

Proving expected sensitivity of probabilistic programs

GILLES BARTHE, Imdea Software

THOMAS ESPITAU, Sorbonne Universités, UPMC Paris 6

BENJAMIN GRÉGOIRE, Inria

JUSTIN HSU, University of Pennsylvania

PIERRE-YVES STRUB, École Polytechnique

Program sensitivity, also known as *Lipschitz continuity*, describes how small changes in a program's input lead to bounded changes in the output. We propose an average notion of program sensitivity for probabilistic programs—*expected sensitivity*—that averages a distance function over a probabilistic coupling of two output distributions from two nearby inputs. By varying the distance, expected sensitivity captures useful notions of probabilistic function sensitivity, including algorithmic stability of machine learning algorithms and convergence of Markov chains.

Furthermore, expected sensitivity satisfies clean compositional properties and is amenable to formal verification. We develop a relational program logic called ExSeL for proving expected sensitivity properties. Our logic features two key ideas. First, relational pre-conditions and post-conditions are expressed using *distances*, which can be seen as a real-valued generalization of typical boolean-valued (relational) assertions. Second, judgments are interpreted in terms of *expectation coupling*, a novel, quantitative generalization of probabilistic couplings which supports compositional reasoning.

We demonstrate our logic on two classes of examples: formalizing uniform stability for the stochastic gradient algorithm from machine learning, and rapid mixing for a model of asexual population dynamics. We also extend our logic with a transitivity principle for expectation couplings to capture the *path coupling* proof technique by Bubley and Dyer [13], and we demonstrate our extension by proving rapid mixing of the Glauber dynamics from statistical physics.

ACM Reference format:

Gilles Barthe, Thomas Espitau, Benjamin Grégoire, Justin Hsu, and Pierre-Yves Strub. 2017. Proving expected sensitivity of probabilistic programs. 1, 1, Article 1 (July 2017), 37 pages.

<https://doi.org/10.1145/nnnnnnn.nnnnnnn>

1 INTRODUCTION

Sensitivity, also known as *Lipschitz continuity*, is a fundamental property in mathematics and computer science. Sensitivity describes how small changes in inputs can affect outputs. Formally, sensitivity of a function $g : A \rightarrow B$ is defined relative to two metrics d_A and d_B on A and B respectively. We say that f is α -sensitive, where $\alpha \in \mathbb{R}^+$, iff for every two inputs x_1 and x_2 , $d_B(g(x_1), g(x_2)) \leq \alpha \cdot d_A(x_1, x_2)$. Bounded sensitivity is a fundamental property in the semantics of programming languages and also plays a central role in many other fields, including hybrid and dynamical systems, and in differential privacy. These and other applications have motivated a broad range verification methods for bounding sensitivity for programs.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2017 Association for Computing Machinery.

XXXX-XXXX/2017/7-ART1 \$15.00

<https://doi.org/10.1145/nnnnnnn.nnnnnnn>

We consider *expected* (or average) sensitivity, a natural generalization of sensitivity for the probabilistic setting, and develop a program logic for proving expected sensitivity of probabilistic programs. We work with a mild generalization of sensitivity. First, we consider arbitrary non-negative real-valued functions instead of metrics; for convenience, we call such functions *distances*. Second, we consider f -sensitivity rather than α -sensitivity, where f is a non-negative affine function the form $z \mapsto \alpha \cdot z + \beta$, with $\alpha, \beta \in \mathbb{R}^+$. Formally, let $\mathfrak{d} : A \times A \rightarrow \mathbb{R}^+$ and $\mathfrak{d}' : B \times B \rightarrow \mathbb{R}^+$ and let f be a non-negative affine function. We say that $g : A \rightarrow B$ is f -sensitive iff for every two inputs x_1 and x_2 , $\mathfrak{d}_B(g(x_1), g(x_2)) \leq f(\mathfrak{d}_A(x_1, x_2))$. This notion of f -sensitivity can model multiplicative and additive bounds between the outputs from two inputs. Furthermore, using affine functions f enables sensitivity to compose cleanly in the probabilistic case.

1.1 Expected sensitivity

Let us now consider the case where g is a probabilistic function, i.e. $g : A \rightarrow \mathbb{D}(B)$. The first obstacle to defining sensitivity is that g produces distributions over B , rather than elements of B . One possibility is to define distances directly on the space of distributions; however, such distances can be complex and difficult to reason about. We consider an alternative approach: lift a distance \mathfrak{d}_B on elements to a distance on distributions, by averaging \mathfrak{d}_B over some distribution μ on pairs $B \times B$. To derive information about the two output distributions μ_1 and μ_2 , we require that μ models μ_1 and μ_2 in a probabilistic sense; namely, its first and second marginals must be equal to μ_1 and μ_2 , respectively. Such a distribution μ is known as a *probabilistic coupling* of μ_1 and μ_2 (we refer the reader to Lindvall [28], Thorisson [34] for an overview of the rich theory of probabilistic couplings).

Formally, we say that a probabilistic function $g : A \rightarrow \mathbb{D}(B)$ is *expected f -sensitive* iff for every two inputs x_1 and x_2 , there exists a coupling μ of $g(x_1)$ and $g(x_2)$, such that:

$$\mathbb{E}_{(y_1, y_2) \sim \mu} [\mathfrak{d}_B(y_1, y_2)] \leq f(\mathfrak{d}_A(x_1, x_2)). \quad (1)$$

This notion is inspired by the *Wasserstein* metric, a well-studied distance on distributions [36]. Our notion of expected sensitivity has several appealing features. First, it is quite general—we can capture many probabilistic notions of sensitivity by varying the distance.

Example 1.1 (Average sensitivity). If the outputs (y_1, y_2) are numbers, then a natural notion of sensitivity bounds the difference in *average* outputs in terms of the distance between inputs (x_1, x_2) . By setting the distance $\mathfrak{d}_B(y_1, y_2) \triangleq |y_1 - y_2|$, a direct consequence of expected f -sensitivity is the bound

$$\left| \mathbb{E}_{y_1 \sim \mu_1} [y_1] - \mathbb{E}_{y_2 \sim \mu_2} [y_2] \right| \leq f(\mathfrak{d}_A(x_1, x_2)).$$

This inequality shows that two distributions μ_1 and μ_2 have similar average values when the inputs (x_1, x_2) are close. This type of bound can imply *algorithmic stability*, a useful property for machine learning algorithms [12].

Example 1.2 (Probabilistic sensitivity). Suppose that the output distance \mathfrak{d}_B is bounded away from zero: $\mathfrak{d}_B(y_1, y_2) < 1$ implies $y_1 = y_2$; for instance, \mathfrak{d}_B could be an integer-valued metric. Then, a consequence of expected f -sensitivity is the bound

$$\left| \Pr_{y_1 \sim \mu_1} [y_1 \in E] - \Pr_{y_2 \sim \mu_2} [y_2 \in E] \right| \leq f(\mathfrak{d}_A(x_1, x_2))$$

for every set of outputs E . This inequality shows that the distributions μ_1 and μ_2 are close in a pointwise sense, and can imply that two sequences of distributions converge to one another.

Another appealing feature of expected sensitivity is that it is compositional: The sequential composition of an f -sensitive function with an f' -sensitive function yields an $f' \circ f$ -sensitive

function, as long as f and f' are non-negative affine functions. As we will see, this property makes expected sensitivity a good target for formal verification.

1.2 Expected sensitivity from expectation couplings

The definition of expected sensitivity averages the distance over *any* probabilistic coupling of μ_1 and μ_2 . In general, there are multiple probabilistic couplings between any two distributions, leading to different expected distances.

To better reason about these couplings and their expected distances, we develop a quantitative generalization of probabilistic coupling that captures the inequality from Eq. (1); namely, if a distribution μ on pairs satisfies the bound for expected sensitivity, we call μ an *expectation coupling* of μ_1 and μ_2 with respect to \mathfrak{d}_B and δ , where $\delta = f(\mathfrak{d}_A(x_1, x_2))$. We show that expectation couplings satisfy several natural properties, including a sequential composition theorem.

1.3 ExSEL: A program logic for proving expected sensitivity bounds

By leveraging their composition principles, we can bound expected sensitivity by building an expectation coupling for the output distributions from every pair of nearby inputs. To make this idea concrete, we develop a relational program logic ExSEL to build expectation couplings for pairs of programs. ExSEL judgments have the form

$$\{\Phi; \mathfrak{d}\} s_1 \sim_f s_2 \{\Psi; \mathfrak{d}'\},$$

where s_1 and s_2 are probabilistic imperative programs—often, the same program—the pre- and post-conditions $\Phi, \Psi : \mathcal{M} \times \mathcal{M} \rightarrow \mathbb{B}$ are relational assertions over pairs of memories, $\mathfrak{d}, \mathfrak{d}' : \mathcal{M} \times \mathcal{M} \rightarrow \mathbb{R}^+$ are non-negative distances on memories, and $f(z) = \alpha \cdot z + \beta$ is a non-negative affine function with $\alpha, \beta \geq 0$. ExSEL judgments state that for any pair of related input memories (m_1, m_2) satisfying the pre-condition Φ , there exists an expectation coupling μ of the output distributions μ_1, μ_2 such that all pairs of output memories (m'_1, m'_2) with positive probability satisfy the post-condition Ψ , and the expected distance is bounded:

$$\mathbb{E}_{(m'_1, m'_2) \sim \mu} [\mathfrak{d}'(m'_1, m'_2)] \leq f(\mathfrak{d}(m_1, m_2)) = \alpha \cdot \mathfrak{d}(m_1, m_2) + \beta.$$

We call the function f a *distance transformer*, as it bounds the (average) *post-distance* \mathfrak{d}' in terms of the *pre-distance* \mathfrak{d} . As a consequence, each ExSEL proof verifies a bound on expected sensitivity for s_1 and s_2 .

We give a rich Hoare-style proof system for ExSEL, internalizing various compositional properties of expectation couplings. For the case of sequential composition of programs, given two judgments

$$\{\Phi; \mathfrak{d}\} s_1 \sim_f s_2 \{\Xi; \mathfrak{d}'\} \quad \text{and} \quad \{\Xi; \mathfrak{d}'\} s'_1 \sim_{f'} s'_2 \{\Psi; \mathfrak{d}''\},$$

we can compose to give the judgment

$$\{\Phi; \mathfrak{d}\} s_1; s'_1 \sim_{f' \circ f} s_2; s'_2 \{\Psi; \mathfrak{d}''\}$$

Note that the pre- and post-conditions and the distances compose cleanly, while the distance transformers combine by function composition. As a result, we can reason about the sequential composition of two sub-programs by building expectation couplings for each sub-program.

1.4 Applications

We illustrate the expressiveness of the proof system with two classes of examples from the recent randomized algorithms literature.

Stability of stochastic gradient method. In machine learning, stability [12, 19] measures how changes in the training set influence the quality of an algorithm’s prediction. A stable algorithm does not depend too much on the particular training set, and its performance generalizes from the training set to unseen examples. Recently, Hardt, Recht, and Singer [20] show stability of the Stochastic Gradient Method (SGM), an widely used optimization algorithm for training in machine learning. We verify the quantitative stability claim for several variants of SGM within our logic.

Rapid mixing for population dynamics. Randomized algorithms are a useful modeling tool for biological or social phenomena (see, e.g., Jansen [22]). They can be used to analyze population dynamics, both in the infinite population setting where evolution is *deterministic*, and in the finite population setting where evolution can be *stochastic*. We formally analyze a variant of the so-called RSM (Replication-Selection-Mutate) model, which captures the evolution of an unstructured, asexual haploid population (see, e.g., Hartl and Clark [21]). Recently, a series of papers prove rapid mixing of the RSM model under some mild conditions [17, 31, 37]. We formally verify rapid mixing in a simplified setting, where the evolution function is strictly contractive. This example relies on an advanced proof rule internalizing the *maximal* coupling of two multinomial distributions; in some sense, the coupling that minimizes the expected distance between samples.

1.5 Extension: path coupling

Once we have set the core logic, we extend the rules to capture more advanced reasoning about expectation couplings. We consider the *path coupling* method due Bubley and Dyer [13], a theoretical tool for building couplings; the resulting couplings can be used to show fast convergence of two Markov chains in numerous domains, including statistical mechanics, molecular evolution, and security. More concretely, let Φ a binary relation and suppose that the state space of the Markov chain is equipped with a *path metric* d_Φ , i.e., where the distance between two elements is the length of the shortest Φ -path between them. We say that two states are *adjacent* if their distance is 1. The main idea of path coupling is that if we can give a coupling for the distributions after one step of the random process started in neighboring states, then we can combine these pieces to give a coupling for the distributions started from *any* two states. Moreover, if for every two initial states at distance 1 under d_Φ there is an expectation coupling of the output distributions with expected distance at most γ , then for every two initial states at distance k under d_Φ , path couplings gives an expectation coupling with expected distance at most $k \cdot \gamma$.

From a logical point of view, path coupling is a transitivity principle for expectation couplings: we take a coupling for inputs related by Φ , and construct a coupling for inputs related by Φ^k . In ExSEL, we internalize this composition principle by a structural transitivity rule, allowing a family of relational judgments to be chained together. We formally prove rapid mixing for a classical example called the Glauber dynamics, a Markov chain for drawing approximately uniform samplings from the set of proper colorings of a graph [13].

1.6 Outline and core contributions

After illustrating our approach on a simple example (§ 2) and reviewing relevant mathematical preliminaries, we present the following contributions.

- A novel abstraction called expectation couplings for reasoning about probabilistic sensitivity, supporting natural composition properties (§ 3).
- A probabilistic relational program logic ExSEL for constructing expectation couplings, along with a proof of soundness (§ 4).
- A formal proof of uniform stability for two versions of the Stochastic Gradient Method, relying on proof rules to perform probabilistic case analysis (§ 5).

- A formal proof of rapid mixing for a Markov chain simulating a finite model of population evolution, relying on a proof rule internalizing the *maximal* coupling of two multinomial distributions (§ 6).
- A formal proof of rapid mixing for the Glauber dynamics from statistical physics relying on an advanced proof rule internalizing the path coupling principle (§ 7).

We implemented our logic in the EasyCrypt [5], a general-purpose proof assistant for reasoning about probabilistic programs, and machine-checked our main examples (§ 8). We conclude by surveying related work (§ 9) and presenting promising future directions (§ 10).

2 STABILITY OF STOCHASTIC GRADIENT METHOD

To give a taste of our approach, let's consider a property from machine learning. In a typical learning setting, we have a space of possible *examples* Z , a *parameter space* \mathbb{R}^d , and a *loss function* $\ell : Z \rightarrow \mathbb{R}^d \rightarrow [0, 1]$. An algorithm A takes a finite set $S \in Z^n$ of *training examples*—assumed to be drawn independently from some unknown distribution \mathcal{D} —and produces a parameter $w \in \mathbb{R}^d$ such that the expected loss of $\ell(-, w)$ on a fresh sample from \mathcal{D} should be as small as possible. When the algorithm is randomized, we think of $A : Z^n \rightarrow \mathbb{D}(\mathbb{R}^d)$ as mapping the training examples to a distribution over parameters.

In order to minimize the loss, a natural idea is to find some parameter w that minimizes the average error on the training set. When the loss function ℓ is well-behaved this optimization problem, known as *empirical risk minimization* in the literature, can be solved efficiently. However there is no guarantee that these parameters *generalize* to the true distribution—even if w has low loss on the training set, it may have high loss on fresh samples from the true distribution. Roughly speaking, the algorithm may select parameters that are too specific to the training sample.

This issue—also known as *overfitting*—is a serious problem in machine learning. To avoid overfitting, Bousquet and Elisseeff [12] proposed a technical property of the learning algorithm: the algorithm should produce similar outputs when executed on two training sets that differ in a single example, so that the output does not depend too much on any single training example.

Definition (Bousquet and Elisseeff [12]). Let $A : Z^n \rightarrow \mathbb{D}(\mathbb{R}^d)$ be an algorithm for some loss function $\ell : Z \rightarrow \mathbb{R}^d \rightarrow [0, 1]$. The algorithm A is said to be ϵ -uniformly stable if for all input sets $S, S' \in Z^n$ that differ in a single element,¹ we have

$$\mathbb{E}_{w \sim A(S)}[\ell(z, w)] - \mathbb{E}_{w \sim A(S')}[\ell(z, w)] \leq \epsilon$$

for all $z \in Z$, where $\mathbb{E}_{x \sim \mu}[f(x)]$ denotes the expected value of $f(x)$ when x is sampled from μ .

By the following observation, ϵ -uniform stability follows if A satisfies an expected sensitivity condition, if we take the distance on the input space Z^n to be the number of differing elements in (S, S') , and the distance on output parameters to be a function of the loss ℓ .

Fact. For every pair of training sets $S, S' \in Z^n$ that differ in a single element, suppose that we can find a joint distribution $\mu(S, S') \in \mathbb{D}(\mathbb{R}^d \times \mathbb{R}^d)$ such that $\pi_1(\mu) = A(S)$ and $\pi_2(\mu) = A(S')$, where $\pi_1, \pi_2 : \mathbb{D}(\mathbb{R}^d \times \mathbb{R}^d) \rightarrow \mathbb{D}(\mathbb{R}^d)$ give the first and second marginals. If

$$\mathbb{E}_{(w, w') \sim \mu(S, S')}[|\ell(z, w) - \ell(z, w')|] \leq \epsilon$$

for every $z \in Z$, then A is ϵ -uniformly stable.

In a bit more detail, the joint distribution $\mu(S, S')$ is an example of a *expectation coupling* of $A(S)$ and $A(S')$, where $|\ell(z, w) - \ell(z, w')|$ is viewed as a *distance* on pairs of samples (w, w') . Then, if we view the distance on training sets Z^n to be the symmetric distance (number of differing elements

¹In other words, S and S' have the same cardinality and their symmetric difference contains exactly two elements.

between training sets), ϵ -uniform stability follows from expected sensitivity of the function A . To prove stability, then, we will prove expected sensitivity by (i) finding an expectation coupling and (ii) reasoning about the expected value of the distance function. Our logic ExSeL includes elements to handle both tasks.

To demonstrate our logic, we will show ϵ -stability of several versions of the *Stochastic Gradient Method* (SGM), following recent work by Hardt et al. [20]. SGM is a simple and classical optimization algorithm commonly used in machine learning. Typically, the parameter space is \mathbb{R}^d (i.e., that the algorithm is trying to learn d real parameters). SGM maintains a parameter w and iteratively updates this parameter to reduce the loss. Each iteration, SGM selects a uniformly random example z from the input training set S and computes the *gradient* vector g of the function $\ell(z, -) : \mathbb{R}^d \rightarrow [0, 1]$ evaluated at the current parameter w —this vector indicates the direction to move w to decrease the loss. Then, SGM updates w to step along g . After running T iterations, the algorithm returns w as the final parameter choice. We can implement SGM in an imperative language as follows.

```

w ← w0;
t ← 0;
while t < T do
  i ←⊆ [n];
  g ← ∇ℓ(S[i], -)(w);
  w ← w - αt · g;
  t ← t + 1;
return w

```

The program firsts initialize the parameters w to some default value w_0 and initialize the loop counter t to 0. Then, it runs T iterations of the main loop. The first step in the loop samples a uniformly random element index i from $[n] = \{0, 1, \dots, n-1\}$, while the second step computes the gradient g . We will model the gradient operator ∇ as a higher-order function with type $(\mathbb{R}^d \rightarrow [0, 1]) \rightarrow (\mathbb{R}^d \rightarrow \mathbb{R}^d)$.² The third step in the loop updates w to try to decrease the loss. The step size α_t determines how far the algorithm moves; it is a real number that may depend on the iteration t .

