read against the exact results above, and

specifies the prototype that would replace the modeled costs of the simulation with measured ones.

8.1 Setup of the Simulation

Two workers each hold the whole model, a multilayer perceptron with 32 inputs, two hidden layers of width 128 with GELU activations, and 8 outputs. Regression targets come from a fixed teacher network of the same shape with added noise. The two workers draw inputs from Gaussian distributions with different means, a mild form of the heterogeneity analyzed in Section 6, and the validation set is drawn equally from both. The simulation is implemented in PyTorch (version 2.15.0, CPU only). The local optimizer is AdamW with learning rate 2×10^{-3} and decoupled weight decay 0.01 [16], and each worker retains its optimizer state from one round to the next. This places the persistent-optimizer-state case, which Section 6 lists as an open proposition, inside the experiment, so that the simulation probes a regime the exact results do not cover.

A round proceeds as in Section 4. Both workers load the common checkpoint, take H local steps on their own shard, and return a dense shift. The root applies one of two outer rules. Under plain averaging, the new parameter point is the old point plus the mean shift. Under Nesterov momentum [21], the pseudo-gradient is $g = -\bar{\Delta}$, the velocity updates as $m \leftarrow \mu m + g$, and the parameter point moves to $\theta - \lambda(g + \mu m)$, where λ is the outer learning rate and μ the momentum coefficient. The default setting is $\lambda = 0.7$ and $\mu = 0.9$, the values used with an adaptive outer optimizer in low-communication language model training [5]. Both workers complete every round. There is no deadline and no quorum, so every run lies within the full-participation hypotheses of Theorem 1, with single-precision arithmetic in place of exact arithmetic.

Time is modeled and not measured. A round costs $Hs + c$ seconds, where $s = 0.25$ is the modeled time of one local step and $c = 2P/B + 2\ell$ is the cost of sending a shift up and a checkpoint down, with payload $P = 2000$ MB, bandwidth B and latency ℓ. The local-area link has $B = 10$ Gbit/s and $\ell = 0.5$ ms, so communication costs 3.2 s per round. The wide-area link has $B = 100$ Mbit/s and $\ell = 50$ ms, so communication costs 320.1 s per round. The payload is set independently of the size of the small model that is actually trained. This separates the communication cost of a large model from the optimization dynamics of a small one, and it leaves untested whether the dynamics of a small model carry over to a large one. The budget is 3000 local steps per worker. For each seed the target validation loss is 1.10 times the final loss of a reference run that averages parameters after every local step, with zero communication cost. The reference applies a separate AdamW step at each worker and then averages parameters, so it is per-step parameter averaging and is distinct from synchronous gradient averaging. Time to target is the modeled time at the first round boundary where the validation loss falls to the target or below, and a run that does not reach the target within the budget is recorded as a miss. Results are averaged over two seeds, each of which fixes a teacher, shard distributions and initialization.

8.2 Results across Synchronization Intervals

Table 2 gives the modeled cost of each horizon, and Table 3 reports the three horizons $H \in \{50, 200, 500\}$ under both outer rules on both links.

H	Rounds	Moved (GB)	LAN wait (s)	WAN wait (s)
50	60	480	192	19206
200	15	120	48	4802
500	6	48	19	1921

Table 2: Modeled cost of each horizon over the full budget of 3000 steps per worker. Rounds, bytes moved (both workers, both directions) and communication wait are functions of H, the link and the payload only, and do not depend on the outer rule or on the data.

Link	Outer rule	H	Reached	Time to target (s)	Final loss
LAN	average	50	2/2	581	0.0132
LAN	average	200	2/2	585	0.0134
LAN	average	500	2/2	641	0.0137
LAN	Nesterov	50	2/2	463	0.0114
LAN	Nesterov	200	2/2	612	0.0133
LAN	Nesterov	500	0/2	miss	0.0151
WAN	average	50	2/2	12306	0.0132
WAN	average	200	2/2	4071	0.0134
WAN	average	500	2/2	2226	0.0137
WAN	Nesterov	50	2/2	9812	0.0114
WAN	Nesterov	200	2/2	4256	0.0133
WAN	Nesterov	500	0/2	miss	0.0151

Table 3: Two-worker simulation with outer learning rate 0.7 and momentum 0.9 for the Nesterov rows. Times are modeled, with the costs of Table 2. Final loss is the validation loss at the end of the budget. All entries are means over two seeds.

