

Probe-Based Directional Credit Assignment

A Research Note

Richard Vermillion

2026-03-30

We propose a framework for delayed credit assignment that sits between scalar eligibility traces and full gradient transport. The central idea is to attach a small probe dictionary to each perturbable module, instantiate a batch of nearby counterfactual trajectories by selecting probe directions across modules, and use delayed modulatory signals to reinforce or suppress the sampled directions. The framework is intended for settings in which full end-to-end gradients are unavailable, undesirable, biologically implausible, too expensive, or simply not the right abstraction. Its core objects are: (1) per-module probe dictionaries, (2) an assignment matrix defining which probe each module uses in each sampled trajectory, (3) directional eligibility traces that store sampled local directions over time, and (4) a scheduler that allocates limited perturbation budget across modules and trajectories. This note is a proposal rather than a validated empirical claim. Its purpose is to articulate the framework clearly, position it relative to related ideas, and define a concrete experimental agenda.

Table of contents

1	Introduction	2
1.1	Status	3
2	High-Level View	3
3	Probe Dictionaries	4
4	Assignment Matrix	4
4.1	Why the assignment matrix matters	5
4.2	Counterfactual trajectories, not isolated local probes	5
4.3	Measuring Probe Diversity and Entanglement	5
4.3.1	Per-module diversity	6
4.3.2	Cross-module entanglement	6
4.4	Design interpretation	6
5	Delayed Modulatory Credit Assignment	7
5.1	Why “scalar” modulation may still be enough	7
5.2	Designed routing first, learned routing later	8

5.3	Optional extension: local amplification or participation factors	8
6	Resource Allocation as a First-Class Problem	8
7	ProbedLinear: A Concrete Implementation Primitive	9
7.1	Why this implementation matters	10
8	Scheduler: The Global Perturbation Policy	10
8.1	Inputs to the scheduler	10
8.2	Outputs of the scheduler	10
8.3	High-level scheduler goals	11
8.4	Simple initial schedulers	11
9	Relationship to Prior Work	11
9.1	Eligibility traces and three-factor rules	11
9.2	Perturbation learning in temporally extended tasks	12
9.3	Evolution strategies and low-rank batched perturbations	12
9.4	Forward-gradient and local-loss methods	12
9.5	Claimed delta	13
10	Minimal Algorithmic Skeleton	13
11	Open Questions and Experimental Agenda	14
12	Limitations	15
13	Conclusion	15
	References	15

1 Introduction

Many learning systems face delayed credit assignment without having convenient access to full formal gradients. This includes recurrent systems, modular architectures, embodied agents, predictive-coding systems, hybrid differentiable/non-differentiable systems, and settings in which backward transport is too expensive or too structurally brittle. Existing alternatives each cover only part of this space: three-factor and eligibility-trace methods preserve locality and delayed modulation, but typically do not provide explicit parameter-space directional information; perturbation-based methods provide directional information, but are usually framed as zeroth-order optimization rather than as temporally persistent credit-assignment mechanisms (Bellec et al. 2020; Murray 2019; Saito et al. 2011; Zügel et al. 2023; Salimans et al. 2017).

The gap can be stated simply:

In many settings, we want more than “this mattered,” but less than a full gradient.

This note proposes a framework intended to fill that gap. The proposal is to combine local perturbative probes with delayed modulatory signals in order to form **directional eligibility**

traces. These traces preserve the temporal role of eligibility mechanisms while adding coarse directional information in parameter space.

The resulting view of learning is different from passive gradient reception. A learning system becomes an active experimenter: it allocates perturbation budget, instantiates nearby counterfactual trajectories, records which local directions were involved, and later consolidates the directions that proved useful under delayed evaluation.

1.1 Status

This document is a **research note / proposal**. It is intended to state the framework precisely enough to be discussed, cited, critiqued, and implemented before full empirical validation and ablation studies are complete.

2 High-Level View

The framework has five main components.

1. **Probe dictionaries** express each module’s currently available local perturbation directions.
2. The **assignment matrix** expresses the system’s experimental design across the batch.
3. **Directional traces** store how a module was recently perturbed, not merely that it was active.
4. A **modulatory signal** provides delayed evaluative feedback to each module.
5. The **update rule** synthesizes the resulting weighted evidence into a parameter update.

