Proving expected sensitivity of probabilistic programs

Gilles Barthe, Thomas Espitau, Benjamin Grégoire, Justin Hsu, Pierre-Yves Strub

Proving Expected Sensitivity of Probabilistic Programs

GILLES BARTHE, Imdea Software Institute, Spain
THOMAS ESPITAU, Sorbonne Universités, UPMC, France
BENJAMIN GRÉGOIRE, Inria Sophia Antipolis–Méditerranée, France
JUSTIN HSU, University College London, UK
PIERRE-YVES STRUB, École Polytechnique, France

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 similar inputs. By varying the distance, expected sensitivity recovers useful notions of probabilistic function sensitivity, including 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 EpRHL for proving expected sensitivity properties. Our logic features two key ideas. First, relational pre-conditions and post-conditions are expressed using *distances*, 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 examples beyond the reach of prior relational logics. Our main example formalizes uniform stability of the stochastic gradient method. Furthermore, we prove rapid mixing for a probabilistic model of 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 [1997], and formalize rapid mixing of the Glauber dynamics from statistical physics.

CCS Concepts: • **Theory of computation → Pre- and post-conditions**; **Program verification**; **Program analysis**;

Additional Key Words and Phrases: Program sensitivity, relational program logics, Kantorovich distance

ACM Reference Format:
Gilles Barthe, Thomas Espitau, Benjamin Grégoire, Justin Hsu, and Pierre-Yves Strub. 2018. Proving Expected Sensitivity of Probabilistic Programs. *Proc. ACM Program. Lang.* 2, POPL, Article 57 (January 2018), 38 pages. https://doi.org/10.1145/3158145

1 INTRODUCTION

Sensitivity is a fundamental property in mathematics and computer science, describing how small changes in inputs can affect outputs. Formally, the sensitivity of a function $g : A \to B$ is defined relative to two metrics $\mathfrak{d}_A$ and $\mathfrak{d}_B$ on A and B respectively. We say that f is α-sensitive if for every two inputs x_1 and x_2, the outputs are a bounded distance apart: $\mathfrak{d}_B(g(x_1), g(x_2)) \leq \alpha \cdot \mathfrak{d}_A(x_1, x_2)$. Bounded sensitivity plays a central role in many other fields, motivating broad range verification methods for bounding program sensitivity.

Authors' addresses: Gilles Barthe, Imdea Software Institute, Madrid, Spain; Thomas Espitau, Sorbonne Universités, UPMC, Paris, France; Benjamin Grégoire, Inria Sophia Antipolis–Méditerranée, Nice, France; Justin Hsu, University College London, London, UK; Pierre-Yves Strub, École Polytechnique, Paris, France.

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, called f*-sensitivity*. Formally, let $\mathfrak{d} : A \times A \to \mathbb{R}^+$ and $\mathfrak{d}' : B \times B \to \mathbb{R}^+$ be two distances, and let f be a non-negative affine function of the form $z \mapsto \alpha \cdot z + \beta$ with α, β non-negative. We say that $g : A \to B$ is f-sensitive iff for for every two inputs x_1 and x_2, $\mathfrak{d}_B(g(x_1), g(x_2)) \leq f(\mathfrak{d}_A(x_1, x_2))$. Taking f to be affine will allow f-sensitivity to compose cleanly in the probabilistic case, while still being expressive enough to model multiplicative and additive bounds on the output distance in terms of the input distance.

1.1 Expected Sensitivity

Let us now consider the case where g is probabilistic, i.e., $g : A \to \mathbb{D}(B)$. Since g produces distributions over B rather than elements of B, we have a choice of what output distance to take. One possibility is to allow arbitrary distances between distributions; however, such distances can be complex and difficult to reason about. We consider an alternative approach: lifting a distance $\mathfrak{d}_B$ on elements to a distance on distributions by averaging $\mathfrak{d}_B$ over some distribution μ on pairs $B \times B$. For any two nearby inputs of g leading to output distributions μ_1 and μ_2, we require this distribution μ to model μ_1 and μ_2 in a probabilistic sense; namely, its first and second marginals must be equal to μ_1 and μ_2. Such a distribution μ is known as a *probabilistic coupling* of μ_1 and μ_2 (we refer the reader to Lindvall [2002] and Thorisson [2000] for overviews of the rich theory of probabilistic couplings).

Formally, a probabilistic function g is *expected* f*-sensitive* if for every two inputs x_1 and x_2, there exists a coupling μ of $g(x_1)$ and $g(x_2)$, such that

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

The left-hand side is the *expected value* of the function $\mathfrak{d}_B$ over μ (the average distance between pairs drawn from μ), inspired by the *Wasserstein* metric, a well-studied distance on distributions in the theory of optimal transport [Villani 2008]. 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). When the outputs (y_1, y_2) are numbers, a natural notion of sensitivity bounds the difference between *average* outputs in terms of the distance between inputs (x_1, x_2). Taking the distance $\mathfrak{d}_B(y_1, y_2) \triangleq |y_1 - y_2|$, expected f-sensitivity implies

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

In other words, the two output distributions μ_1 and μ_2 have similar averages when the inputs (x_1, x_2) are close. This type of bound can imply *algorithmic stability*, a useful property for machine learning algorithms [Bousquet and Elisseeff 2002].

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

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

for every subset 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 closure under composition: the sequential (Kleisli) composition of an f-sensitive function with an f'-sensitive function yields an $f' \circ f$-sensitive function. As we will see, this property makes expected sensitivity a good target for formal verification.

1.2 Expected Sensitivity from Expectation Couplings

To bound expected distance, it suffices to find a coupling of the output distributions and show that the expected distance is sufficiently small. In general, there are multiple probabilistic couplings between any two distributions, leading to different expected distances.

To better reason about couplings and their expected distances, we develop a quantitative generalization of probabilistic coupling that captures 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 closure under sequential composition and transitivity.

1.3 $\mathbb{E}$pRHL: A Program Logic for Expected Sensitivity Bounds

By leveraging these principles, we can bound expected sensitivity by compositionally building an expectation coupling between output distributions from pairs of nearby inputs. Concretely, we develop a relational program logic $\mathbb{E}$pRHL to construct expectation couplings between pairs of programs. $\mathbb{E}$pRHL 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 $\Phi, \Psi : \mathcal{M} \times \mathcal{M} \to \mathbb{B}$ are relational assertions over pairs of memories, $\mathfrak{d}, \mathfrak{d}' : \mathcal{M} \times \mathcal{M} \to \mathbb{R}^+$ are non-negative distances on memories, and $f(z) = \alpha \cdot z + \beta$ is a non-negative affine function with $\alpha, \beta \geq 0$. $\mathbb{E}$pRHL 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 such that all pairs of output memories (m_1', m_2') in the support of μ (i.e., 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 f a *distance transformer*, as it bounds the (average) *post-distance* $\mathfrak{d}'$ in terms of the *pre-distance* $\mathfrak{d}$. When s_1 and s_2 are the same program s, for instance, a $\mathbb{E}$pRHL judgment establishes an expected sensitivity bound for s.

We give a rich Hoare-style proof system for $\mathbb{E}$pRHL, internalizing various composition properties of expectation couplings. For instance, given two judgments

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

the sequential composition rule in $\mathbb{E}$pRHL yields

$$\{\Phi; \mathfrak{d}\}\ s_1; s_1' \sim_{f' \circ f} s_2; s_2'\ \{\Psi; \mathfrak{d}''\}.$$

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

1.4 Applications

We illustrate the expressiveness of our proof system on several novel examples.

Stability of Stochastic Gradient Method. In machine learning, *stability* [Bousquet and Elisseeff 2002; Elisseeff et al. 2005] 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, so that its performance on the training set generalizes to new, unseen examples; in other words, it does not overfit. Recently, Hardt et al. [2016] show a quantitative stability bound for the Stochastic Gradient Method (SGM), a widely used optimization algorithm for training in machine learning. We verify their result for several variants of SGM within our logic, contributing to the expanding field of formal verification for machine learning algorithms [Huang et al. 2017; Katz et al. 2017; Selsam et al. 2017].

Rapid Mixing for Population Dynamics. Randomized algorithms are a useful tool for modeling biological and social phenomena (see, e.g., Jansen [2013]). 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 *RSM* (Replication-Selection-Mutate) model, which captures the evolution of an unstructured, asexual haploid population (see, e.g., Hartl and Clark [2006]). Recently, a series of papers prove rapid mixing of the RSM model under mild conditions [Dixit et al. 2012; Panageas et al. 2016; Vishnoi 2015]. 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 to Bubley and Dyer [1997], a theoretical tool for building couplings on Markov chains. Let Φ be a binary relation and suppose that the state space of the Markov chain is equipped with a *path metric* $\mathfrak{d}_\Phi$, i.e., 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 starting from neighboring states, then we can combine these pieces to give a coupling for the distributions started from *any* two states. More concretely, if for every two initial states at distance 1 under $\mathfrak{d}_\Phi$ 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 $\mathfrak{d}_\Phi$, 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: given a coupling for inputs related by Φ, we get a coupling for inputs related by Φ^k. In $\mathbb{E}$pRHL, we internalize this 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 proper colorings of a graph [Bubley and Dyer 1997].

1.6 Outline and Core Contributions

After illustrating our approach on a simple example (§ 2) and reviewing 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 $\mathbb{E}$pRHL 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 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 have implemented our logic in the EasyCrypt [Barthe et al. 2013], 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* $\ell : Z \to \mathbb{R}^d \to [0,1]$. 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 parameters $w \in \mathbb{R}^d$ aiming to minimize the expected loss of $\ell(-,w)$ on a fresh sample from $\mathcal{D}$. When the algorithm is randomized, we think of $A : Z^n \to \mathbb{D}(\mathbb{R}^d)$ as mapping the training examples to a distribution over parameters.

In order to minimize the loss on the true distribution $\mathcal{D}$, a natural idea is to use parameters that minimize the average error on the available training set. When the loss function ℓ is well-behaved this optimization problem, known as *empirical risk minimization*, can be solved efficiently. However there is no guarantee that these parameters *generalize* to the true distribution—even if they have low loss on the training set, they may induce high loss on fresh samples from the true distribution. Roughly speaking, the algorithm may select parameters that are too specific to the inputs, *overfitting* to the training set.

To combat overfitting, Bousquet and Elisseeff [2002] considered a technical property of the learning algorithm: the algorithm should produce similar outputs when executed on any 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 [2002]). Let $A : Z^n \to \mathbb{D}(\mathbb{R}^d)$ be an algorithm for some loss function $\ell : Z \to \mathbb{R}^d \to [0,1]$. 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 from an expected sensitivity condition, taking 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 the difference between losses ℓ on any single example.

Fact. For every pair of training sets $S, S' \in Z^n$ that differ in a single element, suppose there exists 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 $\pi_1, \pi_2 : \mathbb{D}(\mathbb{R}^d \times \mathbb{R}^d) \to \mathbb{D}(\mathbb{R}^d)$ 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.

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

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 parameters (w, w'). If we take the distance on training sets Z^n to be the symmetric distance (the number of differing elements between training sets), ϵ-uniform stability follows from expected sensitivity of the function A. To prove stability, then, we will establish expected sensitivity by (i) finding an expectation coupling and (ii) reasoning about the expected value of the distance function. Our logic EpRHL is designed to handle both tasks.