On the local-area link, the interval has almost no effect on time to target, because communication is already inexpensive and the only visible consequence of a longer horizon is a slightly higher final loss. On the wide-area link, communication dominates and the interval decides the outcome. Under plain averaging, time to target falls from 12306 s at $H = 50$ to 4071 s at $H = 200$ and 2226 s at $H = 500$, a factor of about 5.5 between the shortest and longest horizon, while the final loss rises from 0.0132 to 0.0137, a difference that two seeds cannot distinguish from noise, and the bytes moved fall from 480 GB to 48 GB. The saving in bytes and waiting is fixed by the cost model, since the number of rounds is the budget divided by H and each round has a fixed price. The empirical content of the table is the optimization side: that the target is still reached, and at what cost in final loss, when exchanges become ten times rarer.

The Nesterov rows show a different pattern. At $H = 50$ momentum helps, with a time to target of 9812 s against 12306 s and a final loss of 0.0114 against 0.0132. At $H = 200$ it is slightly slower than averaging, and at $H = 500$ it misses the target in both seeds, with a final

loss of 0.0151. The longest horizon is the one that matters most on the wide-area link, which is why the miss is more than a curiosity.

8.3 Outer Optimizer Sweep at the Longest Horizon

A miss at $H = 500$ could reflect the local shifts, the outer step, or their interaction, so the outer parameters were swept at that horizon on the wide-area link with the same seeds and budget. With 3000 steps and $H = 500$ there are six rounds of 445.1 modeled seconds each, so times to target fall on a grid of multiples of 445.1 s.

Outer rule	(λ, μ)	Reached	Time to target (s)	Final loss
average	—	2/2	2226	0.0137
Nesterov	$(0.1, 0.0)$	0/2	miss	0.1013
Nesterov	$(0.1, 0.5)$	0/2	miss	0.0470
Nesterov	$(0.1, 0.9)$	0/2	miss	0.0452
Nesterov	$(0.3, 0.0)$	0/2	miss	0.0243
Nesterov	$(0.3, 0.5)$	0/2	miss	0.0151
Nesterov	$(0.3, 0.9)$	0/2	miss	0.0370
Nesterov	$(0.7, 0.0)$	2/2	2671	0.0142
Nesterov	$(0.7, 0.5)$	2/2	2226	0.0135
Nesterov	$(0.7, 0.9)$	0/2	miss	0.0151

Table 4: Outer parameter sweep at $H = 500$ on the wide-area link, with six rounds of 445.1 modeled seconds. The average row is the control. Times of 2226 s and 2671 s correspond to the fifth and sixth rounds. Entries are means over two seeds.

No setting beats the control. One setting matches it, with $(\lambda, \mu) = (0.7, 0.5)$ reaching the target at the fifth round in both seeds and ending at a final loss of 0.0135 against 0.0137, a difference too small to distinguish with two seeds. At $\lambda = 0.7$ momentum does help relative to its absence, since $\mu = 0$ needs a sixth round, and a moderate coefficient helps while a large one fails. The settings with $\lambda = 0.1$ or 0.3 miss in both seeds. These misses are consistent with slow progress under a six-round budget, since a damped outer step covers less distance per round, and the experiment does not establish that damping alone caused them. The miss at $(0.7, 0.9)$ is different, since the outer learning rate equals that of the successful runs and only the momentum differs. In steady state heavy-ball momentum multiplies the effective step by roughly $1/(1 - \mu)$, which is ten at $\mu = 0.9$ and two at $\mu = 0.5$, so overshoot of the outer iteration is the natural account of that miss. The data are consistent with this account and do not establish it.

8.4 The Fixed-Point Diagnostic

The framework suggests a diagnostic that separates the explanations left open by the outer-parameter sweep, and it can be stated and verified symbolically. Fix a reference solution $\theta^\star$, taken to be a minimizer of the declared global objective, together with the full eligible roster, a specified data stream for each site, and a specified initial optimizer state at each site. Write $\Delta_i^H(\theta^\star)$ for the shift that site i returns after H local steps from $\theta^\star$ under that specification,

and define

$$D_H(\theta^\star) = \frac{\sum_i p_i \, \Delta_i^H(\theta^\star)}{\sum_i p_i},$$

the full-participation mean of the local shifts, with the prescribed weights p_i.