Our goal is to verify that this program is ϵ -uniformly stable. At a high level, suppose we have two training sets S_{\triangleleft} and S_{\triangleright} differing in a single example. Viewing the sets as lists, we suppose that the two lists have the same length and $S[i]_{\triangleleft} = S[i]_{\triangleright}$ for all indices i except for a particular index $j \in [n]$. Then, we construct an expectation coupling between the two distributions on output parameters, bounding the expected distance between the outputs w_{\triangleleft} and w_{\triangleright} . Assuming that $\ell(z, -)$ is a Lipschitz function, i.e. $|\ell(z, w) - \ell(z, w')| \leq L\|w - w'\|$ for all $w, w' \in \mathbb{R}^d$ where $\|\cdot\|$ is the usual Euclidean distance, bounding the expected distance between the parameters also bounds the expected losses, implying uniform stability.

Now, let's see how to carry out this verification in our logic. ExSeL is a relational program logic with judgments of the form

$$\vdash \{\Phi; \mathfrak{d}\} s_1 \sim_f s_2 \{\Psi; \mathfrak{d}'\}.$$

Here, s_1, s_2 are two imperative programs, the formulas Φ and Ψ are assertions over pairs of memories $(m_1, m_2) \in \mathcal{M} \times \mathcal{M}$, the distances $\mathfrak{d}, \mathfrak{d}'$ are maps $\mathcal{M} \times \mathcal{M} \rightarrow \mathbb{R}^+$, and $f : \mathbb{R}^+ \rightarrow \mathbb{R}^+$ is a non-negative affine function (i.e., of the form $x \mapsto ax + b$ for $a, b \in \mathbb{R}^+$). The judgment above states that for any two initial memories (m_1, m_2) satisfying the pre-condition Φ , there is an expectation coupling of the output distributions from executing s_1, s_2 on m_1, m_2 respectively such that the expected value of \mathfrak{d}'

²Strictly speaking, this operation is only well-defined if the input function is differentiable; this holds for most of the loss functions considered in the machine learning literature.

on the coupling is at most $f(\mathfrak{d}(m_1, m_2))$ and all pairs of output memories with positive probability satisfy Ψ .

To sketch the verification, we focus on the loop. Let s_a be the sampling command and let s_b be the remainder of the loop body. First, we can show

$$\vdash \{\Phi; \|w_a - w_b\|\} s_a \sim_{\text{id}} s_a \{i_a = i_b; \|w_a - w_b\|\}. \quad (2)$$

We consider each part in turn. In the pre-condition, we abbreviate trivial invariants like $t_a = t_b$ and $\text{Adj}(S_a, S_b)$ as Φ . The post-condition $i_a = i_b$ indicates that the coupling assumes both executions sample the same index i . Finally, the distances $\|w_a - w_b\|$ in the pre- and post-conditions indicate that the expected value of the distance does not grow—this is clear because s_a does not modify w .

Now, we know that the training sets S_a and S_b differ in a single example, say at index j . There are two cases: either we have sampled $i_a = i_b = j$, or we have sampled $i_a = i_b \neq j$. In the first case, we can apply properties of the loss function ℓ and gradient operator ∇ to prove:

$$\vdash \{S[i]_a \neq S[i]_b; \|w_a - w_b\|\} s_b \sim_{+\gamma} s_b \{\Phi; \|w_a - w_b\|\} \quad (3)$$

where $+\gamma$ is the function $x \mapsto x + \gamma$ for a constant γ —since we are considering different examples in the two executions, the resulting parameters may grow a bit farther apart. In the second case we know that the example $S[i]$ is the same in both executions, so we can prove:

$$\vdash \{S[i]_a = S[i]_b; \|w_a - w_b\|\} s_b \sim_{\text{id}} s_b \{\Phi; \|w_a - w_b\|\} \quad (4)$$

That is, the expected distance does not increase. To combine these two cases, we note that the first case happens with probability $1/n$ —this is the probability of sampling index j —while the second case happens with probability $1 - 1/n$. Our logic allows us to scale the bounds accordingly when composing s_a and s_b , yielding

$$\vdash \{\Phi; \|w_a - w_b\|\} s_a; s_b \sim_{+\gamma/n} s_a; s_b \{\Phi; \|w_a - w_b\|\}, \quad (5)$$

since $x \mapsto (1/n) \cdot (x + \gamma) + (1 - 1/n) \cdot x = x + \gamma/n$.

Now that we have a bound on how the distance grows in the body, we can apply the loop rule. Roughly speaking, for a loop running T iterations, this rule simply takes the T -fold composition f^T of the bounding function f ; since f is the linear function $+\gamma/n$, f^T is the linear function $+T\gamma/n$, and we have:

$$\vdash \{\Phi; \|w_a - w_b\|\} \text{sgm} \sim_{+T\gamma/n} \text{sgm} \{\Phi; \|w_a - w_b\|\}. \quad (6)$$

Assuming that the loss function $\ell(-, z)$ is Lipschitz, we know that $|\ell(w, z) - \ell(w', z)| \leq L\|w - w'\|$ for some constant L and so

$$\vdash \{\Phi; \|w_a - w_b\|\} \text{sgm} \sim_{+LT\gamma/n} \text{sgm} \{\Phi; |\ell(w_a, z) - \ell(w_b, z)|\} \quad (7)$$

for every example $z \in Z$. Since w_a and w_b are initialized to the same value w_0 , this judgment gives a μ of the output distributions such that

$$\mathbb{E}_\mu[|\ell(w_a, z) - \ell(w_b, z)|] \leq \|w_0 - w_0\| + LT\gamma/n = LT\gamma/n.$$

Since the left side is larger than $\mathbb{E}_\mu[|\ell(w_a, z) - \ell(w_b, z)|]$, SGM is $LT\gamma/n$ -uniform stable.

3 EXPECTED SENSITIVITY

Before we can present our logic, we first review basic definitions and notations from probability theory related to expected values and probabilistic couplings. Then, we introduce our notions of expected sensitivity and expectation coupling.

3.1 Mathematical preliminaries

Linear and affine functions. We let \mathcal{A} be the set of non-negative affine functions, mapping $z \mapsto \alpha \cdot z + \beta$ where $\alpha, \beta \in \mathbb{R}^+$; $\mathcal{L} \subseteq \mathcal{A}$ be the set of non-negative linear functions, mapping $z \mapsto \alpha \cdot z$; and $\mathcal{C} \subseteq \mathcal{A}$ be the set of non-negative constant functions, mapping $z \mapsto \beta$. We will use the metavariables f for \mathcal{A} and bolded letters (e.g., β) for \mathcal{C} .

Non-negative affine functions can be combined in several ways. Let $f, f' \in \mathcal{A}$ map z to $\alpha \cdot z + \beta$ and $\alpha' \cdot z + \beta'$ respectively, and let $\gamma \in \mathbb{R}^+$.

- *sequential composition:* the function $f' \circ f$ maps z to $(\alpha\alpha') \cdot z + \alpha'\beta + \beta'$;
- *addition:* the function $f + f'$ maps z to $f(z) + f'(z)$;
- *scaling:* the function $(\gamma \cdot f)$ maps z to $\gamma \cdot f(z)$
- *translation:* the function $f + \gamma$ maps z to $f(z) + \gamma$.

We will use shorthand for particularly common functions. For scaling, we write $\bullet\gamma$ for the function mapping z to $\gamma \cdot z$. For translation, we write $+\gamma$ for the function mapping z to $z + \gamma$. The identity function will be simply id (equivalently, $\bullet 1$ or $+0$).

Distances. A *distance* function \mathfrak{d} is a map $A \times A \rightarrow \mathbb{R}^+$, where \mathbb{R}^+ denote the set of non-negative real numbers. Note that we use the term “distance” rather loosely—the core logic does not require distances to satisfy any axioms, like reflexivity, symmetry, triangle inequality, etc. Distances are partially ordered using the pointwise order inherited from the extended reals: we write $\mathfrak{d} \leq \mathfrak{d}'$ if $\mathfrak{d}(a_1, a_2) \leq \mathfrak{d}'(a_1, a_2)$ for all $(a_1, a_2) \in A \times A$.

Distributions. Programs in our language are interpreted in terms of sub-distributions. A (discrete) *sub-distribution* over a set A is a map $\mu : A \rightarrow \mathbb{R}^+$ such that its *support*

$$\text{supp}(\mu) \triangleq \{a \in A \mid \mu(a) \neq 0\}$$

is discrete and its *weight* $|\mu| \triangleq \sum_{a \in \text{supp}(\mu)} \mu(a)$ is well-defined and satisfies $|\mu| \leq 1$. We let $\mathbb{D}(A)$ denote the set of discrete sub-distributions over A . Note that $\mathbb{D}(A)$ is partially ordered using the pointwise inequality inherited from reals. Similarly, equality of distributions is defined extensionally: two distributions are equal if they assign the same value (i.e., *probability*) to each element in their domain. *Events* are map $E : A \rightarrow \mathbb{B}$, where \mathbb{B} denotes the set of booleans. The probability of an event E w.r.t. a sub-distribution μ , written as $\Pr_\mu[E]$, is defined as $\sum_{x \mid E(x)} \mu(x)$.

The *expectation* of a function $f : A \rightarrow \mathbb{R}^+$ w.r.t. a sub-distribution $\mu \in \mathbb{D}(A)$, written $\mathbb{E}_{x \sim \mu}[f(x)]$ or $\mathbb{E}_\mu[f]$ for short, is defined as $\sum_x \mu(x) \cdot f(x)$ when this sum exists, and $+\infty$ otherwise. Expectation satisfies the usual linearity properties: $\mathbb{E}_\mu[f + g] = \mathbb{E}_\mu[f] + \mathbb{E}_\mu[g]$ and $\mathbb{E}_\mu[k \cdot f] = k \cdot \mathbb{E}_\mu[f]$, where addition and scaling of functions is defined in the usual way.

Discrete sub-distributions support several useful constructions. First, they can be given a monadic structure. Let $x \in A$, $\mu \in \mathbb{D}(A)$ and $M : A \rightarrow \mathbb{D}(B)$. Then:

$$\begin{aligned} \text{unit } x &\triangleq a \mapsto \mathbb{1}[x = a] \\ \text{bind } \mu M &\triangleq b \mapsto \sum_{a \in A} \mu(a) \cdot M(a)(b). \end{aligned}$$

Intuitively, $\text{bind } \mu M$ is the distribution from first sampling from from μ and applying M to the sample; in particular, it is a distribution over B . We will write δ_x for the Dirac distribution $\text{unit } x$, and abusing notation, we write $\mathbb{E}_{x \sim \mu}[M]$ and $\mathbb{E}_\mu[M]$ for $\text{bind } \mu M$.

Given a distribution μ over pairs in $A \times B$, we can define the usual projections $\pi_1 : \mathbb{D}(A \times B) \rightarrow \mathbb{D}(A)$ and $\pi_2 : \mathbb{D}(A \times B) \rightarrow \mathbb{D}(B)$ as:

$$\pi_1(\mu)(a) \triangleq \sum_{b \in B} \mu(a, b) \quad \text{and} \quad \pi_2(\mu)(b) \triangleq \sum_{a \in A} \mu(a, b).$$

A *probabilistic coupling* is a distribution over the product space of two distributions, such that its first and second marginals coincide with the first and second distributions. Formally, two sub-distributions $\mu_a \in \mathbb{D}(A)$ and $\mu_b \in \mathbb{D}(B)$ are *coupled* by $\mu \in \mathbb{D}(A \times B)$, written $\mu_a \langle \mu \rangle \mu_b$, iff $\pi_1(\mu) = \mu_a$ and $\pi_2(\mu) = \mu_b$.

3.2 Expected f -sensitivity

Our system is structured around a probabilistic version of sensitivity. Let $f \in \mathcal{A}$, and let \mathfrak{d}_A and \mathfrak{d}_B be distances on A and B respectively.

Definition 3.1. We say that a probabilistic function $g : A \rightarrow \mathbb{D}(B)$ is *expected f -sensitive* (with respect to \mathfrak{d}_A and \mathfrak{d}_B) iff for every $x_1, x_2 \in A$, we have the bound

$$\mathbb{E}_{(y_1, y_2) \sim \mu} [\mathfrak{d}_B(y_1, y_2)] \leq f(\mathfrak{d}_A(x_1, x_2))$$

for some $\mu \in \mathbb{D}(B \times B)$ with marginals $\pi_1(\mu) = g(x_1)$ and $\pi_2(\mu) = g(x_2)$. When f maps z to $\alpha \cdot z + \beta$, we sometimes say that g is *expected (α, β) -sensitive*.

By carefully selecting the distances \mathfrak{d}_A and \mathfrak{d}_B on the input and output spaces, we can recover different notions of probabilistic sensitivity as a consequence of expected f -sensitivity. We derive two particularly useful results here, which we first saw in the introduction.

PROPOSITION 3.2 (AVERAGE SENSITIVITY). *Suppose that $g : A \rightarrow \mathbb{D}(\mathbb{R})$ is expected f -sensitive with respect to distances \mathfrak{d}_A and $|\cdot|$. Then for any two inputs $a_1, a_2 \in A$, we have*

$$|\mathbb{E}_{y_1 \sim g(a_1)}[y_1] - \mathbb{E}_{y_2 \sim g(a_2)}[y_2]| \leq f(\mathfrak{d}_A(a_1, a_2)).$$

PROOF. Let $a_1, a_2 \in A$ be two inputs. Since g is expected f -sensitive, there exists a coupling $g(a_1) \langle \mu \rangle g(a_2)$ such that the expected distance over μ is at most $f(\mathfrak{d}_A(a_1, a_2))$. We can bound:

$$\begin{aligned} |\mathbb{E}_{y_1 \sim g(a_1)}[y_1] - \mathbb{E}_{y_2 \sim g(a_2)}[y_2]| &= |\mathbb{E}_{(y_1, y_2) \sim \mu}[y_1] - \mathbb{E}_{(y_1, y_2) \sim \mu}[y_2]| && \text{(Coupling)} \\ &= |\mathbb{E}_{(y_1, y_2) \sim \mu}[y_1 - y_2]| && \text{(Linearity)} \\ &\leq \mathbb{E}_{(y_1, y_2) \sim \mu}[|y_1 - y_2|] && \text{(Triangle ineq.)} \\ &\leq f(\mathfrak{d}_A(a_1, a_2)) && \text{(Ex. coupling)} \end{aligned}$$

□

PROPOSITION 3.3 (PROBABILISTIC SENSITIVITY). *Suppose that $g : A \rightarrow \mathbb{D}(B)$ is expected f -sensitive with respect to distances \mathfrak{d}_A and \mathfrak{d}_B , where $\mathfrak{d}_B(b_1, b_2) < \beta$ if and only if $b_1 = b_2$. Then for any two inputs $a_1, a_2 \in A$, we have*

$$TV(g(a_1), g(a_2)) \leq f(\mathfrak{d}_A(a_1, a_2))/\beta,$$

where the total variation distance is defined as

$$TV(g(a_1), g(a_2)) \triangleq \max_{E \subseteq B} \left| \Pr_{b_1 \sim g(a_1)} [b_1 \in E] - \Pr_{b_2 \sim g(a_2)} [b_2 \in E] \right|.$$

PROOF. Let $a_1, a_2 \in A$ be two inputs. Since g is expected f -sensitive, there exists a coupling $g(a_1) \langle \mu \rangle g(a_2)$ such that the expected distance \mathfrak{d}_B over μ is at most $f(\mathfrak{d}_A(a_1, a_2))$. We can bound:

$$\begin{aligned} \Pr_{(b_1, b_2) \sim \mu} [b_1 \neq b_2] &= \mathbb{E}_{(b_1, b_2) \sim \mu} [\mathbb{1}[b_1 \neq b_2]] \\ &\leq \mathbb{E}_{(b_1, b_2) \sim \mu} [\mathfrak{d}_B(b_1, b_2)/\beta] && (b_1 \neq b_2 \rightarrow \mathfrak{d}_B \geq \beta) \\ &\leq f(\mathfrak{d}_A(a_1, a_2))/\beta. && \text{(Linearity, ex. coupling)} \end{aligned}$$

By a classical theorem about couplings (see, e.g., Lindvall [28]), the TV distance is at most the probability on the first line. □

Next, we show that expected f -sensitive functions are closed under sequential composition.

PROPOSITION 3.4. *Let $f, f' \in \mathcal{A}$ be non-negative affine functions, and let $\mathfrak{d}_A, \mathfrak{d}_B$ and \mathfrak{d}_C be distances on A, B and C respectively. Assume that $g : A \rightarrow \mathbb{D}(B)$ is expected f -sensitive and $h : B \rightarrow \mathbb{D}(C)$ is expected f' -sensitive. Then the (monadic) composition $a \mapsto \mathbb{E}_{b \sim g(a)}[h(b)] : A \rightarrow \mathbb{D}(C)$ of g and h is expected $f' \circ f$ -sensitive.*

PROOF. By unfolding definitions and applying linearity of expectation. In more detail, let $a_1, a_2 \in A$ be any pair of inputs. Since g is expected f -sensitive, there is some distribution $\mu \in \mathbb{D}(A \times A)$ such that

$$\mathbb{E}_\mu[\mathfrak{d}_B] \leq f(\mathfrak{d}_A(a_1, a_2)) \quad (8)$$

with marginals $\pi_1(\mu) = g(a_1), \pi_2(\mu) = g(a_2)$. Similarly, for every $b_1, b_2 \in B$, there is some distribution $M(b_1, b_2) \in \mathbb{D}(C \times C)$ such that

$$\mathbb{E}_{M(b_1, b_2)}[\mathfrak{d}_C] \leq f'(\mathfrak{d}_B(b_1, b_2)) \quad (9)$$

with marginals $\pi_1(M(b_1, b_2)) = h(b_1)$ and $\pi_2(M(b_1, b_2)) = h(b_2)$, since h is f' -sensitive.

Define the distribution $\mu' \triangleq \mathbb{E}_\mu[M]$. It is straightforward to check the marginals $\pi_1(\mu')(a_1) = \mathbb{E}_{g(a_1)}[h]$ and $\pi_2(\mu')(a_2) = \mathbb{E}_{g(a_2)}[h]$. To check the distance condition, we can bound the expected distance as desired:

$$\begin{aligned} \mathbb{E}_{\mu'}[\mathfrak{d}_C] &= \sum_{c_1, c_2} \mathfrak{d}_C(c_1, c_2) \cdot \sum_{b_1, b_2} \mu(b_1, b_2) \cdot M(b_1, b_2)(c_1, c_2) \\ &= \sum_{b_1, b_2} \mu(b_1, b_2) \sum_{c_1, c_2} \mathfrak{d}_C(c_1, c_2) \cdot M(b_1, b_2)(c_1, c_2) \\ &\leq \sum_{b_1, b_2} \mu(b_1, b_2) f'(\mathfrak{d}_B(b_1, b_2)) \quad (\text{Eq. (9)}) \\ &\leq f' \left(\sum_{b_1, b_2} \mu(b_1, b_2) \cdot \mathfrak{d}_B(b_1, b_2) \right) \quad (\text{Linearity, } f' \text{ affine}) \\ &\leq f'(f(\mathfrak{d}_A(a_1, a_2))) \quad (\text{Eq. (8), } f' \text{ non-decreasing}) \\ &= f' \circ f(\mathfrak{d}_A(a_1, a_2)). \end{aligned}$$

□

Taking the pre- and post-distances to be the same yields a useful consequence.

PROPOSITION 3.5. *Let \mathfrak{d} be a distance over A and let $f \in \mathcal{F}$. Let $g : A \rightarrow \mathbb{D}(A)$ be an expected f -sensitive function. Then for every $T \in \mathbb{N}$, the T -fold (monadic) composition g^T of g is expected f^T -sensitive, i.e. for every $x_1, x_2 \in A$, there exists a coupling μ of $g^T(x_1)$ and $g^T(x_2)$ such that*

$$\mathbb{E}_\mu[\mathfrak{d}] \leq f^T(\mathfrak{d}(x_1, x_2))$$

An important consequence of this proposition is a variant of Banach's fixed point theorem. Informally, the theorem states that, under some reasonable conditions on \mathfrak{d} , contractive probabilistic maps $g : A \rightarrow \mathbb{D}(A)$ have a unique stationary distribution, where a probabilistic map is contractive iff it is expected f -sensitive for a map f of the form $z \mapsto \alpha \cdot z$, with $\alpha < 1$.