To demonstrate, we will show ϵ-stability of the *Stochastic Gradient Method* (SGM), following recent work by Hardt et al. [2016]. SGM is a simple and classical optimization algorithm commonly used in machine learning. Typically, the parameter space is $\mathbb{R}^d$ (i.e., the algorithm learns d real parameters). SGM maintains parameters w and iteratively updates w 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 $\ell(z, -) : \mathbb{R}^d \to [0, 1]$ evaluated at 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 final parameters. We can implement SGM in an imperative language as follows.

$$
\begin{array}{l}
w \leftarrow w_0; \\
t \leftarrow 0; \\
\mathsf{while}\ t < T\ \mathsf{do} \\
\quad i \xleftarrow{\$} [n]; \\
\quad g \leftarrow \nabla \ell(S[i], -)(w); \\
\quad w \leftarrow w - \alpha_t \cdot g; \\
\quad t \leftarrow t + 1; \\
\mathsf{return}\ w
\end{array}
$$

The program first initializes the parameters to some default value w_0. Then, it runs T iterations of the main loop. The first step in the loop samples a uniformly random index i from $[n] = \{0, 1, \ldots, n-1\}$, while the second step computes the gradient g. We will model the gradient operator ∇ as a higher-order function with type $(\mathbb{R}^d \to [0, 1]) \to (\mathbb{R}^d \to \mathbb{R}^d)$.[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; we write $\mathrm{Adj}(S_\triangleleft, S_\triangleright)$. 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 one index $j \in [n]$. We then 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. EpRHL 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 $\mathcal{M} \times \mathcal{M} \to \mathbb{R}^+$, and $f : \mathbb{R}^+ \to \mathbb{R}^+$ 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 μ of

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

the output distributions from executing s_1, s_2 on m_1, m_2 respectively such that the expected value of $\mathfrak{d}'$ on the coupling is at most $f(\mathfrak{d}(m_1, m_2))$ and all pairs of output memories in the support of μ satisfy Ψ.

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

$$\vdash \{\Phi; \|w_\triangleleft - w_\triangleright\|\}\ s_a \sim_{\mathrm{id}} s_a\ \{i_\triangleleft = i_\triangleright; \|w_\triangleleft - w_\triangleright\|\}. \tag{2}$$

The pre-condition Φ is shorthand for simpler invariants, including $t_\triangleleft = t_\triangleright$ and $\mathrm{Adj}(S_\triangleleft, S_\triangleright)$. The post-condition $i_\triangleleft = i_\triangleright$ indicates that the coupling assumes both executions sample the same index i. Finally, the pre- and post-distances indicate that the expected value of $\|w_\triangleleft - w_\triangleright\|$ does not grow—this is clear because s_a does not modify w.

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

$$\vdash \{\Phi \wedge S[i]_\triangleleft \neq S[i]_\triangleright; \|w_\triangleleft - w_\triangleright\|\}\ s_b \sim_{+\gamma} s_b\ \{\Phi; \|w_\triangleleft - w_\triangleright\|\} \tag{3}$$

where $+\gamma$ is the function $x \mapsto x + \gamma$ for some constant γ that depends on L and the α_t's—since the algorithm selects different examples in the two executions, the resulting parameters may grow a bit farther apart. In the second case, the chosen example $S[i]$ is the same in both executions so we can prove:

$$\vdash \{\Phi \wedge S[i]_\triangleleft = S[i]_\triangleright; \|w_\triangleleft - w_\triangleright\|\}\ s_b \sim_{\mathrm{id}} s_b\ \{\Phi; \|w_\triangleleft - w_\triangleright\|\}. \tag{4}$$

The identity distance transformer id indicates that 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 blend the distance transformers when composing s_a and s_b, yielding

$$\vdash \{\Phi; \|w_\triangleleft - w_\triangleright\|\}\ s_a; s_b \sim_{+\gamma/n} s_a; s_b\ \{\Phi; \|w_\triangleleft - w_\triangleright\|\}, \tag{5}$$

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

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:

$$\vdash \{\Phi; \|w_\triangleleft - w_\triangleright\|\}\ \mathsf{sgm} \sim_{+T\gamma/n} \mathsf{sgm}\ \{\Phi; \|w_\triangleleft - w_\triangleright\|\}. \tag{6}$$

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

$$\vdash \{\Phi; \|w_\triangleleft - w_\triangleright\|\}\ \mathsf{sgm} \sim_{+LT\gamma/n} \mathsf{sgm}\ \{\Phi; |\ell(w_\triangleleft, z) - \ell(w_\triangleright, z)|\} \tag{7}$$

for every example $z \in Z$. Since $w_\triangleleft$ and $w_\triangleright$ are initialized to the same value w_0, the initial pre-distance is zero so this judgment gives a coupling μ of the output distributions such that

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

Since the left side is larger than

$$\mathbb{E}_\mu[\ell(w_\triangleleft, z) - \ell(w_\triangleright, z)] = \mathbb{E}_{\mu_1}[\ell(w, z)] - \mathbb{E}_{\mu_2}[\ell(w, z)],$$

where μ_1 and μ_2 are the output distributions of sgm, SGM is $LT\gamma/n$-uniform stable.

3 EXPECTED SENSITIVITY

Before we 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 $z \mapsto \alpha \cdot z$; $\mathcal{L}^{\geq} \subseteq \mathcal{L}$ be the set of non-contractive linear functions, mapping $z \mapsto \alpha \cdot z$ with $\alpha \geq 1$; and $C \subseteq \mathcal{A}$ 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., $\boldsymbol{\beta}$) for 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* $\mathfrak{d}$ is a map $A \times A \to \mathbb{R}^+$. We use the term "distance" rather loosely—we do not assume any axioms, like reflexivity, symmetry, triangle inequality, etc. Distances are partially ordered by the pointwise order inherited from the 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 will be interpreted in terms of sub-distributions. A (discrete) *sub-distribution* over a set A is a map $\mu : A \to \mathbb{R}^+$ such that its *support*

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

is discrete and its *weight* $|\mu| \triangleq \sum_{a \in \mathrm{supp}(\mu)} \mu(a)$ 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 maps $E : A \to \mathbb{B}$, where $\mathbb{B}$ denotes the set of booleans. The probability of an event E w.r.t. a sub-distribution μ, written $\Pr_\mu[E]$, is defined as $\sum_{x|E(x)} \mu(x)$.

The *expectation* of a function $f : A \to \mathbb{R}^+$ 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. Expectations are linear: $\mathbb{E}_\mu[f + g] = \mathbb{E}_\mu[f] + \mathbb{E}_\mu[g]$ and $\mathbb{E}_\mu[k \cdot f] = k \cdot \mathbb{E}_\mu[f]$, where addition and scaling of functions are 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 $M : A \to \mathbb{D}(B)$. Then:

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

Intuitively, $\mathsf{bind}\ \mu\ M$ is the distribution from first sampling 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 $\mathsf{unit}\ x$, and abusing notation, $\mathbb{E}_{x\sim\mu}[M]$ and $\mathbb{E}_\mu[M]$ for $\mathsf{bind}\ \mu\ M$.

Given a distribution μ over pairs in $A \times B$, we can define the usual projections $\pi_1 : \mathbb{D}(A \times B) \to \mathbb{D}(A)$ and $\pi_2 : \mathbb{D}(A \times B) \to \mathbb{D}(B)$ 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 joint distribution over pairs, 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 $\mu_a \ \langle \mu \rangle \ \mu_b$, if $\pi_1(\mu) = \mu_a$ and $\pi_2(\mu) = \mu_b$.

3.2 Expected f-sensitivity

The core concept in our system is 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.

Definition 3.1. A probabilistic function $g : A \to \mathbb{D}(B)$ is *expected f-sensitive* (with respect to $\mathfrak{d}_A$ and $\mathfrak{d}_B$) if 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 coupling $g(x_1) \ \langle \mu \rangle \ g(x_2)$. When f maps $z \mapsto \alpha \cdot z + \beta$, 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 $g : A \to \mathbb{D}(\mathbb{R})$ 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*

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

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:

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

□

Proposition 3.3 (Probabilistic sensitivity). *Suppose that $g : A \to \mathbb{D}(B)$ 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:

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

By a classical theorem about couplings (see, e.g., Lindvall [2002]), the total-variation distance is at most the probability on the first line. □

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 $g : A \to \mathbb{D}(B)$ is expected f-sensitive and $h : B \to \mathbb{D}(C)$ is expected f'-sensitive. Then the (monadic) composition of g and h is expected $f' \circ f$-sensitive.*

Proof. Let $a_1, a_2 \in A$ be any pair of inputs. Since g is expected f-sensitive, there is a coupling $g(a_1)\ \langle\mu\rangle\ g(a_2)$ such that

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

Similarly for every $b_1, b_2 \in B$, there is a coupling $h(b_1)\ \langle M(b_1,b_2)\rangle\ h(b_2)$ such that

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

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 $\pi_2(\mu')(a_2) = \mathbb{E}_{g(a_2)}[h]$. We can bound the expected distance:

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

□

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

Proposition 3.5. *Let $\mathfrak{d}$ be a distance over A and let $f \in \mathcal{A}$. Let $g : A \to \mathbb{D}(A)$ 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 $g^T(x_1)\ \langle\mu\rangle\ g^T(x_2)$ such that*

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

This proposition can be seen as a variant of the Banach fixed point theorem. Informally, under some reasonable conditions on $\mathfrak{d}$, contractive probabilistic maps $g : A \to \mathbb{D}(A)$ have a unique stationary distribution, where a probabilistic map is *contractive* if 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

Expected f-sensitivity is a property of a probabilistic function. It will be useful to factor out the condition on distributions. To this end, we introduce *expectation couplings* a quantitative extension of probabilistic couplings where an average distance over the coupling is bounded.

Definition 3.6 (Expectation couplings). Let $\mathfrak{d} : A \times B \to \mathbb{R}^+$ 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 $(\mathfrak{d}, \delta)$-*expectation coupling* (or simply, an *expectation coupling*) for μ_a and μ_b if $\mu_a \langle\mu\rangle \mu_b$ and $\mathbb{E}_\mu[\mathfrak{d}] \leq \delta$.

We write $\mu_a \langle\mu\rangle^{\Phi}_{\mathfrak{d} \leq \delta} \mu_b$ when μ is an expectation coupling with support $\mathrm{supp}(\mu)$ 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 $g : A \to \mathbb{D}(B)$ is* expected f-sensitive *(with respect to $\mathfrak{d}_A$ and $\mathfrak{d}_B$) if 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 $\delta = f(\mathfrak{d}_A(x_1, x_2))$.*

Much like expected f-sensitive functions, expectation couplings are closed under sequential composition: given an expectation coupling between two distributions μ_a and μ_b, two functions $M_a : A \to \mathbb{D}(A')$ and $M_b : B \to \mathbb{D}(B')$ 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 $\Phi \subseteq A \times B$, $\mathfrak{d} : A \times B \to \mathbb{R}^+$, $\Psi \subseteq A \times B$, $\mathfrak{d}' : A \times B \to \mathbb{R}^+$, $\delta \in \mathbb{R}^+$, and $f \in \mathcal{A}$. Let $\mu_a \in \mathbb{D}(A)$, $M_a : A \to \mathbb{D}(A')$, and let $\mu'_a = \mathbb{E}_{\mu_a}[M_a]$. Let $\mu_b \in \mathbb{D}(B)$, $M_b : B \to \mathbb{D}(B')$, and set $\mu'_b = \mathbb{E}_{\mu_b}[M_b]$. Suppose we have functions $\mu \in \mathbb{D}(A \times B)$ and $M : (A \times B) \to \mathbb{D}(A' \times B')$ such that:*

(1) $\mu_a \langle\mu\rangle^{\Phi}_{\mathfrak{d} \leq \delta} \mu_b$ and

(2) $M_a(a) \langle M(a, b)\rangle^{\Psi}_{\mathfrak{d}' \leq f(\mathfrak{d}(a,b))} M_b(b)$ for every $(a, b) \in \Phi$.