Proposition 7 (Fixed-point diagnostic)**.** *Under the stated specification the plain averaging round map satisfies $G_H(\theta^\star) = \theta^\star + D_H(\theta^\star)$. (i) If $D_H(\theta^\star) \neq 0$, one round of plain averaging moves the system away from $\theta^\star$. The first round of outer momentum started from zero velocity also moves it, by $\lambda(1+\mu)D_H(\theta^\star)$. (ii) If $D_H(\theta^\star) = 0$, then $\theta^\star$ is a fixed point of the parameter map for the stated initialization. (iii) If the sites carry optimizer state m_i across rounds, the round acts on the full state $(\theta, m_1, \ldots, m_n)$, and D_H depends on the optimizer states. A fixed point of the full system requires $D_H(\theta^\star; m) = 0$ together with the reproduction of every optimizer state, in the sense that each m_i at the end of the round equals m_i at its start. For stateless local rules this condition is vacuous. For AdamW it asks that the moment estimates reach a stationary regime within the round, and under random data it can hold at most in distribution. Stability of such a fixed point concerns the response of the full state to perturbations, and $D_H = 0$ does not settle it.*

Proof. By definition the plain averaging step adds the weighted mean shift to the current point, which at $\theta^\star$ is $D_H(\theta^\star)$. This gives the identity and the first two statements. For outer momentum with zero initial velocity the pseudo-gradient is $g = -D_H(\theta^\star)$, the velocity becomes g, and the step is $-\lambda(g+\mu g) = \lambda(1+\mu)D_H(\theta^\star)$. With persistent optimizer state the round map is $(\theta, m) \mapsto (\theta + D_H(\theta; m), m^{\mathrm{end}}(\theta; m))$, and a point is fixed exactly when both components are fixed. □

When the local data stream is random, D_H is a random vector and the statements apply to its expectation, so a vanishing expected value does not make $\theta^\star$ a fixed point of the stochastic process. The scalar example of Section 6 shows the distinctions exactly. With equal weights, no persistent optimizer memory, and $x^\star = (au + bv)/(a + b)$,

$$D_H(x^\star) = \frac{(u - v)\big(b \, d_a(H) - a \, d_b(H)\big)}{2(a + b)},$$

and since G_H is affine with slope $1 - \frac{1}{2}(d_a + d_b)$, the displacement of its fixed point satisfies $x_H - x^\star = 2D_H(x^\star)/(d_a + d_b)$. The diagnostic vanishes at $H = 1$. For $a \neq b$ and $u \neq v$ it is nonzero for every $H \geq 2$, because $d_a(H)/a$ is strictly decreasing in a once $H \geq 2$. In the numerical example of Table 1 it equals -0.0480, -0.1690, -0.2954 and -0.3000 at $H = 2$, 5, 20 and 100, which have the sign and ordering of the tabulated displacements. For the idealized control-variate round of Proposition 5 every site's shift is proportional to $F'(x)$, so the diagnostic is zero at $x^\star$ for every H. A nonzero value at the reference solution is therefore a statement about the round map and holds for every outer optimizer whose stationary points are the zeros of the aggregate shift, which separates bias from the transient and stability effects of the outer step. In practice $\theta^\star$ is known only approximately, for instance as the final iterate of a long per-step reference run. A nonzero D_H at such a point then mixes two effects, continued descent from a point that is not stationary and drift away from a point that is. The sign of the change in the objective under the mean shift separates them. A decrease indicates the first, and an increase beyond the noise of the estimate indicates the second. The diagnostic runs below estimate D_H in this way.

8.5 Diagnostic and Heterogeneity Runs

Two further configurations address the limits of the first simulation, and each is run with and without optimizer memory. The strong configuration separates the shards more sharply: the input means of the two workers are drawn at four times the previous scale, and each worker's targets come from its own teacher, the common teacher plus a perturbation of relative scale 0.5 on every weight, so that the local objectives have different minimizers and different curvatures. The memoryless configuration replaces AdamW by plain stochastic gradient descent with learning rate 0.05, no momentum and no weight decay, so that no optimizer state is carried between rounds. Every other element is as before: plain averaging, a budget of 3000 steps, the wide-area cost model, and a target of 1.10 times the final loss of the per-step reference.

