Local Inconsistency Resolution: The Interplay between Attention and Control in Probabilistic Models

Anonymous Author
Anonymous Institution

Abstract

We present a generic algorithm for learning and approximate inference with an intuitive epistemic interpretation: iteratively focus on a subset of the model and resolve inconsistencies using the parameters under control. This framework, which we call Local Inconsistency Resolution (LIR) is built upon Probabilistic Dependency Graphs (PDGs), which provide a flexible representational foundation capable of capturing inconsistent beliefs. We show how LIR unifies and generalizes a wide variety of important algorithms in the literature, including the Expectation-Maximization (EM) algorithm, belief propagation, adversarial training, GANs, and GFlowNets. Each of these methods can be recovered as a specific instance of LIR by choosing a procedure to direct focus (attention and control). We implement this algorithm for discrete PDGs and study its properties on synthetically generated PDGs, comparing its behavior to the global optimization semantics of the full PDG.

1 Introduction

What causes one to change their mind, to learn, act, and draw inferences? From a neurocognitive perspective, the mind operates by resolving internal inconsistencies, whether they arise from conflicting new information or from new awareness of contradictions among prior beliefs. Theories of cognitive dissonance (Festinger, 1962) famously explain much of human behavior as targeted adjustments to dispel the discomfort of psychological inconsistency. Predictive coding theories (Rao and Ballard, 1999; Friston, 2005) posit that discrepancies between expected and actual sensory inputs—known as prediction errors—are treated as inconsistency signals that drive learning via local adjustments to neural representation. Inspired by this neurocognitive principle, we introduce a computational framework for local inconsistency resolution (LIR) in probabilistic models.

The locality—which comes in two flavors—is key for those of us with cognitive limitations. First, inconsistency can be difficult to detect, and impossible to address constructively unless we are aware of it. A further complication is that one seldom has uniform control over one's epistemic state; we are reluctant to revise our direct observations or bedrock principles, forcing us to instead make adjustments to more pliable beliefs elsewhere. So in practice, we resolve inconsistencies *locallly*—by looking at only a small part of the picture, and changing another small part of it. This process isn't guaranteed to work: some inconsistencies simply cannot be seen with a narrow focus, and fixing one inconsistency can easily create others out of view.

Our approach leans heavily on the theory of Probabilistic Dependency Graphs (PDGs), which are very flexible graphical models that allow for arbitrary (even inconsistent) probabilistic information, weighted by confidence (Richardson and Halpern, 2021). There is a natural way to measure how inconsistent a PDG is, and many standard loss functions can be viewed as measuring the inconsistency of a PDG that describes the appropriate situation (Richardson, 2022). LIR operationalizes training models as the process of adjusting parameters to resolve this inconsistency.

Computing a PDG's degree of inconsistency is easy in some cases, but it is often intractable (e.g., the number of solutions to SAT problem and the log evidence of a latent variable model can both be represented as PDG inconsistencies), and is NP-hard in general. The problem is equivalent to inference in PDGs, which, like for other graphical modes, can be achieved in polynomial time under the assumption of bounded tree-width (Richardson et al., 2023). Variational inference (Blei et al., 2017; Kingma and Welling, 2014; Jordan et al.,

Preliminary work. Under review by AISTATS 2026. Do not distribute.

1999) can be understood as the practice of adopting extra beliefs to get an overapproximation of inconsistency that is easier to calculate (Richardson, 2022). Our approach also enables the opposite: focusing on small parts of the graph at a time to address tractable underapproximations of the global inconsistency. Nevertheless, we will show that LIR provides a powerful recipe for learning and (approximate) inference and demonstrate that many foundational techniques in the literature arise naturally as instances of it.

2 Preliminaries and Parametric PDGs

Variables and Probability. We write $\mathcal{V}X$ for the set of values that a (random) variable X can take on, and $\Delta\mathcal{V}X$ for the set of distributions over $\mathcal{V}X$. A conditional probability distribution (cpd) is a map $p(Y|X) : \mathcal{V}X \to \Delta\mathcal{V}Y$. If $\mathcal{X} = \{X_1, X_2, \dots\}_{i \in I}$ is an (indexed) set of variables, we regard $\mathcal{X}$ itself as a variable that can take on joint settings $\mathcal{V}\mathcal{X} = \prod_{i \in I} \mathcal{V}X_i$. If $\mu \in \Delta\mathcal{V}\mathcal{X}$ is a joint distribution and $Y \subseteq \mathcal{X}$, we write $\mu(Y)$ for the marginal of μ on the variables Y.