3.3 Continuity from expectation couplings

Expectation couplings are a quantitative extension of probabilistic couplings, where the expected distance over μ is bounded.

Definition 3.6 (Expectation couplings). Let $\mathfrak{d} : A \times B \rightarrow \mathbb{R}^+$ be a distance and let $\delta \in \mathbb{R}^+$ be a constant. Moreover, let $\mu_a \in \mathbb{D}(A)$, $\mu_b \in \mathbb{D}(B)$ and $\mu \in \mathbb{D}(A \times B)$. Then μ is an (\mathfrak{d}, δ) -*expectation coupling* (or simply, an *expectation coupling*) for μ_a and μ_b if $\mu_a \langle \mu \rangle_{\mathfrak{d} \leq \delta} \mu_b$ and $\mathbb{E}_\mu[\mathfrak{d}] \leq \delta$.

We write $\mu_a \langle \mu \rangle_{\mathfrak{d} \leq \delta}^\Phi \mu_b$ when μ is an expectation coupling with support $\text{supp}(\mu)$ contained in a binary relation $\Phi \subseteq A \times B$. We omit Φ when it is the trivial (always true) relation.

Expectation couplings are closely linked to expected f -sensitivity.

PROPOSITION 3.7. *A probabilistic function $g : A \rightarrow \mathbb{D}(B)$ is expected f -sensitive (with respect to \mathfrak{d}_A and \mathfrak{d}_B) iff for every $x_1, x_2 \in A$, there exists μ such that $g(x_1) \langle \mu \rangle_{\mathfrak{d} \leq \delta} g(x_2)$, where $\delta = f(\mathfrak{d}_A(x_1, x_2))$.*

Expectation couplings are closed under sequential composition: given an expectation coupling between two distributions μ_a and μ_b , two functions $M_a : A \rightarrow \mathbb{D}(A')$ and $M_b : B \rightarrow \mathbb{D}(B')$ and a function M mapping pairs of samples in $(a, b) \in A \times B$ to an expectation coupling of $M_a(a)$ and $M_b(b)$, we have an expectation coupling of the two distributions from sampling μ_a and μ_b and running M_a and M_b , respectively.

PROPOSITION 3.8 (COMPOSITION OF EXPECTATION COUPLINGS). *Let $\Phi \subseteq A \times B$, $\mathfrak{d} : A \times B \rightarrow \mathbb{R}^+$, $\Psi \subseteq A \times B$, $\mathfrak{d}' : A \times B \rightarrow \mathbb{R}^+$, $\delta \in \mathbb{R}^+$, and $f \in \mathcal{A}$. Let $\mu_a \in \mathbb{D}(A)$, $M_a : A \rightarrow \mathbb{D}(A')$, and let $\mu'_a = \mathbb{E}_{\mu_a}[M_a]$. Let $\mu_b \in \mathbb{D}(B)$, $M_b : B \rightarrow \mathbb{D}(B')$, and set $\mu'_b = \mathbb{E}_{\mu_b}[M_b]$. Suppose we have functions $\mu \in \mathbb{D}(A \times B)$ and $M : (A \times B) \rightarrow \mathbb{D}(A' \times B')$ such that:*

- (1) $\mu_a \langle \mu \rangle_{\mathfrak{d} \leq \delta}^\Phi \mu_b$ and
- (2) $M_a(a) \langle M(a, b) \rangle_{\mathfrak{d}' \leq f(\mathfrak{d}(a, b))}^\Psi M_b(b)$ for every $(a, b) \in \Phi$.

Then $\mu'_a \langle \mu' \rangle_{\mathfrak{d}' \leq f(\delta)}^\Psi \mu'_b$, where μ' is the monadic composition $\mathbb{E}_{(a, b) \sim \mu}[M(a, b)]$.

PROOF SKETCH. By unfolding definitions and checking the support, marginal, and expected distance properties. The support and marginal conditions follow by the support and marginal conditions for the premises, while the expected distance condition follows by an argument similar to Proposition 3.4 using that f is affine and non-decreasing. We defer full details to the appendix. \square

4 PROGRAM LOGIC

As we have seen, expectation couplings can be composed together and the existence of an expectation coupling implies expected sensitivity. Accordingly, we can give a program logic to reason about expectation couplings in a structured way.

4.1 Programming language

We base our development on **PWHILE**, a core language with deterministic assignments, probabilistic assignments, conditionals, and loops. The syntax of statements is defined by the grammar:

$$s ::= x \leftarrow e \mid x \stackrel{e}{\leftarrow} g \mid s; s \mid \text{skip} \mid \text{if } e \text{ then } s \text{ else } s \mid \text{while } e \text{ do } s$$

where x , e , and g range over variables in \mathcal{V} , expressions in \mathcal{E} and distribution expressions in \mathcal{D} respectively. \mathcal{E} is defined inductively from \mathcal{V} and operators, while \mathcal{D} consists of parametrized distributions—for instance, the uniform distribution $[n]$ over the set $\{0, \dots, n-1\}$ or the Bernoulli distribution $\text{Bern}(p)$, where the numeric parameter $p \in [0, 1]$ is the probability of returning true. We will write *if e then s* as shorthand for *if e then s else skip*. We implicitly assume that programs are well-typed w.r.t. a standard typing discipline; for instance, the guard expressions of conditionals and loops are booleans, operations on expressions are applied to arguments of the correct type, etc.

Following the seminal work of Kozen [25], the denotational semantics of programs is given as sub-distribution transformers over memories. To keep measure-theoretic technicalities to a minimum, we limit our focus to discrete sub-distributions. Memories are type-preserving mappings

$$\begin{aligned}
\llbracket \text{skip} \rrbracket_m &= \delta_m & \llbracket x \leftarrow^{\$} g \rrbracket_m &= \mathbb{E}_{v \sim \llbracket g \rrbracket_m} [\delta_{m[x:=v]}] \\
\llbracket x \leftarrow e \rrbracket_m &= \delta_{m[x:=\llbracket e \rrbracket_m]} & \llbracket \text{if } e \text{ then } s_1 \text{ else } s_2 \rrbracket_m &= \text{if } \llbracket e \rrbracket_m \text{ then } \llbracket s_1 \rrbracket_m \text{ else } \llbracket s_2 \rrbracket_m \\
\llbracket s_1; s_2 \rrbracket_m &= \mathbb{E}_{\xi \sim \llbracket s_1 \rrbracket_m} [\llbracket s_2 \rrbracket_{\xi}] & \llbracket \text{while } b \text{ do } s \rrbracket_m &= \lim_{n \rightarrow \infty} \llbracket (\text{if } b \text{ then } s)_{|-b}^n \rrbracket_m
\end{aligned}$$

Fig. 1. Denotational semantics of programs

from variables to values—formally, we define an interpretation for each type and require that a variable of type T is mapped to an element of the interpretation of T . We let \mathcal{M} denote the set of memories. Then, the semantics $\llbracket e \rrbracket_m$ of a (well-typed) expression e is defined in the usual way as an element of the interpretation of the type of e , and parametrized by a memory m . The interpretation of distribution expressions is defined and denoted likewise. Now, we can define the semantics of statements.

Definition 4.1 (Semantics of statements).

- The semantics $\llbracket s \rrbracket_m$ of a statement s w.r.t. to some initial memory m is a sub-distribution over states, and is defined by the clauses of Fig. 1.
- The (lifted) semantics $\llbracket s \rrbracket_{\mu}$ of a statement s w.r.t. to some initial sub-distribution μ over memories is a sub-distribution over states, and is defined as $\llbracket s \rrbracket_{\mu} \triangleq \mathbb{E}_{m \sim \mu} [\llbracket s \rrbracket_m]$.

The semantics of programs given in Figure 1 is standard. The most interesting case is for loops, where the interpretation of a while loop is the limit of the interpretations of its finite unrollings. Formally, the n^{th} truncated iterate of the loop while b do s is defined as

$$\overbrace{\text{if } b \text{ then } s; \dots; \text{if } b \text{ then } s; \text{if } b \text{ then abort}}^{n \text{ times}}$$

which we represent using the shorthand $(\text{if } b \text{ then } s)_{|-b}^n$. For any initial sub-distribution μ , applying the truncated iterates yields an increasing and bounded sequence of sub-distributions. The limit of this sequence is well-defined, and gives the semantics of the while loop.

4.2 Proof system

EXSEL is a Hoare-style logic augmented to consider two programs instead of one (a so-called *relational logic*). EXSEL judgments are of the form

$$\{\Phi; \mathfrak{d}\} s_1 \sim_f s_2 \{\Psi; \mathfrak{d}'\}$$

for programs s_1, s_2 , assertions $\Phi, \Psi : \mathcal{M} \times \mathcal{M} \rightarrow \mathbb{B}$, distances $\mathfrak{d}, \mathfrak{d}' : \mathcal{M} \times \mathcal{M} \rightarrow \mathbb{R}^+$, and a non-negative affine function $f \in \mathcal{A}$. We will refer to f as a *distance transformer*.

Definition 4.2. A judgment $\{\Phi; \mathfrak{d}\} s_1 \sim_f s_2 \{\Psi; \mathfrak{d}'\}$ is valid iff for every memories m_1, m_2 s.t. $(m_1, m_2) \models \Phi$, there exists μ such that

$$\llbracket s_1 \rrbracket_{m_1} \langle \mu \rangle_{\mathfrak{d}' \leq f(\mathfrak{d}(m_1, m_2))}^{\Psi} \llbracket s_2 \rrbracket_{m_2}$$

The notion of validity is closely tied to expected f -sensitivity. For instance, if the judgment

$$\{\top; \mathfrak{d}\} s \sim_f s \{\top; \mathfrak{d}'\}$$

is valid, then the program s interpreted as a function $\llbracket s \rrbracket : \mathcal{M} \rightarrow \mathbb{D}(\mathcal{M})$ is expected f -sensitive with respect to distances \mathfrak{d} and \mathfrak{d}' . In fact, the pre- and post-conditions Φ and Ψ can also be interpreted as

distances. If we map Φ to the pre-distance $\mathfrak{d}_\Phi(m_1, m_2) \triangleq \mathbb{1}[(m_1, m_2) \notin \Phi]$, and Ψ to the post-distance $\mathfrak{d}_\Psi(m_1, m_2) \triangleq \mathbb{1}[(m_1, m_2) \notin \Psi]$, then the judgment

$$\{\top; \mathfrak{d}_\Phi\} s_1 \sim_{\text{id}} s_2 \{\top; \mathfrak{d}_\Psi\}$$

is equivalent to

$$\{\Phi; -\} s_1 \sim_- s_2 \{\Psi; -\}$$

where dashes stand for arbitrary distances and distance transformers.

Now, we introduce some notation and then present the rules of the logic. First, note that each boolean expression e naturally yields two assertions e_{\leftarrow} and e_{\rightarrow} , resp. called its left and right injections:

$$\begin{aligned} m_1 \models e &\iff m_1, m_2 \models e_{\leftarrow} \\ m_2 \models e &\iff m_1, m_2 \models e_{\rightarrow} \end{aligned}$$

The notation naturally extends to mappings from memories to booleans. Second, several rules use substitutions. Given a memory m , variable x and expression e such that the types of x and e agree, we let $m[x := e]$ denote the unique memory m' such that $m(y) = m'(y)$ if $y \neq x$ and $m'(x) = \llbracket e \rrbracket_m$. Then, given a variable x (resp. x'), an expression e (resp. e'), and an assertion Φ , we define the assertion $\Phi[x_{\leftarrow}, x'_{\rightarrow} := e_{\leftarrow}, e'_{\rightarrow}]$ by the clause:

$$\Phi[x_{\leftarrow}, x'_{\rightarrow} := e_{\leftarrow}, e'_{\rightarrow}](m_1, m_2) \triangleq \Phi(m_1[x := e], m_2[x' := e'])$$

Substitution of distances is defined similarly. One can also define one-sided substitutions, for instance $\Phi[x_{\leftarrow} := e_{\leftarrow}]$.

We now turn to the rules of the proof system in Fig. 2. The rules can be divided into two groups: two-sided rules relate programs with the same structure, while structural rules apply to two programs of any shape. The full logic ExSEL also features one-sided rules for relating a program with a fixed shape to a program of unknown shape; later we will show that many of these rules are derivable. We briefly comment on each of the rules, starting with the two-sided rules.

The [ASSG] rule is similar to the usual rule for assignments, and substitutes the assigned expressions into the pre-condition and pre-distance.

The [RAND] rule is similar to the pRHL rule for random assignments; again, one substitutes into the pre-condition and distance. Informally the rule requires to exhibit the existence of a coupling, given as a bijection between their support, between the two distributions used for sampling in the left and right program.

The [SEQCASE] rule combines sequential composition with a case analysis on properties satisfied by intermediate memories, i.e. after executing the programs s_1 and s_2 . Informally, the rule considers events $e_1 \dots e_n$ such that Ψ entails $\bigvee_i e_{i_{\leftarrow}}$. Provided one can relate for every i the programs s'_1 and s'_2 with distance transformer f_i , pre-condition $\Psi \wedge e_{i_{\leftarrow}}$; \mathfrak{d}' and post-condition Ψ' ; \mathfrak{d}'' , one can conclude that $s_1; s'_1$ and $s_2; s'_2$ are related under distance transformer f , where f upper bounds the functions f_i weighted by the probability of each case.

The [WHILE] rule for while loops considers two loops that execute synchronously, and whose loop bodies satisfy the invariant Ψ ; \mathfrak{d}' . The rule additionally requires that both loops perform exactly n steps, and that there exists a variant i initially set to n and decreasing by 1 at each iteration. Assuming that f_k denotes the distance transformer corresponding to the $(n - k)$ th iteration, i.e. the iteration where the variant i is equal to k , the distance transformer for the while loops is the function composition of the distance transformers: $f_1 \circ \dots \circ f_n$.

The remaining rules are structural rules. The [CONSEQ] rule captures the fact that validity is preserved by weakening the post-conditions, strengthening the pre-conditions, and increasing the distance transformer.

The [STRUCT] rule captures the fact that validity is preserved by replacing programs by equivalent ones. Fig. 4 gives the rules for proving two programs s, s' equivalent under some relational assertion Φ ; the judgments are of the form $\Phi \vdash s \equiv s'$. We keep the notion of structural equivalence as simple as possible.

The [FRAME-D] rule is a structural rule, analogous to a typical frame rule, that allows to modify the distance in a judgment. Assuming that the distance δ'' is not modified by the statements of the judgments and f is a linear function such that $x \leq f(x)$ for all x , validity is preserved when adding δ'' to the pre-distance and post-distance of the judgment. Formally, $MV(s)$ denotes the set of modified variables of s and the notation $\delta'' \# MV(s_1), MV(s_2)$ states that for every memories m_1 and m'_1 that coincide on the non-modified variables of s_1 and m_2 and m'_2 that coincide on the non-modified variables of s_2 , we have $\delta''(m_1, m_2) = \delta''(m'_1, m'_2)$.

THEOREM 4.3 (SOUNDNESS). *For every derivable judgment $\vdash \{\Phi; \delta\} s_1 \sim_f s_2 \{\Psi; \delta'\}$ and initial memories m_1 and m_2 such that $(m_1, m_2) \models \Phi$, there exists μ such that*

$$\llbracket s_1 \rrbracket_{m_1} \langle \mu \rangle_{\delta' \leq f(\delta(m_1, m_2))}^{\Psi} \llbracket s_2 \rrbracket_{m_2}.$$

PROOF. By induction on the derivation. We defer the details to the appendix. \square

4.3 Derived rules and weakest pre-condition

Fig. 3 presents some useful derived rules of our logic, including rules for standard sequential composition and conditionals, and one-sided rules.

The [SEQ] rule for sequential composition simply composes the two product programs in sequence. This rule reflects the compositional property of couplings. It can be derived from the rule [SEQCASE] by taking e_1 to be true.

The [COND] rule for conditional statements requires that the two guards of the left and right programs are equivalent under the pre-condition, and then that each branch is related.

The [CASE] rule allows proving a judgment by case analysis; specifically, the validity of a judgment can be established from the validity of two judgments, one where the boolean-valued pre-condition is strengthened with e and the other where the pre-condition is strengthened with $\neg e$.

The [ASSG-L] is the left one-sided rule for assignment. It can be derived from the assignment rule using structural equivalence. The full version of the logic also has similar one-sided rules for other constructs, notably random assignments and conditionals. Using one sided-rules, one can define a relational weakest pre-condition calculus wp , taking as inputs two loop-free and deterministic programs, a post-condition, and a distance, and returning a pre-condition and a distance.

PROPOSITION 4.4. *Let $(\Phi', \delta') = \text{wp}(s_1, s_2, \Psi, \delta')$. Assume $\Phi \implies \Phi'$ and $\delta(m_1, m_2) \leq \delta'(m_1, m_2)$ for every $(m_1, m_2) \models \Phi$. Then $\vdash \{\Phi; \delta\} s_1 \sim_{\text{id}} s_2 \{\Psi; \delta'\}$.*

5 UNIFORM STABILITY OF STOCHASTIC GRADIENT METHOD, REVISITED

Now that we have described the logic, let's return to the Stochastic Gradient Method we first saw in § 2. Recall that the loss function has type $\ell : Z \rightarrow \mathbb{R}^d \rightarrow [0, 1]$. We consider two versions: one where the loss function $\ell(z, -)$ is convex, and one where $\ell(z, -)$ may be non-convex. The algorithm is the same in both cases, but the stability properties require different proofs. For convenience, we

$$\begin{array}{c}
 \text{[ASSG]} \frac{}{\vdash \{\Psi[x_{1_{\leftarrow}} := e_1, x_{2_{\rightarrow}} := e_2]; \delta'[x_{1_{\leftarrow}} := e_1, x_{2_{\rightarrow}} := e_2]\} x_1 \leftarrow e_1 \sim_{\text{id}} x_2 \leftarrow e_2 \{\Psi; \delta'\}} \\
 \\
 \text{[RAND]} \frac{h : \text{supp}(g_1) \stackrel{\perp}{\mapsto} \text{supp}(g_2) \quad \delta \triangleq \mathbb{E}_{v \sim g_1}[\delta'[x_{1_{\leftarrow}} := v, x_{2_{\rightarrow}} := h(v)]] \quad \forall v \in \text{supp}(g_1). g_1(v) = g_2(h(v))}{\vdash \{\forall v \in \text{supp}(g_1), \Psi[x_{1_{\leftarrow}} := v, x_{2_{\rightarrow}} := h(v)]; \delta\} x_1 \stackrel{\perp}{\sim} g_1 \sim_{\text{id}} x_2 \stackrel{\perp}{\sim} g_2 \{\Psi; \delta'\}} \\
 \\
 \text{[SEQCASE]} \frac{\begin{array}{l} \vdash \{\Phi; \delta\} s_1 \sim_{f_0} s_2 \{\Psi; \delta'\} \quad \forall i \in I, \vdash \{\Psi \wedge e_{i_{\leftarrow}}; \delta'\} s'_1 \sim_{f_i} s'_2 \{\Psi'; \delta''\} \\ \forall m_1, m_2 \models \Phi, (\sum_{i \in I} \text{Pr}_{[s_1]_{m_1}}[e_i] \cdot f_i) \circ f_0 \leq f \quad \models \Psi \implies \bigvee_{i \in I} e_{i_{\leftarrow}} \end{array}}{\vdash \{\Phi; \delta\} s_1; s'_1 \sim_f s_2; s'_2 \{\Psi'; \delta''\}} \\
 \\
 \text{[WHILE]} \frac{\forall 0 < k \leq n. \vdash \{\Psi \wedge e_{1_{\leftarrow}} \wedge i_{\leftarrow} = k; \delta_k\} s_1 \sim_{f_k} s_2 \{\Psi \wedge i_{\leftarrow} = k - 1; \delta_{k-1}\} \quad \models \Psi \implies e_{\leftarrow} = e_{\rightarrow} \wedge (i_{\leftarrow} \leq 0 \iff \neg e_{\leftarrow})}{\vdash \{\Psi \wedge i_{\leftarrow} = n; \delta_n\} \text{ while } e_1 \text{ do } s_1 \sim_{f_1 \circ \dots \circ f_n} \text{ while } e_2 \text{ do } s_2 \{\Psi \wedge i_{\leftarrow} = 0; \delta_0\}} \\
 \\
 \text{[CONSEQ]} \frac{\begin{array}{l} \vdash \{\Phi; \delta\} s_1 \sim_f s_2 \{\Psi; \delta'\} \\ \models \Phi' \implies \Phi \quad \models \Psi \implies \Psi' \quad \models \Phi' \implies f(\delta) \leq f(\delta'') \quad \models \Psi \implies \delta''' \leq \delta' \end{array}}{\vdash \{\Phi'; \delta'''\} s_1 \sim_{f'} s_2 \{\Psi'; \delta'''\}} \\
 \\
 \text{[STRUCT]} \frac{\begin{array}{l} \vdash \{\Phi; \delta\} s_1 \sim_f s_2 \{\Psi; \delta'\} \\ \Phi_1 \vdash s_1 \equiv s'_1 \quad \Phi_2 \vdash s_2 \equiv s'_2 \quad \forall (m_1, m_2) \models \Phi, \Phi_1(m_1) \wedge \Phi_2(m_2) \end{array}}{\vdash \{\Phi; \delta\} s'_1 \sim_f s'_2 \{\Psi; \delta'\}} \\
 \\
 \text{[FRAME-D]} \frac{\begin{array}{l} f \in \mathcal{L} \quad \delta'' \# \text{MV}(s_1), \text{MV}(s_2) \quad \models \Phi \implies \delta'' \leq f(\delta'') \\ \vdash \{\Phi; \delta\} s_1 \sim_f s_2 \{\Psi; \delta'\} \end{array}}{\vdash \{\Phi; \delta + \delta''\} s_1 \sim_f s_2 \{\Psi; \delta' + \delta''\}}
 \end{array}$$