For each configuration and seed, the final global iterate of the per-step reference run is taken as the reference solution $\theta^\star$, an approximate stationary point and not an exact minimizer. The quantity $D_H(\theta^\star)$ of the preceding subsection is estimated from 16 independent draws of the local data streams. In each draw both workers start at $\theta^\star$ and take H local steps, and the mean shift is recorded. For AdamW the per-worker optimizer state at the end of the reference run is restored, which is the persistent-state regime of the simulation, and a second set of draws starts from fresh optimizer state. Table 5 reports the norm $\|\bar{D}_H\|$ of the mean over draws, the standard error SE of that mean taken as the root of the summed coordinate variances divided by $\sqrt{16}$, and the relative change $\Delta L/L$ of the validation loss when the mean shift is applied to $\theta^\star$. When the true mean is zero, $\|\bar{D}_H\|$ is of the order of SE, so a ratio near one is consistent with no systematic shift. The AdamW rows use the persisted state, and the fresh-state draws give the same signs and ordering, with norms that differ by up to a factor of 1.6.

Shards	Local rule	H	Reached	Time (s)	Final	Ref.	$\|\bar{D}_H\|$	SE	$\Delta L/L$
mild	AdamW	50	2/2	12306	0.0132	0.0133	0.43	0.15	−8.5%
mild	AdamW	200	2/2	4071	0.0134	0.0133	0.90	0.23	−9.0%
mild	AdamW	500	2/2	2226	0.0137	0.0133	1.65	0.28	−9.8%
mild	SGD	50	2/2	12971	0.0471	0.0470	0.023	0.004	−0.5%
mild	SGD	200	2/2	3886	0.0475	0.0470	0.085	0.007	−1.6%
mild	SGD	500	2/2	2003	0.0482	0.0470	0.20	0.008	−3.5%
strong	AdamW	50	0/2	miss	0.0416	0.0334	0.30	0.10	+6.6%
strong	AdamW	200	0/2	miss	0.1046	0.0334	0.79	0.16	+65.5%
strong	AdamW	500	0/2	miss	0.2042	0.0334	1.61	0.20	+171.8%
strong	SGD	50	2/2	14967	0.0733	0.0713	0.17	0.017	+1.8%
strong	SGD	200	0/2	miss	0.1243	0.0713	0.22	0.015	+3.8%
strong	SGD	500	0/2	miss	0.2144	0.0713	0.32	0.015	+7.5%

Table 5: Plain averaging on the wide-area link at three horizons, under mild and strong shard heterogeneity, with AdamW and with memoryless SGD. Time is the modeled time to target, final is the final validation loss, and ref. is the final loss of the per-step reference. The remaining columns estimate $D_H(\theta^\star)$ at the reference solution from 16 data-stream draws. Entries are means over two seeds. The mild AdamW rows reproduce Table 3.

Under mild heterogeneity the mean shift at the reference is nonzero in every row and grows with H, with a ratio to its standard error between about three and twenty-four. Applying it lowers the validation loss in every row, by 0.5 to 3.5 percent with SGD and by 8 to 10 percent with AdamW. The reference iterate is therefore not stationary, and the nonzero shift

is continued descent. It is not evidence of bias. Under strong heterogeneity the sign reverses. Applying the mean shift raises the validation loss in every row, by 1.8 to 7.5 percent with SGD and by 7 to 172 percent with AdamW, and the increase grows with H. The final losses move the same way. With AdamW the final loss is 0.0416, 0.1046 and 0.2042 at $H = 50$, 200 and 500 against a reference of 0.0334, and the target is missed at every horizon. With SGD the final loss is 0.0733, 0.1243 and 0.2144 against 0.0713, and the target is reached at $H = 50$ and missed at the two longer horizons. The memoryless runs show the same pattern as the AdamW runs, so the effect does not depend on optimizer memory. This is the behaviour that the scalar construction predicts when curvatures and minimizers differ across sites.