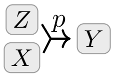
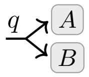
A *directed hypergraph* $(N, \mathcal{A})$ is a set of nodes N and a set of arcs $\mathcal{A}$, each $a \in \mathcal{A}$ of which is associated with a set $S_a \subseteq N$ of source nodes, and $T_a \subseteq N$ target nodes. We also write $S \xrightarrow{a} T \in \mathcal{A}$ to specify an arc a together with its sources $S = S_a$ and targets $T = T_a$.

Geometry. For our purposes, a *pointed parameter space* Θ is a convex subset of $\mathbb{R}^n$ for some $n \geq 0$ (that may differ between parameter spaces) with a distinguished default value $\theta_0 \in \Theta$. A *vector field* over Θ is a differentiable map X assigning to each $\theta \in \Theta$ a vector $X_\theta \in \mathbb{R}^n$. The *gradient* of a twice differentiable map $f : \Theta \to \mathbb{R}$, which we write $\nabla_\Theta f(\Theta)$, is a vector field. Given a vector field X and an initial point $\theta_0 \in \Theta$, there is a unique trajectory $y(t)$ that solves the ordinary differential equation (ODE) $\{\frac{\mathrm{d}}{\mathrm{d}t} y(t) = X_{y(t)}, y(0) = \theta_0\}$. To refer to that solution compactly, we adopt the notatation $\exp_{\theta_0}(X) := y(1)$. Although exp may appear to give us access only to $y(1)$, it is easily verified that $\exp_{\theta_0}(tX) = y(t)$ for all $t \geq 0$. Putting all the pieces together: the map $t \mapsto \exp_\theta(t \nabla_\Theta f(\Theta))$ is the smooth path beginning at θ that follows the gradient of f. It is known as *gradient flow*.

Probabilistic Dependency Graphs. A PDG is a directed (hyper)graph whose arcs carry probabilistic and causal information, weighted by confidence (Richardson and Halpern, 2021). We define an unweighted (but for our purposes, equivalent) variant whose explicit parametric nature will prove useful.

Definition 1. An *unweighted parametric PDG* $\mathcal{M}(\Theta) = (\mathcal{X}, \mathcal{A}, \Theta = \{\Theta_a\}_{a \in \mathcal{A}}, \mathbb{P} = \{\mathbb{P}_a\}_{a \in \mathcal{A}})$ is a directed hypergraph $(\mathcal{X}, \mathcal{A})$ whose nodes $\mathcal{X}$ are variables, and whose arcs $a \in \mathcal{A}$ are each associated with a parameter space Θ_a and a map $\mathbb{P}_a : \Theta_a \times \mathcal{V}S_a \to \Delta\mathcal{V}T_a$ that gives a probability distribution $\mathbb{P}_a(T_a|S_a;\theta)$ over a's target variables given values of its sources and a parameter setting $\theta \in \Theta_a$.

An unweighted PDG is the object obtained by fixing the parameters; thus, a joint setting $\boldsymbol{\theta} = (\theta_a)_{a \in \mathcal{A}}$ yields a PDG $\mathcal{M} = \mathcal{M}(\boldsymbol{\theta})$. Thus, at a syntactic level, an unweighted PDG is just an arbitrary collection of cpds. When constructing PDGs from known cpds (e.g., $p(Y|X)$), we use the mathematical symbol representing that cpd (e.g., p) for both a and $\mathbb{P}_a$. We typically depict (parametric) PDGs in graphical notation, specifying a hypergraph

$p(Y|X,Z)$ as  and $q(A,B)$ as .

It is not hard to see that a PDG can be faithfully viewed as the special case of a parametric PDG in which every $\Theta_a = \{\theta_0\}$ is a singleton containing only the default value. Conversely, a parametric PDG may be viewed as a PDG by adding each Θ_a as an explicit variable.

PDG Semantics and Inconsistency. The power of PDGs comes from their semantics, which sew their (possibly inconsistent) cpds and confidences together into joint probabilistic information.

A PDG contains two kinds of information: qualitative information about the types of causal mechanisms (the graph $\mathcal{A}$), and observational data (the cpds $\mathbb{P}$). In the standard presentation, a PDG also comes with weight vectors $\boldsymbol{\alpha}, \boldsymbol{\beta} \in \overline{\mathbb{R}}^{\mathcal{A}}$ over the arcs, encoding confidence in information of each type: α_a can be thought of as the number of independent causal mechanisms by which S_a determines T_a, while β_a can be thought of as the effective number of independent reports endorsing that the probability of T_a given S_a is $\mathbb{P}_a$. Corresponding to these two types of information, PDG semantics provide a way of scoring compatibility of a joint distribution $\mu(\mathcal{X})$ with information of each type.