Then $\mu'_a \langle\mu'\rangle^{\Psi}_{\mathfrak{d}' \leq f(\delta)} \mu'_b$, 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. We defer the details to the appendix. □

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:

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

where x, e, and g range over variables $\mathcal{V}$, expressions $\mathcal{E}$ and distribution expressions $\mathcal{D}$ 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 $\{0, \ldots, n-1\}$ or the Bernoulli distribution $\mathbf{Bern}(p)$, 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

$$\begin{aligned}
[\![\mathsf{skip}]\!]_m &= \delta_m & [\![x \xleftarrow{\$} g]\!]_m &= \mathbb{E}_{v\sim[\![g]\!]_m}[\delta_{m[x:=v]}] \\
[\![x \leftarrow e]\!]_m &= \delta_{m[x:=[\![e]\!]_m]} & [\![\text{if } e \text{ then } s_1 \text{ else } s_2]\!]_m &= \text{if } [\![e]\!]_m \text{ then } [\![s_1]\!]_m \text{ else } [\![s_2]\!]_m \\
[\![s_1; s_2]\!]_m &= \mathbb{E}_{\xi\sim[\![s_1]\!]_m}[[\![s_2]\!]_\xi] & [\![\text{while } b \text{ do } s]\!]_m &= \lim_{n\to\infty} [\![(\text{if } b \text{ then } s)^n_{|\neg b}]\!]_m
\end{aligned}$$

(Note that $\mathbb{E}$ is the monadic bind.)

Fig. 1. Denotational semantics of programs

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 [1979], probabilistic programs can be given a monadic denotational semantics, taking a memory as input and producing a sub-distribution on output memories. To avoid measure-theoretic technicalities, we limit our focus to discrete sub-distributions. Memories are type-preserving maps 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 $[\![e]\!]_m$ 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, the semantics $[\![s]\!]_m$ of a statement s w.r.t. to some initial memory m is the sub-distribution over states defined by the clauses of Fig. 1. 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

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

which we represent using the shorthand $(\text{if } b \text{ then } s)^n_{|\neg b}$. For any initial memory m, applying the truncated iterates yields an pointwise-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

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

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

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

Definition 4.1. A judgment $\{\Phi; \mathfrak{d}\}\ s_1 \sim_f s_2\ \{\Psi; \mathfrak{d}'\}$ is valid if for every memories m_1, m_2 s.t. $(m_1, m_2) \models \Phi$, there exists μ such that

$$[\![s_1]\!]_{m_1}\ \langle\mu\rangle^{\Psi}_{\mathfrak{d}' \le f(\mathfrak{d}(m_1,m_2))}\ [\![s_2]\!]_{m_2}$$

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 $[\![s]\!] : \mathcal{M} \to \mathbb{D}(\mathcal{M})$ 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

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

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 $e_\triangleleft$ and $e_\triangleright$, resp. called its left and right injections:

$$m_1 \models e \iff m_1, m_2 \models e_\triangleleft$$
$$m_2 \models e \iff m_1, m_2 \models e_\triangleright$$

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 $m[x \coloneqq e]$ denote the unique memory m' such that $m(y) = m'(y)$ if $y \neq x$ and $m'(x) = [\![e]\!]_m$. Then, given a variable x (resp. x'), an expression e (resp. e'), and an assertion Φ, we define the assertion $\Phi[x_\triangleleft, x'_\triangleright \coloneqq e_\triangleleft, e'_\triangleright]$ by the clause

$$\Phi[x_\triangleleft, x'_\triangleright \coloneqq e_\triangleleft, e'_\triangleright](m_1, m_2) \triangleq \Phi(m_1[x \coloneqq e], m_2[x' \coloneqq e']).$$

Substitution of distances is defined similarly. One can also define one-sided substitutions, for instance $\Phi[x_\triangleleft \coloneqq e_\triangleleft]$.

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 EpRHL 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 a bit more subtle. Informally, the rule selects a coupling, given as a bijection between supports, between the two sampled distributions in the left and right program.

The [SeqCase] rule combines sequential composition with a case analysis on properties satisfied by intermediate memories after executing s_1 and s_2. Informally, the rule considers events $e_1, \ldots, e_n$ such that Ψ entails $\bigvee_i e_{i\triangleleft}$. If for every i we can relate the programs s'_1 and s'_2 with distance transformer f_i, pre-condition $\Psi \wedge e_{i\triangleleft}; \mathfrak{d}'$ and post-condition $\Psi'; \mathfrak{d}''$, we 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 considers two loops synchronously, where the loop bodies preserve the invariant Ψ. 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: $f_1 \circ \cdots \circ f_n$.

The remaining rules are structural rules. The [Conseq] rule weakens the post-conditions, strengthens the pre-conditions, and relaxes the distance bounds.

The [Struct] rule replaces programs by equivalent programs. Figure 4 gives 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.

$$[\textsc{Assg}]\ \frac{}{\vdash \{\Psi[x_{1_\triangleleft} := e_1, x_{2_\triangleright} := e_2]; \mathfrak{d}'[x_{1_\triangleleft} := e_1, x_{2_\triangleright} := e_2]\}\ x_1 \leftarrow e_1 \sim_{\mathrm{id}} x_2 \leftarrow e_2\ \{\Psi; \mathfrak{d}'\}}$$

$$[\textsc{Rand}]\ \frac{h : \mathrm{supp}(g_1) \xrightarrow{1\text{-}1} \mathrm{supp}(g_2) \qquad \mathfrak{d} \triangleq \mathbb{E}_{v\sim g_1}[\mathfrak{d}'[x_{1_\triangleleft} := v, x_{2_\triangleright} := h(v)]] \qquad \forall v \in \mathrm{supp}(g_1).\, g_1(v) = g_2(h(v))}{\vdash \{\forall v \in \mathrm{supp}(g_1).\, \Psi[x_{1_\triangleleft} := v, x_{2_\triangleright} := h(v)]; \mathfrak{d}\}\ x_1 \xleftarrow{\$} g_1 \sim_{\mathrm{id}} x_2 \xleftarrow{\$} g_2\ \{\Psi; \mathfrak{d}'\}}$$

$$[\textsc{SeqCase}]\ \frac{\forall m_1, m_2 \models \Phi.\, (\sum_{i\in I} \Pr_{[\![s_1]\!]_{m_1}}[e_i] \cdot f_i) \circ f_0 \leq f \qquad \models \Psi \implies \bigvee_{i\in I} e_{i_\triangleleft} \qquad \vdash \{\Phi; \mathfrak{d}\}\ s_1 \sim_{f_0} s_2\ \{\Psi; \mathfrak{d}'\} \qquad \forall i \in I.\ \vdash \{\Psi \wedge e_{i_\triangleleft}; \mathfrak{d}'\}\ s_1' \sim_{f_i} s_2'\ \{\Psi'; \mathfrak{d}''\}}{\vdash \{\Phi; \mathfrak{d}\}\ s_1; s_1' \sim_f s_2; s_2'\ \{\Psi'; \mathfrak{d}''\}}$$

$$[\textsc{While}]\ \frac{\models \Psi \implies e_\triangleleft = e_\triangleright \wedge (i_\triangleleft \leq 0 \iff \neg e_\triangleleft) \qquad \forall 0 < k \leq n.\ \vdash \{\Psi \wedge e_{1_\triangleleft} \wedge i_\triangleleft = k; \mathfrak{d}_k\}\ s_1 \sim_{f_k} s_2\ \{\Psi \wedge i_\triangleleft = k-1; \mathfrak{d}_{k-1}\}}{\vdash \{\Psi \wedge i_\triangleleft = n; \mathfrak{d}_n\}\ \mathsf{while}\ e_1\ \mathsf{do}\ s_1 \sim_{f_1\circ\cdots\circ f_n} \mathsf{while}\ e_2\ \mathsf{do}\ s_2\ \{\Psi \wedge i_\triangleleft = 0; \mathfrak{d}_0\}}$$

$$[\textsc{Conseq}]\ \frac{\models \Phi' \implies \Phi \qquad \models \Psi \implies \Psi' \qquad \vdash \{\Phi; \mathfrak{d}\}\ s_1 \sim_f s_2\ \{\Psi; \mathfrak{d}'\} \qquad \models \Phi' \implies f(\mathfrak{d}) \leq f'(\mathfrak{d}'') \qquad \models \Psi \implies \mathfrak{d}''' \leq \mathfrak{d}'}{\vdash \{\Phi'; \mathfrak{d}''\}\ s_1 \sim_{f'} s_2\ \{\Psi'; \mathfrak{d}'''\}}$$

$$[\textsc{Struct}]\ \frac{\Phi_1 \vdash s_1 \equiv s_1' \qquad \Phi_2 \vdash s_2 \equiv s_2' \qquad \forall (m_1, m_2) \models \Phi.\, \Phi_1(m_1) \wedge \Phi_2(m_2) \qquad \vdash \{\Phi; \mathfrak{d}\}\ s_1 \sim_f s_2\ \{\Psi; \mathfrak{d}'\}}{\vdash \{\Phi; \mathfrak{d}\}\ s_1' \sim_f s_2'\ \{\Psi; \mathfrak{d}'\}}$$

$$[\textsc{Frame-D}]\ \frac{f \in \mathcal{L}^{\geq} \qquad \mathfrak{d}''\#\,\mathrm{MV}(s_1), \mathrm{MV}(s_2) \qquad \models \Phi \implies \mathfrak{d}'' \leq f(\mathfrak{d}'') \qquad \vdash \{\Phi; \mathfrak{d}\}\ s_1 \sim_f s_2\ \{\Psi; \mathfrak{d}'\}}{\vdash \{\Phi; \mathfrak{d} + \mathfrak{d}''\}\ s_1 \sim_f s_2\ \{\Psi; \mathfrak{d}' + \mathfrak{d}''\}}$$

Fig. 2. Selected proof rules

The [Frame-D] rule generalizes the typical frame rule, to preserve distances. Assuming that the distance $\mathfrak{d}''$ is not modified by the statements of the judgments and f is a non-contractive linear function (i.e., such that $x \leq f(x)$ for all x), validity is preserved when adding $\mathfrak{d}''$ to the pre-and post-distances of the judgment. Formally, $\mathrm{MV}(s)$ denotes the set of modified variables of s and the notation $\mathfrak{d}''\#\,\mathrm{MV}(s_1), \mathrm{MV}(s_2)$ states that for all memories m_1 and m_1' that coincide on the non-modified variables of s_1, and all memories m_2 and m_2' that coincide on the non-modified variables of s_2, we have $\mathfrak{d}''(m_1, m_2) = \mathfrak{d}''(m_1', m_2')$.