8.6 Reading the Results against the Theory

Section 6 shows that scaling the averaged shift and applying momentum to it leave the zeros of that shift unchanged. The outer-parameter sweep therefore varies the transient and the stability of the outer iteration and cannot move a fixed point that the horizon induces. A miss at $H = 500$ is consistent with a biased fixed point, with an unstable or slow outer iteration, or with both, and the sweep alone does not separate them. The result under mild heterogeneity is narrow. Plain averaging at $H = 500$ brought the validation loss into the prescribed sublevel set, within ten percent of the per-step reference, in both seeds and within six rounds. That is a statement about where an iterate entered a set. It does not bound the distance to a limiting fixed point, establish that a limit exists, or show that the persistent AdamW state has equilibrated. The final losses under averaging, 0.0132 at $H = 50$ and 0.0137 at $H = 500$, are consistent with a small residual bias and equally with seed variation.

The diagnostic runs supply what the sweep could not. In the mild configuration the sign of the loss change under the mean shift indicates remaining descent from an unconverged reference, and the run supports no claim of bias. In the strong configuration the mean shift moves the iterate away from the reference and raises the loss, and the loss grows with the horizon for both local rules. Because a stateless local rule is covered by the diagnostic proposition in expectation, the memoryless rows show that this drift arises from the interaction of local horizon and heterogeneity, as the exact example predicts, and not from AdamW memory. The exact example gives a scale for comparison. In Table 1 the curvatures differ by a factor of four and the minimizers are one unit apart, and the diagnostic of the scalar example reaches -0.30 against a minimizer at 0.80. The strong configuration is the closer analogue, and the mild one is milder in both respects.

Several limits of the evidence deserve to be stated plainly. The reference solution is the final iterate of a stochastic run and not an exact minimizer, so the diagnostic is read through the sign of the loss change and not through the vanishing of the shift. Each estimate uses 16 draws, and some ratios of the shift to its standard error are only about three. The experiment has two seeds and six rounds at the longest horizon, so differences of a few percent in final loss carry no weight, and time to target is quantized to round boundaries. The modeled communication savings are a consequence of fewer exchanges under the chosen cost model and would hold for any run with the same number of rounds. A dense shift at the same precision has the same size as the parameter vector, so the simulation says nothing about compressed shifts, and actual transfer times are left to the prototype.

8.7 The Prototype

The prototype replaces modeled costs with measured ones and exercises the protocol layer that the simulation omits. It consists of three programs: a worker that loads a checkpoint, trains locally with AdamW and reports a based shift, a relay that validates, reduces and forwards summaries and that can also run as the root, and a checkpoint store. The first topology has three trusted workers, two behind a relay and one connected directly to the root, with a model small enough to fit on each worker. The root applies a configurable outer update that supports both plain averaging and Nesterov momentum, with momentum placed at the root only, so that the optimizer remains the one analyzed in the preceding sections. The initial contribution weight is the number of completed examples, recorded with each committed checkpoint together with the participating workers. By Section 5 that weight is a claim by the worker about its own training, and the first prototype accepts it because its workers are trusted. The certificate therefore records the first fault-model regime, in which relays and workers are trusted to report faithfully.

The relay interface carries tagged summaries, with a bitmap over the roster accompanying each weighted sum and total weight, and a merge is defined only for disjoint tags. The structural rule of Section 3, in which each relay accepts only identifiers assigned to its subtree, is available as an alternative configuration. A checkpoint is identified by a digest of the full tuple of Section 2, and round identifiers and contribution identifiers make retransmission idempotent. Completion follows the Null Convention discipline of Section 5: a branch is NULL until it holds a complete contribution, a received zero tensor is DATA, and the closure record assigns ABSENT(c) to identifiers excluded at the cutoff and states whether closure was by full receipt or by quorum.

Three acceptance criteria govern the first version. The first is routing invariance. The same saved worker shifts are fed through a flat topology and through the two-level topology, and the aggregates must agree within the configured numerical tolerance. A negative control accompanies it. A worker's contribution is retransmitted through both relays, and the test confirms that the untagged weighted sum counts it twice, as in the example following Proposition 2, while the tagged merge rejects it. If bitwise replay is required, the reduction order and the accumulation precision are fixed, as Section 4 states. The second criterion is closure under failure. A worker is disconnected, and the remaining branches must close the round under the quorum policy. A shift that arrives late must leave the committed checkpoint unchanged and appear only in the historical record, and the disconnected worker must be able to rejoin from the latest checkpoint. The third criterion is the measured comparison. Synchronization intervals and outer optimizers are compared on a bandwidth-limited link, and the measurements are the wall-clock time to a fixed validation loss, the bytes transferred across sites, and the time the accelerators spend idle. These are the quantities that Table 3 models, so the comparison tests whether the modeled ordering of horizons survives real transfer times.