Fig. 2. Selected proof rules

$$\begin{array}{c}
 \text{[SEQ]} \frac{\vdash \{\Phi; \delta\} s_1 \sim_f s_2 \{\Xi; \delta'\} \quad \vdash \{\Xi; \delta'\} s'_1 \sim_{f'} s'_2 \{\Psi; \delta''\}}{\vdash \{\Phi; \delta\} s_1; s'_1 \sim_{f \circ f'} s_2; s'_2 \{\Psi; \delta''\}} \\
 \\
 \text{[CASE]} \frac{\vdash \{\Phi \wedge e_{\leftarrow}; \delta\} s_1 \sim_f s_2 \{\Psi; \delta'\} \quad \vdash \{\Phi \wedge \neg e_{\leftarrow}; \delta\} s_1 \sim_f s_2 \{\Psi; \delta'\}}{\vdash \{\Phi; \delta\} s_1 \sim_f s_2 \{\Psi; \delta'\}} \\
 \\
 \text{[COND]} \frac{\begin{array}{l} \models \Phi \implies e_{1_{\leftarrow}} = e_{2_{\rightarrow}} \\ \vdash \{\Phi \wedge e_{1_{\leftarrow}}; \delta\} s_1 \sim_f s_2 \{\Psi; \delta'\} \quad \vdash \{\Phi \wedge \neg e_{1_{\leftarrow}}; \delta\} s'_1 \sim_f s'_2 \{\Psi; \delta'\} \end{array}}{\vdash \{\Phi; \delta\} \text{ if } e_1 \text{ then } s_1 \text{ else } s'_1 \sim_f \text{ if } e_2 \text{ then } s_2 \text{ else } s'_2 \{\Psi; \delta'\}} \\
 \\
 \text{[ASSG-L]} \frac{}{\vdash \{\Psi[x_{1_{\leftarrow}} := e_1]; \delta[x_{1_{\leftarrow}} := e_1]\} x_1 \leftarrow e_1 \sim_{\text{id}} \text{skip} \{\Psi; \delta'\}}
 \end{array}$$

Fig. 3. Selected derived rules

$$\begin{array}{c}
\frac{}{\Phi \vdash s \equiv s} \quad \frac{\Phi \vdash s_1 \equiv s_2}{\Phi \vdash s_2 \equiv s_1} \quad \frac{}{\Phi \vdash x \stackrel{\$}{\leftarrow} \delta_x \equiv \text{skip}} \quad \frac{\Phi \implies x = e}{\Phi \vdash x \leftarrow e \equiv \text{skip}} \quad \frac{}{\Phi \vdash s; \text{skip} \equiv s} \\
\frac{}{\Phi \vdash \text{skip}; s \equiv s} \quad \frac{\Phi \vdash s_1 \equiv s'_1}{\Phi \vdash s_1; s_2 \equiv s'_1; s_2} \quad \frac{\top \vdash s_2 \equiv s'_2}{\Phi \vdash s_1; s_2 \equiv s_1; s'_2} \quad \frac{\Phi \implies e}{\Phi \vdash \text{if } e \text{ then } s \text{ else } s' \equiv s} \\
\frac{\Phi \implies \neg e}{\Phi \vdash \text{if } e \text{ then } s \text{ else } s' \equiv s'} \quad \frac{\Phi \wedge e \vdash s_1 \equiv s_2 \quad \Phi \wedge \neg e \vdash s'_1 \equiv s'_2}{\Phi \vdash \text{if } e \text{ then } s_1 \text{ else } s'_1 \equiv \text{if } e \text{ then } s_2 \text{ else } s'_2}
\end{array}$$

Fig. 4. Equivalence rules

reproduce the code sgm:

```

w ← w0;
t ← 0;
while t < T do
  i ←$ [n];
  g ← ∇ℓ(S[i], -)(w);
  w ← w - αt · g;
  t ← t + 1;
return w

```

We will assume that $\ell(z, -)$ is L -Lipschitz for all z : for all $w, w' \in \mathbb{R}^d$, we can bound $|\ell(z, w) - \ell(z, w')| \leq L\|w - w'\|$ where $\|\cdot\|$ is the usual Euclidean norm on \mathbb{R}^d :

$$\|x\| \triangleq \left(\sum_{i=1}^d x_i^2 \right)^{1/2}$$

Furthermore, we will assume that the loss function is β -smooth: the gradient $\nabla\ell(z, -) : \mathbb{R}^d \rightarrow \mathbb{R}^d$ must be β -Lipschitz.

5.1 SGM with convex loss

Suppose that the function $\ell(z, -)$ is a convex function for every z , i.e., we have: $\langle (\nabla\ell(z, -))(w) - (\nabla\ell(z, -))(w'), w - w' \rangle \geq 0$ where $\langle x, y \rangle$ is the inner product between two vectors $x, y \in \mathbb{R}^d$:

$$\langle x, y \rangle \triangleq \sum_{i=1}^d x_i \cdot y_i.$$

When the step sizes satisfy $0 \leq \alpha_t \leq 2/\beta$, we can prove uniform stability of SGM in this case by following the strategy outlined in § 2. We refer back to the judgments there, briefly describing how to apply the rules (for lack of space, we defer details to the appendix). Let s_a be the sampling command, and s_b be the rest of the loop body. We will prove the following judgment:

$$\vdash \{\Phi; \|w_{\triangleleft} - w_{\triangleright}\|\} \text{sgm} \sim_{+\gamma} \text{sgm} \{\Phi; |\ell(w_{\triangleleft}, z) - \ell(w_{\triangleright}, z)|\},$$

where $\Phi \triangleq \text{Adj}(S_{\triangleleft}, S_{\triangleright}) \wedge (w_0)_{\triangleleft} = (w_0)_{\triangleright} \wedge t_{\triangleleft} = t_{\triangleright}$ and

$$\gamma \triangleq \frac{2L^2}{n} \sum_{t=0}^{T-1} \alpha_t.$$

By soundness (Theorem 4.3), this implies that SGM is γ -uniformly stable.

As before, we will first establish a simpler judgment:

$$\vdash \{\Phi; \|w_a - w_b\|\} \text{sgm} \sim_{+\gamma/L} \text{sgm} \{\Phi; \|w_a - w_b\|\}.$$

As we proceed through the proof, we will indicate the corresponding step from the proof outline in § 2. Let j be the index such that the $S[j]_a \neq S[j]_b$; this is the index of the differing example. First, we couple the samplings in s_a with the identity coupling, using rule [RAND] with $h = \text{id}$ (Eq. (2)). Next, we perform a case analysis on whether we sample the differing vertex or not. We can define guards $e_ = \triangleq i = j$ and $e_\neq \triangleq i \neq j$, and then apply the probabilistic case rule [SEQCASE]. In the case $e_ =$, we can use the Lipschitz property of $\ell(z, -)$ and some properties of the norm $\|\cdot\|$ to prove

$$\vdash \{\Phi \wedge e_ =; \|w_a - w_b\|\} s_b \sim_{+2\alpha_t L} s_b \{\Phi; \|w_a - w_b\|\};$$

this corresponds to Eq. (3). In the case e_\neq , we know that the examples are the same in both runs. So, can use the Lipschitz, smoothness, and convexity of $\ell(z, -)$ to prove:

$$\vdash \{\Phi \wedge e_\neq; \|w_a - w_b\|\} s_b \sim_{\text{id}} s_b \{\Phi; \|w_a - w_b\|\};$$

this corresponds to Eq. (4). Applying [SEQCASE], noting that the probability of e_\neq is $1 - 1/n$ and the probability of $e_ =$ is $1/n$, we can bound the expected distance for the loop body (Eq. (5)). Applying the rule [WHILE], we can bound the distance for the whole loop (Eq. (6)). Finally, we can use the Lipschitz property of $\ell(z, -)$ and the rule [CONSEQ] to prove the desired judgment.

5.2 SGM with non-convex loss

When the loss function is non-convex, the previous proof no longer goes through. However, we can still verify the uniform stability bound by Hardt et al. [20]. Roughly, their proof proceeds by showing that with sufficiently high probability, SGM does not select the differing example until many iterations have already passed. If the step size α_t is taken to be rapidly decreasing, SGM will be contracting when it visits the differing example.

Technically, Hardt et al. [20] prove uniform stability by dividing the proof into two pieces. First they show that with sufficiently high probability, the algorithm does not select the differing example before a fixed cutoff time t_0 . In particular, with high probability the parameters w_a and w_b are equal up to iteration t_0 . Then, they prove a uniform stability bound for SGM started at iteration t_0 , assuming $w_a = w_b$.

This proof can also be carried out in ExSEL, with some extensions. First, we split the SGM program into two loops: iterations before t_0 , and iterations after t_0 . The probability of $w_a \neq w_b$ is precisely the expected value of the indicator function $\mathbb{1}[w_a \neq w_b]$, which is 1 if the parameters are not equal and 0 otherwise. Thus, we can bound the probability for the first loop by bounding this expected value in ExSEL. For the second loop, we can proceed much like we did for standard SGM: assume that the parameters are initially equal, and then bound the expected distance on parameters.

The most difficult part is gluing these two pieces together. Roughly, we want to perform case analysis on $w_a = w_b$ but this event depends on both sides—the existing probabilistic case rule [SEQCASE] does not apply. However, we can give an advanced probabilistic case rule, [SEQCASE-A], that does the trick. We defer the details to the appendix.

6 POPULATION DYNAMICS

Our second example comes from the field of evolutionary biology. Consider an infinite population separated into $m \in \mathbb{N}$ classes of organisms. The population at time t is described by a probability vector $\vec{x}_t = (x_1, \dots, x_m)$, where x_i represents the fraction of the population belonging to the class i . In the RSM model, the evolution is described by a function f —called the *step function*—which

$$[\text{MULT-MAX}] \frac{}{\vdash \{\top; \|\vec{p}_a - \vec{p}_b\|_1\} \vec{x}_a \stackrel{s}{\leftarrow} \mathbf{Mult}(\vec{p}_a) \sim_{\text{id}} \vec{x}_b \stackrel{s}{\leftarrow} \mathbf{Mult}(\vec{p}_b) \{ \vec{x}_a, \vec{x}_b \in \{0, 1\}^m; \|\vec{x}_a - \vec{x}_b\|_1 \}}$$

Fig. 5. Maximal coupling rule for multinomial

updates the probability vectors. More precisely, the population at time $t + 1$ is given as the average of $N \in \mathbb{N}$ samples according to the distribution $f(\vec{x}_t)$. A central question is whether this process mixes rapidly: starting from two possibly different population distributions, how fast do the populations converge?

We will verify a probabilistic property that is the main result needed to show rapid mixing: there is a coupling of the population distributions such that the expected distance between the two populations decreases exponentially quickly. Concretely, let $m \in \mathbb{N}$ be the number of different classes. We will work with real vectors $\vec{x} = (x_1, \dots, x_m) \in \mathbb{R}^m$, along with the associated norm $\|\vec{x}\|_1 \triangleq \sum_{i=1}^m |x_i|$. Let the simplex Δ_m be the set of non-negative vectors with norm 1:

$$\Delta_m \triangleq \{\vec{x} \in \mathbb{R}^m \mid x_i \geq 0, \|\vec{x}\|_1 = 1\}$$

Elements of Δ_m can be viewed as probability distributions over the classes $\{1, \dots, m\}$; this is how we will encode the distribution of species in the population.

In the RSM model, the population evolution is governed by two vectors: the true class frequencies, and the current empirical frequencies. In each timesteps, we apply a function $\text{step} : \Delta_m \rightarrow \Delta_m$ to the empirical frequencies to get the updated true frequencies; we will assume that the step function is contractive, i.e., it is L -Lipschitz

$$\|\text{step}(\vec{x}) - \text{step}(\vec{y})\|_1 \leq L \cdot \|\vec{x} - \vec{y}\|_1$$

for $L < 1$. Then, we draw N samples from the distribution given by the true frequency and update the empirical frequencies. We can model the evolutionary process as a simple probabilistic program $\text{popdyn}(T)$ which repeats T iterations of the evolutionary step.

```

 $\vec{x} \leftarrow x_0; t \leftarrow 0;$ 
while  $t < T$  do
   $\vec{p} \leftarrow \text{step}(\vec{x});$ 
   $\vec{x} \leftarrow \vec{0}; j \leftarrow 0;$ 
  while  $j < N$  do
     $\vec{z} \stackrel{s}{\leftarrow} \mathbf{Mult}(\vec{p});$ 
     $\vec{x} \leftarrow \vec{x} + (1/N) \cdot \vec{z};$ 
     $j \leftarrow j + 1;$ 
   $t \leftarrow t + 1$ 

```

The vector \vec{x} stores the current empirical frequencies (the distribution of each class in our current population), while the vector \vec{p} represents the true frequencies for the current step.

The sampling instruction is new. We write $\mathbf{Mult}(\vec{p})$ for the *multinomial distribution* with parameters \vec{p} ; this distribution can be thought of as generalizing a Bernoulli (biased coin toss) distribution to m outcomes, where each outcome has some probability p_i such that $\sum_i p_i = 1$. We represent samples from the multinomial distribution as binary vectors in Δ_m : with probability p_i , the sampled vector has the i th entry set to 1 and all other entries 0.

To analyze the sampling instruction, we introduce the rule [MULT-MAX] in Fig. 5. This rule encodes the *maximal coupling*—a standard coupling construction that minimizes the probability of returning different samples—of two multinomial distributions; in the appendix, we show that this rule is sound. The post-condition $\vec{x}_a, \vec{x}_b \in \{0, 1\}^m$ states that the samples are always binary vectors

of length m , while the distances indicate that the expected distance between the sampled vectors $\|\vec{x}_a - \vec{x}_b\|_1$ are at most the distance between the parameters $\|\vec{p}_a - \vec{p}_b\|_1$.

Given two possibly different initial frequencies $(x_0)_a, (x_0)_b \in \Delta_m$, we want to show that the resulting distributions on empirical frequencies from $\text{popdyn}(T)$ converge as T increases. We will construct an expectation coupling where the expected distance between the empirical distributions x_a and x_b decays exponentially in the number of steps T ; by Proposition 3.3, this implies that the TV distance between the distributions of x_a and x_b decreases exponentially quickly. Formally, we prove the following judgement:

$$\vdash \{\Phi; \|(\vec{x}_0)_a - (\vec{x}_0)_b\|_1\} \text{popdyn}(T) \sim_{\bullet L T} \text{popdyn}(T) \{\Phi; \|\vec{x}_a - \vec{x}_b\|_1\} \quad (10)$$

where Φ is short for the relational assertion

$$\Phi \triangleq \|\vec{x}_a - \vec{x}_b\|_1 < 1/N \rightarrow \vec{x}_a = \vec{x}_b.$$

Φ is an invariant throughout because every entry of \vec{x}_a and \vec{x}_b is an integer multiple of $1/N$.

To prove the inner judgment, let s_{out} and s_{in} be the outer and inner loops, and let w_{out} and w_{in} be the respective loop bodies. We can handle the verification in two steps. In the inner loop, we want

$$\vdash \{\Phi; \|\vec{p}_a - \vec{p}_b\|_1\} s_{in} \sim_{id} s_{in} \{\Phi; \|\vec{x}_a - \vec{x}_b\|_1\} \quad (11)$$

hiding invariants asserting j, t are equal in both runs. By the loop rule [WHILE], it suffices to prove:

$$\vdash \{e_a = k \wedge \Phi; \mathfrak{d}_k\} w_{in} \sim_{id} w_{in} \{e_a = k - 1 \wedge \Phi; \mathfrak{d}_{k-1}\} \quad (12)$$

for each $0 < k \leq N$, where $\mathfrak{d}_k \triangleq \|x_a - x_b\|_1 + (k/N) \cdot \|p_a - p_b\|_1$ and the decreasing variant is $e \triangleq N - j$. Let the sampling command be w'_{in} , and the remainder of the loop body be w''_{in} . By applying the multinomial rule [MULT-MAX] and using the rule of consequence to scale the distances by $1/N$, we have

$$\vdash \{\Phi; (1/N) \cdot \|\vec{p}_a - \vec{p}_b\|_1\} w'_{in} \sim_{id} w'_{in} \{\Phi; (1/N) \cdot \|\vec{z}_a - \vec{z}_b\|_1\}.$$

Since the sampling command does not modify the vectors \vec{x}, \vec{p} , we can add the distance \mathfrak{d}_{k-1} to the pre-condition and the post-condition by the frame rule [FRAME-D] (noting that the distance transformer id is a linear function). Since $\mathfrak{d}_k = \mathfrak{d}_{k-1} + (1/N) \cdot \|\vec{p}_a - \vec{p}_b\|_1$ by definition

$$\vdash \{\Phi; \mathfrak{d}_k\} w'_{in} \sim_{id} w'_{in} \{\Phi; \mathfrak{d}_{k-1} + (1/N) \cdot \|\vec{z}_a - \vec{z}_b\|_1\}. \quad (13)$$

For the deterministic commands w''_{in} , the assignment rule [ASSG] gives

$$\vdash \{\Phi; \mathfrak{d}_{k-1}[\vec{x} := (\vec{x} + (1/N) \cdot \vec{z})]\} w''_{in} \sim_{id} w''_{in} \{\Phi; \mathfrak{d}_{k-1}\}, \quad (14)$$

where the substitution is made on the respective sides. Applying the rule of consequence with the triangle inequality in the pre-condition, we can combine this judgment (Eq. (14)) with the judgment for w'_{in} (Eq. (13)) to verify the inner loop body (Eq. (12)). The rule [WHILE] gives the desired judgment for the inner loop s_{in} (Eq. (11)).