Theorem 4.2 (Soundness). *For every derivable judgment* $\vdash \{\Phi; \mathfrak{d}\}\ s_1 \sim_f s_2\ \{\Psi; \mathfrak{d}'\}$ *and initial memories* m_1 *and* m_2 *such that* $(m_1, m_2) \models \Phi$*, there exists* μ *such that*

$$[\![s_1]\!]_{m_1}\ \langle\mu\rangle^{\Psi}_{\mathfrak{d}' \leq f(\mathfrak{d}(m_1,m_2))}\ [\![s_2]\!]_{m_2}.$$

Proof. By induction on the derivation. We defer the details to the appendix. □

$$[\textsc{Seq}]\ \frac{\vdash \{\Phi;\mathfrak{d}\}\ s_1 \sim_f s_2\ \{\Xi;\mathfrak{d}'\} \qquad \vdash \{\Xi;\mathfrak{d}'\}\ s_1' \sim_{f'} s_2'\ \{\Psi;\mathfrak{d}''\}}{\vdash \{\Phi;\mathfrak{d}\}\ s_1;s_1' \sim_{f'\circ f} s_2;s_2'\ \{\Psi;\mathfrak{d}''\}}$$

$$[\textsc{Case}]\ \frac{\vdash \{\Phi \wedge e_{\triangleleft};\mathfrak{d}\}\ s_1 \sim_f s_2\ \{\Psi;\mathfrak{d}'\} \qquad \vdash \{\Phi \wedge \neg e_{\triangleleft};\mathfrak{d}\}\ s_1 \sim_f s_2\ \{\Psi;\mathfrak{d}'\}}{\vdash \{\Phi;\mathfrak{d}\}\ s_1 \sim_f s_2\ \{\Psi;\mathfrak{d}'\}}$$

$$[\textsc{Cond}]\ \frac{\models \Phi \implies e_{1_{\triangleleft}} = e_{2_{\triangleright}} \qquad \vdash \{\Phi \wedge e_{1_{\triangleleft}};\mathfrak{d}\}\ s_1 \sim_f s_2\ \{\Psi;\mathfrak{d}'\} \qquad \vdash \{\Phi \wedge \neg e_{1_{\triangleleft}};\mathfrak{d}\}\ s_1' \sim_f s_2'\ \{\Psi;\mathfrak{d}'\}}{\vdash \{\Phi;\mathfrak{d}\}\ \text{if } e_1 \text{ then } s_1 \text{ else } s_1' \sim_f \text{if } e_2 \text{ then } s_2 \text{ else } s_2'\ \{\Psi;\mathfrak{d}'\}}$$

$$[\textsc{Assg-L}]\ \frac{}{\vdash \{\Psi[x_{1_{\triangleleft}} := e_1];\mathfrak{d}'[x_{1_{\triangleleft}} := e_1]\}\ x_1 \leftarrow e_1 \sim_{\mathrm{id}} \mathsf{skip}\ \{\Psi;\mathfrak{d}'\}}$$

Fig. 3. Selected derived rules

4.3 Derived Rules and Weakest Pre-condition

Figure 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 that corresponding branches can be 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 also 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 4.3. *Let* $(\Phi'', \mathfrak{d}'') = \mathsf{wp}(s_1, s_2, \Psi, \mathfrak{d}')$. *Assume* $\Phi \implies \Phi''$ *and* $\mathfrak{d}(m_1, m_2) \leq \mathfrak{d}''(m_1, m_2)$ *for every* $(m_1, m_2) \models \Phi$. *Then* $\vdash \{\Phi;\mathfrak{d}\}\ s_1 \sim_{\mathrm{id}} s_2\ \{\Psi;\mathfrak{d}'\}$.

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 $\ell : Z \to \mathbb{R}^d \to [0,1]$. 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

$$\frac{}{\Phi \vdash s \equiv s} \qquad \frac{\Phi \vdash s_1 \equiv s_2}{\Phi \vdash s_2 \equiv s_1} \qquad \frac{}{\Phi \vdash x \xleftarrow{\$} \delta_x \equiv \mathsf{skip}} \qquad \frac{\models \Phi \implies x = e}{\Phi \vdash x \leftarrow e \equiv \mathsf{skip}} \qquad \frac{}{\Phi \vdash s; \mathsf{skip} \equiv s}$$

$$\frac{}{\Phi \vdash \mathsf{skip}; s \equiv s} \qquad \frac{\Phi \vdash s_1 \equiv s_1'}{\Phi \vdash s_1; s_2 \equiv s_1'; s_2} \qquad \frac{\top \vdash s_2 \equiv s_2'}{\Phi \vdash s_1; s_2 \equiv s_1; s_2'} \qquad \frac{\models \Phi \implies e}{\Phi \vdash \mathsf{if}\ e\ \mathsf{then}\ s\ \mathsf{else}\ s' \equiv s}$$

$$\frac{\models \Phi \implies \neg e}{\Phi \vdash \mathsf{if}\ e\ \mathsf{then}\ s\ \mathsf{else}\ s' \equiv s'} \qquad \frac{\Phi \wedge e \vdash s_1 \equiv s_2 \qquad \Phi \wedge \neg e \vdash s_1' \equiv s_2'}{\Phi \vdash \mathsf{if}\ e\ \mathsf{then}\ s_1\ \mathsf{else}\ s_1' \equiv \mathsf{if}\ e\ \mathsf{then}\ s_2\ \mathsf{else}\ s_2'}$$

Fig. 4. Equivalence rules

reproduce the code sgm:

$w \leftarrow w_0;$
$t \leftarrow 0;$
while $t < T$ do
  $i \xleftarrow{\$} [n];$
  $g \leftarrow \nabla\ell(S[i], -)(w);$
  $w \leftarrow w - \alpha_t \cdot g;$
  $t \leftarrow t + 1;$
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 $\nabla\ell(z, -) : \mathbb{R}^d \to \mathbb{R}^d$ must be β-Lipschitz.

5.1 SGM with Convex Loss

Suppose that the loss $\ell(z, -)$ 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 some 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:

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

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

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

By soundness (Theorem 4.2), this will imply that SGM is γ-uniformly stable.

As before, we will first establish a simpler judgment:

$$\vdash \{\Phi; \|w_\triangleleft - w_\triangleright\|\}\ \mathrm{sgm} \sim_{+\gamma/L} \mathrm{sgm}\ \{\Phi; \|w_\triangleleft - w_\triangleright\|\}.$$

As we proceed through the proof, we will indicate the corresponding step from the outline in § 2. Let j be the index such that the $S[j]_\triangleleft \neq S[j]_\triangleright$; this is the index of the differing example. First, we couple the samplings in s_a with the identity coupling, using rule [Rand] with $h = \mathrm{id}$ (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 use the Lipschitz property of $\ell(z, -)$ and some properties of the norm $\|\cdot\|$ to prove

$$\vdash \{\Phi \wedge e_=; \|w_\triangleleft - w_\triangleright\|\}\ s_b \sim_{+2\alpha_t L} s_b\ \{\Phi; \|w_\triangleleft - w_\triangleright\|\};$$

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

$$\vdash \{\Phi \wedge e_\neq; \|w_\triangleleft - w_\triangleright\|\}\ s_b \sim_{\mathrm{id}} s_b\ \{\Phi; \|w_\triangleleft - w_\triangleright\|\};$$

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 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. [2016]. Technically, they 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 carefully chosen time t_0. In particular, with high probability the parameters $w_\triangleleft$ and $w_\triangleright$ are equal up to iteration t_0. Then, they prove a uniform stability bound for SGM started at iteration t_0, assuming $w_\triangleleft = w_\triangleright$; if the step size α_t is taken to be rapidly decreasing, SGM will be already be contracting by iteration t_0.

This proof can also be carried out in EpRHL, with some extensions. First, we split the SGM program into two loops: iterations before t_0, and iterations after t_0. The probability of $w_\triangleleft \neq w_\triangleright$ is is precisely the expected value of the indicator function $\mathbb{1}[w_\triangleleft \neq w_\triangleright]$, 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 EpRHL. 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 a case analysis on $w_\triangleleft = w_\triangleright$ but this event depends on both sides—the existing probabilistic case rule [SeqCase] does not suffice. 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 $\vec{x}_t = (x_1, \ldots, x_m)$, where x_i represents the fraction of the population belonging to class i. In the Replication-Selection-Mutate (RSM) model, the evolution is described by a function f—called the *step function*—which 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

$$[\textsc{Mult-Max}]\ \frac{}{\vdash \{\top; \|\vec{p}_\triangleleft - \vec{p}_\triangleright\|_1\}\ \vec{x}_\triangleleft \xleftarrow{\$} \mathbf{Mult}(\vec{p}_\triangleleft) \sim_{\mathrm{id}} \vec{x}_\triangleright \xleftarrow{\$} \mathbf{Mult}(\vec{p}_\triangleright)\ \{\vec{x}_\triangleleft, \vec{x}_\triangleright \in \{0,1\}^m; \|\vec{x}_\triangleleft - \vec{x}_\triangleright\|_1\}}$$

Fig. 5. Maximal coupling rule for multinomial

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, we take the 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 $\{1, \ldots, m\}$; this is how we will encode the distribution of species.

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 $step : \Delta_m \to \Delta_m$ 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

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

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

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

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.

In the sampling instruction, $\mathbf{Mult}(\vec{p})$ represents 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 and $\sum_i p_i = 1$. We represent samples from the multinomial distribution as binary vectors in Δ_m: with probability p_i, the sampled vector has the ith 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 $\vec{x}_\triangleleft, \vec{x}_\triangleright \in \{0,1\}^m$ states that the samples are always binary vectors of length m, while the distances indicate that the expected distance between the sampled vectors $\|\vec{x}_\triangleleft - \vec{x}_\triangleright\|_1$ is at most the distance between the parameters $\|\vec{p}_\triangleleft - \vec{p}_\triangleright\|_1$.

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

$$\vdash \{\Phi; \|(\vec{x_0})_\triangleleft - (\vec{x_0})_\triangleright\|_1\}\ \mathsf{popdyn}(T) \sim_{\bullet L^T}\ \mathsf{popdyn}(T)\ \{\Phi; \|\vec{x}_\triangleleft - \vec{x}_\triangleright\|_1\} \tag{10}$$

where

$$\Phi \triangleq \|\vec{x}_\triangleleft - \vec{x}_\triangleright\|_1 < 1/N \implies \vec{x}_\triangleleft = \vec{x}_\triangleright.$$

Φ is an invariant throughout because every entry of $\vec{x}_\triangleleft$ and $\vec{x}_\triangleright$ 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 their loop bodies. We proceed in two steps. In the inner loop, we want

$$\vdash \{\Phi; \|\vec{p}_\triangleleft - \vec{p}_\triangleright\|_1\}\ s_{in} \sim_{\mathrm{id}}\ s_{in}\ \{\Phi; \|\vec{x}_\triangleleft - \vec{x}_\triangleright\|_1\} \tag{11}$$

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

$$\vdash \{e_\triangleleft = k \wedge \Phi; \mathfrak{d}_k\}\ w_{in} \sim_{\mathrm{id}}\ w_{in}\ \{e_\triangleleft = k-1 \wedge \Phi; \mathfrak{d}_{k-1}\} \tag{12}$$

for each $0 < k \leq N$, where $\mathfrak{d}_k \triangleq \|x_\triangleleft - x_\triangleright\|_1 + (k/N) \cdot \|p_\triangleleft - p_\triangleright\|_1$ 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

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

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-and the post-conditions by the frame rule [Frame-D] (noting that the distance transformer id is non-contractive). Since $\mathfrak{d}_k = \mathfrak{d}_{k-1} + (1/N) \cdot \|\vec{p}_\triangleleft - \vec{p}_\triangleright\|_1$ by definition, we have

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

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

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

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:

$$\vdash \{\Phi; \|\vec{x}_\triangleleft - \vec{x}_\triangleright\|_1\}\ w_{out} \sim_{\bullet L}\ w_{out}\ \{\Phi; \|\vec{x}_\triangleleft - \vec{x}_\triangleright\|_1\}.$$