With respect to a PDG $\mathcal{M}$, the *observational incompatibility* of a joint probability measure $\mu \in \Delta\mathcal{V}\mathcal{X}$ is given by a weighted sum of relative entropies

$$\mathit{OInc}_{\mathcal{M}}(\mu) := \sum_{S \xrightarrow{a} T \in \mathcal{A}} \beta_a \, \boldsymbol{D}\Big(\mu(T,S) \,\Big\|\, \mathbb{P}_a(T|S)\mu(S)\Big), \quad (1)$$

and can be thought of as the excess cost of using codes optimized for our beliefs $\mathbb{P}$ weighted by the confidence we have in them, when in fact $\mathcal{X} \sim \mu$. If $\boldsymbol{\beta} > \mathbf{0}$, then $\mathit{OInc}_{\mathcal{M}}(\mu) = 0$ if and only if μ satisfies the constraints imposed by every cpd of $\mathbb{P}$. Intuitively, the cpds of $\mathbb{P}$ are inconsistent when constraints are not simultaneously satisfiable, in which case the *observational inconsistency* $\langle\!\langle \mathcal{M} \rangle\!\rangle_0 := \inf_\mu \mathit{OInc}_{\mathcal{M}}(\mu)$ is an important measure the magnitude of the unavoidable internal conflict between the probabilistic data in $\mathcal{M}$.

Modern machine learning has a tendency to prize observational data above all else, and in particular above structural informaion such as causal influence or qualitative independencies; for this reason, the scoring function *OInc* and the corresponding observational inconsistency $\langle\!\langle \mathcal{M} \rangle\!\rangle_0$ will suffice for most of our examples. That said, we can also score μ by its incompatibility with the structural information in the weighted hypergraph $(\mathcal{A}, \boldsymbol{\alpha})$. This *structural deficiency* is given by:[1]

$$\mathit{SDef}_{\mathcal{A},\boldsymbol{\alpha}}(\mu) := -\operatorname{H}(\mathcal{X}) + \sum_{a \in \mathcal{A}} \alpha_a \operatorname{H}(T_a \mid S_a), \qquad (2)$$

and, roughly speaking, measures μ's failure to arise as a result of an independent causal mechanism along each hyperarc (Richardson et al., 2024). If $\mathcal{A}$ is a qualitative Bayesian Network and $\boldsymbol{\alpha} = \mathbf{1}$, for instance, then $\mathit{SDef}_{\mathcal{A},\boldsymbol{\alpha}}(\mu) \geq 0$ with equality iff μ has the conditional independencies of $\mathcal{A}$. With confidence $\gamma \geq 0$ in the structural information overall, the γ-*inconsistency* of $\mathcal{M}$ is the smallest possible overall incompatibility of any distribution with $\mathcal{M}$, and denoted

$$\langle\!\langle \mathcal{M} \rangle\!\rangle_\gamma := \inf_\mu \Big(\mathit{OInc}_{\mathcal{M}}(\mu) + \gamma\, \mathit{SDef}_{(\mathcal{A},\boldsymbol{\alpha})}(\mu) \Big). \qquad (3)$$

Richardson (2022) argues that this inconsistency measure (3) is a "universal" loss function, largely by showing how it specializes to standard loss functions across a wide breadth of contexts. It follows that, at least at an abstract level, much of machine learning can be viewed as inconsistency resolution. We now take this idea a few steps further, by operationalizing the resolution process, and allowing it to be done "locally". But what exactly do we mean by that?

3 The Local Inconsistency Resolution (LIR) Algorithm

There are two distinct senses in which inconsistency resolution can be *local*: we can restrict what we can see, or what we can do about it. Correspondingly, there are two "focus" knobs for our algorithm: one restricts our **attention** to the inconsistency of a subset of arcs $A \subseteq \mathcal{A}$, and the other restricts our **control** to only the parameters of a subset $C \subseteq \mathcal{A}$ as we resolve that inconsistency. The former makes for an underestimate of the inconsistency that is easier to calculate, while the latter makes for an easier optimization problem. These restrictions are not just cheap approximations, though: they are also appropriate modeling assumptions for actors that cannot see and control everything at once.

Attention and control need not be black and white. A more general approach would be to choose an *attention mask* $\varphi \in \mathbb{R}^{\mathcal{A}}$ and a *control mask* $\chi \in [0, \infty]^{\mathcal{A}}$. Large $\varphi(a)$ makes a salient while $\varphi(a) = 0$ keeps it out of mind; similarly, large $\chi(a)$ gives significant freedom to change a's parameters, small $\chi(a)$ affords only minor adjustments, and $\chi(a) = 0$ prevents change altogether.