Turning to the outer loop, we first prove a judgment for the loop bodies:

$$\vdash \{\Phi; \|\vec{x}_a - \vec{x}_b\|_1\} w_{out} \sim_{\bullet L} w_{out} \{\Phi; \|\vec{x}_a - \vec{x}_b\|_1\}.$$

By the sequence and assignment rules and the judgment for the inner loop (Eq. (11)), we have

$$\vdash \{\Phi; \|\text{step}(\vec{x}_a) - \text{step}(\vec{x}_b)\|_1\} w_{out} \sim_{id} w_{out} \{\Phi; \|\vec{x}_a - \vec{x}_b\|_1\}.$$

Applying the fact that step is L -Lipschitz, the rule of consequence gives

$$\vdash \{\Phi; \|\vec{x}_a - \vec{x}_b\|_1\} w_{out} \sim_{\bullet L} w_{out} \{\Phi; \|\vec{x}_a - \vec{x}_b\|_1\}$$

for the outer loop body. We can then apply the rule [WHILE] to conclude the desired judgment for the whole program (Eq. (10)).

This judgment shows that the distributions of \vec{x} in the two runs converge exponentially quickly. More precisely, let ν_a, ν_b be the distributions of \vec{x} after T steps of popdyn, starting from any two initial frequencies $(x_0)_a, (x_0)_b \in \Delta_m$. Eq. (10) implies that there is an expectation coupling

$$\nu_a \langle \nu \rangle_{\|\cdot\|_1 \leq \delta}^\Phi \nu_b,$$

where $\delta = L^T \cdot \|(x_0)_a - (x_0)_b\|_1$. All pairs of vectors (v_1, v_2) in the support of ν with $v_1 \neq v_2$ are at distance at least $1/N$, so Proposition 3.3 implies the bound

$$\text{TV}(\nu_a, \nu_b) \leq N \cdot L^T.$$

Since $L < 1$, the distributions converge exponentially fast as T increases.

7 PATH COUPLING AND GRAPH COLORING

Path coupling [13] is a powerful method for proving rapid mixing of Markov chains. We review the central claim of path coupling from the perspective of expected sensitivity. Then, we define an extension of our program logic that incorporates the central idea of path coupling. Finally, we apply of our logic to verify a classic example using the path coupling method.

7.1 Path coupling and local expected sensitivity

So far, we have assumed very little structure on our distances; essentially they may be arbitrary non-negative functions from $A \times A$ to the real numbers. Commonly used distances tend to have more structure. For integer-valued distances, we can define a weakening of sensitivity that only considers pairs of inputs at distance 1, rather than arbitrary pairs of inputs. We call the resulting property *local* expected sensitivity.

Definition 7.1. Let \mathfrak{d}_A be an integer-valued distance over A and \mathfrak{d}_B be a distance over B . Moreover, let $f \in \mathcal{L}$. We say that a probabilistic function $g : A \rightarrow \mathbb{D}(B)$ is *locally expected f -sensitive* (with respect to \mathfrak{d}_A and \mathfrak{d}_B) iff for every $x_1, x_2 \in A$ such that $\mathfrak{d}_A(x_1, x_2) = 1$, we have:

$$\mathbb{E}_{(y_1, y_2) \sim \mu}[\mathfrak{d}_B(y_1, y_2)] \leq f(1)$$

for some coupling $g(x_1) \langle \mu \rangle g(x_2)$.

In general, local expected f -sensitivity is weaker than expected f -sensitivity. However, both notions coincide under some mild conditions on the distances. We introduce a pair of conditions:

$$\begin{cases} \forall x, y. \mathfrak{d}(x, y) = 0 \implies x = y \\ \forall x, y. \mathfrak{d}(x, y) = n + 1 \implies \exists z. \mathfrak{d}(x, z) = 1 \wedge \mathfrak{d}(z, y) = n \end{cases} \quad (\text{P})$$

$$\begin{cases} \forall x. \mathfrak{d}(x, x) = 0 \\ \forall x, y, z. \mathfrak{d}(x, z) \leq \mathfrak{d}(x, y) + \mathfrak{d}(y, z) \end{cases} \quad (\text{H})$$

In condition (P), \mathfrak{d} is an integer-valued distance. The first condition is standard for metrics. The second condition is more interesting: if two points are at distance 2 or greater, we can find a strictly intermediate point. We will soon see an important class of distances—*path metrics*—that satisfy these conditions (Definition 7.3). Condition (H) is more standard: the distance \mathfrak{d} should assign distance 0 to two equal points, and satisfy the triangle inequality. Every metric satisfies these properties; in general, such a distance is called a *hemimetric*.

When the pre-distance satisfies (P) and the post-distance satisfies (H), local expected sensitivity is equivalent to expected sensitivity for linear distance transformers.

PROPOSITION 7.2. *Let \mathfrak{d}_A be an integer-valued distance over A satisfying (P), and \mathfrak{d}_B be distance over B satisfying (H). Let $f \in \mathcal{L}$ and $g : A \rightarrow \mathbb{D}(B)$. Then g is locally expected f -sensitive iff it is expected f -sensitive (both with respect to \mathfrak{d}_A and \mathfrak{d}_B).*

PROOF. The reverse direction is immediate. The forward implication is proved by induction on $\mathfrak{d}_A(x_1, x_2)$. For the base case, where $\mathfrak{d}_A(x_1, x_2) = 0$, we have $x_1 = x_2$ and hence $g(x_1) = g(x_2)$. Let μ be the identity coupling for $g(x_1)$ and $g(x_2)$. We have $\mathbb{E}_\mu[\mathfrak{d}_B] = \sum_{y \in B} \mathfrak{d}_B(y, y) = 0$, since $\mathfrak{d}_B(y, y) = 0$ for every y . For the inductive step, assume that $\mathfrak{d}_A(x_1, x_2) = n + 1$. Then there exists x_0 such that $\mathfrak{d}_A(x_1, x_0) = 1$ and $\mathfrak{d}_A(x_0, x_2) = n$. Therefore, there exists expectation couplings μ_0 and μ_n satisfying the distance conditions

$$\mathbb{E}_{\mu_0}[\mathfrak{d}_B] \leq f(\mathfrak{d}_B(x_1, x_0)) \quad \text{and} \quad \mathbb{E}_{\mu_n}[\mathfrak{d}_B] \leq f(\mathfrak{d}_B(x_0, x_2)).$$

Define μ as

$$\mu(x, y) \triangleq \sum_{z \in A} \frac{\mu_0(x, z) \cdot \mu_n(z, y)}{g(x_0)(z)}.$$

We treat terms with zero in the denominator as 0; note that since μ_0 and μ_n satisfy the marginal conditions, we have $\pi_2(\mu_0) = \pi_1(\mu_n) = g(x_0)$, so $g(x_0)(z) = 0$ implies that $\mu_0(x, z) = \mu_n(z, y) = 0$, so the numerator is also zero in these cases.

Now, the marginal conditions $\pi_1(\mu) = g(x_1)$ and $\pi_2(\mu) = g(x_2)$ follow from the marginal conditions for μ_0 and μ_n . The distance condition $\mathbb{E}_\mu[\mathfrak{d}_B] \leq f(\mathfrak{d}_A(x_1, x_2))$ is more involved. Indeed,

$$\begin{aligned} \mathbb{E}_\mu[\mathfrak{d}_B] &= \sum_{x, y} \mu(x, y) \mathfrak{d}_B(x, y) \\ &= \sum_{x, y} \sum_z \left(\frac{\mu_0(x, z) \mu_n(z, y)}{g(x_0)(z)} \right) \mathfrak{d}_B(x, y) \\ &\leq \sum_{x, y, z} \left(\frac{\mu_0(x, z) \mu_n(z, y)}{g(x_0)(z)} \right) \mathfrak{d}_B(x, z) + \sum_{x, y, z} \left(\frac{\mu_0(x, z) \mu_n(z, y)}{g(x_0)(z)} \right) \mathfrak{d}_B(z, y) \quad (\text{triangle ineq.}) \\ &= \sum_{y, z} \left(\sum_x \frac{\mu_0(x, z)}{g(x_0)(z)} \right) \mu_n(z, y) \mathfrak{d}_B(z, y) + \sum_{x, z} \left(\sum_y \frac{\mu_n(z, y)}{g(x_0)(z)} \right) \mu_0(x, z) \mathfrak{d}_B(x, z) \\ &= \sum_{x, z} \mu_0(x, z) \mathfrak{d}_B(x, z) + \sum_{y, z} \mu_n(z, y) \mathfrak{d}_B(z, y) \quad (\text{marginals}) \\ &= \mathbb{E}_{\mu_0}[\mathfrak{d}_B] + \mathbb{E}_{\mu_n}[\mathfrak{d}_B] \\ &\leq f(\mathfrak{d}_A(x_1, x_0)) + f(\mathfrak{d}_A(x_0, x_2)) \quad (\text{distances}) \\ &= f(\mathfrak{d}_A(x_1, x_0) + \mathfrak{d}_A(x_0, x_2)) \quad (f \text{ linear}) \\ &= f(\mathfrak{d}_A(x_1, x_2)). \end{aligned}$$

Thus, we have an expectation coupling $g(x_1) \langle \mu \rangle_{\mathfrak{d}_B \leq \delta} g(x_2)$ for $\delta = f(\mathfrak{d}_A(x_1, x_2))$. This completes the inductive step, so g is expected f -sensitive. \square

One important application of our result is for path metrics.

Definition 7.3 (Path metric). Let Φ be a binary relation over A , and let Φ^* denote its reflexive-transitive closure and Φ^n denote its n -fold composition. Assume that for every $a, a' \in A$, we have $(a, a') \in \Phi^*$. The *path metric* pd_Φ of Φ is the distance defined by the clause

$$\text{pd}_\Phi(a, a') = \min_n \{(a, a') \in \Phi^n\}$$

Note that the set is non-empty by assumption, and hence the minimum exists.

$$[\text{TRANS}] \frac{f \in \mathcal{L} \quad \mathfrak{d}' \text{ satisfies (H)} \quad \models \Psi^* \implies \Psi \quad \vdash \{\Phi \wedge \Phi'; -\} s \sim_{f(1)} s \{\Psi; \mathfrak{d}'\}}{\vdash \{\Phi; \text{pd}_{\Phi \wedge \Phi'}\} s \sim_f s \{\Psi; \mathfrak{d}'\}}$$

Fig. 6. Rule for path coupling

Path metrics evidently satisfy (P). Since they are also metrics, they also satisfy (H). The fundamental theorem of path coupling is then stated—in our terminology—as follows.

COROLLARY 7.4 (BUBLEY AND DYER [13]). *Let $\mathfrak{d} = \text{pd}_{\Phi}$ for some a binary relation Φ over A . Let $g : A \rightarrow \mathbb{D}(A)$ be a locally expected f -sensitive function, where $f \in \mathcal{L}$. Then for every $T \in \mathbb{N}$, the T -fold (monadic) composition g^T of g is expected f^T -sensitive, i.e. for every $x_1, x_2 \in A$, there exists a coupling μ of $g^T(x_1)$ and $g^T(x_2)$ such that*

$$\mathbb{E}_{\mu}[\mathfrak{d}] \leq f^T(\mathfrak{d}(x_1, x_2))$$

PROOF. The proof follows from the equivalence between local expected sensitivity and sensitivity, and the composition theorem of expected sensitive functions. \square

7.2 Program logic

One could express an analogue of local expected f -sensitivity for expectation couplings. For convenience, we instead formulate a proof rule inspired from local expected sensitivity, in Figure 6. Let us first consider the conclusion of the rule. Intuitively, Φ plays the role of a standard precondition, while Φ' defines the path distance—the pre-distance is a path metric for $\Phi \wedge \Phi'$. The first and second premises of the rule also require that the post-distance is a hemimetric, i.e. satisfies the condition (H), and that the distance transformer is a linear function f ; these two requirements are inherited from Proposition 7.2. The third premise requires that the post-condition Ψ is reflexive and transitive. Then, the main premise of the rule considers a constant distance transformer that always returns $f(1)$ —for this reason, we do not need to specify a pre-distance for the premise. The precondition of this premise states that the two initial memories are related by $\Phi \wedge \Phi'$, corresponding to the case that the two memories are adjacent (that is, at path distance 1). Since the post-condition Ψ is transitive, it is preserved.

THEOREM 7.5 (SOUNDNESS). *The rule [TRANS] is sound, i.e. for every instance of the rule concluding $\{\Phi; \mathfrak{d}\} s_1 \sim_f s_2 \{\Psi; \mathfrak{d}'\}$ and initial memories m_1 and m_2 such that $(m_1, m_2) \models \Phi$, there exists μ such that*

$$\llbracket s_1 \rrbracket_{m_1} \langle \mu \rangle_{\mathfrak{d}' \leq f(\mathfrak{d}(m_1, m_2))}^{\Psi} \llbracket s_2 \rrbracket_{m_2}.$$

7.3 Example: Glauber dynamics

The *Glauber dynamics* is a randomized algorithm for approximating uniform samples from the valid colorings of a finite graph. This algorithm is a prime example of algorithm where rapid mixing can be established using the path coupling method [13].

Before detailing this example, we recall some basic definitions and notations. Consider a graph G with a finite set of vertices V and a symmetric relation $E \subseteq V \times V$ representing the edges, and let C be a finite set of colors. A *coloring* of G is a map $w : V \rightarrow C$; a coloring is *valid* if neighboring vertices receive different colors: if $(a, b) \in E$, then $w(a) \neq w(b)$. We write $w(V')$ for the set of colors at vertices V' .

For a graph G and a fixed set of colors C , there may be multiple (or perhaps zero) valid colorings. Jerrum [23] proposed a simple Markov chain, called the Glauber dynamics, for sampling a uniformly

random coloring. Beginning at any valid coloring w , it draws a uniform vertex v and a uniform color c , and then change the color of v to c in w if this gives a valid coloring. The Glauber dynamics repeats this process for some finite number of steps T and returns the final coloring. We can model this process with the following program $\text{glauber}(T)$:

```

i ← 0;
while i < T do
  v ←$ V;
  c ←$ C;
  if  $\mathcal{V}_G(w, (v, c))$  then  $w \leftarrow w[v \mapsto c]$ ;
  i ← i + 1;
return w

```

The guard $\mathcal{V}_G(w, (v, c))$ is true when the vertex v in coloring w can be colored c . Jerrum [23] proved that the distribution on outputs for this process converges rapidly to the uniform distribution on valid colorings of G as we take more and more steps. While the original proof was quite technical, Buble and Dyer [13] gave a much simpler proof of the convergence by applying their *path coupling* technique.

Roughly, suppose that for every two colorings that differ in exactly *one* vertex coloring, we can couple the distributions obtained by executing one step of the transition function of the Markov process (i.e. the loop body of the program above) such that the expected distance (measuring in how many vertices the colorings differ) is at most $\beta < 1$. Then, the path coupling machinery gives a coupling of the processes started at two colorings at any distance, and concludes that after T steps the expected distance between two executions started with colorings at distance k is upper bounded by $\beta^T \cdot k$.

In ExSEL, this final property corresponds to the following judgment:

$$\vdash \{\Phi; \text{pd}_{\text{Adj}}\} \text{glauber}(T) \sim_{\bullet, \beta^T} \text{glauber}(T) \{\top; \text{pd}_{\text{Adj}}\}$$

In the judgment above, Adj holds on two states iff the colorings (stored in the variable w) differ in the color of a single vertex, and $\text{pd}' \triangleq \text{pd}_{\text{Adj}}$ counts the number of vertices with $w_a(v) \neq w_b(v)$. In addition, Φ captures some properties of the graph; in particular, Φ states that Δ is the maximal degree of vertices in G , i.e., each vertex in G has at most Δ neighbors. Finally, β is a constant determined by the graph; in certain parameter ranges, β is strictly less than 1 and the Markov chain converges quickly from any initial state.

Working backwards from the conclusion, the outline of the proof is as follows. We first apply the [WHILE] rule, then the [TRANS] rule on the loop bodies. We apply the rule with Φ , $\Psi \triangleq i_a = i_b$ and $\Phi' \triangleq \text{Adj}$, and $f \triangleq \bullet\beta$. We must prove the premise of the rule, i.e.

$$\vdash \{\Phi \wedge \text{Adj}; -\} s \sim_{\beta} s \{\Phi; \text{pd}_{\text{Adj}}\},$$

where s denotes the loop body. In this case, we apply the [SEQCASE] rule with the first judgment for s_{samp} , consisting of the two random samplings in the loop body, and the second judgment for the deterministic statement s_{rest} , consisting of the conditional statement and the updates. For the first judgment, let v_δ be the vertex that is colored differently (a and b respectively) in the two input states. We will first couple the vertex samplings with the identity coupling so that $v_a = v_b$, using the rule [RAND] with $h = \text{id}$. This gives:

$$\vdash \{\Phi \wedge \text{Adj}; -\} v \stackrel{\$}{\leftarrow} V \sim_{\beta} v \stackrel{\$}{\leftarrow} V \{\Phi \wedge v_a = v_b; \text{pd}_{\text{Adj}}\}.$$

Next, we can perform a case analysis on v_a using the rule [CASE]. If v_a is not a neighbor of v_δ , then we couple samplings so that $c_a = c_b$ with [RAND] with $h = \text{id}$. Otherwise, we couple $c_a = \pi^{ab}(c_b)$,

where π^{ab} swaps a and b and leaves all other colors unchanged. Combined:

$$\vdash \{\Phi \wedge \text{Adj}; -\} s_{\text{samp}} \sim_{\beta} s_{\text{samp}} \{\Theta; \text{pd}_{\text{Adj}}\},$$

where s_{samp} are the two sampling commands, and

$$\Theta \triangleq \Phi \wedge v_{\triangleleft} = v_{\triangleright} \wedge \begin{cases} v_{\triangleleft} = v_{\delta} \implies c_{\triangleleft} = \pi^{ab}(c_{\triangleright}) \\ v_{\triangleleft} \neq v_{\delta} \implies c_{\triangleleft} = c_{\triangleright}. \end{cases}$$

Now, we combine the sampling commands s_{samp} with the remaining commands s_{rest} using the rule [SEQCASE]. We distinguish three mutually exclusive cases, depending on how the distance changes under the coupling. Let q_b, q_g, q_n be the probability of the three cases.

- In the *bad case*, the distance grows to 2. We use the guard $e_b \triangleq v \in \mathcal{N}_G(v_{\delta}) \wedge c = b \wedge b \notin w(\mathcal{N}_G(v_{\delta}))$, and the assignment and consequence rules gives:

$$\vdash \{\Theta \wedge e_{b_{\triangleleft}}; \text{pd}_{\text{Adj}}\} s_{\text{rest}} \sim_{\bullet 2} s_{\text{rest}} \{i_{\triangleleft} = i_{\triangleright}; -\}.$$

Note that $q_b \leq \Delta/|V||C|$ since for e_b to hold, we must select a neighbor of v_{δ} and the color b in the first side.

- In the *good case*, the distance shrinks to zero. We use the guard $e_g \triangleq v = v_{\delta} \wedge c \notin w(\mathcal{N}_G(v))$. By applying the assignment and consequence rules, we can prove:

$$\vdash \{\Theta \wedge e_{g_{\triangleleft}}; \text{pd}_{\text{Adj}}\} s_{\text{rest}} \sim_{\bullet 0} s_{\text{rest}} \{i_{\triangleleft} = i_{\triangleright}; \text{pd}_{\text{Adj}}\}.$$

We will use a *lower* bound on the probability of this case: since we must choose the differing vertex and a color different from its neighbors, and there are at most Δ neighbors, $q_g \geq (|C| - \Delta)/|C||V|$.

- In the *neutral case*, the distance stays unchanged. We use the guard $e_n \triangleq v \in \mathcal{N}_G(v_{\delta}) \cup v_{\delta} \rightarrow c \in w(\mathcal{N}_G(v))$, and the assignment rule gives:

$$\vdash \{\Theta \wedge e_{n_{\triangleleft}}; \text{pd}_{\text{Adj}}\} s_{\text{rest}} \sim_{\text{id}} s_{\text{rest}} \{i_{\triangleleft} = i_{\triangleright}; \text{pd}_{\text{Adj}}\}.$$

To put everything together, we need to bound the average change in distance. Since the cases are mutually exclusive and at least one case holds, we know $q_n = 1 - q_b - q_g$. Combining the three cases, we need to bound the function $x \mapsto (q_n + 2 \cdot q_b) \cdot x = (1 - q_g + q_b) \cdot x$. By the upper bound on q_b and the lower bound on q_g , we can conclude

$$\vdash \{\Phi \wedge \text{Adj}; -\} s_{\text{samp}}; s_{\text{rest}} \sim_{\beta} s_{\text{samp}}; s_{\text{rest}} \{\Phi; \text{pd}_{\text{Adj}}\},$$

where

$$\beta \triangleq 1 - \frac{1}{|V|} + \frac{2\Delta}{|C||V|}.$$

When the number of colors $|C|$ is strictly larger than 2Δ , the constant β is strictly less than 1 and the Glauber dynamics is rapidly mixing.