These components separate concerns that are often blurred together:

- what local changes are available,
- which of them are actually tested,
- how recent local experiments are remembered,
- what counts as success or failure,
- and how limited exploration budget is allocated.

That separation is one of the main points of the framework. It turns delayed credit assignment into a structured process of local experimentation rather than a monolithic optimization step.

Here the batch dimension is used in a nonstandard but increasingly common way: not as a collection of unrelated training examples, but as a parallel evaluation axis over structured counterfactual branches. Each batch member represents a different sampled perturbation bundle propagated through the same shared computation, allowing nearby perturbed trajectories to be evaluated efficiently in parallel. This is what makes the framework computationally plausible: the same forward machinery can be reused while the sampled branches provide the directional contrasts needed for delayed credit assignment.

3 Probe Dictionaries

Let module m have a local probe dictionary

$$\mathcal{D}_m = \{\delta_{m,0}, \delta_{m,1}, \dots, \delta_{m,K_m}\},$$

where:

- $\delta_{m,0} = 0$ is the **null probe**,
- $\delta_{m,k}$ for $k > 0$ are non-null local perturbation directions.

These directions may be low-rank parameter perturbations, structured adapter updates, or other compact module-local modifications. A LoRA-style low-rank parameterization is a natural implementation substrate because it provides inexpensive per-module directions and efficient batched recombination (Hu et al. 2021).

The null probe matters conceptually. A module is not “outside” the framework when it receives no directional variation in a given window. Rather, it is a degenerate case of the same formalism: its assignments do not vary across the sampled batch.

A probe dictionary can be interpreted as a rolling local basis of candidate update directions. Several regimes are possible:

1. **Pure probes:** ephemeral random directions.
2. **Semi-persistent probes:** directions are retained for multiple windows if they continue to look informative.
3. **Learned subspaces:** a slowly changing local basis from which concrete probes are drawn.

The framework does not require choosing the most ambitious option immediately. A semi-persistent dictionary is a natural practical starting point.

4 Assignment Matrix

The entire batchwise perturbation design can be represented by an **assignment matrix**

$$A = (A_{b,m})_{b=1,\dots,B}^{m=1,\dots,M}, \quad A_{b,m} \in \{0, \dots, K_m\}.$$

where:

- B is the batch size, interpreted here as the number of sampled nearby trajectories,
- M is the number of perturbable modules,
- K_m is the number of non-null directions in the probe dictionary for module m ,
- $A_{b,m}$ specifies which probe direction module m uses in trajectory b .

Each row of A defines one **joint perturbation bundle**, i.e. one nearby counterfactual world. Each column defines the variation seen by a particular module across the sampled batch. Trajectory b uses perturbation

$$\delta_{m,A_b,m}$$

at module m .

Thus A is a $B \times M$ assignment matrix whose m -th column takes values in $\{0, \dots, K_m\}$, with 0 denoting the null probe.

4.1 Why the assignment matrix matters

This matrix unifies several earlier distinctions:

- “Which modules are on?” becomes: which columns vary across rows.
- “Which probe directions are being used?” becomes: which labels appear in each column.
- “Which joint perturbation combinations are being tested?” becomes: the set of rows.

The entire experimental design is therefore the assignment matrix together with the probe dictionaries from which its entries are drawn.

4.2 Counterfactual trajectories, not isolated local probes

When several modules vary simultaneously, a row of A does not represent an isolated probe of one module. It represents a **joint hypothesis** about how several modules should move together. A successful trajectory therefore provides evidence for a coordinated perturbation bundle, not for fully isolated module-level effects. This is important because any resulting parameter update is not justified as a standalone local move, but as one component of a bundle of changes that was actually evaluated together in the sampled trajectory.

This is both a strength and a limitation:

- bundled trajectories respect cross-module interactions and support coordinated updates based on perturbation bundles that were actually tested together, but
- they reduce local identifiability unless the assignment design deliberately provides disentangling contrasts.

That tradeoff is explicit in the framework rather than hidden.

4.3 Measuring Probe Diversity and Entanglement

The assignment matrix supports principled reasoning about what information a batch can provide.