In full generality, we refine the types of φ and χ one step further by breaking the control and attention in each arcs into natural subcomponents. The behavior of the attention mask φ described in the previous paragraph matches the role played by $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ in PDG semantics—which why we needed only to define unweighted PDGs as our basic objects, but require weighted semantics for them. The parameter γ also describes a form of attention, to the overall qualitative information in the network. Thus we take our control mask $\varphi = (\boldsymbol{\alpha}, \boldsymbol{\beta}, \gamma)$ to be a vector of dimension $2|\mathcal{A}| + 1$, and write $\langle\!\langle \varphi \odot \mathcal{M} \rangle\!\rangle := \langle\!\langle \mathcal{M}, \boldsymbol{\alpha}, \boldsymbol{\beta} \rangle\!\rangle_\gamma$ to denote the γ-inconsistency of the weighted PDG $(\mathcal{M}, \boldsymbol{\alpha}, \boldsymbol{\beta})$. Because we will typically set $\gamma = 0$, rendering $\boldsymbol{\alpha}$ irrelevant (except in Section 4.4 where $\gamma = 1$ and $\boldsymbol{\alpha} = \boldsymbol{\beta}$), we assume that $\boldsymbol{\alpha} = \boldsymbol{\beta} = \mathbf{1}$ unless otherwise specified. We also allow $\chi(a)$ to be a vector of dimension $\dim \Theta_a$, so overall χ is a vector of dimension $\sum_{a \in \mathcal{A}} \dim \Theta_a$.

Algorithm 1 Local Inconsistency Resolution (LIR)

```
Require: procedure REFOCUS
Input: knowledge base $\mathcal{M}(\Theta)$,
Initialize $\boldsymbol{\theta}^{(0)} \leftarrow 0$;
for $t = 0, 1, 2, \ldots$ do
    $\varphi, \chi \leftarrow$ REFOCUS();
    $\boldsymbol{\theta}^{(t+1)} \leftarrow \exp_{\boldsymbol{\theta}^{(t)}} \Big\{ -\chi \odot \nabla_{\boldsymbol{\theta}} \Big\langle\!\!\Big\langle \varphi \odot \mathcal{M}(\boldsymbol{\theta}) \Big\rangle\!\!\Big\rangle \Big\}$;
```

The full procedure, formalized in Algorithm 1, is a heuristic algorithm that adjusts the parameters of a parametric PDG $\mathcal{M}(\Theta)$ as to locally resolve inconsistencies as follows. First, accept a belief state in the form of a parametric PDG $\mathcal{M}(\Theta)$ and initialize the parameters to their default values. In each iteration, choose *focus* consisting of an attention mask $\varphi = (\boldsymbol{\alpha}, \boldsymbol{\beta}, \gamma)$ and a control mask χ. Calculate $\langle\!\langle \varphi \odot \mathcal{M}(\Theta) \rangle\!\rangle_\gamma$, the inconsistency of the combined context and memory, weighted by attention. In many cases (e.g., when $\boldsymbol{\beta} \geq \gamma \boldsymbol{\alpha}$) these problems can be solved with techniques in conic optimization (Richardson et al., 2023), but this may be intractable unless the attention is narrow enough or one can find a formula for it in closed form. Finally, mitigate this local inconsistency by updating mutable memory via (an approximation to) gradient flow, chang-

[1] If the underlying variables are continuous, then we redefine entropy as follows. Assume each variable comes with a base measure λ, like the Lebesgue or counting measure. Then define $\operatorname{H}_\mu(X) := \mathbb{E}_\mu[\log \frac{\mathrm{d}\mu(X)}{\mathrm{d}\lambda_X}]$, where $\frac{\mathrm{d}\mu}{\mathrm{d}\lambda}$ is the Radon-Nikodym derivative of $\mu(X)$ with respect to λ_X.

ing the parameters associated with a in proportion to the degree of control $\chi(a)$.

Before we can run Algorithm 1 we must select the procedure Refocus that provides attention and control masks. We focus primarily on the case where Refocus chooses non-deterministically from a fixed finite set of attention-control mask pairs $\mathbf{F} = \{(\varphi_i, \chi_i)\}_{i=1}^{n}$, which we call *foci*.

The ODE described in the final line may be approximated with an inner loop running an iterative gradient-based optimization algorithm. Alternatively, if Refocus produces small χ, then it is well-approximated by a single gradient descent step of size χ. At the other extreme, if χ is infinite in every component, then we typically expect the final line to reduce to

$$\boldsymbol{\theta}^{(t+1)} \leftarrow \arg\min_{\boldsymbol{\theta}} \langle\!\langle \varphi \odot \mathcal{M}(\boldsymbol{\theta}) \rangle\!\rangle\,, \tag{4}$$

at least in many cases of interest. For example:

Proposition 1. *If $\mathcal{M}(\Theta)$ is an unweighted parametric PDG whose parameterizations $\mathbb{P}$ are either constant or unconditional and log-concave and $\boldsymbol{\beta} \geq \gamma\boldsymbol{\alpha}$, the map $\boldsymbol{\theta} \mapsto \langle\!\langle \mathcal{M}(\boldsymbol{\theta}), \boldsymbol{\alpha}, \boldsymbol{\beta} \rangle\!\rangle_\gamma$ is convex.*[2]

link to proof

To our surprise (and contradicting a prior version of this paper), we have found that inconsistency is not always convex in the parameters of *conditional* probability distributions—yet we conjecture that (4) holds nevertheless. Determining whether or not $\langle\!\langle \mathcal{M} + p(Y|X) \rangle\!\rangle$ is quasi-convex in p remains a key open question in the theory of PDGs. In the remaining sections, we give a sample of some historically important algorithms that are instances of LIR.