8 PROTOTYPE IMPLEMENTATION

We have developed a prototype implementation of our program logic on top of EasyCrypt, a general-purpose proof assistant for reasoning about probabilistic programs, and formalized stability of the convex version of Stochastic Gradient Method and convergence of population dynamics.

- For some rules, we implement stronger versions that are required for formalization of the examples. For instance, our implementation of the [CONSEQ] rule supports scaling of distances.
- The ambient higher-order logic of EasyCrypt is used both for specifying distributions and for reasoning about their properties. Likewise, the logic is used for defining distances, Lipschitz continuity, and affine functions, and for proving their basic properties.

- We axiomatize the gradient operator and postulate its main properties. Defining gradients from first principles and proving their properties is technically possible, but beyond the scope of the paper. Similarly, we axiomatize norms and state relevant properties as axioms. A small collection of standard facts, including the Cauchy-Schwartz inequality, are assumed.

The formalization of the examples is reasonably straightforward. The formalization of stability for the Stochastic Gradient Method is 338 lines; about one third is devoted to proving mathematical facts. The formalization of convergence of the population dynamics is 144 lines.

We have not implemented the [TRANS] rule, which is required for path coupling, and we have not interfaced the prototype with the rich set of program transformations supported by EasyCrypt, e.g. code motion, loop unrolling, loop range splitting, which are required for the non-convex version of Stochastic Gradient Method. Implementing these features should not pose any difficulty, and is left for future work.

9 RELATED WORK

There is a long tradition of using non-expansive (α -sensitive functions, with $\alpha < 1$) maps over metric spaces for defining the denotational semantics of deterministic and probabilistic programs; e.g., [3, 15, 16, 25, 35]. It is also common to interpret programs as functions between ultrametric spaces, a special class of metric spaces where max is used instead of addition in the triangular inequality; e.g. [1, 2, 27, 29].

Lipschitz continuity has also been considered extensively in the setting of program verification: Chaudhuri, Gulwani, and Lublinerman [14] develop a SMT-based analysis for proving programs robust, in the setting of a core imperative language; Reed and Pierce [32] develop a linear type system for proving sensitivity and differential privacy in a higher-order language.

There is also a long tradition of verifying expectation properties of probabilistic programs; seminal works include PPDL [26] and pGCL [30]. Recently, Kaminski, Katoen, Matheja, and Olmedo [24] have developed a method based on similar ideas to reason about expected runtime of probabilistic programs. This line of work is focused on non-relational properties, such as proving upper bounds on errors, whereas expected sensitivity is intrinsically relational.

Finally, there has been a significant amount of work on the relational verification of probabilistic programs. Barthe and collaborators develop relational program logics for reasoning about the provable security of cryptographic constructions [9] and differential privacy of algorithms [10]. ExSEL subsumes the relational program logic considered in Barthe et al. [9]; indeed, one can prove that the two-sided rules of pRHL are essentially equivalent to the fragment of ExSEL where the pre-distance and post-distance are the null functions. In contrast, the relational program logic APRHL considered by Barthe et al. [10] is not comparable with ExSEL. APRHL uses a notion of approximate coupling tailored to applications to differential privacy, while expectation couplings are more broadly applicable and designed for average versions of quantitative relational properties. Generally speaking, APRHL considers *pointwise* notions of distance between distributions, without assuming a distance on the sample space. In contrast, ExSEL works with distances on the underlying space, proving fundamentally different properties with a substantially different style of reasoning.

There have been some works that apply formal verification to specific examples of relational expectation properties. For instance, the standard target property in *masking* implementations in cryptography is a variant of probabilistic non-interference, known as *probing security*. Recent work introduces quantitative masking strength [18], a quantitative generalization that measures average leakage of the programs. Similarly, the bounded moment model [4] is a qualitative, expectation-based non-interference property for capturing security of parallel implementations against differential

power analyses. Current verification technology for the bounded moment model is based on a meta-theorem which reduces security in the bounded moment model to probing security, and a custom program logic inspired from pRHL for proving probing security. It would be interesting to develop a custom program logic based on ExSeL to verify a broader class of parallel implementations.

For another example, there are formal verification techniques for verifying *incentive properties* in mechanism design. These properties are relational, and when the underlying mechanism is randomized (or when the inputs are randomized), incentive properties describe the expected payoff of an agent in two executions. Barthe, Gaboardi, Arias, Hsu, Roth, and Strub [6, 7] show how to use a relational type system to verify these properties. While their approach is also based on couplings, they reason about expectations only at the top level, as a consequence of a particular coupling. In particular, it is not possible to compose reasoning about expected values like in ExSeL .

Lastly, there have been prior efforts to verify rapid mixing for Markov chains via the path coupling method. In particular, Barthe, Grégoire, Hsu, and Strub [8] use $\times\text{pRHL}$, a proof-relevant variant of pRHL , to extract a product program that can then be verified (using an external system) in order to prove rapid mixing. This approach can be used for proving formally rapid convergence of Markov chains, and in particular it has been applied to prove convergence of the Glauber dynamics. Our system improves upon this work in two respects. First, we can internalize the path coupling principle as a rule in our logic. Second, the probabilistic reasoning in our system is confined to a handful of side-conditions (e.g., in the $[\text{SEQCASE}]$ rule).

10 CONCLUSION

We have introduced the notion of expected f -sensitivity for reasoning about algorithmic stability and convergence of probabilistic processes, and proved some of its basic properties. Moreover, we have introduced expectation couplings for reasoning about a broader class of relational expectation properties, and proposed a relational program logic for proving such properties. We have illustrated the expressiveness of the logic with recent and challenging examples from machine learning, evolutionary biology, and statistical physics.

There are several directions for future work. On the foundational side, it would be interesting to develop semantic foundations for advanced fixed point-theorems and convergence criteria that arise in probabilistic analysis. There is a wealth of results to consider, for instance, see the survey by Bharucha-Reid et al. [11]. On the practical side, it would be interesting to formalize more advanced examples featuring relational and probabilistic analysis. For instance, it would be extremely interesting to formally verify a recent result by Shamir [33], which proves convergence of a practical variant of the Stochastic Gradient Method, or algorithms for regret-minimization in learning theory and algorithmic game theory. Another goal would be to verify more general results about population dynamics, including the general case from Panageas et al. [31].

REFERENCES

- [1] Martín Abadi and Gordon D. Plotkin. 1990. A per model of polymorphism and recursive types. In *IEEE Symposium on Logic in Computer Science (LICS), Philadelphia, Pennsylvania*. 355–365. <https://doi.org/10.1109/LICS.1990.113761>
- [2] Pierre America and Jan J. M. M. Rutten. 1987. Solving reflexive domain equations in a category of complete metric spaces. In *Conference on the Mathematical Foundations of Programming Semantics (MFPS), New Orleans, Louisiana (Lecture Notes in Computer Science)*, Michael G. Main, Austin Melton, Michael W. Mislove, and David A. Schmidt (Eds.), Vol. 298. Springer-Verlag, 254–288. https://doi.org/10.1007/3-540-19020-1_13
- [3] André Arnold and Maurice Nivat. 1980. Metric interpretations of infinite trees and semantics of non deterministic recursive programs. *Theoretical Computer Science* 11 (1980), 181–205. [https://doi.org/10.1016/0304-3975\(80\)90045-6](https://doi.org/10.1016/0304-3975(80)90045-6)
- [4] Gilles Barthe, François Dupressoir, Sebastian Faust, Benjamin Grégoire, François-Xavier Standaert, and Pierre-Yves Strub. 2016. Parallel implementations of masking schemes and the bounded moment leakage model. *IACR Cryptology ePrint Archive* 2016 (2016), 912. <http://eprint.iacr.org/2016/912>

- [5] Gilles Barthe, François Dupressoir, Benjamin Grégoire, César Kunz, Benedikt Schmidt, and Pierre-Yves Strub. 2013. Easycrypt: A tutorial. In *Foundations of Security Analysis and Design VII (FOSAD) (Lecture Notes in Computer Science)*, Vol. 8604. Springer-Verlag, 146–166. Tutorial Lectures.
- [6] Gilles Barthe, Marco Gaboardi, Emilio Jesús Gallego Arias, Justin Hsu, Aaron Roth, and Pierre-Yves Strub. 2015. Higher-order approximate relational refinement types for mechanism design and differential privacy. In *ACM SIGPLAN–SIGACT Symposium on Principles of Programming Languages (POPL)*, Mumbai, India. 55–68.
- [7] Gilles Barthe, Marco Gaboardi, Emilio Jesús Gallego Arias, Justin Hsu, Aaron Roth, and Pierre-Yves Strub. 2016. Computer-aided verification in mechanism design. In *Conference on Web and Internet Economics (WINE)*, Montréal, Québec. <http://arxiv.org/abs/1502.04052>
- [8] Gilles Barthe, Benjamin Grégoire, Justin Hsu, and Pierre-Yves Strub. 2017. Coupling proofs are probabilistic product programs. In *ACM SIGPLAN–SIGACT Symposium on Principles of Programming Languages (POPL)*, Paris, France. <http://arxiv.org/abs/1607.03455>
- [9] Gilles Barthe, Benjamin Grégoire, and Santiago Zanella-Béguelin. 2009. Formal certification of code-based cryptographic proofs. In *ACM SIGPLAN–SIGACT Symposium on Principles of Programming Languages (POPL)*, Savannah, Georgia. New York, 90–101. <http://certicrypt.gforge.inria.fr/2013.Journal.pdf>
- [10] Gilles Barthe, Boris Köpf, Federico Olmedo, and Santiago Zanella Béguelin. 2012. Probabilistic relational reasoning for differential privacy. In *ACM SIGPLAN–SIGACT Symposium on Principles of Programming Languages (POPL)*, Philadelphia, Pennsylvania. 97–110.
- [11] AT Bharucha-Reid et al. 1976. Fixed point theorems in probabilistic analysis. *Bull. Amer. Math. Soc.* 82, 5 (1976), 641–657.
- [12] Olivier Bousquet and André Elisseeff. 2002. Stability and generalization. *Journal of Machine Learning Research* 2 (2002), 499–526. <http://www.jmlr.org/papers/v2/bousquet02a.html>
- [13] Russ Bubley and Martin Dyer. 1997. Path coupling: A technique for proving rapid mixing in Markov chains. In *IEEE Symposium on Foundations of Computer Science (FOCS)*, Miami Beach, Florida. 223–231.
- [14] Swarat Chaudhuri, Sumit Gulwani, and Roberto Lublinerman. 2010. Continuity analysis of programs. In *ACM SIGPLAN–SIGACT Symposium on Principles of Programming Languages (POPL)*, Madrid, Spain. 57–70.
- [15] Arthur Azevedo de Amorim, Marco Gaboardi, Justin Hsu, Shin-ya Katsumata, and Ikram Cherigui. 2017. A semantic account of metric preservation. In *ACM SIGPLAN–SIGACT Symposium on Principles of Programming Languages (POPL)*, Paris, France. 545–556.
- [16] J. W. de Bakker and Jeffrey I. Zucker. 1982. Denotational semantics of concurrency. In *ACM SIGACT Symposium on Theory of Computing (STOC)*, San Francisco, California, Harry R. Lewis, Barbara B. Simons, Walter A. Burkhard, and Lawrence H. Landweber (Eds.). 153–158. <https://doi.org/10.1145/800070.802188>
- [17] Narendra M Dixit, Piyush Srivastava, and Nisheeth K Vishnoi. 2012. A finite population model of molecular evolution: Theory and computation. *Journal of Computational Biology* 19, 10 (2012), 1176–1202.
- [18] Hassan Eldib, Chao Wang, Mostafa M. I. Taha, and Patrick Schaumont. 2015. Quantitative masking strength: Quantifying the power side-channel resistance of software code. *IEEE Transactions on CAD of Integrated Circuits and Systems* 34, 10 (2015), 1558–1568. <https://doi.org/10.1109/TCAD.2015.2424951>
- [19] André Elisseeff, Theodoros Evgeniou, and Massimiliano Pontil. 2005. Stability of randomized learning algorithms. *Journal of Machine Learning Research* 6 (2005), 55–79. <http://www.jmlr.org/papers/v6/elisseeff05a.html>
- [20] Moritz Hardt, Ben Recht, and Yoram Singer. 2016. Train faster, generalize better: Stability of stochastic gradient descent. In *International Conference on Machine Learning (ICML)*, New York, NY (*Journal of Machine Learning Research*), Vol. 48. JMLR.org, 1225–1234. <http://jmlr.org/proceedings/papers/v48/hardt16.html>
- [21] Daniel L. Hartl and Andrew G. Clark. 2006. *Principles of Population Genetics* (fourth ed.). Sinauer Associates.
- [22] Thomas Jansen. 2013. *Analyzing Evolutionary Algorithms: The Computer Science Perspective*. Springer-Verlag. <https://doi.org/10.1007/978-3-642-17339-4>
- [23] Mark Jerrum. 1995. A very simple algorithm for estimating the number of k -colorings of a low-degree graph. *Random Structures and Algorithms* 7, 2 (1995), 157–166. <https://doi.org/10.1002/rsa.3240070205>
- [24] Benjamin Lucien Kaminski, Joost-Pieter Katoen, Christoph Matheja, and Federico Olmedo. 2016. Weakest precondition reasoning for expected run-times of probabilistic programs. In *European Symposium on Programming (ESOP)*, Eindhoven, The Netherlands (*Lecture Notes in Computer Science*), Vol. 9632. Springer-Verlag, 364–389. https://doi.org/10.1007/978-3-662-49498-1_15
- [25] Dexter Kozen. 1979. Semantics of probabilistic programs. In *IEEE Symposium on Foundations of Computer Science (FOCS)*, San Juan, Puerto Rico. 101–114.
- [26] Dexter Kozen. 1985. A probabilistic PDL. *J. Comput. System Sci.* 30, 2 (1985), 162–178.
- [27] Neelakantan R. Krishnaswami and Nick Benton. 2011. Ultrametric semantics of reactive programs. In *IEEE Symposium on Logic in Computer Science (LICS)*, Toronto, Ontario. 257–266.
- [28] Torgny Lindvall. 2002. *Lectures on the coupling method*. Courier Corporation.

- [29] David B. MacQueen, Gordon D. Plotkin, and Ravi Sethi. 1984. An ideal model for recursive polymorphic types. In *ACM Symposium on Principles of Programming Languages (POPL), Salt Lake City, Utah*, Ken Kennedy, Mary S. Van Deusen, and Larry Landweber (Eds.), 165–174. <https://doi.org/10.1145/800017.800528>
- [30] Carroll Morgan, Annabelle McIver, and Karen Seidel. 1996. Probabilistic predicate transformers. *ACM Transactions on Programming Languages and Systems* 18, 3 (1996), 325–353.
- [31] Ioannis Panageas, Piyush Srivastava, and Nisheeth K. Vishnoi. 2016. Evolutionary dynamics in finite populations mix rapidly. In *ACM–SIAM Symposium on Discrete Algorithms (SODA), Arlington, Virginia*. 480–497. <https://doi.org/10.1137/1.9781611974331.ch36>
- [32] Jason Reed and Benjamin C Pierce. 2010. Distance makes the types grow stronger: A calculus for differential privacy. In *ACM SIGPLAN International Conference on Functional Programming (ICFP), Baltimore, Maryland*. <http://dl.acm.org/citation.cfm?id=1863568>
- [33] Ohad Shamir. 2016. Without-replacement sampling for stochastic gradient methods: Convergence results and application to distributed optimization. *CoRR* abs/1603.00570 (2016). <http://arxiv.org/abs/1603.00570>
- [34] Hermann Thorisson. 2000. *Coupling, Stationarity, and Regeneration*. Springer-Verlag.
- [35] Franck van Breugel. 1997. *Comparative Metric Semantics of Programming Languages: Nondeterminism and Recursion*. Birkhauser.
- [36] Cédric Villani. 2008. *Optimal transport: Old and new*. Springer-Verlag.
- [37] Nisheeth K. Vishnoi. 2015. The speed of evolution. In *ACM–SIAM Symposium on Discrete Algorithms (SODA), San Diego, California*. 1590–1601. <https://doi.org/10.1137/1.9781611973730.105>

A SOUNDNESS

First, we can show that the equivalence judgment $\Phi \vdash s_1 \equiv s_2$ shows that programs s_1, s_2 have equal denotation under any memory satisfying the (non-relation) pre-condition Φ .

LEMMA A.1. *If $\Phi \vdash s_1 \equiv s_2$, then for any $m \models \Phi$, $\llbracket s_1 \rrbracket_m = \llbracket s_2 \rrbracket_m$.*

PROOF. Direct induction on $\Phi \vdash s_1 \equiv s_2$, using the semantics in Fig. 1 for the base cases. \square

Next, we prove the key lemma showing composition of expectation couplings (Proposition 3.8).

PROOF OF COMPOSITION OF EXPECTATION COUPLINGS (PROPOSITION 3.8). We check each of the conditions in turn.

For the support condition, $\text{supp}(\mu) \subseteq \Phi$ by the first premise, and for every $(a, b) \in \Phi$ we have $\text{supp}(M(a, b)) \subseteq \Psi$ by the second premise, so

$$\text{supp}(\mu') = \text{supp}(\mathbb{E}_\mu[M]) \subseteq \Psi.$$

For the marginal condition, we have $\pi_1(\mu) = \mu_a$ and $\pi_2(\mu) = \mu_b$ by the first premise, and for every $(a, b) \in \Phi$ we have $\pi_1(M(a, b)) = M_a(a)$ and $\pi_2(M(a, b)) = M_b(b)$ by the second premise. We can directly calculate the marginals of μ' . For instance, for every $a' \in A$ the first marginal is

$$\begin{aligned} \pi_1(\mu')(a') &= \pi_1(\mathbb{E}_\mu[M])(a') \\ &= \sum_{b' \in B} \sum_{(a, b) \in A \times B} \mu(a, b) \cdot M(a, b)(a', b') \\ &= \sum_{b' \in B} \sum_{(a, b) \in \Phi} \mu(a, b) \cdot M(a, b)(a', b') && \text{(Support of } \mu) \\ &= \sum_{(a, b) \in \Phi} \mu(a, b) \cdot \pi_1(M(a, b))(a') \\ &= \sum_{(a, b) \in \Phi} \mu(a, b) \cdot M_a(a)(a') && \text{(Marginal of } M(a, b)) \\ &= \sum_{a \in A} M_a(a)(a') \sum_{b \in B} \mu(a, b) && \text{(Support of } \mu) \\ &= \sum_{a \in A} M_a(a)(a') \cdot \mu_a(a) && \text{(Marginal of } \mu) \\ &= \mu_a(a'). \end{aligned}$$

The second marginal $\pi_2(\mu') = \mu_b$ is similar.