4.3.1 Per-module diversity

For module m , define the empirical assignment distribution

$$p_m(k) = \frac{1}{B} \sum_{b=1}^B \mathbf{1}[A_{b,m} = k].$$

This gives at least two useful quantities:

- **support size**: how many probe options actually appeared,
- **entropy**: how evenly those options were represented.

A constant column has support size 1 and entropy 0, regardless of whether the repeated value is null or non-null. Such a module contributes no within-batch directional information in that window.

4.3.2 Cross-module entanglement

Two modules may each have high local diversity yet still be difficult to disentangle if their assignments are strongly correlated.

For modules i and j , define the empirical joint distribution

$$p_{i,j}(k, \ell) = \frac{1}{B} \sum_{b=1}^B \mathbf{1}[A_{b,i} = k, A_{b,j} = \ell].$$

Then mutual information

$$I(i; j) = \sum_{k, \ell} p_{i,j}(k, \ell) \log \frac{p_{i,j}(k, \ell)}{p_i(k)p_j(\ell)}$$

measures **assignment entanglement**: how much knowing one module's probe choice reduces uncertainty about another's.

4.4 Design interpretation

The assignment matrix therefore carries at least three kinds of information:

1. **coverage**: how many probe options participated,
2. **balance**: how evenly those options were allocated across the batch,
3. **entanglement**: how strongly assignments are bundled across modules.

This turns probing into a design-of-experiments problem rather than an ad hoc perturbation scheme.

5 Delayed Modulatory Credit Assignment

The minimal formulation assumes that each sampled counterfactual branch carries its own delayed evaluative signal for each module. For module m , let

$$\mu_{m,b,t} \in \mathbb{R}$$

denote the modulatory signal assigned at time or window index t to branch b . Although each $\mu_{m,b,t}$ is scalar, the collection

$$(\mu_{m,1,t}, \dots, \mu_{m,B,t})$$

forms a branch-indexed pattern of evaluative feedback over the sampled perturbation ensemble.

This is the key point: the directional information does not have to be encoded inside the modulator itself. It arises from the fact that different counterfactual branches receive different scalar evaluations, and those evaluations are paired with different sampled perturbation directions.

We therefore define the directional trace for module m at time t directly as

$$e_{m,t} = \frac{1}{B} \sum_{b=1}^B \mu_{m,b,t} \delta_{m,A_{b,m}}.$$

This is the core object. It records a weighted combination of the local directions that were actually sampled, with branchwise weights determined by delayed or globally modulated evaluative signals.

Over multiple windows, a module maintains a history of such traces,

$$e_{m,t}, e_{m,t-1}, \dots, e_{m,t-P+1},$$

which together form its directional eligibility trace history.

The parameter update is then

$$\Delta\theta_m = -\alpha \sum_{j=0}^{P-1} \lambda^j e_{m,t-j},$$

where: - α is the learning rate, - $\lambda \in [0, 1]$ is trace decay, - P is the trace horizon.

This should be read as follows:

- the sampled perturbation ensemble provides a local basis of candidate directions,
- branch-specific delayed evaluative signals determine which sampled branches were better or worse,
- the resulting update lies in the span of directions that were actually probed.

5.1 Why “scalar” modulation may still be enough

At first glance, a scalar modulatory signal can appear too weak. But this is misleading. The framework does not rely on a single scalar per module in isolation. It relies on a pattern of scalar signals across branches, each paired with a distinct sampled perturbation direction.

For module m , the effective update is built from the joint structure of: - the sampled directions $\delta_{m,A_{b,m}}$, - the branchwise modulatory coefficients $\mu_{m,b,t}$, - and the temporal accumulation of those weighted directional traces.

The modulator therefore does not need to encode direction internally if the directional basis is already supplied by the perturbation ensemble. A batch of branch-specific scalars distributed over distinct probe directions is already rich enough to synthesize a structured update in parameter space.

There is a second sense in which the “scalar” description can be misleading. Even if each branch contributes only a scalar coefficient at the point where it weights a sampled direction, that scalar may itself be produced by a much richer routing process. A module may receive a structured vector of task-level, graph-level, or temporally extended error information and project that richer signal down to a branch-specific scalar value for a module. The final modulation is scalar only at the last step of the local update rule, not necessarily in its upstream computation.