4 Unifying Classical Algorithms as Instances of LIR

4.1 The Classification Setting

Consider a parametric classifier $p_\theta(Y|X)$, perhaps arising from a neural network whose final layer is a softmax, and suppose we also have a labeled example (x, y). Let's capture this situation with a parametric PDG. Since $\mathcal{V}Y$ is a discrete label space space, we regard labels as unconditional distributions over Y, viewing "hard" labels such as our $y \in \mathcal{V}Y$ as vertices of this simplex, while also allowing for label smoothing (see Müller et al., 2019, for an overview). Doing the same for x may be intractable, but fortunately is not necessary; in the typical case where the space of inputs $\mathcal{V}X$ is itself a manifold (e.g., color images in $[0,1]^{W\times H\times 3}$), we can regard a value $x \in \mathcal{V}X$ as parameterizing a deterministic unconditional probability over X. To communicate this deterministic parameterization visually, we use a double-headed arrow, writing $\overset{x}{\twoheadrightarrow}\boxed{X}$. Combining the parametric classifier $p_\theta(Y|X)$ with the sample (x, y), we get a parametric PDG $\mathcal{M}(x, y, \theta) :=$

$$\begin{tikzcd} {} \arrow[r, two heads, "x"] & \boxed{X} \arrow[r, "p_\theta"] & \boxed{Y} & {} \arrow[l, "y"'] \end{tikzcd} \quad\cong\quad \begin{tikzcd} {} \arrow[r, two heads, "\theta"] & \boxed{\Theta} \arrow[dr, dash] & & & \\ & & {} \arrow[r, "p"] & \boxed{Y} & {} \arrow[l, "y"'] \\ {} \arrow[r, two heads, "x"] & \boxed{X} \arrow[ur, dash] & & & \end{tikzcd} \tag{5}$$

whose observational inconsistency is $-\log p_\theta(y|x)$, the standard training objective for such a classifier (Richardson, 2022). Each cpd plays major role in this inconsistency. What happens when we resolve this it with control over different arcs?

- Adjusting θ amounts to **training** the network in the standard way. In this case, the value χ of the control mask corresponds roughly to the product of the learning rate and the number of optimization iterations spent on the example (x, y).
- Adjusting y amounts to **inference**: it adjusts y to match distribution $p_\theta(Y|x)$.
- Adjusting x amounts to **forming an adversarial example**: it makes small changes to the input until the (fixed) network gives it the desired label.

Some readers may find the last point surprising. While adversarial examples (Goodfellow et al., 2014) are often presented as weird quirk of neural networks, they are, together with training and inference, one of the three basic resolutions ways of resolving this simple inconsistency. The sustained interest of the machine learning community on adversarial examples may appear to be a cultural phenomenon, but at a mathematical level, it is no accident (Shafahi et al., 2018). At this level of abstraction, there is no difference between the network parameters and inputs. Making the parameterization of p explicit and adding L2 regularization, the symmetry becomes striking (see Figure 1, right).

With minor modifications to $\mathcal{M}(x, y, \theta)$, we can capture variations of training procedures.

Stochastic Gradient Descent (SGD). Take the mutable state to be the classifier p as before. Define Refresh so that it draws a batch of samples $\{(x_i, y_i)\}_{i=1}^{m}$, and returns a PDG with a single arc describing their emperical distribution $d(X, Y)$, and let Refocus be such that $\varphi(d) = \infty$ (reflecting high confidence in the data). If $\eta := \chi(p)\varphi(p)$ is small, then LIR reduces to SGD with batch size m and learning rate η.

Adversarial training. Suppose we want to slightly alter x to obtain x' that is classified as y' instead of y. Adding x' and y' to $\mathcal{M}$ and relaxing $\mathbb{P}_x$ to be a Gaussian centered x rather than a point mass, we get the PPDG on the left of Figure 1. A LIR iteration whose focus consists of the edges marked in green (with

[2] All proofs can be found in the appendix.

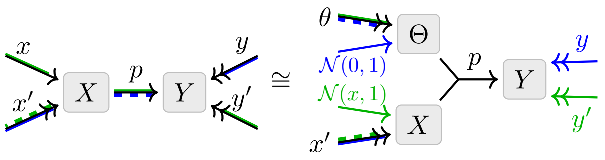


Figure 1: Two equivalent illustrations of adversarial training.