The prototype is specified and has not been built, so the paper reports no measurements from it. The criteria are stated in advance so that each result will confirm or limit a specific proposition of the preceding sections.

8.8 Remaining Diagnostics

Four further diagnostics follow from the framework and have not been run. The first sweeps the local horizon with and without a control-variate correction while holding the outer optimizer fixed, which separates the effect of local trajectories from the effect of the outer step. The second varies participation through the deadline, with sites assigned to the slow and fast groups according to their data, and compares the received aggregate with the full aggregate at the same checkpoint. The third measures the byte overhead of tagged summaries relative to the dense shift, which tests the claim of the remark in Section 3. The fourth replays a recorded history from its certificates under each fault-model regime and reports which claims can be reconstructed. Each of these either confirms or limits a specific proposition of the preceding sections, and none requires the symbolic development to be extended.

The paper uses the simulation as a running example alongside the scalar construction. The simulation exhibits behaviour in a nonlinear model with optimizer memory. The quadratic construction isolates a mechanism exactly. The formal protocol describes the conditions under which either experiment can be replayed as an authorized history, and the prototype is the setting in which that replay is carried out on real links. Each method answers a question that the others leave open.

9 The Scope of the Claims

The results of the paper differ in status, and the differences are worth stating together. The routing theorem is a theorem about exact summation over a fixed accepted set, and with tagged summaries its uniqueness hypothesis is enforced by the interface. The descent proposition is a theorem about the powerset construction and does not yet extend to enriched domains with causal structure, and it concerns submission records only, since receipt, rejection and timeout are observer-indexed events that enter neither the sheaf nor the information order. The certified counting proposition is exact for a tree evaluated by one party from validated leaves, and it does not cover remote summaries, whose calculation claim rests on relay evidence. The certificate binds the transition, and completeness of the root's observations is a further claim that it does not establish. The fixed-point diagnostic is exact for deterministic local maps and applies to the expected shift when local data are random. The inclusion and exclusion identity is exact and shows that the plain summary interface is insufficient. The statements about certificates are conditional on a fault model, and the paper identifies three regimes without proving results about the second and third. The scalar obstruction and its idealized correction are exact for two quadratic sites, with a straightforward extension of the fixed point to n sites. The decomposition of the aggregate error is a framework for stating obligations, and the corresponding bounds for smooth non-quadratic objectives, intermittent participation and optimizer memory remain open. The simulation results are modeled and cover two seeds. On the mildly heterogeneous task, plain averaging at a long interval brought the validation loss into the prescribed sublevel set. On strongly heterogeneous shards the same runs missed it at long intervals, and the mean shift at an approximate reference solution increased the validation loss, with the same pattern for memoryless local SGD. The runs use an approximate reference and not an exact minimizer, estimate the diagnostic from 16 draws, and say nothing about

partial participation or real transfer times. The prototype of Section 8 is specified and not yet implemented.

Conclusion

A worldwide training system can maintain a common state while permitting substantial local independence. The formal condition is that each site's contribution has a declared origin, each relay preserves the information required by its parent, and each commit records the evidence under which the system advanced. In the affine setting, the resulting reduction has a compositional structure that supports arbitrary rearrangement of pure relays over a fixed accepted set, provided that the relays forward evidence of disjointness along with their sums. Sheaf-theoretic gluing accounts for exact agreement among overlapping observations, an information order accounts for the accumulation of incomplete evidence, and dependent types carry the resulting conditions into the implementation. Null Convention Logic gives absence an explicit role in the protocol without confusing it with any numerical value, and the commit certificate records which assumptions about relays and participants its claims rest on.