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

$$\vdash \{\Phi; \|\mathit{step}(\vec{x}_\triangleleft) - \mathit{step}(\vec{x}_\triangleright)\|_1\}\ w_{out} \sim_{\mathrm{id}}\ w_{out}\ \{\Phi; \|\vec{x}_\triangleleft - \vec{x}_\triangleright\|_1\}.$$

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

$$\vdash \{\Phi; \|\vec{x}_\triangleleft - \vec{x}_\triangleright\|_1\}\ w_{out} \sim_{\bullet L}\ w_{out}\ \{\Phi; \|\vec{x}_\triangleleft - \vec{x}_\triangleright\|_1\}$$

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 $\nu_\triangleleft, \nu_\triangleright$ be the distributions of $\vec{x}$ after two executions of popdyn(T) from initial frequencies $(x_0)_\triangleleft, (x_0)_\triangleright \in \Delta_m$. Eq. (10) implies that there is an expectation coupling

$$\nu_\triangleleft \ \langle \nu \rangle^{\Phi}_{\|\cdot\|_1 \leq \delta} \ \nu_\triangleright,$$

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

$$\mathrm{TV}(\nu_\triangleleft, \nu_\triangleright) \leq N \cdot L^T.$$

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

7 PATH COUPLING AND GRAPH COLORING

Path coupling is a powerful method for proving rapid mixing of Markov chains [Bubley and Dyer 1997]. 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 classical 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 $g : A \to \mathbb{D}(B)$ is *locally expected f-sensitive* (with respect to $\mathfrak{d}_A$ and $\mathfrak{d}_B$) if for every $x_1, x_2 \in A$ such that $\mathfrak{d}_A(x_1, x_2) = 1$, we have

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

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:

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

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

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 let $\mathfrak{d}_B$ be a distance over B satisfying (H). Let $f \in \mathcal{L}$ and $g : A \to \mathbb{D}(B)$. 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 $g(x_1) = g(x_2)$. Letting 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, establishing the base case. For the inductive step, assume that $\mathfrak{d}_A(x_1, x_2) = n + 1$. Then there exists x' such that $\mathfrak{d}_A(x_1, x') = 1$ and $\mathfrak{d}_A(x', x_2) = n$. By induction, there exist two expectation couplings μ_1 and μ_n satisfying the distance conditions

$$\mathbb{E}_{\mu_1}[\mathfrak{d}_B] \le f(\mathfrak{d}_B(x_1, x')) \quad \text{and} \quad \mathbb{E}_{\mu_n}[\mathfrak{d}_B] \le f(\mathfrak{d}_B(x', x_2)).$$

Define μ as

$$\mu(x, y) \triangleq \sum_{z \in A} \frac{\mu_1(x, z) \cdot \mu_n(z, y)}{g(x')(z)},$$

where we treat terms with zero in the denominator as 0; note that since μ_1 and μ_n satisfy the marginal conditions, we have $\pi_2(\mu_1) = \pi_1(\mu_n) = g(x')$, so $g(x')(z) = 0$ implies that $\mu_1(x, z) = \mu_n(z, y) = 0$, 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_1 and μ_n. The distance condition $\mathbb{E}_\mu[\mathfrak{d}_B] \le f(\mathfrak{d}_A(x_1, x_2))$ is a bit more involved:

$$\begin{aligned}
\mathbb{E}_\mu[\mathfrak{d}_B] &= \sum_{x,y} \mu(x, y)\ \mathfrak{d}_B(x, y) \\
&= \sum_{x,y} \sum_{z} \left(\frac{\mu_1(x, z)\ \mu_n(z, y)}{g(x')(z)} \right) \mathfrak{d}_B(x, y) \\
&\le \sum_{x,y,z} \left(\frac{\mu_1(x, z)\ \mu_n(z, y)}{g(x')(z)} \right) \mathfrak{d}_B(x, z) + \sum_{x,y,z} \left(\frac{\mu_1(x, z)\ \mu_n(z, y)}{g(x')(z)} \right) \mathfrak{d}_B(z, y) && \text{(triangle ineq.)} \\
&= \sum_{y,z} \left(\sum_{x} \frac{\mu_1(x, z)}{g(x')(z)} \right) \mu_n(z, y)\ \mathfrak{d}_B(z, y) + \sum_{x,z} \left(\sum_{y} \frac{\mu_n(z, y)}{g(x')(z)} \right) \mu_1(x, z)\ \mathfrak{d}_B(x, z) \\
&= \sum_{x,z} \mu_1(x, z)\ \mathfrak{d}_B(x, z) + \sum_{y,z} \mu_n(z, y)\ \mathfrak{d}_B(z, y) && \text{(marginals)} \\
&= \mathbb{E}_{\mu_1}[\mathfrak{d}_B] + \mathbb{E}_{\mu_n}[\mathfrak{d}_B] \\
&\le f(\mathfrak{d}_A(x_1, x')) + f(\mathfrak{d}_A(x', x_2)) && \text{(distances)} \\
&= f(\mathfrak{d}_A(x_1, x') + \mathfrak{d}_A(x', x_2)) && (f \text{ linear}) \\
&= f(\mathfrak{d}_A(x_1, x_2)).
\end{aligned}$$

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

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 transitive closure and Φ^n denote the union of its n-fold compositions for $n \ge 1$. Assume that for every $a, a' \in A$, we have $(a, a') \in \Phi^*$. The *path metric* of Φ is the distance

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

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

$$[\textsc{Trans}]\ \frac{\begin{array}{c} f \in \mathcal{L} \qquad \mathfrak{d} : \mathbb{N} \qquad \mathfrak{d}' \text{ satisfies (H)} \\ \models \Phi \implies \mathrm{PathCompat}(\mathfrak{d}, \Phi) \qquad \models \Psi^* \implies \Psi \\ \vdash \{\Phi \wedge \mathfrak{d} = 0; -\}\ s \sim_0 s\ \{\Psi; \mathfrak{d}'\} \qquad \vdash \{\Phi \wedge \mathfrak{d} = 1; -\}\ s \sim_{f(1)} s\ \{\Psi; \mathfrak{d}'\} \end{array}}{\vdash \{\Phi; \mathfrak{d}\}\ s \sim_f s\ \{\Psi; \mathfrak{d}'\}}$$

Fig. 6. Transitivity rule

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

Corollary 7.4 (Bubley and Dyer [1997]). *Let $\mathfrak{d} = pd_\Phi$ for a binary relation Φ over A. Let $g : A \to \mathbb{D}(A)$ 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 $g^T(x_1)\ \langle\mu\rangle\ g^T(x_2)$ 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. □

7.2 Program Logic

We formulate a proof rule inspired from local expected sensitivity, in Fig. 6. Let us first consider the premises of the rule. The first three conditions are inherited from Proposition 7.2: the distance transformer f is linear, the pre-distance $\mathfrak{d}$ is $\mathbb{N}$-valued, and the post-distance satisfies condition (H). The new two conditions deal with the pre- and post-conditions, respectively. First, the pre-condition Φ and the pre-distance $\mathfrak{d}$ satisfy the following condition:

$$\begin{aligned} \mathrm{PathCompat}(\Phi, \mathfrak{d}) \triangleq \forall m_1, m_2, n \in \mathbb{N}.\, \Phi(m_1, m_2) \wedge \mathfrak{d}(m_1, m_2) = n + 1 \\ \implies \exists m'.\, \mathfrak{d}(m_1, m') = 1 \wedge \mathfrak{d}(m', m_2) = n \wedge \Phi(m_1, m') \wedge \Phi(m', m_2). \end{aligned}$$

This condition implies that $\mathfrak{d}$ satisfies condition (P) (needed for Proposition 7.2), but it is stronger: when the distance $\mathfrak{d}$ is at least 1, we can find some memory m' such that the pre-condition can also be split into $\Phi(m_1, m')$ and $\Phi(m', m_2)$. We call this condition *path compatibility*; intuitively, it states that the pre-condition is compatible with the path structure on the pre-distance. Likewise, the post-condition Ψ must be transitively closed; the transitivity rule represents a finite sequence of judgments with post-condition Ψ.

The main premises cover two cases: either the initial memories are at distance 0, or they are at distance 1. Given these two judgments, the conclusion gives a judgment for two input memories at *any* distance. In this way, the rule [Trans] models a transitivity principle for expectation couplings.

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

$$[\![s_1]\!]_{m_1}\ \langle\mu\rangle^{\Psi}_{\mathfrak{d}' \leq f(\mathfrak{d}(m_1, m_2))}\ [\![s_2]\!]_{m_2}.$$

Proof. By a similar argument as Proposition 7.2, with careful handling for the pre- and post-conditions. We defer details to the appendix. □

7.3 Example: Glauber Dynamics

The *Glauber dynamics* is a randomized algorithm for approximating uniform samples from the valid colorings of a finite graph. It is a prime example of an algorithm where rapid mixing can be established using the path coupling method [Bubley and Dyer 1997].

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 $w : V \to C$; 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 a set of vertices $V' \subseteq V$.

For a graph G and a fixed set of colors C, there may be multiple (or perhaps no) valid colorings. Jerrum [1995] proposed a simple Markov chain for sampling a uniformly random coloring. Beginning at any coloring w, it draws a uniform vertex v and a uniform color c, and then changes the color of v to c in w if no neighbor of v is colored c. The Glauber dynamics repeats this process for T steps T and returns the final coloring. We can model this process with the following program glauber(T):

$$
\begin{array}{l}
i \leftarrow 0; \\
\mathsf{while}\ i < T\ \mathsf{do} \\
\quad v \xleftarrow{\$} V; \\
\quad c \xleftarrow{\$} C; \\
\quad \mathsf{if}\ \mathcal{V}_G(w, v, c)\ \mathsf{then}\ w \leftarrow w[v \mapsto c]; \\
\quad i \leftarrow i + 1; \\
\mathsf{return}\ w
\end{array}
$$

The guard $\mathcal{V}_G(w, v, c)$ is true when the vertex v in coloring w can be colored c. Jerrum [1995] proved that the distribution over outputs for this process converges rapidly to the uniform distribution on valid colorings of G as we take more and more steps, provided we start with a valid coloring. While the original proof was quite technical, Bubley and Dyer [1997] gave a much simpler proof of the convergence by *path coupling*.

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 above) such that the expected distance (how many vertices are colored differently) is at most β. Then, the path coupling machinery gives a coupling of the processes started from any two colorings, 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 EpRHL, this final property corresponds to the following judgment:

$$\vdash \{\Phi_G; \mathrm{pd}_{\mathrm{Adj}}\}\ \mathsf{glauber}(T) \sim_{\bullet\beta^T} \mathsf{glauber}(T)\ \{\top; \mathrm{pd}_{\mathrm{Adj}}\}$$

Above, Adj holds on two states iff the colorings (stored in the variable w) differ in the color of a single vertex, and $\mathfrak{d}' \triangleq \mathrm{pd}_{\mathrm{Adj}}$ counts the number of vertices with $w_{\triangleleft}(v) \neq w_{\triangleright}(v)$. The pre-condition Φ_G captures properties of the graph; in particular, Φ_G states that Δ is the maximal degree in G, i.e., each vertex in G has at most Δ neighbors. Finally, β is a constant determined by the graph and the number of colors; in certain parameter ranges, β is strictly less than 1 and the Markov chain converges quickly from any initial state.