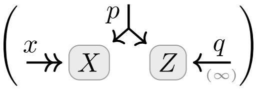
control over the dashed green edge) is an adversarial attack with Euclidean distance (Biggio et al., 2013). The blue focus, by contrast, "patches" the adversarial example by adjusting the model parameters to again classify it correctly. Thus, LIR that alternates between the two and refreshes (x, y, y') is adversarial training, a standard defense to adversarial attacks (Goodfellow et al., 2014). Thus, it is just as sensible to train the inputs, as the network (Kishore et al., 2021).

4.2 The EM Algorithm and VAEs

Suppose we have a latent variable model $p_\theta(Z, X)$ describing the joint distribution of an observable variable X and a latent one Z. Given observations of the form $X{=}x$, how can we learn model parameters θ despite the missing data z? There is an overwhelmingly standard answer that has independently arisen in many different domains: the EM algorithm (Dempster et al., 1977; McLachlan and Krishnan, 2008). The EM algorithm iteratively improves an estimate of θ as follows: first estimate an **E**xpectation over the missing values z (fixing the current estimate θ), and then perform a **M**aximization over θ to update the parameters; altogether this amounts to computing:

$$\theta^{(t+1)}_{\text{EM}} := \arg\max_{\theta} \; \mathbb{E}_{z\sim p(Z|x,\theta^{(t)}_{\text{EM}})}\big[\, \log p(x, z|\theta) \,\big].$$

(unproven!)

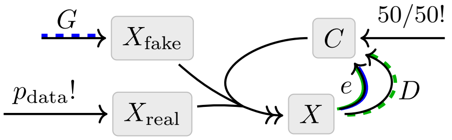
Proposition 2. LIR  *in which* Refocus *re-samples x and alternates between full control of p and q implements EM, in that $\theta^{(t)}_{EM} = \theta^{(2t)}_{LIR}$.*[3]

This result is closely related to one due to Neal and Hinton (1998), who view it as an intuitive explanation of why the EM algorithm works. Indeed, it is obvious in this form that every adjustment reduces the overall inconsistency, and that the M step need not fully solve the maximization problem, but merely make progress.

Variational inference as LIR. This form of the EM algorithm is closely related to variational inference. Indeed, the inconsistency of the PDG in Proposition 2 is equal to the evidence lower bound (ELBO), the optimization target for variational methods. Furthermore, allowing q to depend on x and breaking up $p(X, Z) = p(Z)d(X|Z)$ into two separate arcs yields a PDG containing the data of a variational autoencoder, or VAE (Kingma and Welling, 2014), and the resulting inconsistency is the variant of the ELBO used to train VAEs (Richardson, 2022). It follows that LIR where we control the encoder and decoder (but not the prior) and attend to various mini-batches of samples, trains a VAE with stochastic gradient descent. *Mean-field* variational methods are ones where the latents $Z = (Z_i)_{i=1}^n$ are real-valude, and the variational family is restricted to those of the form $q(Z) = \prod_{i=1}^n q_i(Z)$.

4.3 Generative Adversarial Networks

LIR can also capture the procedure for training Generative Adversarial Networks, or GANs (Goodfellow et al., 2020). The goal is to train a generator network G to generate images that cannot be distinguished from real ones. Formally, let X be either a generated image $X_{\text{fake}} \sim G$ or one from a dataset $X_{\text{real}} \sim p_{\text{data}}$, based on the outcome of a fair coin C. A discriminator D then predicts the outcome C of the coin flip from the image X. The only additional ingredient we will need that is not explicit in the original formulation is a belief e that the coin is equally likely heads as tails given X—intuitively, representing the goal of the generator and the opposite of the goal of the discriminator. All of these pieces of probabilistic information can be summarized by the PDG below.

$m(\Theta) :=$ 

GAN training is often written as a 2-player game $\min_G \max_D \mathcal{L}^{\text{GAN}}(G, D)$, where

$$\mathcal{L}^{\text{GAN}}(G, D) = \mathbb{E}_{x\sim p_{\text{data}}}[\log D(x)] + \mathbb{E}_{x'\sim G}[\log(1-D(x'))].$$

The Discriminator's Focus. The discriminator has full control over D, and attends to everything but e. That inconsistency of this PDG is what might be called the discriminator's objective: the expected KL divergence from D to the optimal discriminator. If D also disbelieves that any image is equally likely to be fake as real (by chosing $\varphi(e) = -1$), then the inconsistency becomes $-\mathcal{L}^{\text{GAN}}$.