The optimization question begins where those guarantees end. Independently trained trajectories can reconcile into a valid global transition and still lead the system away from the intended objective, as the scalar example shows exactly and the strongly heterogeneous simulation illustrates. The research programme is therefore to determine which properties survive composition, which depend on the selected participants, and which require a dynamical bound on the local transformations. Geographic reach is then a property of the communication design, while useful global learning is a property of the entire composed system.

References

[1] P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, 2008.

[2] S. Awodey. *Category Theory*. Oxford University Press, second edition, 2010.

[3] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth. Practical Secure Aggregation for Privacy-Preserving Machine Learning. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, 2017.

[4] M. Castro and B. Liskov. Practical Byzantine Fault Tolerance. In *Proceedings of the Third Symposium on Operating Systems Design and Implementation*, 1999.

[5] A. Douillard, Q. Feng, A. A. Rusu, R. Chhaparia, Y. Donchev, A. Kuncoro, M. Ranzato, A. Szlam, and J. Shen. DiLoCo: Distributed Low-Communication Training of Language Models. arXiv:2311.08105, 2023.

[6] K. M. Fant and S. A. Brandt. NULL Convention Logic: A Complete and Consistent Logic for Asynchronous Digital Circuit Synthesis. In *Proceedings of the International Conference on Application-Specific Systems, Architectures and Processors*, pages 261–273, 1996.

[7] K. M. Fant. *Logically Determined Design: Clockless System Design with NULL Convention Logic*. Wiley, 2005.

[8] D. Goldberg. What Every Computer Scientist Should Know About Floating-Point Arithmetic. *ACM Computing Surveys*, 23(1):5–48, 1991.

[9] The Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations of Mathematics*. Institute for Advanced Study, 2013.

[10] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh. SCAFFOLD: Stochastic Controlled Averaging for Federated Learning. In *Proceedings of the International Conference on Machine Learning*, pages 5132–5143, 2020.

[11] M. M. Kim. *Null Convention Logic Circuits for Asynchronous Computer Architecture*. PhD thesis, RMIT University, 2020. `https://doi.org/10.25439/rmt.27581082`.

[12] L. Lamport. Time, Clocks, and the Ordering of Events in a Distributed System. *Communications of the ACM*, 21(7):558–565, 1978.

[13] L. Lamport, R. Shostak, and M. Pease. The Byzantine Generals Problem. *ACM Transactions on Programming Languages and Systems*, 4(3):382–401, 1982.

[14] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith. Federated Optimization in Heterogeneous Networks. In *Proceedings of Machine Learning and Systems*, 2:429–450, 2020.

[15] L. Liu, J. Zhang, S. H. Song, and K. B. Letaief. Client-Edge-Cloud Hierarchical Federated Learning. In *IEEE International Conference on Communications*, 2020.

[16] I. Loshchilov and F. Hutter. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*, 2019.

[17] S. Mac Lane. *Categories for the Working Mathematician*. Springer, second edition, 1998.

[18] S. Mac Lane and I. Moerdijk. *Sheaves in Geometry and Logic: A First Introduction to Topos Theory*. Springer, 1992.

[19] P. Martin-Löf. *Intuitionistic Type Theory*. Bibliopolis, 1984.

[20] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. Aguera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, pages 1273–1282, 2017.

[21] Y. Nesterov. A Method of Solving a Convex Programming Problem with Convergence Rate $O(1/k^2)$. *Soviet Mathematics Doklady*, 27(2):372–376, 1983.

[22] S. Reddi, Z. Charles, M. Zaheer, Z. Garrett, K. Rush, J. Konecny, S. Kumar, and H. B. McMahan. Adaptive Federated Optimization. In *International Conference on Learning Representations*, 2021.

[23] D. Scott. Data Types as Lattices. *SIAM Journal on Computing*, 5(3):522–587, 1976.

[24] M. Shapiro, N. Preguica, C. Baquero, and M. Zawirski. Conflict-Free Replicated Data Types. In *Proceedings of the International Symposium on Stabilization, Safety, and Security of Distributed Systems*, 2011.

[25] S. U. Stich. Local SGD Converges Fast and Communicates Little. In *International Conference on Learning Representations*, 2019.

[26] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor. Tackling the Objective Inconsistency Problem in Heterogeneous Federated Optimization. In *Advances in Neural Information Processing Systems*, 33:7611–7623, 2020.