Now, we present the proof. Since the graph G is not modified in the program, we keep Φ_G as an implicit invariant throughout. We begin with the loop body s. We apply the rule [Trans] with pre- and post-condition $\Phi, \Psi \triangleq i_{\triangleleft} = i_{\triangleright}$, distances $\mathfrak{d}, \mathfrak{d}' \triangleq \mathrm{pd}_{\mathrm{Adj}}$, and $f \triangleq \bullet\beta$. The side-conditions are clear: f is linear, the $\mathfrak{d}'$ satisfies condition (H), the pre-condition Φ is compatible with the path

distance $\mathfrak{d}$, and the post-condition Ψ is transitively closed. The first main premise

$$\vdash \{\Phi \wedge \mathrm{pd}_{\mathrm{Adj}} = 0; -\}\ s \sim_0 s\ \{\Psi; \mathrm{pd}_{\mathrm{Adj}}\}$$

is easy to show: the initial states have $w_\triangleleft = w_\triangleright$, so simply coupling using the identity bijection in [Rand] preserves $w_\triangleleft = w_\triangleright$ and keeps the states at distance 0. The second main premise

$$\vdash \{\Phi \wedge \mathrm{pd}_{\mathrm{Adj}} = 1; -\}\ s \sim_{\boldsymbol{\beta}} s\ \{\Psi; \mathrm{pd}_{\mathrm{Adj}}\},$$

is more complicated. Note that $\mathrm{pd}_{\mathrm{Adj}} = 1$ is equivalent to Adj: the two initial colorings $w_\triangleleft$ and $w_\triangleright$ must differ in the color at a single vertex. So, Adj implies the invariant

$$\Xi(a, b, v_\delta) \triangleq \forall z \in V.\ \begin{cases} z = v_\delta \implies a = w_\triangleleft(z) \wedge b = w_\triangleright(z) \\ z \neq v_\delta \implies w_\triangleleft(z) = w_\triangleright(z) \end{cases}$$

for some differing vertex v_δ, which is colored as $a \triangleq w_\triangleleft(v_\delta)$ and $b \triangleq w_\triangleright(v_\delta)$ in the two respective colorings. By the [Case] rule, it suffices to show

$$\vdash \{\Phi \wedge \Xi(a, b, v_\delta); -\}\ s \sim_{\boldsymbol{\beta}} s\ \{\Psi; \mathrm{pd}_{\mathrm{Adj}}\}$$

for every $a, b \in C$ and $v_\delta \in V$.

We apply the [SeqCase] rule with s_{samp}, consisting of the two random samplings in the loop body, and s_{rest}, consisting of the conditional statement and the updates. For the first judgment, we first couple the vertex samplings with the identity coupling so that $v_\triangleleft = v_\triangleright$, using the rule [Rand] with $h = \mathrm{id}$. This gives:

$$\vdash \{\Phi \wedge \Xi(a, b, v_\delta); -\}\ v \xleftarrow{\$} V \sim_{\boldsymbol{\beta}} v \xleftarrow{\$} V\ \{\Phi \wedge \Xi(a, b, v_\delta) \wedge v_\triangleleft = v_\triangleright; \mathrm{pd}_{\mathrm{Adj}}\}.$$

Next, we can perform a case analysis on $v_\triangleleft$ using the rule [Case]. If $v_\triangleleft$ is not a neighbor of v_δ, then we couple samplings so that $c_\triangleleft = c_\triangleright$ with [Rand] with $h = \mathrm{id}$. Otherwise, we couple $c_\triangleleft = \pi^{ab}(c_\triangleright)$, where π^{ab} swaps a and b and leaves all other colors unchanged. This gives

$$\vdash \{\Phi \wedge \Xi(a, b, v_\delta); -\}\ s_{samp} \sim_{\boldsymbol{\beta}} s_{samp}\ \{\Theta; \mathrm{pd}_{\mathrm{Adj}}\},$$

where

$$\Theta \triangleq \Phi \wedge \Xi(a, b, v_\delta) \wedge v_\triangleleft = v_\triangleright \wedge \begin{cases} v_\triangleleft \in \mathcal{N}_G(v_\delta) \implies c_\triangleleft = \pi^{ab}(c_\triangleright) \\ v_\triangleleft \notin \mathcal{N}_G(v_\delta) \implies c_\triangleleft = c_\triangleright. \end{cases}$$

Continuing with [SeqCase], we distinguish the following three mutually exclusive cases with probabilities q_b, q_g, and q_n, depending on how the distance changes under the coupling:

- In the *bad case*, the distance may grow to 2. Taking the guard

  $$e_b \triangleq v \in \mathcal{N}_G(v_\delta) \wedge c = b,$$

  the assignment and consequence rules give

  $$\vdash \{\Theta \wedge e_{b_\triangleleft}; \mathrm{pd}_{\mathrm{Adj}}\}\ s_{rest} \sim_{\bullet 2} s_{rest}\ \{\Psi; \mathrm{pd}_{\mathrm{Adj}}\}.$$

  (In fact, this judgment can be proved without the guard $e_{b_\triangleleft}$ in the pre-condition since the path distance increases from 1 to at most 2, but we will need to bound the probability of the guard being true in order to apply [SeqCase].) The probability of this case is at most $\Delta/|V||C|$ since we must select a neighbor of v_δ and the color b in the first side, so $q_b \leq \Delta/|V||C|$.
- In the *good case*, the distance shrinks to zero. We take 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:

  $$\vdash \{\Theta \wedge e_{g_\triangleleft}; \mathrm{pd}_{\mathrm{Adj}}\}\ s_{rest} \sim_{\bullet 0} s_{rest}\ \{\Psi; \mathrm{pd}_{\mathrm{Adj}}\}.$$

We will later need a *lower* bound on the probability of this case: since we must choose the differing vertex v_δ 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*, we take the guard $e_n \triangleq \neg e_b \wedge \neg e_g$. The assignment rule gives

$$\vdash \{\Theta \wedge e_{n\lhd}; \mathrm{pd}_{\mathrm{Adj}}\}\ s_{rest} \sim_{\mathrm{id}} s_{rest}\ \{\Psi; \mathrm{pd}_{\mathrm{Adj}}\},$$

  showing that the distance remains unchanged.

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, [SeqCase] gives

$$\vdash \{\Phi \wedge \Xi(a, b, v_\delta); -\}\ s_{samp}; s_{rest} \sim_\beta s_{samp}; s_{rest}\ \{\Psi; \mathrm{pd}_{\mathrm{Adj}}\},$$

for every $a, b \in C$ and $v_\delta \in V$,where

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

So, we also have

$$\vdash \{\Phi \wedge \mathrm{pd}_{\mathrm{Adj}} = 1; -\}\ s_{samp}; s_{rest} \sim_\beta s_{samp}; s_{rest}\ \{\Psi; \mathrm{pd}_{\mathrm{Adj}}\}$$

and the rule [Trans] gives

$$\vdash \{\Phi; \mathrm{pd}_{\mathrm{Adj}}\}\ s_{samp}; s_{rest} \sim_\beta s_{samp}; s_{rest}\ \{\Psi; \mathrm{pd}_{\mathrm{Adj}}\}.$$

Finally, we apply the rule [While] with invariant $\Phi = \Psi$ and the assignment rule [Assg] to conclude the desired judgment

$$\vdash \{\Phi_G; \mathrm{pd}_{\mathrm{Adj}}\}\ \mathrm{glauber}(T) \sim_{\bullet\beta^T} \mathrm{glauber}(T)\ \{\top; \mathrm{pd}_{\mathrm{Adj}}\}.$$

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 and Glauber 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 are assumed.

The formalization of the examples is reasonably straightforward. The formalization of stability for the Stochastic Gradient Method is about 400 lines; about one third is devoted to proving mathematical facts. The formalization of convergence for the population dynamics about is 150 lines, while formalization of convergence for the Glauber dynamics is about 550 lines.

We have not yet interfaced current 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

Lipschitz continuity has also been considered extensively in the setting of program verification: Chaudhuri et al. [2010] develop a SMT-based analysis for proving programs robust, in the setting of a core imperative language; Reed and Pierce [2010] develop a linear type system for proving sensitivity and differential privacy in a higher-order language [Azevedo de Amorim et al. 2014, 2017; Gaboardi et al. 2013; Winograd-Cort et al. 2017].

There is also a long tradition of verifying expectation properties of probabilistic programs; seminal works include PPDL [Kozen 1985] and pGCL [Morgan et al. 1996]. Recently, Kaminski et al. [2016] have developed a method to reason about the expected running time 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.

There has also 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 [Barthe et al. 2009] and differential privacy of algorithms [Barthe et al. 2012]. $\mathbb{E}$pRHL subsumes the relational program logic considered by Barthe et al. [2009]; indeed, one can prove that the two-sided rules of pRHL are essentially equivalent to the fragment of $\mathbb{E}$pRHL where the pre-distance and post-distance are the zero function. In contrast, the relational program logic apRHL considered by Barthe et al. [2012] and developed in subsequent work [Barthe et al. 2017a, 2016b,d; Barthe and Olmedo 2013; Hsu 2017; Sato 2016] is not comparable with $\mathbb{E}$pRHL. apRHL uses a notion of approximate coupling targeting differential privacy, while expectation couplings are designed for average versions of quantitative relational properties. In particular, apRHL considers *pointwise* notions of distance between distributions without assuming a distance on the sample space, while $\mathbb{E}$pRHL works with distances on the underlying space, proving fundamentally different properties.

There have been a few works on more specific 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 [Eldib et al. 2015], a quantitative generalization that measures average leakage of the programs. Similarly, the bounded moment model [Barthe et al. 2016a] 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 for proving probing security. It would be interesting to develop a program logic based on $\mathbb{E}$pRHL 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 compare the expected payoff of an agent in two executions. Barthe et al. [2015, 2016c] 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 $\mathbb{E}$pRHL, and it is also not possible to carry the analyses required for our examples.

Lastly, Barthe et al. [2017b] use ×pRHL, a proof-relevant variant of pRHL, to extract a product program for the Glauber dynamics. In a second step, they analyze the product program to prove rapid mixing; their analysis is performed directly on the semantics of the product program. Our system improves upon this two-step approach 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 the side-condition in the [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 are a wealth of results to consider, for instance, see the survey by Bharucha-Reid et al. [1976]. On the practical side, it would be interesting to formalize more advanced examples featuring relational and probabilistic analysis, like the recent result by Shamir [2016] proving 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. [2016].

ACKNOWLEDGMENTS

We thank the anonymous reviewers for useful comments on this work. This work is partially supported by the European Research Council under Grant No. 679127, the National Science Foundation CNS under Grant No. 1513694, and the Simons Foundation under Grant No. 360368 to Justin Hsu.

REFERENCES

Arthur Azevedo de Amorim, Marco Gaboardi, Emilio Jesús Gallego Arias, and Justin Hsu. 2014. Really natural linear indexed type-checking. In *Symposium on Implementation and Application of Functional Programming Languages (IFL), Boston, Massachusetts*. ACM Press, 5:1–5:12. http://arxiv.org/abs/1503.04522

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.

Gilles Barthe, François Dupressoir, Sebastian Faust, Benjamin Grégoire, François-Xavier Standaert, and Pierre-Yves Strub. 2016a. Parallel Implementations of Masking Schemes and the Bounded Moment Leakage Model. *IACR Cryptology ePrint Archive* 2016 (2016), 912. http://eprint.iacr.org/2016/912

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.

Gilles Barthe, Thomas Espitau, Justin Hsu, Tetsuya Sato, and Pierre-Yves Strub. 2017a. ⋆-Liftings for differential privacy. In *International Colloquium on Automata, Languages and Programming (ICALP), Warsaw, Poland (Leibniz International Proceedings in Informatics)*, Vol. 80. Schloss Dagstuhl–Leibniz Center for Informatics, 102:1–102:12. https://arxiv.org/abs/1705.00133

Gilles Barthe, Noémie Fong, Marco Gaboardi, Benjamin Grégoire, Justin Hsu, and Pierre-Yves Strub. 2016b. Advanced Probabilistic Couplings for Differential Privacy. In *ACM SIGSAC Conference on Computer and Communications Security (CCS), Vienna, Austria*. 55–67. https://doi.org/10.1145/2976749.2978391

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.