5.2 Designed routing first, learned routing later

A longer-term extension is to learn how structured spatiotemporal error information should be routed and projected into the branch-specific modulatory coefficients $\mu_{m,b,t}$ seen by each module. But the right place to start is with a designed projection based on architectural heuristics such as topology, precision weighting, temporal decay, and distance from recent error sources. That isolates the core question of whether branchwise directional traces help at all before introducing a second difficult bootstrapping problem.

5.3 Optional extension: local amplification or participation factors

Some architectures may benefit from an additional local factor

$$\ell_{m,b,t} \in \mathbb{R},$$

used to amplify or attenuate the contribution of a sampled branch before it is added to the directional trace. In that case one may write

$$e_{m,t} = \frac{1}{B} \sum_{b=1}^B \mu_{m,b,t} \ell_{m,b,t} \delta_{m,A_b,m}.$$

Here $\ell_{m,b,t}$ is not the primary source of directional credit. That role is already played by the branchwise modulatory pattern $\mu_{m,b,t}$. Instead, $\ell_{m,b,t}$ serves as an optional local gain term, capturing factors such as activation magnitude, module participation, local sensitivity, or other architecture-specific notions of how strongly module m was engaged in branch b at time t .

This extension may be useful when one wants the global or delayed evaluative signal to be further filtered by local information about how much the module actually participated in the branch’s behavior. But it is not required for the minimal framework.

6 Resource Allocation as a First-Class Problem

Once learning is treated as active experimentation, resource allocation becomes central rather than incidental.

The system cannot:

- vary every module equally at every step,
- maintain arbitrarily many useful probe directions everywhere,
- instantiate all possible joint perturbation bundles,
- or spend the batch budget uniformly if only some parts of the system are worth exploring.

There are therefore at least three distinct budgets:

1. **module budget**: which modules should receive variation,
2. **local basis budget**: how many and what kinds of directions each module should maintain,
3. **trajectory budget**: which combinations of local directions should be allocated space in the finite batch.

The framework’s components work together to allocate these budgets. Probe dictionaries manage local exploratory capacity. The assignment matrix expresses the current experiment. Modulatory signals and traces determine which experiments paid off. Updates consolidate the returns. This is one of the main reasons the framework feels more like a laboratory than a passive optimizer.

7 ProbedLinear: A Concrete Implementation Primitive

To make the framework implementable, it is useful to introduce a reusable module-level primitive such as **ProbedLinear**.

A **ProbedLinear** module contains:

- a shared base weight matrix W ,
- a probe dictionary $\{\delta_0, \delta_1, \dots, \delta_K\}$,
- a mechanism for selecting one probe per batch member,
- a mechanism for collapsing credited probe directions back into an update on the shared weight.

A natural implementation is LoRA-like. For example, a probe direction may be represented by low-rank factors for a linear projection:

$$\Delta W_k = U_k V_k^\top,$$

so that batch member b uses

$$W + \Delta W_{A_b}$$

or equivalently computes a shared base projection plus a probe-specific correction (Hu et al. 2021). Low-rank perturbation schemes are also attractive because related zeroth-order methods have shown that low-rank perturbations can dramatically reduce the cost of population-based exploration at scale (Sarkar et al. 2025).

In practice, the efficient implementation is to:

1. compute the shared full-rank path once using a batched `matmul`,
2. compute batched low-rank corrections in parallel,
3. add them according to the batch assignments.

This primitive can be dropped into MLPs, attention projections, recurrent transitions, and other linear-heavy components.

7.1 Why this implementation matters

`ProbedLinear` turns the framework from a training rule on paper into reusable infrastructure.

- probe directions become explicit module-local objects,
- the batch dimension naturally instantiates nearby trajectories,
- replacing stale probes becomes easy,
- updates to the base weight are naturally expressed as linear combinations of credited probe directions.

8 Scheduler: The Global Perturbation Policy

The final major object is the **scheduler**, which chooses the assignment matrix over time.

The scheduler determines how limited perturbation budget is allocated across modules, directions, and trajectories.