The Generator's Focus. The generator has control over G. If it ignores D attends only to e, the inconsistency is the Jensen-Shannon Divergence between G

[3] The result can also be readily adapted to an entire dataset by replacing x with a high confidence empirical distribution, or mini-batched as in Section 4.1.

and p_{data}. If the generator also disbelieves the discriminator D (i.e., $\varphi(D) = -1$), then the inconsistency becomes $+\mathcal{L}^{\text{GAN}}$.

Standard practice is to use small $\chi(G)$ and large $\chi(D)$.

4.4 Message Passing

A *factor graph* over a set of variables $\mathcal{X}$ is a set of factors $\Phi = \{\phi_a : \mathbf{X}_a \to \mathbb{R}_{\geq 0}\}_{a\in\mathcal{A}}$, where each $\mathbf{X}_a \subseteq \mathcal{X}$ is called the *scope* of a. Conversely, for $X \in \mathcal{X}$, let $\partial X := \{a \in \mathcal{A} : X \in \mathbf{X}_a\}$ be the set of factors with X in scope. The factor graph Φ specifies a distribution $\Pr_\Phi(\mathcal{X}) \propto \prod_a \phi_a(\mathbf{X}_a)$, and corresponds to a PDG

$$\boldsymbol{m}_\Phi = \left\{ \xrightarrow[(\alpha,\,\beta=1)]{\propto \phi_a} \boxed{\mathbf{X}_a} \right\}_{a\in\mathcal{A}}$$

that specifies the same distribution $\Pr_\Phi(\mathcal{X})$ when $\gamma{=}1$.

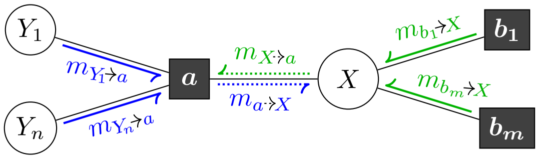
Sum-product belief propogation (Kschischang et al., 2001) aims to approximate marginals of $\Pr_\Phi$ with only local computations: messages sent between factors and the variables they have in scope. Its state consists "messages" $m_{X\to a}$ and $m_{a\to X}$ both (unnormalized) distributions over X, for each variable X and factor $a \in \partial X$ adjacent to it. After initialization, belief propogation repeatedly recomputes:

$$m_{X\to a}(x) :\propto \prod_{b\in\partial X\setminus a} m_{b\to X}(x) \tag{6}$$

$$m_{a\to X}(x) :\propto \sum_{\mathbf{y}\in\mathcal{V}(\mathbf{X}_a\setminus X)} \phi_a(x,\mathbf{y}) \prod_{Y\in\mathbf{X}_a\setminus X} m_{Y\to a}(Y(\mathbf{y})), \tag{7}$$

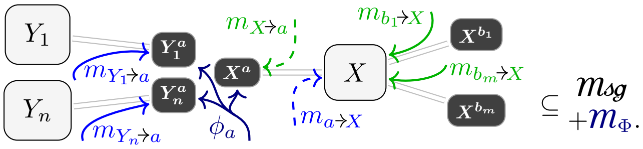
where $Y(\mathbf{y})$ is the value of Y in the joint setting $\mathbf{y}$. Observe that every calculation is a (marginal of) a product of factors, and thus amounts to inference in a "local" factor graph.

The usual schematic illustration people draw to depict messages moving between variables and factors according to Equations (6) and (7) is as follows:



,

While this standard diagram is only a schematic, simply writing down the messages as unconditional probability distributions, we get a PDG $\mathcal{M}\mathit{sg}$ that can be made to look very similar. Formally, a variable X^a for every pair (X, a) with $X \in \mathbf{X}_a$ along with edges asserting that $X^a = X$, we obtain the equivalent PDG



Indeed, it can be shown that (6,7) minimize inconsistency of the dotted components in their appropriate contexts (shown in green and blue above, and formalized in Appendix A).

Finally, variable marginals $\{b_X\}_{X\in\mathcal{X}}$, which we regard as another PDG, $\mathcal{B}$, are computed from the messages according to $b_X(x) \propto \prod_{a\in\partial X} m_{a\to X}(x)$.

Proposition 3. *If* Refocus *selects a focus nondeterministically from* $\{a{\to}X, X{\to}a, X\}_{X\in\mathcal{X}, a\in\partial X}$ *(see illustration above; details in Appendix A), then the possible runs of* $\text{LIR}(\boldsymbol{m}_\Phi, \mathcal{M}\mathit{sg} + \mathcal{B})$ *are precisely those of BP for different message schedules.*

link to proof

The same construction works for general cluster graphs, and we suspect the same is true of the many other message passing algorithms that are generated from the α-divergences (Minka, 2005), because of the close relationship those divergences have with simple PDGs (Richardson, 2022, §5).