Gilles Barthe, Marco Gaboardi, Emilio Jesús Gallego Arias, Justin Hsu, Aaron Roth, and Pierre-Yves Strub. 2016c. Computer-aided verification in mechanism design. In *Conference on Web and Internet Economics (WINE), Montréal, Québec*. http://arxiv.org/abs/1502.04052

Gilles Barthe, Marco Gaboardi, Benjamin Grégoire, Justin Hsu, and Pierre-Yves Strub. 2016d. Proving Differential Privacy via Probabilistic Couplings. In *IEEE Symposium on Logic in Computer Science (LICS), New York, New York*. 749–758. https://doi.org/10.1145/2933575.2934554

Gilles Barthe, Benjamin Grégoire, Justin Hsu, and Pierre-Yves Strub. 2017b. 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

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

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.

Gilles Barthe and Federico Olmedo. 2013. Beyond Differential Privacy: Composition Theorems and Relational Logic for f-divergences between Probabilistic Programs. In *International Colloquium on Automata, Languages and Programming (ICALP), Riga, Latvia (Lecture Notes in Computer Science)*, Vol. 7966. Springer-Verlag, 49–60. http://certicrypt.gforge.inria.fr/2013.ICALP.pdf

AT Bharucha-Reid et al. 1976. Fixed point theorems in probabilistic analysis. *Bull. Amer. Math. Soc.* 82, 5 (1976), 641–657.

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

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.

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.

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.

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 Transansactions on CAD of Integrated Circuits and Systems* 34, 10 (2015), 1558–1568. https://doi.org/10.1109/TCAD.2015.2424951

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

Marco Gaboardi, Andreas Haeberlen, Justin Hsu, Arjun Narayan, and Benjamin C. Pierce. 2013. Linear dependent types for differential privacy. In *ACM SIGPLAN–SIGACT Symposium on Principles of Programming Languages (POPL), Rome, Italy*. 357–370. http://dl.acm.org/citation.cfm?id=2429113

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

Daniel L. Hartl and Andrew G. Clark. 2006. *Principles of Population Genetics* (fourth ed.). Sinauer Associates.

Justin Hsu. 2017. *Probabilistic Couplings for Probabilistic Reasoning*. Ph.D. Dissertation. University of Pennsylvania. arXiv:cs.LO/1710.09951 https://arxiv.org/abs/1710.09951

Xiaowei Huang, Marta Kwiatkowska, Sen Wang, and Min Wu. 2017. Safety Verification of Deep Neural Networks. In *International Conference on Computer Aided Verification (CAV), Heidelberg, Germany (Lecture Notes in Computer Science)*, Rupak Majumdar and Viktor Kuncak (Eds.), Vol. 10426. Springer-Verlag, 3–29. https://doi.org/10.1007/978-3-319-63387-9_1

Thomas Jansen. 2013. *Analyzing Evolutionary Algorithms: The Computer Science Perspective*. Springer-Verlag. https://doi.org/10.1007/978-3-642-17339-4

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

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

Guy Katz, Clark W. Barrett, David L. Dill, Kyle Julian, and Mykel J. Kochenderfer. 2017. Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks. In *International Conference on Computer Aided Verification (CAV), Heidelberg, Germany*

(Lecture Notes in Computer Science), Rupak Majumdar and Viktor Kuncak (Eds.), Vol. 10426. Springer-Verlag, 97–117. https://doi.org/10.1007/978-3-319-63387-9_5

Dexter Kozen. 1979. Semantics of probabilistic programs. In *IEEE Symposium on Foundations of Computer Science (FOCS), San Juan, Puerto Rico*. 101–114.

Dexter Kozen. 1985. A Probabilistic PDL. *J. Comput. System Sci.* 30, 2 (1985), 162–178.

Torgny Lindvall. 2002. *Lectures on the coupling method*. Courier Corporation.

Carroll Morgan, Annabelle McIver, and Karen Seidel. 1996. Probabilistic Predicate Transformers. *ACM Transactions on Programming Languages and Systems* 18, 3 (1996), 325–353.

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

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

Tetsuya Sato. 2016. Approximate Relational Hoare Logic for Continuous Random Samplings. In *Conference on the Mathematical Foundations of Programming Semantics (MFPS), Pittsburgh, Pennsylvania*. http://arxiv.org/abs/1603.01445

Daniel Selsam, Percy Liang, and David L. Dill. 2017. Developing Bug-Free Machine Learning Systems With Formal Mathematics. In *International Conference on Machine Learning (ICML), Sydney, Australia (Proceedings of Machine Learning Research)*, Doina Precup and Yee Whye Teh (Eds.), Vol. 70. 3047–3056. http://proceedings.mlr.press/v70/selsam17a.html

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

Hermann Thorisson. 2000. *Coupling, Stationarity, and Regeneration*. Springer-Verlag.

Cédric Villani. 2008. *Optimal transport: Old and new*. Springer-Verlag.

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

Daniel Winograd-Cort, Andreas Haeberlen, Aaron Roth, and Benjamin C. Pierce. 2017. A framework for adaptive differential privacy. In *ACM SIGPLAN International Conference on Functional Programming (ICFP), Oxford, England*. 10:1–10:29. https://dl.acm.org/citation.cfm?id=3110254

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 $m \models \Phi$, $[\![s_1]\!]_m = [\![s_2]\!]_m$.*

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

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

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

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

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

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

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

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

□

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 $[\![s_1]\!]_{m_1} \langle\eta\rangle^{\Psi}_{\mathfrak{d}'\leq\delta} [\![s_2]\!]_{m_2}$. We use η for the coupling of the conclusion. We already know that $\forall i \in \{1,2\}.\ \pi_i(\mu) = [\![s_i]\!]_{m_i}$ and that $\mathrm{supp}(\mu) \subseteq \Psi \subseteq \Psi''$. Finally,

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

[Struct] Immediate consequence of Lemma A.1.

[Assg] & [Assg-L] Immediate.

[Rand] Let $m_1, m_2 \models \forall v \in \mathrm{supp}(g_1).\ \Psi[x_1, x_2 := v, h(v)]$ and $\mu_i \triangleq \mathbb{E}_{v\sim g_i}[\delta_{m_i[x_i:=v]}]$ for $i \in \{1,2\}$. Since h is a one to one mapping from $\mathrm{supp}(g_1)$ to $\mathrm{supp}(g_2)$ that preserves the mass, we have $|\mu_1| = |\mu_2|$ and $\mu_2 = \mathbb{E}_{v\sim g_1}[\delta_{m_2[x_2:=h(v)]}]$. Let

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

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

$$\begin{aligned}
\mathbb{E}_{\overline{m}\sim\mu}[\mathfrak{d}'] &= \mathbb{E}_{v\sim g_1}[\mathbb{E}_{\overline{m}\sim\delta_{(m_1[x_1:=v],\, m_2[x_2:=h(v)])}}[\mathfrak{d}']]\\
&= \mathbb{E}_{v\sim g_1}[\mathfrak{d}'(m_1[x_1 := v], m_2[x_2 := h(v)])]\\
&= \mathbb{E}_{v\sim g_1}[\mathfrak{d}'[(x_1)_\triangleleft, (x_2)_\triangleright := v, h(v)]].
\end{aligned}$$

[Seq] Let $(m_1, m_2) \models \Phi$ and, for $i \in \{1,2\}$, let $\mu_i \triangleq [\![c_i]\!]_{m_i}$ and $\eta_i(m) \triangleq [\![c_i']\!]_m$. 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 $\mathrm{supp}(\eta) \models \Xi$, 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 $\mathrm{supp}(\eta_m) \models \Psi$, 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 $\mathrm{supp}(\mu) \models \Psi$. We are left to prove that $\mathbb{E}_\mu[\mathfrak{d}''] \leq (f' \circ f)(\mathfrak{d}(m_1, m_2))$:

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

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

[Cond] Immediate consequence of [Case] and [Struct], using the synchronicity of both guards.

[SeqCase] For $\overline{m} \models \Psi \wedge \exists i.\, e_{i_\triangleleft}$, we denote by $\iota(\overline{m})$ an index i s.t. $\overline{m} \models e_{i_\triangleleft}$, and by $\eta_{\overline{m}}$ the coupling from

$$\vdash \{\Psi \wedge e_{\iota(\overline{m})_\triangleleft}; \mathfrak{d}'\}\ s_1'\ \sim_{f_i}\ s_2'\ \{\Psi'; \mathfrak{d}''\},$$

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

$$\begin{aligned}
\pi_i(\eta) = \pi_i(\mathbb{E}_{\overline{m} \sim \mu}[\eta_{\overline{m}}]) &= \mathbb{E}_{\overline{m} \sim \mu}[\underbrace{\pi_i(\eta_{\overline{m}})}_{[\![s_i']\!]_{\pi_i(\overline{m})}}] \\
&= \mathbb{E}_{m \sim \pi_i(\mu)}[[\![s_i']\!]_m] = \mathbb{E}_{m \sim [\![s_i]\!]_{m_i}}[[\![s_i']\!]_m] \\
&= m \mapsto [\![s_i; s_i']\!]_m.
\end{aligned}$$

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

$$\overline{p}_i = \Pr_{\overline{m} \sim \mu}[\iota(\overline{m}) = i] \leq \Pr_{\overline{m} \sim \mu}[[\![e_i]\!]_{\pi_1(\overline{m})}] = \Pr_{\underbrace{m \sim \pi_1(\mu)}_{m \sim [\![s_1]\!]_{m_1}}}[[\![e_i]\!]_m].$$

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

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

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

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

Hence,

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

where the last step is by the premise.

[While] We proceed by induction on n. For $i \in \{1, 2\}$, let $\overline{s_i} \triangleq \mathsf{while}\ e\ \mathsf{do}\ s_i$. For $n \in \mathbb{N}$, let $\Psi_n \triangleq \Psi \wedge (i_\triangleleft = n)$ and $\overline{f}_n \triangleq f_1 \circ \cdots \circ f_n$. If $n = 0$, under $m_1, m_2 \models \Psi$, we have $[\![e]\!]_{m_1} = [\![e]\!]_{m_2} = \bot$. Hence, for $i \in \{1, 2\}$, $[\![\overline{s_i}]\!]_{m_i} = \delta_{m_i}$ 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:

$$\begin{aligned}
&\vdash \{\Psi_{n+1} \wedge e_{1_\triangleleft}; \mathfrak{d}'_{n+1}\}\ s_1\ \sim_{f_{n+1}}\ s_2\ \{\Psi_n; \mathfrak{d}'_n\} \\
&\vdash \{\Psi_n; \mathfrak{d}'_n\}\ \overline{s_1}\ \sim_{\overline{f}_n}\ \overline{s_2}\ \{\Psi_0; \mathfrak{d}'_0\}
\end{aligned}$$

By reasoning similar to [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 $i \in \{1,2\}$, $[\![s_i; \overline{s_i}]\!]_{m_i} = [\![\overline{s_i}]\!]_{m_i}$. Hence, by reasoning similar to the one of [Struct], we obtain $\vdash \{\Psi_{n+1} \wedge e_{1_\triangleleft}; \mathfrak{d}'_{n+1}\}\ \overline{s_1} \sim_{\overline{f}_{n+1}} \overline{s_2}\ \{\Psi_0; \mathfrak{d}'_0\}$. By $\Psi_{n+1} \iff (\Psi_{n+1} \wedge e_{1_\triangleleft})$, 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, there is a coupling η s.t. $[\![s_1]\!]_{m_1}\ \langle\eta\rangle^{\Psi}_{\mathfrak{d}' \le \delta}\ [\![s_2]\!]_{m_2}$, where $\delta \triangleq f(\mathfrak{d}(m_1, m_2))$. Now, we have

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

For $\overline{m}_1, \overline{m}_2 \in \mathrm{supp}(\eta)$, from $\pi_i(\eta) = [\![s_i]\!]_{m_i}$ and $\mathfrak{d}'' \# \mathrm{MV}(s_1), \mathrm{MV}(s_2)$, we have $\mathfrak{d}''(\overline{m}_1, \overline{m}_2) = \mathfrak{d}''(m_1, m_2)$. The last line is because f is a non-contractive linear function, $f \in \mathcal{L}^{\ge}$. Hence,

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

Hence, η is a coupling s.t. $[\![s_1]\!]_{m_1}\ \langle\eta\rangle^{\Psi}_{\mathfrak{d}' + \mathfrak{d}'' \le \delta'}\ [\![s_2]\!]_{m_2}$, 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:

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

This coupling is called the *maximal* or *optimal* coupling (see, e.g., Thorisson [2000]).