8.1 Inputs to the scheduler

A scheduler may use signals such as:

- recent per-module trace magnitudes,
- per-module uncertainty or surprise,
- per-module assignment entropy,
- pairwise or global assignment entanglement statistics,
- recency or staleness of probe directions,
- topological proximity to recent error sources,
- architecture-specific salience signals.

8.2 Outputs of the scheduler

At each window, the scheduler emits an assignment matrix A_t , i.e. a set of sampled perturbation bundles.

This implicitly chooses:

- which modules receive variation,
- how much variation each receives,
- whether the batch emphasizes disentangled local evidence or coherent joint hypotheses,
- how much null probing versus non-null probing to include.

8.3 High-level scheduler goals

A useful scheduler likely trades off at least four objectives:

1. **information value**: probe where the system expects to learn something,
2. **learning value**: probe where the system expects actual improvement,
3. **coverage**: ensure modules and directions do not go stale,
4. **controllability**: avoid so much simultaneous variation that trajectories become uninterpretable.

8.4 Simple initial schedulers

The first empirical tests should begin with simple scheduler classes:

- random sampling,
- round-robin probing across modules or blocks,
- blockwise bundled probing,
- uncertainty-weighted probing,
- diversity-seeking sampling with entanglement penalties.

More adaptive global scheduling can come later.

9 Relationship to Prior Work

This proposal sits at the intersection of four nearby literatures.

9.1 Eligibility traces and three-factor rules

The closest conceptual ancestor is the family of local-learning rules built around eligibility traces and delayed modulatory signals. In spiking and recurrent settings, e-prop shows that eligibility traces can preserve temporal locality while allowing later learning signals to shape updates (Bellec et al. 2019, 2020). Related biologically motivated approaches such as RFLO likewise keep local state and delayed feedback, but they do not explicitly maintain a bank of parameter-space directions sampled as local counterfactual hypotheses (Murray 2019).

The point of departure here is therefore not delayed modulation itself, but the proposal to make the trace *directional*: a module stores a weighted summary of recently sampled perturbation directions rather than only scalar recency or activity information.

9.2 Perturbation learning in temporally extended tasks

The second neighboring literature is perturbation-based credit assignment. Saito et al. explicitly combine node perturbation with eligibility traces and analyze structural and temporal credit assignment together, making that work one of the closest prior neighbors to the present proposal (Saito et al. 2011). More recently, Züge et al. compare weight perturbation and node perturbation in temporally extended tasks and show that weight perturbation can be competitive or superior in such settings (Züge et al. 2023).

These works are important precedents, but they still differ in emphasis from the present framework. The proposal here is not simply to use perturbations for zeroth-order optimization, nor merely to combine perturbation with scalar eligibility. Rather, it is to treat sampled perturbation directions as a persistent local record that can later be reinforced or suppressed by delayed modulatory signals. In that sense, the intended abstraction is a *directional eligibility trace*, not just perturbation-based learning with a temporal baseline.

9.3 Evolution strategies and low-rank batched perturbations

A third neighboring literature is evolution strategies and related zeroth-order methods. Classic ES already shows how parameter-space perturbations can scale through batched parallel evaluation (Salimans et al. 2017). EGGROLL is especially relevant for implementation because it demonstrates how low-rank batched perturbations can make large-scale ES far more practical (Sarkar et al. 2025).

The present proposal borrows this implementation insight directly: low-rank probe dictionaries and batched probe evaluation are natural practical substrates for the framework. But the conceptual role of perturbations is different. In standard ES, perturbations are principally population members used to estimate an optimizer step. Here, perturbations are also part of a *credit-assignment instrument*: the assignment matrix, directional traces, and delayed modulatory signals turn batched perturbations into a structured experimental design over local hypotheses.

9.4 Forward-gradient and local-loss methods

A fourth neighboring literature is forward-gradient and local-loss learning. Baydin et al. reintroduced forward gradients as an unbiased gradient estimator computed from directional derivatives without backpropagation (Baydin et al. 2022). Ren et al. then showed how forward-gradient methods can be made substantially more practical by combining them with local losses and architectural choices that reduce estimator variance (Ren et al. 2023).