Finally, we check the distance condition. By the premises, we have

$$\begin{aligned} \mathbb{E}_\mu[\mathfrak{d}] &\leq \delta \\ \mathbb{E}_{M(a, b)}[\mathfrak{d}'] &\leq f(\mathfrak{d}(a, b)) \quad \text{for every } (a, b) \in \Phi. \end{aligned}$$

Then, we can bound

$$\begin{aligned} \mathbb{E}_{\mu'}[\mathfrak{d}'] &= \sum_{(a', b') \in A \times B} \mathfrak{d}'(a', b') \cdot \mu'(a', b') \\ &= \sum_{(a', b') \in A \times B} \mathfrak{d}'(a', b') \sum_{(a, b) \in A \times B} \mu(a, b) \cdot M(a, b)(a', b') \\ &= \sum_{(a, b) \in \Phi} \mu(a, b) \mathbb{E}_{M(a, b)}[\mathfrak{d}'] && \text{(Support of } \mu) \end{aligned}$$

$$\begin{aligned}
&\leq \sum_{(a,b) \in \Phi} \mu(a,b) \cdot f(\mathfrak{d}(a,b)) && \text{(Expectation of } M(a,b)) \\
&= \mathbb{E}_\mu[f(\mathfrak{d})] && \text{(Support of } \mu) \\
&\leq f(\mathbb{E}_\mu[\mathfrak{d}]) && \text{(Linearity of expectation)} \\
&\leq f(\delta). && \text{(Monotonicity of } f, \text{ expectation of } \mu)
\end{aligned}$$

□

We now move to the soundness of the logic.

PROOF OF THE SOUNDNESS OF THE LOGIC. We prove that each rule is sound.

[CONSEQ] Let $m_1, m_2 \models \Phi''$. Hence, $m_1, m_2 \models \Phi$, and there exists η such that $\llbracket s_1 \rrbracket_{m_1} \langle \eta \rangle_{\mathfrak{d}' \leq \delta} \llbracket s_2 \rrbracket_{m_2}$. We use η for the coupling of the conclusion. We already know that $\forall i \in \{1, 2\}. \pi_i(\mu) = \llbracket s_i \rrbracket_{m_i}$ and that $\text{supp}(\mu) \subseteq \Psi \subseteq \Psi''$. Finally,

$$\begin{aligned}
\mathbb{E}_\mu[\mathfrak{d}'''] &\leq \mathbb{E}_\mu[\mathfrak{d}'] && \text{(E monotone)} \\
&\leq f(\mathfrak{d}(m_1, m_2)) && (\mu \text{ coupling}) \\
&\leq f'(\mathfrak{d}''(m_1, m_2)) && \text{(premise).}
\end{aligned}$$

[STRUCT] Immediate consequence of Lemma A.1.

[ASSG] & [ASSG-L] Immediate.

[RAND] Let $m_1, m_2 \models \forall v \in \text{supp}(g_1). \Psi[x_{1\leftarrow} v, x_{2\leftarrow} h(v)]$ and, for $i \in \{1, 2\}$, let $\mu_i \triangleq \mathbb{E}_{v \sim g_i}[\delta_{m_i[v:=x_i]}]$. Since h is a one to one mapping from $\text{supp}(g_1)$ to $\text{supp}(g_2)$ that preserves the mass, we have $|\mu_1| = |\mu_2|$ and $\mu_2 = \mathbb{E}_{v \sim g_1}[\delta_{m_2[h(v):=x_2]}]$. Let

$$\mu \triangleq \mathbb{E}_{v \sim g_1}[\delta_{(m_1[v:=x_1], m_2[h(v):=x_2])}].$$

By construction, for $i \in \{1, 2\}$, we have $\pi_i(\mu) = \mu_i$. Let $\bar{m} \in \text{supp}(\mu)$. By definition, there exists a $v \in \text{supp}(g_1)$ s.t. $\bar{m} = (m_1[v := x_1], m_2[h(v) := x_2])$. Hence, $\bar{m} \models \Psi$. Last,

$$\begin{aligned}
\mathbb{E}_{\bar{m} \sim \mu}[\mathfrak{d}'] &= \mathbb{E}_{v \sim g_1}[\mathbb{E}_{\bar{m} \sim \delta_{(m_1[v:=x_1], m_2[h(v):=x_2])}}[\mathfrak{d}']] \\
&= \mathbb{E}_{v \sim g_1}[\mathfrak{d}'(m_1[v := x_1], m_2[h(v) := x_2])] \\
&= \mathbb{E}_{v \sim g_1}[\mathfrak{d}'[(x_{1\leftarrow} v, (x_{2\leftarrow} h(v) \rightarrow h(v))]].
\end{aligned}$$

[SEQ] Let $(m_1, m_2) \models \Phi$ and, for $i \in \{1, 2\}$, let $\mu_i \triangleq \llbracket c_i \rrbracket_{m_i}$ and $\eta_i(m) \triangleq \llbracket c'_i \rrbracket_m$. From the first premise, we know that there exists an η such that $\mu_1 \langle \eta \rangle_{\mathfrak{d}' \leq \delta} \mu_2$ and $\text{supp}(\eta) \models \Xi$, where $\delta \triangleq f(\mathfrak{d}(m_1, m_2))$. Likewise, from the second premise, for $m \triangleq (m'_1, m'_2) \models \Xi$, there exists an η_m such that $\eta_1(m) \langle \eta_m \rangle_{\mathfrak{d}' \leq \delta'(m)} \eta_2(m)$ and $\text{supp}(\eta_m) \models \Psi$, where $\delta'(m) \triangleq f'(\mathfrak{d}'(m))$.

Let $\mu \triangleq \mathbb{E}_{m \sim \eta}[\eta_m \mid \Xi]$. By Proposition 3.8, we already know that $\mathbb{E}_{\mu_1}[\eta_1] \langle \mu \rangle \mathbb{E}_{\mu_2}[\eta_2]$ and that $\text{supp}(\mu) \models \Psi$. We are left to prove that $\mathbb{E}_\mu[\mathfrak{d}'''] \leq (f' \circ f)(\mathfrak{d}(m_1, m_2))$:

$$\begin{aligned}
\mathbb{E}_\mu[\mathfrak{d}'''] &= \mathbb{E}_{m \sim \eta}[\mathbb{E}_{\eta_m}[\mathfrak{d}'''] \mid \Xi] \\
&\leq \mathbb{E}_{m \sim \eta}[f'(\mathfrak{d}'(m))] && \text{(monotonicity of } \mathbb{E}) \\
&\leq f'(\mathbb{E}_\eta[\mathfrak{d}']) && \text{(Linearity of expectation)} \\
&\leq f'(f(\mathfrak{d}(m_1, m_2))). && (f' \text{ is increasing})
\end{aligned}$$

[CASE] Let $m_1, m_2 \models \Phi$. We do a case analysis on $\llbracket e_1 \rrbracket_{m_1}$ and conclude from we one of the two premises.

[COND] Immediate consequence of [CASE] and [STRUCT], using the synchronicity of both guards.

[SEQCASE] For $\bar{m} \models \Psi \wedge \exists i. e_{i_{\triangleleft}}$, we denote by $\iota(\bar{m})$ an index i s.t. $\bar{m} \models e_{i_{\triangleleft}}$, and by $\eta_{\bar{m}}$ the coupling obtained, for \bar{m} seen as an initial relational memory, from:

$$\vdash \{\Psi \wedge e_{\iota(\bar{m})_{\triangleleft}}; \mathfrak{d}'\} s'_1 \sim_{f_i} s'_2 \{\Psi'; \mathfrak{d}''\},$$

i.e. $\eta_{\bar{m}}$ is s.t. $\llbracket s'_1 \rrbracket_{\pi_1(\bar{m})} \langle \eta_{\bar{m}} \rangle_{\mathfrak{d}'}^{\Psi'} \llbracket s'_2 \rrbracket_{\pi_2(\bar{m})}$, where $\delta_{\bar{m}} \triangleq f_{\iota(\bar{m})}(\mathfrak{d}'(\bar{m}))$. Let $m_1, m_2 \models \Phi$ and μ s.t. $\llbracket s_1 \rrbracket_{m_1} \langle \mu \rangle_{\mathfrak{d}'}^{\Psi} \llbracket s_2 \rrbracket_{m_2}$, where $\delta \triangleq f_0(\mathfrak{d}(m_1, m_2))$ —such a coupling is obtained from the premise $\vdash \{\Phi; \mathfrak{d}\} s_1 \sim_{f_0} s_2 \{\Psi; \mathfrak{d}'\}$. Let $\eta \triangleq \mathbb{E}_{\bar{m} \sim \mu}[\eta_{\bar{m}}]$. The distribution η is well-defined if for any $\bar{m} \in \text{supp}(\mu)$, $\bar{m} \models \Psi \wedge \exists i. e_{i_{\triangleleft}}$. By definition of μ , we already know that $\text{supp}(\mu) \subseteq \Psi$. Moreover, from the premise $\Psi \implies \bigvee_{i \in I} e_i$, we obtain the existence of a $i \in I$ s.t. $\pi_1(\bar{m}) \models e_i$, i.e. such that $\bar{m} \models e_{i_{\triangleleft}}$. It is immediate that $\text{supp}(\eta) \subseteq \Psi'$ since for any $\bar{m} \in \text{supp}(\mu)$, by definition of $\eta_{\bar{m}}$, we know that $\eta_{\bar{m}} \subseteq \Psi'$. Now, for $i \in \{1, 2\}$, we have:

$$\begin{aligned} \pi_i(\eta) &= \pi_i(\mathbb{E}_{\bar{m} \sim \mu}[\eta_{\bar{m}}]) = \mathbb{E}_{\bar{m} \sim \mu}[\underbrace{\pi_i(\eta_{\bar{m}})}_{\llbracket s'_i \rrbracket_{\pi_i(\bar{m})}}] \\ &= \mathbb{E}_{m \sim \pi_i(\mu)}[\llbracket s'_i \rrbracket_m] = \mathbb{E}_{m \sim \llbracket s_i \rrbracket_{m_i}}[\llbracket s'_i \rrbracket_m] \\ &= m \mapsto \llbracket s_i; s'_i \rrbracket_m. \end{aligned}$$

We are left to prove the bounding property of η . For $i \in I$, we denote by \bar{p}_i the quantity $\Pr_{\bar{m} \sim \mu}[\iota(\bar{m}) = i]$. Then,

$$\bar{p}_i = \Pr_{\bar{m} \sim \mu}[\iota(\bar{m}) = i] \leq \Pr_{\bar{m} \sim \mu}[\llbracket e_i \rrbracket_{\pi_1(\bar{m})}] = \Pr_{m \sim \pi_1(\mu)}[\underbrace{\llbracket e_i \rrbracket_m \rrbracket}_{m \sim \llbracket s_1 \rrbracket_{m_1}}].$$

Denote this last quantity by p_i . By the law of total expectation:

$$\begin{aligned} \mathbb{E}_{\mu}[\mathfrak{d}'] &= \mathbb{E}_{\bar{m} \sim \mu}[\mathbb{E}_{\mu_{\bar{m}}}[\mathfrak{d}']] \\ &= \sum_{i \in I} \bar{p}_i \cdot \mathbb{E}_{\bar{m} \sim \mu}[\mathbb{E}_{\mu_{\bar{m}}}[\mathfrak{d}'] \mid \iota(\bar{m}) = i] \\ &\leq \sum_{i \in I} p_i \cdot \mathbb{E}_{\bar{m} \sim \mu}[\mathbb{E}_{\mu_{\bar{m}}}[\mathfrak{d}'] \mid \iota(\bar{m}) = i]. \end{aligned}$$

Now, for $\bar{m} \in \text{supp}(\mu)$ s.t. $\iota(\bar{m}) = i$, we have:

$$\mathbb{E}_{\mu_{\bar{m}}}[\mathfrak{d}'] \leq \delta_{\bar{m}} = f_i(\mathfrak{d}'(\bar{m})).$$

Hence,

$$\begin{aligned} \mathbb{E}_{\mu}[\mathfrak{d}'] &\leq \sum_{i \in I} p_i \cdot \mathbb{E}_{\bar{m} \sim \mu}[f_i(\mathfrak{d}'(\bar{m})) \mid \iota(\bar{m}) = i] \\ &\leq \sum_{i \in I} p_i \cdot \mathbb{E}_{\bar{m} \sim \mu}[f_i(\mathfrak{d}'(\bar{m}))] = \sum_{i \in I} p_i f_i(\mathbb{E}_{\mu}[\mathfrak{d}']) \\ &\leq \sum_{i \in I} p_i \cdot f_i(f_0(\mathfrak{d}(m_1, m_2))) = \bar{f}(\mathfrak{d}(m_1, m_2)) \end{aligned}$$

where the last step is by the premise.

[WHILE] We proceed by induction on n . For $i \in \{1, 2\}$, let $\bar{s}_i \triangleq$ while e do s_i . For $n \in \mathbb{N}$, let $\Psi_n \triangleq \Psi \wedge (i_{\triangleleft} = n)$ and $\bar{f}_n \triangleq f_1 \circ \dots \circ f_n$. If $n = 0$, under $m_1, m_2 \models \Psi$, we have $\llbracket e \rrbracket_{m_1} = \llbracket e \rrbracket_{m_2} = \perp$. Hence, for $i \in \{1, 2\}$, $\llbracket \bar{s}_i \rrbracket_{m_i} = \delta_{m_i}$ and we are in a case similar to **[SKIP]**. Otherwise, assume that the rule is valid for n . From the premises and the induction hypothesis, we have:

$$\vdash \{\Psi_{n+1} \wedge e_{i_{\triangleleft}}; \mathfrak{d}'_{n+1}\} s_1 \sim_{f_{n+1}} s_2 \{\Psi_n; \mathfrak{d}'_n\}$$

$$\vdash \{\Psi_n; \mathfrak{d}'_n\} \overline{s}_1 \sim_{\overline{f}_n} \overline{s}_2 \{\Psi_0; \mathfrak{d}'_0\}$$

Hence, by a reasoning similar to the one of [SEQ], we have $\vdash \{\Psi_{n+1}; \mathfrak{d}'_{n+1}\} s_1; \overline{s}_1 \sim_{\overline{f}_{n+1}} s_2; \overline{s}_2 \{\Psi_0; \mathfrak{d}'_0\}$. Now, under $m_1, m_2 \models \Psi$, we have, for $i \in \{1, 2\}$, $\llbracket s_i; \overline{s}_i \rrbracket_{m_i} = \llbracket \overline{s}_i \rrbracket_{m_i}$. Hence, by a reasoning similar to the one of [STRUCT], we obtain $\vdash \{\Psi_{n+1} \wedge e_{1_{\leftarrow}}; \mathfrak{d}'_{n+1}\} \overline{s}_1 \sim_{\overline{f}_{n+1}} \overline{s}_2 \{\Psi_0; \mathfrak{d}'_0\}$. Last, from $\Psi_{n+1} \iff (\Psi_{n+1} \wedge e_{1_{\leftarrow}})$, we conclude that $\vdash \{\Psi_{n+1}; \mathfrak{d}'_{n+1}\} \overline{s}_1 \sim_{\overline{f}_{n+1}} \overline{s}_2 \{\Psi_0; \mathfrak{d}'_0\}$.

[FRAME-D] Let $m_1, m_2 \models \Phi$. From the premise we know the existence of a coupling η s.t. $\llbracket s_1 \rrbracket_{m_1} \langle \eta \rangle_{\mathfrak{d}' \leq \delta}^{\Psi} \llbracket s_2 \rrbracket_{m_2}$, where $\delta \triangleq f(\mathfrak{d}(m_1, m_2))$. Now, we have

$$\mathbb{E}_{\eta}[\mathfrak{d}' + \mathfrak{d}'''] = \mathbb{E}_{\eta}[\mathfrak{d}'] + \mathbb{E}_{\eta}[\mathfrak{d}'''] \leq f(\mathfrak{d}(m_1, m_2)) + \mathbb{E}_{\eta}[\mathfrak{d}'''].$$

For $\overline{m}_1, \overline{m}_2 \in \text{supp}(\eta)$, from $\pi_i(\eta) = \llbracket s_i \rrbracket_{m_i}$ and $\mathfrak{d}'' \# \text{MV}(s_1), \text{MV}(s_2)$, we have $\mathfrak{d}''(\overline{m}_1, \overline{m}_2) = \mathfrak{d}''(m_1, m_2)$. Hence,

$$\begin{aligned} \mathbb{E}_{\eta}[\mathfrak{d}' + \mathfrak{d}'''] &\leq f(\mathfrak{d}(m_1, m_2)) + \mathbb{E}_{\overline{m}_1, \overline{m}_2 \sim \eta}[\mathfrak{d}''(m_1, m_2)] \\ &= f(\mathfrak{d}(m_1, m_2)) + |\eta| \cdot \mathfrak{d}''(m_1, m_2) \leq f(\mathfrak{d}(m_1, m_2)) + \mathfrak{d}''(m_1, m_2) \\ &\leq f(\mathfrak{d}(m_1, m_2)) + f(\mathfrak{d}''(m_1, m_2)) = f(\mathfrak{d}(m_1, m_2) + \mathfrak{d}''(m_1, m_2)). \end{aligned}$$

Hence, η is a coupling s.t. $\llbracket s_1 \rrbracket_{m_1} \langle \eta \rangle_{\mathfrak{d}' + \mathfrak{d}'' \leq \delta'}^{\Psi} \llbracket s_2 \rrbracket_{m_2}$, where $\delta' \triangleq f((\mathfrak{d} + \mathfrak{d}'')(m_1, m_2))$.

[MULT-MAX] A basic result about couplings is that for any two distributions η_1, η_2 over the same set, there exists a coupling η such that:

$$\Pr_{(a_1, a_2) \sim \eta} [a_1 \neq a_2] = \|\eta_1 - \eta_2\|_{TV}.$$

This coupling is called the *maximal* or *optimal* coupling (see, e.g., Thorisson [34]).