To show soundness of the rule, let (m_1, m_2) two memories and, for $i \in \{1,2\}$, let $\mu_i \triangleq [\![\vec{x} \xleftarrow{\$} \mathbf{Mult}(\vec{p})]\!]_{m_i}$. Let ν_i be the distributions $[\![\mathbf{Mult}(\vec{p})]\!]_{m_i}$. Let μ be a coupling of μ_1 and μ_2 such that the projection of μ on the variables $x_\triangleleft$ and $x_\triangleright$ is a maximal coupling of ν_1 and ν_2; note that the projection of μ_1 onto $x_\triangleleft$ is ν_1, and the projection of μ_2 onto $x_\triangleright$ is ν_2. Now, we can prove the inequality on distances:

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

By definition we have:

$$\begin{aligned}\mathbb{E}_{(m'_1, m'_2) \sim \mu}[\|[\![\vec{x}]\!]_{m'_1} - [\![\vec{x}]\!]_{m'_2}\|_1] &= \sum_{m'_1, m'_2} \mu(m'_1, m'_2) \cdot \|[\![\vec{x}]\!]_{m'_1} - [\![\vec{x}]\!]_{m'_2}\|_1 \\ &= 2 \sum_{m'_1, m'_2} \sum_{a \neq b} \mathbb{1}[[\![\vec{x}]\!]_{m'_1} = a] \mathbb{1}[[\![\vec{x}]\!]_{m'_2} = b] \mu(m'_1, m'_2) \\ &\qquad \text{(distance is 0 or 2)} \\ &= 2 \sum_{a \neq b} \sum_{m'_1, m'_2} \mathbb{1}[[\![\vec{x}]\!]_{m'_1} = a] \mathbb{1}[[\![\vec{x}]\!]_{m'_2} = b] \mu(m'_1, m'_2) \\ &= 2 \sum_{a \neq b} \Pr_{(m'_1, m'_2) \sim \mu}[[\![\vec{x}]\!]_{m'_1} = a, [\![\vec{x}]\!]_{m'_2} = b] \\ &= 2 \cdot \Pr_{(m'_1, m'_2) \sim \mu}[[\![\vec{x}]\!]_{m'_1} \neq [\![\vec{x}]\!]_{m'_2}] \\ &= 2\mathrm{TV}(\nu_1, \nu_2) \qquad \text{(maximal coupling)} \\ &= \|[\![\vec{p}]\!]_{m_1} - [\![\vec{p}]\!]_{m_2}\|_1.\end{aligned}$$

[Trans] We prove by induction on $n \in \mathbb{N}$ that for every two memories m_1 and m_2 such that $\mathfrak{d}(m_1, m_2) = n$, there exists a coupling μ such that

$$[\![s]\!]_{m_1}\ \langle\mu\rangle^{\Psi}_{\mathfrak{d}' \le f(\mathfrak{d}(m_1, m_2))}\ [\![s]\!]_{m_2}.$$

For the base case $\mathfrak{d}(m_1, m_2) = 0$, the inductive hypothesis on the premise

$$\vdash \{\Phi \wedge \mathfrak{d} = 0; -\}\ s \sim_0 s\ \{\Psi; \mathfrak{d}'\}$$

give the desired coupling.
For the inductive step $\mathfrak{d}(m_1, m_2) = n + 1$, by path compatibility $\mathrm{PathCompat}(\Phi, \mathfrak{d})$ there exists m' with $\mathfrak{d}(m_1, m') = 1$ and $\mathfrak{d}(m', m_2) = n$ such that $\Phi(m_1, m')$ and $\Phi(m', m_2)$. By induction on n, there exists μ_n such that $[\![s]\!]_{m'}\ \langle \mu_n \rangle^{\Psi^*}_{\mathfrak{d}' \leq \delta_n}\ [\![s]\!]_{m_2}$, where $\delta_n \triangleq f(\mathfrak{d}(m', m_2))$. By the inductive hypothesis on premise

$$\vdash \{\Phi \wedge \mathfrak{d} = 1; -\}\ s \sim_{f(1)} s\ \{\Psi; \mathfrak{d}'\},$$

there exists μ_1 such that $[\![s]\!]_{m_1}\ \langle \mu_1 \rangle^{\Psi}_{\mathfrak{d}' \leq f(1)}\ [\![s]\!]_{m'}$, Define the coupling

$$\mu(m_1, m_2) \triangleq \sum_m \frac{\mu_1(m_1, m) \cdot \mu_n(m, m_2)}{M(m)}$$

where $M(m) \triangleq \pi_1(\mu_n)(m) = \pi_2(\mu_1)(m) = [\![s]\!]_{m'}$ and we drop terms with $M = 0$. By induction, $\mathrm{supp}(\mu) \models \Psi^2 \subseteq \Psi^* \subseteq \Psi$ since Ψ is transitive. The marginal conditions are straightforward: for any m_1,

$$\begin{aligned}
\pi_1(\mu)(m_1) &= \sum_{m_2} \mu(m_1, m_2) \\
&= \sum_m \left(\frac{\mu_1(m_1, m)}{\pi_1(\mu_n)(m)} \cdot \underbrace{\sum_{m_2} \mu_n(m, m_2)}_{\pi_1(\mu_n)(m)} \right) \\
&= \sum_m \mu_1(m_1, m) = \pi_1(\mu_1)(m_1) = [\![s]\!]_{m_1}.
\end{aligned}$$

Similarly, $\pi_2(\mu) = [\![s]\!]_{m_2}$. Finally, we show the expected distance condition:

$$\begin{aligned}
\mathbb{E}_\mu[\mathfrak{d}'] &= \sum_{m_1, m_2} \mu(m_1, m_2) \cdot \mathfrak{d}'(m_1, m_2) \\
&= \sum_{m_1, m_2, m} \frac{\mu_1(m_1, m) \cdot \mu_n(m, m_2)}{M(m)} \cdot \mathfrak{d}'(m_1, m_2) \\
&\leq \sum_{m_1, m} \mu_1(m_1, m) \cdot \mathfrak{d}'(m_1, m) \sum_{m_2} \frac{\mu_n(m, m_2)}{M(m)} \\
&\qquad + \sum_{m, m_2} \mu_n(m, m_2) \cdot \mathfrak{d}'(m, m_2) \sum_{m_1} \frac{\mu_1(m_1, m)}{M(m)} && (\mathfrak{d}' \text{ satisfies (H)}) \\
&= \mathbb{E}_{\mu_1}[\mathfrak{d}'] + \mathbb{E}_{\mu_n}[\mathfrak{d}'] \\
&\leq f(\mathfrak{d}(m_1, m')) + f(\mathfrak{d}(m', m_2)) && (\text{induction hypotheses}) \\
&= f(\mathfrak{d}(m_1, m_2)). && (f \in \mathcal{L})
\end{aligned}$$

□

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:

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

We can directly bound:

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

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

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

when $S[i]_\triangleleft = S[i]_\triangleright$. This follows from a calculation similar to the proof by Hardt et al. [2016, Lemma 3.7.2]:

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

The first inequality follows since convexity and Lipschitz gradient implies that

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

The second inequality follows from $0 \leq \alpha_t \leq 2/\beta$. Thus, we can take $f = \mathrm{id}$ 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 β-Lipschitz gradient. 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:

$$\vdash \{\mathrm{Adj}(S_\triangleleft, S_\triangleright); -\}\ \mathtt{sgm} \sim_\epsilon \mathtt{sgm}\ \{\top; |\ell(w_\triangleleft, z) - \ell(w_\triangleright, z)|\}$$

where

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

This example uses an advanced analysis from Hardt et al. [2016, 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:

$t \leftarrow 0;$
while $t < T \wedge t < t_0$ do
  $i \xleftarrow{\$} [n];$
  $w \leftarrow w - \alpha_t \cdot (\nabla \ell(S[i], -))(w);$
  $t \leftarrow t + 1;$
while $t < T$ do
  $i \xleftarrow{\$} [n];$
  $w \leftarrow w - \alpha_t \cdot (\nabla \ell(S[i], -))(w);$
  $t \leftarrow t + 1;$
return w

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

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

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

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

This boils down to showing:

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

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

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

$$\{t_\lhd = t_\rhd \wedge S[i]_\lhd = S[i]_\rhd ; \mathbb{1}[w_\lhd \neq w_\rhd]\}\ w_< \sim_{\mathrm{id}} w_< \ \{t_\lhd = t_\rhd ; \mathbb{1}[w_\lhd \neq w_\rhd]\}.$$

This boils down to showing:

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

assuming that $S[i]_\lhd = S[i]_\rhd$. But this is clear also—if $w_\lhd \neq w_\rhd$ then there is nothing to prove, otherwise if $w_\lhd = w_\rhd$ 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:

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

as desired.

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

$$\{t_\lhd = t_\rhd \wedge t_\lhd \geq t_0; \|w_\lhd - w_\rhd\|\}\ w_\geq \sim_{f_c} w_\geq \ \{t_\lhd = t_\rhd \wedge t_\lhd \geq t_0; \|w_\lhd - w_\rhd\|\}$$

where

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

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

We can then apply the loop rule to show:

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

where

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

Let the last term be ρ. The first inequality uses $\sum_{t=a+1}^{b} 1/t \le \log(b/a)$ and the second inequality uses $\sum_{t=a+1}^{b} 1/t^c \le a^{1-c}/(c-1)$ 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:

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

for every example $z \in Z$.

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

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

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

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

Note that setting

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

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

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

$$[\textsc{SeqCase-A}]\ \frac{\vdash \{\Phi;-\}\ s_1 \sim_{\boldsymbol{\gamma}} s_2\ \{\Theta;\mathbb{1}[e]\} \qquad \vdash \{\Theta \wedge e;-\}\ s_1' \sim_{f} s_2'\ \{\Psi;\mathfrak{d}\} \qquad \vdash \{\Theta \wedge \neg e;-\}\ s_1' \sim_{f_\neg} s_2'\ \{\Psi;\mathfrak{d}\}}{\vdash \{\Phi;-\}\ s_1; s_1' \sim_{\boldsymbol{\gamma}\cdot f + f_\neg} s_2; s_2'\ \{\Psi;\mathfrak{d}\}}$$

Fig. 7. Advanced sequential case rule