These methods are adjacent because they also use perturbative or directional information without full reverse-mode gradient transport. The main distinction is temporal. Forward-gradient methods are still primarily about approximating immediate gradients of defined losses. The present proposal is instead aimed at delayed credit settings in which the system may need to preserve a history of sampled local directions and only later decide, via modulatory signals, which of those directions were implicated in success or failure.

9.5 Claimed delta

The claim of this note is therefore intentionally narrow.

It is **not** that eligibility traces are new, perturbation methods are new, low-rank batched perturbations are new, or forward-gradient ideas are new. Rather, the proposed contribution is the explicit synthesis of these ingredients into a framework whose central objects are:

1. per-module **probe dictionaries**,
2. an **assignment matrix** specifying the structured batchwise perturbation design,
3. **directional eligibility traces** that preserve sampled local directions over time,
4. delayed **scalar modulatory signals** that reinforce or suppress those traces,
5. and a **scheduler** that allocates perturbation budget across modules, directions, and trajectories.

Stated more compactly: the proposed delta is to extend scalar eligibility-trace thinking into a framework of **directional eligibility via structured perturbation design**.

10 Minimal Algorithmic Skeleton

A minimal version of the framework can be expressed as follows.

1. **Attach probe dictionaries.**

For each perturbable module m , maintain a probe dictionary

$$\mathcal{D}_m = \{\delta_{m,0}, \delta_{m,1}, \dots, \delta_{m,K_m}\},$$

where $\delta_{m,0} = 0$ is the null probe.

2. **Sample an assignment matrix.**

At each probing window t , the scheduler emits an assignment matrix

$$A_t = (A_{b,m})_{b=1,\dots,B}^{m=1,\dots,M}, \quad A_{b,m} \in \{0, \dots, K_m\},$$

specifying which probe direction each module uses in each sampled branch.

3. **Roll out the sampled counterfactual branches in parallel.**

Use the batch dimension to evaluate the resulting B nearby perturbed trajectories in parallel, with branch b using perturbation $\delta_{m,A_{b,m}}$ at module m .

4. **Obtain branch-specific modulatory signals.**

For each module m and branch b , compute or route a delayed evaluative signal

$$\mu_{m,b,t} \in \mathbb{R},$$

representing how that sampled branch should be reinforced or suppressed at that module.

5. **Form directional traces.**

For each module, construct the directional trace

$$e_{m,t} = \frac{1}{B} \sum_{b=1}^B \mu_{m,b,t} \delta_{m,A_{b,m}}.$$

6. **Accumulate trace history.**

Maintain a recent history

$$e_{m,t}, e_{m,t-1}, \dots, e_{m,t-P+1},$$

with decay parameter λ .

7. **Update base parameters.**

Update the shared parameters of module m by

$$\Delta\theta_m = -\alpha \sum_{j=0}^{P-1} \lambda^j e_{m,t-j}.$$

8. **Refresh or recycle probes.**

Replace stale, low-yield, or redundant probe directions as needed, while retaining useful directions when appropriate.

9. **Choose the next experiment.**

Use the scheduler to allocate the next perturbation budget by choosing a new assignment matrix A_{t+1} .

An optional extension is to include a local amplification or participation factor $\ell_{m,b,t}$, yielding

$$e_{m,t} = \frac{1}{B} \sum_{b=1}^B \mu_{m,b,t} \ell_{m,b,t} \delta_{m,A_b,m},$$

but this is not required for the minimal formulation.

11 Open Questions and Experimental Agenda

The framework remains underdeveloped in several important respects.

1. **Best probe dictionaries:** random, recycled, structured, or learned?
2. **Best local probe-effect summaries:** what signal should define $\ell_{m,b,t}$?
3. **Best scheduler objectives:** should the system prioritize disentanglement, bundled hypotheses, uncertainty reduction, or immediate performance?
4. **Best update-span management:** how should stale or redundant probe directions be replaced?
5. **Interaction with exact gradients:** when should directional-trace updates coexist with ordinary local gradients or scalar traces?
6. **Learned modulatory routing:** when does designed routing stop being enough?