To show soundness of the rule, let (m_1, m_2) two memories and, for $i \in \{1, 2\}$, let $\mu_i \triangleq \llbracket \vec{x} \leftarrow \text{Mult}(\vec{p}) \rrbracket_{m_i}$. Let v_i be the distributions $\llbracket \text{Mult}(\vec{p}) \rrbracket_{m_i}$. Let μ be a coupling of μ_1 and μ_2 such that the projection of μ on the variables x_{\leftarrow} and x_{\rightarrow} is an maximal coupling of v_1 and v_2 ; note that the projection of μ_1 onto x_{\leftarrow} is v_1 , and the projection of μ_2 onto x_{\rightarrow} is v_2 . Now, we can prove the inequality on distances:

$$\mathbb{E}_{(m'_1, m'_2) \sim \mu} [\|\llbracket \vec{x} \rrbracket_{m'_1} - \llbracket \vec{x} \rrbracket_{m'_2}\|_1] \leq \|\llbracket \vec{p} \rrbracket_{m_1} - \llbracket \vec{p} \rrbracket_{m_2}\|_1$$

By definition we have:

$$\begin{aligned} \mathbb{E}_{(m'_1, m'_2) \sim \mu} [\|\llbracket \vec{x} \rrbracket_{m'_1} - \llbracket \vec{x} \rrbracket_{m'_2}\|_1] &= \sum_{m'_1, m'_2} \mu(m'_1, m'_2) \cdot \|\llbracket \vec{x} \rrbracket_{m'_1} - \llbracket \vec{x} \rrbracket_{m'_2}\|_1 \\ &= 2 \sum_{m'_1, m'_2} \sum_{a \neq b} \mathbb{1}[\llbracket \vec{x} \rrbracket_{m'_1} = a] \mathbb{1}[\llbracket \vec{x} \rrbracket_{m'_2} = b] \mu(m'_1, m'_2) \\ &\hspace{15em} (\text{distance is 0 or 2}) \\ &= 2 \sum_{a \neq b} \sum_{m'_1, m'_2} \mathbb{1}[\llbracket \vec{x} \rrbracket_{m'_1} = a] \mathbb{1}[\llbracket \vec{x} \rrbracket_{m'_2} = b] \mu(m'_1, m'_2) \\ &= 2 \sum_{a \neq b} \Pr_{(m'_1, m'_2) \sim \mu} [\llbracket \vec{x} \rrbracket_{m'_1} = a, \llbracket \vec{x} \rrbracket_{m'_2} = b] \\ &= 2 \cdot \Pr_{(m'_1, m'_2) \sim \mu} [\llbracket \vec{x} \rrbracket_{m'_1} \neq \llbracket \vec{x} \rrbracket_{m'_2}] \\ &= 2 \|v_1 - v_2\|_{TV} \hspace{10em} (\text{maximal coupling}) \\ &= \|\llbracket \vec{p} \rrbracket_{m_1} - \llbracket \vec{p} \rrbracket_{m_2}\|_1 \end{aligned}$$

[TRANS] We prove by induction on n that for every two memories m_1 and m_2 such that $\text{pd}_{\Phi \wedge \Phi'}(m_1, m_2) = n$, there exists a coupling μ s.t. $\llbracket s \rrbracket_{m_1} \langle \mu \rangle_{\mathfrak{d}' \leq f(\text{pd}_{\Phi \wedge \Phi'}(m_1, m_2))}^{\Psi} \llbracket s \rrbracket_{m_2}$. If $\text{pd}_{\Phi \wedge \Phi'}(m_1, m_2) = 0$, then $m_1 = m_2 (\triangleq m)$ by definition of path coupling. Let μ the diagonal distribution of $\llbracket s \rrbracket_m$. Since Ψ and $\mathbb{E}_{\mu}[\mathfrak{d}'] = 0$, we have:

$$\llbracket s \rrbracket_{m_1} \langle \mu \rangle_{\mathfrak{d}' \leq f(\text{pd}_{\Phi \wedge \Phi'}(m_1, m_2))}^{\Psi} \llbracket s \rrbracket_{m_2}.$$

Assume now that $\text{pd}_{\Phi \wedge \Phi'}(m_1, m_2) > 0$. Then, by the definition of path-coupling, there exists m s.t. $\text{pd}_{\Phi \wedge \Phi'}(m_1, m_2) = \text{pd}_{\Phi \wedge \Phi'}(m_1, m) + \text{pd}_{\Phi \wedge \Phi'}(m, m_2)$ with and $\text{pd}_{\Phi \wedge \Phi'}(m, m_2) = 1$, that is $m, m_2 \models \Phi \wedge \Phi'$. By induction hypothesis, there exists μ^* such that $\llbracket s \rrbracket_{m_1} \langle \mu^* \rangle_{\mathfrak{d}' \leq \delta^*}^{\Psi^*} \llbracket s \rrbracket_m$, where $\delta^* \triangleq f(\text{pd}_{\Phi \wedge \Phi'}(m_1, m))$. From the premise $\vdash \{\Phi \wedge \Phi'; -\} s \sim_f s \{\Psi; \mathfrak{d}'\}$, there exists μ_1 such that $\llbracket s \rrbracket_m \langle \mu_1 \rangle_{\mathfrak{d}' \leq f(1)}^{\Psi} \llbracket s \rrbracket_{m_2}$. Let

$$\mu : (m_1, m_2) \mapsto \sum_m \frac{1}{M(m)} \cdot \mu^*(m_1, m) \cdot \mu_1(m, m_2)$$

where $M(m) \triangleq \pi_1(\mu_1)(m) = \pi_2(\mu^*)(m)$. It is clear that $\text{supp}(\mu) \models \Psi^{*2} \subseteq \Psi^* \subseteq \Psi$. Moreover, for any m_1 :

$$\begin{aligned} \pi_1(\mu)(m_1) &= \sum_{m_2} \mu(m_1, m_2) \\ &= \sum_m \left(\frac{\mu^*(m_1, m)}{\pi_1(\mu_1)(m)} \cdot \underbrace{\sum_{m_2} \mu_1(m, m_2)}_{\pi_1(\mu_1)(m)} \right) \\ &= \sum_m \mu^*(m_1, m) = \pi_1(\mu^*)(m_1) = \llbracket s \rrbracket_{m_1}. \end{aligned}$$

Likewise, $\forall m_2. \pi_2(\mu)(m_2) = \llbracket s \rrbracket_{m_2}$. Last:

$$\begin{aligned} \mathbb{E}_{\mu}[\mathfrak{d}'] &= \sum_{m_1, m_2} \mu(m_1, m_2) \cdot \mathfrak{d}'(m_1, m_2) \\ &= \sum_{m_1, m_2, m} \frac{\mu^*(m_1, m) \cdot \mu_1(m, m_2)}{M(m)} \cdot \mathfrak{d}'(m_1, m_2) \\ &\leq \sum_{m_1, m} \mu^*(m_1, m) \cdot \mathfrak{d}'(m_1, m) \cdot \frac{\sum_{m_2} \mu_1(m, m_2)}{M(m)} \\ &\quad + \sum_{m, m_2} \mu_1(m, m_2) \cdot \mathfrak{d}'(m, m_2) \cdot \frac{\sum_{m_1} \mu^*(m_1, m)}{M(m)} \\ &\hspace{15em} \text{(Triangle Inequality (\mathfrak{d}' satisfies (H))} \\ &= \mathbb{E}_{\mu^*}[\mathfrak{d}'] + \mathbb{E}_{\mu_1}[\mathfrak{d}'] \\ &\leq f(\text{pd}_{\Phi'}(m_1, m)) + f(1) \hspace{10em} \text{(Induction hypothesis)} \\ &= f(\text{pd}_{\Phi'}(m_1, m) + 1) \hspace{10em} \text{(Linearity of } f) \\ &= f(\text{pd}_{\Phi'}(m_1, m_2)). \end{aligned}$$

□

B DETAILS FOR EXAMPLES

B.1 Convex SGM (§ 5.1)

We detail the bounds in the two cases. In the first case, the selected samples $S[i]_{\triangleleft}$ and $S[i]_{\triangleright}$ may be different. We need to show:

$$\|(\mathbf{w}_{\triangleleft} - \alpha_t \cdot (\nabla \ell(S[i], -))(\mathbf{w}_{\triangleleft})) - (\mathbf{w}_{\triangleright} - \alpha_t \cdot (\nabla \ell(S[i], -))(\mathbf{w}_{\triangleright}))\| \leq \|\mathbf{w}_{\triangleleft} - \mathbf{w}_{\triangleright}\| + 2\alpha_t L.$$

We can directly bound:

$$\begin{aligned} & \|(\mathbf{w}_{\triangleleft} - \alpha_t \cdot (\nabla \ell(S[i], -))(\mathbf{w}_{\triangleleft})) - (\mathbf{w}_{\triangleright} - \alpha_t \cdot (\nabla \ell(S[i], -))(\mathbf{w}_{\triangleright}))\| \\ & \leq \|\mathbf{w}_{\triangleleft} - \mathbf{w}_{\triangleright}\| + \alpha_t \|\nabla \ell(S[i], -))(\mathbf{w}_{\triangleleft})\| + \alpha_t \|\nabla \ell(S[i], -))(\mathbf{w}_{\triangleright})\| \\ & \leq \|\mathbf{w}_{\triangleleft} - \mathbf{w}_{\triangleright}\| + 2\alpha_t L \end{aligned}$$

where the first inequality is by the triangle inequality, and the second follows since $\ell(z, -)$ is L -Lipschitz. Thus, we can take $f = +2\alpha_t L$ in the first case.

The second case boils down to showing

$$\|(\mathbf{w}_{\triangleleft} - \alpha_t \cdot (\nabla \ell(S[i], -))(\mathbf{w}_{\triangleleft})) - (\mathbf{w}_{\triangleright} - \alpha_t \cdot (\nabla \ell(S[i], -))(\mathbf{w}_{\triangleright}))\| \leq \|\mathbf{w}_{\triangleleft} - \mathbf{w}_{\triangleright}\|.$$

when $S[i]_{\triangleleft} = S[i]_{\triangleright}$. This follows from a calculation similar to the proof by Hardt et al. [20, Lemma 3.7.2]:

$$\begin{aligned} & \|(\mathbf{w}_{\triangleleft} - \alpha_t \cdot (\nabla \ell(S[i], -))(\mathbf{w}_{\triangleleft})) - (\mathbf{w}_{\triangleright} - \alpha_t \cdot (\nabla \ell(S[i], -))(\mathbf{w}_{\triangleright}))\|^2 \\ & = \|\mathbf{w}_{\triangleleft} - \mathbf{w}_{\triangleright}\|^2 - 2\alpha_t \langle \nabla \ell(S[i], -))(\mathbf{w}_{\triangleleft}) - \nabla \ell(S[i], -))(\mathbf{w}_{\triangleright}), \mathbf{w}_{\triangleleft} - \mathbf{w}_{\triangleright} \rangle \\ & \quad + \alpha_t^2 \|\nabla \ell(S[i], -))(\mathbf{w}_{\triangleleft}) - \nabla \ell(S[i], -))(\mathbf{w}_{\triangleright})\|^2 \\ & \leq \|\mathbf{w}_{\triangleleft} - \mathbf{w}_{\triangleright}\|^2 - (2\alpha_t/\beta - \alpha_t^2) \|\nabla \ell(S[i], -))(\mathbf{w}_{\triangleleft}) - \nabla \ell(S[i], -))(\mathbf{w}_{\triangleright})\|^2 \\ & \leq \|\mathbf{w}_{\triangleleft} - \mathbf{w}_{\triangleright}\|^2. \end{aligned}$$

The first inequality follows since convexity and Lipschitz gradient implies that

$$\langle \nabla \ell(S[i], -))(\mathbf{w}_{\triangleleft}) - \nabla \ell(S[i], -))(\mathbf{w}_{\triangleright}), \mathbf{w}_{\triangleleft} - \mathbf{w}_{\triangleright} \rangle \geq \frac{1}{\beta} \|\nabla \ell(S[i], -))(\mathbf{w}_{\triangleleft}) - \nabla \ell(S[i], -))(\mathbf{w}_{\triangleright})\|^2.$$

The second inequality follows from $0 \leq \alpha_t \leq 2/\beta$. Thus, we can take $f = \text{id}$ in the second case.

B.2 Non-convex SGM (§ 5.2)

Suppose that the loss function ℓ is bounded in $[0, 1]$, possibly non-convex, but L -Lipschitz and with gradient β -Lipschitz. Suppose that we take non-increasing step sizes $0 \leq \alpha_t \leq \sigma/t$ for some constant $\sigma \geq 0$. Then, we will prove the following judgment:

$$\vdash \{\text{Adj}(S_{\triangleleft}, S_{\triangleright}); -\} \text{sgm} \sim_{\epsilon} \text{sgm} \{\top; |\ell(\mathbf{w}_{\triangleleft}, z) - \ell(\mathbf{w}_{\triangleright}, z)|\}$$

where

$$\epsilon \triangleq (2/n) \left[\left(\frac{2L^2}{\beta(1-1/n)} \right)^{1/(q+1)} T^{q/(q+1)} \right].$$

This example uses an advanced analysis from Hardt et al. [20, Lemma 3.11]. We can't directly express that result in our logic, but we can inline the proof. Roughly, the idea is that with large probability, the first bunch of steps don't see the differing example. By the time we hit the differing example, the step size has already decayed enough. To model this kind of reasoning, we will use the program transformation rules to split the loop into iterations before the critical step, and iterations after the critical step. Then, we will perform a probabilistic case in between, casing on whether we have seen the differing example or not.

To begin, let the critical iteration be

$$t_0 \triangleq \left\lceil \left(\frac{2L^2}{\beta(1-1/n)} \right)^{1/(q+1)} T^{q/(q+1)} \right\rceil,$$

where $q \triangleq \beta\sigma$. We can split the loop in `sgm` into two:

```

t ← 0;
while t < T ∧ t < t_0 do
  i ←$ [n];
  w ← w - α_t · (∇ℓ(S[i], -))(w);
  t ← t + 1;
while t < T do
  i ←$ [n];
  w ← w - α_t · (∇ℓ(S[i], -))(w);
  t ← t + 1;
return w
    
```

Call the loops $c_<$ and $c_>$, with loop bodies $w_<$ and $w_>$. In the first loop, we will bound the probability of $\|w_< - w_>\| > 0$. We want to prove the judgment

$$\{t_< = t_>; \mathbb{1}[w_< \neq w_>]\} w_< \sim_{+1/n} w_< \{t_< = t_>; \mathbb{1}[w_< \neq w_>]\}.$$

Again, we use the identity coupling when sampling i . Then, we case on whether we hit the differing example or not. In the first case, we hit the differing example and we need to prove

$$\{t_< = t_>; \mathbb{1}[w_< \neq w_>]\} w_< \sim_{+1} w_< \{t_< = t_>; \mathbb{1}[w_< \neq w_>]\}.$$

This boils down to showing:

$$\mathbb{1}[w - \alpha_t \cdot (\nabla\ell(S[i], -))(w)_< \neq w - \alpha_t \cdot (\nabla\ell(S[i], -))(w)_>] \leq \mathbb{1}[w_< \neq w_>] + 1$$

but this is clear since the indicator is in $\{0, 1\}$.

In the second case, we hit the same example and need to prove:

$$\{t_< = t_> \wedge S[i]_< = S[i]_>; \mathbb{1}[w_< \neq w_>]\} w_< \sim_{\text{id}} w_< \{t_< = t_>; \mathbb{1}[w_< \neq w_>]\}$$

This boils down to showing:

$$\mathbb{1}[w - \alpha_t \cdot (\nabla\ell(S[i], -))(w)_< \neq w - \alpha_t \cdot (\nabla\ell(S[i], -))(w)_>] \leq \mathbb{1}[w_< \neq w_>]$$

assuming that $S[i]_< = S[i]_>$. But this is clear also—if $w_< \neq w_>$ then there is nothing to prove, otherwise if $w_< = w_>$ then the projections are equal.

Putting these two cases together (noting that they happen with probability $1/n$ and $1 - 1/n$ respectively) and applying the loop rule, we have:

$$\{t_< = t_>; \mathbb{1}[w_< \neq w_>]\} w_< \sim_{+t_0/n} w_< \{t_< = t_>; \mathbb{1}[w_< \neq w_>]\}$$

as desired.

Now, we perform a probabilistic case on $w_< = w_>$. Suppose $w_< = w_>$. In the second loop, we know that $t_< = t_> \geq t_0$. By similar reasoning to the previous sections, we have:

$$\{t_< = t_> \wedge t_< \geq t_0; \|w_< - w_>\|\} w_> \sim_{f_c} w_> \{t_< = t_> \wedge t_< \geq t_0; \|w_< - w_>\|\}$$

where

$$\begin{aligned} f_c(x) &\triangleq (1/n + (1 - 1/n)(1 + \alpha_t\beta))x + 2\alpha_t L/n \\ &\leq (1 + (1 - 1/n)\sigma\beta/t)x + 2\sigma L/t_n \\ &\leq \exp((1 - 1/n)\sigma\beta/t)x + 2\sigma L/t_n. \end{aligned}$$

In the last step, we use $1 + x \leq \exp(x)$.

We can then apply the loop rule to show:

$$\{t_{\triangleleft} = t_{\triangleright} \wedge t_{\triangleleft} \geq t_0 \wedge w_{\triangleleft} = w_{\triangleright}; \|w_{\triangleleft} - w_{\triangleright}\|\} c_{\geq} \sim_f c_{\geq} \{t_{\triangleleft} = t_{\triangleright} \wedge t_{\triangleleft} \geq t_0; \|w_{\triangleleft} - w_{\triangleright}\|\}$$

where

$$\begin{aligned} f(x) &\triangleq x \cdot \prod_{r=t_0+1}^T \exp\left(\left(1 - 1/n\right) \frac{\sigma\beta}{r}\right) + \sum_{s=t_0+1}^T \frac{2\sigma L}{sn} \prod_{r=s+1}^T \exp\left(\left(1 - 1/n\right) \frac{\sigma\beta}{r}\right) \\ &= x \cdot \exp\left(\sigma\beta(1 - 1/n) \sum_{r=t_0+1}^T \frac{1}{r}\right) + \sum_{s=t_0+1}^T \frac{2\sigma L}{sn} \exp\left(\sigma\beta(1 - 1/n) \sum_{r=s+1}^T \frac{1}{r}\right) \\ &\leq x \cdot \exp(\sigma\beta(1 - 1/n) \log(T/t_0)) + \sum_{s=t_0+1}^T \frac{2\sigma L}{sn} \exp(\sigma\beta(1 - 1/n) \log(T/s)) \\ &= x \cdot \exp(\sigma\beta(1 - 1/n) \log(T/t_0)) + \frac{2\sigma L}{n} T^{\beta\sigma(1-1/n)} \sum_{s=t_0+1}^T s^{-\beta\sigma(1-1/n)-1} \\ &\leq x \cdot \exp(\sigma\beta(1 - 1/n) \log(T/t_0)) + \frac{2\sigma L}{n} T^{\beta\sigma(1-1/n)} \cdot \frac{1}{\beta\sigma(1 - 1/n)} t_0^{-\beta\sigma(1-1/n)} \\ &= x \cdot \exp(\sigma\beta(1 - 1/n) \log(T/t_0)) + \frac{2L}{\beta(n-1)} \left(\frac{T}{t_0}\right)^{\beta\sigma(1-1/n)} \\ &\leq x \cdot \exp(\sigma\beta(1 - 1/n) \log(T/t_0)) + \frac{2L}{\beta(n-1)} \left(\frac{T}{t_0}\right)^{\beta\sigma}. \end{aligned}$$

Let the last term be ρ . The first inequality uses $\sum_{t=a+1}^b 1/t \leq \log(b/a)$ and the second inequality uses $\sigma_{t=a+1}^b 1/t^c \leq a^{1-c}/(c-1)$ for $c > 1$; both facts follow from bounding the sum by an integral. By applying the Lipschitz assumption on ℓ and the [CONSEQ] rule, we have:

$$\{t_{\triangleleft} = t_{\triangleright} \wedge t_{\triangleleft} \geq t_0 \wedge w_{\triangleleft} = w_{\triangleright}; -\} c_{\geq} \sim_{L\rho} c_{\geq} \{t_{\triangleleft} = t_{\triangleright} \wedge t_{\triangleleft} \geq t_0; |\ell(w, z)_{\triangleleft} - \ell(w, z)_{\triangleright}|\}$$

for every example $z \in Z$.

In the other case, suppose $w_{\triangleleft} \neq w_{\triangleright}$. Applying the rule of consequence using the fact that the loss function is bounded in $[0, 1]$, we have:

$$\{t_{\triangleleft} = t_{\triangleright} \wedge t_{\triangleleft} \geq t_0 \wedge w_{\triangleleft} \neq w_{\triangleright}; -\} c_{\geq} \sim_1 c_{\geq} \{t_{\triangleleft} = t_{\triangleright} \wedge t_{\triangleleft} \geq t_0; |\ell(w, z)_{\triangleleft} - \ell(w, z)_{\triangleright}|\}.$$

Applying the rule [SEQCASE-A] to link the two loops, we have:

$$\{\text{Adj}(S_{\triangleleft}, S_{\triangleright}); -\} \text{sgm} \sim_{t_0/n + L\rho} \text{sgm} \{\top; |\ell(w, z)_{\triangleleft} - \ell(w, z)_{\triangleright}|\}.$$

Now, we have chosen t_0 to balance the two terms. Note that setting

$$t_0 \geq \delta \triangleq \left(\frac{2L^2}{\beta(1 - 1/n)}\right)^{1/(q+1)} T^{q/(q+1)}$$

gives $t_0/n + L\rho \leq 2t_0/n$ since δ balances the two terms, so we can conclude.

The proof uses an advanced sequential composition rule [SEQCASE-A], shown in Figure 7. This rule combines sequential composition with a case analysis on an event that, compared to [SEQCASE], may depend on both memories.

$$[\text{SEQCASE-A}] \frac{\begin{array}{c} \vdash \{\Phi; -\} s_1 \sim_{\gamma} s_2 \{\Theta; \mathbb{1}[e]\} \\ \vdash \{\Theta \wedge e; -\} s'_1 \sim_f s'_2 \{\Psi; \mathfrak{d}\} \quad \vdash \{\Theta \wedge \neg e; -\} s'_1 \sim_{f_-} s'_2 \{\Psi; \mathfrak{d}\} \end{array}}{\vdash \{\Phi; -\} s_1; s'_1 \sim_{\gamma \cdot f + f_-} s_2; s'_2 \{\Psi; \mathfrak{d}\}}$$

Fig. 7. Additional rules