The first empirical benchmarks should be deliberately small and diagnostic. The key question is not whether the framework beats full backpropagation on every task. The first question is whether directional traces outperform nearby local alternatives in delayed-credit settings where scalar traces are too weak and full gradient transport is unavailable, undesirable, or structurally mismatched. Natural adjacent baselines include scalar eligibility-trace methods, node perturbation, weight perturbation, and local-loss or forward-gradient approaches (Bellec et al. 2020; Saito et al. 2011; Züge et al. 2023; Ren et al. 2023).

12 Limitations

This note is intentionally incomplete.

- It does **not** yet provide experimental validation.
- It does **not** yet show superiority over scalar eligibility traces or standard perturbation methods.
- It does **not** yet provide a fully specified learned scheduler or learned modulatory router.
- It does **not** claim that all important ingredients are individually novel.

Its purpose is narrower: to articulate a coherent, implementable, and potentially general framework clearly enough that it can be tested.

13 Conclusion

The key abstraction is simple.

- modules expose local probe dictionaries,
- a scheduler chooses an assignment matrix,
- the batch dimension instantiates nearby counterfactual trajectories,
- directional traces summarize the sampled local directions,
- delayed modulatory signals reinforce or suppress those traces,
- base parameters update in the span of recently probed directions.

This reframes learning as active local experimentation under delayed evaluation rather than passive consumption of gradients. If it works, it would provide a useful middle ground: richer than scalar “this mattered” traces, cheaper or more broadly applicable than full gradient transport, and concrete enough to implement via reusable module-level primitives such as `ProbedLinear`.

References

- Baydin, Atilim Gunes, Barak A. Pearlmutter, Don Syme, Frank Wood, and Philip H. S. Torr. 2022. *Gradients Without Backpropagation*. <https://doi.org/10.48550/arXiv.2202.08587>.
- Bellec, Guillaume, Franz Scherr, Elias Hajek, Darjan Salaj, Robert Legenstein, and Wolfgang Maass. 2019. “Eligibility Traces Provide a Data-Inspired Alternative to Backpropagation Through Time.” *Advances in Neural Information Processing Systems* 32.
- Bellec, Guillaume, Franz Scherr, Anand Subramoney, et al. 2020. “A Solution to the Learning Dilemma for Recurrent Networks of Spiking Neurons.” *Nature Communications* 11 (1): 3625. <https://doi.org/10.1038/s41467-020-17236-y>.
- Hu, Edward J., Yelong Shen, Phillip Wallis, et al. 2021. “LoRA: Low-Rank Adaptation of Large Language Models.” *arXiv Preprint arXiv:2106.09685*, ahead of print. <https://doi.org/10.48550/arXiv.2106.09685>.

- Murray, James D. 2019. “Local Online Learning in Recurrent Networks with Random Feedback.” *eLife* 8: e43299. <https://doi.org/10.7554/eLife.43299>.
- Ren, Mengye, Simon Kornblith, Renjie Liao, and Geoffrey Hinton. 2023. “Scaling Forward Gradient with Local Losses.” *The Eleventh International Conference on Learning Representations*. <https://openreview.net/forum?id=xde8t1cJXh2>.
- Saito, Hiroshi, Kentaro Katahira, Kazuo Okanoya, and Masato Okada. 2011. “Statistical Mechanics of Structural and Temporal Credit Assignment Effects on Learning in Neural Networks.” *Physical Review E* 83 (5): 051125. <https://doi.org/10.1103/PhysRevE.83.051125>.
- Salimans, Tim, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. 2017. “Evolution Strategies as a Scalable Alternative to Reinforcement Learning.” *arXiv Preprint arXiv:1703.03864*, ahead of print. <https://doi.org/10.48550/arXiv.1703.03864>.
- Sarkar, Bidipta, Mattie Fellows, Juan Agustin Duque, et al. 2025. “Evolution Strategies at the Hyperscale.” *arXiv Preprint arXiv:2511.16652*, ahead of print. <https://doi.org/10.48550/arXiv.2511.16652>.
- Züge, Paul, Christian Klos, and Raoul-Martin Memmesheimer. 2023. “Weight Versus Node Perturbation Learning in Temporally Extended Tasks: Weight Perturbation Often Performs Similarly or Better.” *Physical Review X* 13 (2): 021006. <https://doi.org/10.1103/PhysRevX.13.021006>.