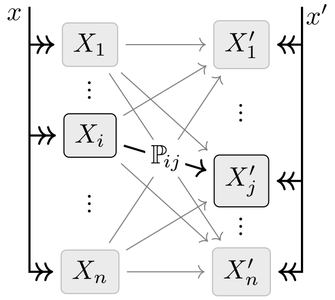
4.5 Transformer Layers

The key innovation of the transformer architecture (Vaswani et al., 2017), the basis of modern language models, is that of (scaled dot-product) (masked) *self-attention.* This notion of attention can be viewed as an instance of our framework.

Suppose that we are looking at a sequence of tokens n, whose current representations at some layer of the model are the vectors $x_1, \ldots, x_n \in \mathbb{R}^d$. The transformation is paramterized by three matricies $W_K, W_Q, W_V \in \mathbb{R}^{d\times d}$, that generate key $k_i = W_K x_i$, query $q_i = W_Q x_i$, and value $v_i = W_V x_i$ representation vectors of each x_i. The result of the transformation are the processed vectors $x'_1, \ldots x'_n$ where $x'_j = \sum_{i=1}^n \alpha_{i|j} v_i$, and $\alpha_{i|j} = \text{softmax}_i(\langle k_i, q_j\rangle)$.

Consider a PDG with variables $\mathcal{X} = \{X_i\}_{i=1}^n \cup \{X'_i\}_{i=1}^n$. We start with a bipartite graph: for each $(i,j) \in [n]^2$, we add an arc with isotropic Gaussian cpd $\mathbb{P}_{ij} = \mathcal{N}(X'_j \mid v_i, I)$; finally, add two additional arcs specifying the joint values of x and x'. If the cpd has attention $\varphi(ij) = \exp\langle k_i, q_j\rangle$.

$\mathcal{M}(K, Q, V, x, x') :=$



Proposition 4. *LIR($\mathcal{M}$) in which attention is set to $\varphi_{ij} = \exp\langle k_i, q_j\rangle$, when controlling x', leads to inference in the transformer layer, i.e., $x'_j = \sum_i \alpha_{i|j} v_i$.*

Notably, this model forces softmax normalization to be row-wise. Controlling the parameters K, Q, V trains the layer, in a sense, but we still do not know whether or not this is equivalent to the standard way of training.

4.6 Generative Flow Networks

this is the last subsection that needs a major rework; I've removed it for now.

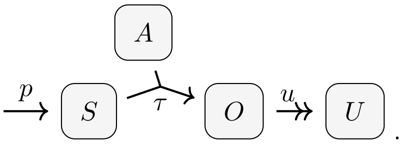
4.7 Decision Making

In this final illustration of the expressive power of LIR, we return to psychological models of agents, as we show how some standard decision rules can be viewed as inconsistency minimization. PDG inconsistencies can represent expected costs (Richardson, 2024, §4, §6), and thus can jointly represent an agent's probabilistic beliefs and utilities. What does it mean to make decisions to as to minimize inconsistency in this context?

Let's formalize the tranditional setup of decision theory with a PDG. Suppose we are trying to choose an action (a setting of the variable A), S is he a variable representing the state of the world (which we are uncertain about), and O is a variable representing the final outcome To complete the standard picture, let's further assume that we have

1. some understanding of how our action A and the state S determine the outcome O, say in the form of a cpd $\tau(O \mid S, A)$,
2. a belief $p(S)$ about the current state of the world, and
3. a utility function $u : \mathcal{V}O \to \mathbb{R}$ on possible outcomes.

It is easy to encode this information in a PDG:



What's missing is the idea that a higher numerical utility is "better" than a lower one. For this, we can add a "soft constraint" (see Richardson, 2024, §4.2.2) that is violated less at higher utilities than lower ones. At a technical level, recall that this means (1) including the variable $\mathtt{T}$ that can technically take on values $\mathcal{V}\mathtt{T} = \{\mathtt{f}, \mathtt{t}\}$, yet happens to always on the value $\mathtt{t}$, and (2) adding a cpd $b(\mathtt{T} \mid U)$ encoding a constraint violation that is more serious the lower the value of U. Let $\mathcal{M}_{(p,\tau,u,b)}$, or simply $\mathcal{M}$, represent the PDG above augmented with such a soft constraint b.

Intuitively, imagine that there is a part of you that you cannot control, which we will call "Faith". Faith engages in wishful thinking: she disbelieves outcomes that are undesirable, creating epistemic conflict if she sees outcomes of low utility. (Technically, Faith refers to the sub-PDG consisting of u, b, and $\mathtt{T}{=}\mathtt{t}$.) If you have no control over Faith, but still have confidence and pay attention to her, then it turns out that selecting actions to minimize your combined inconsistency is a decision rule that interpolates between

(a) maximizing expected utility (when $\beta_p \gg \beta_b$), and
(b) maximizing maximum utility (when $\beta_p \ll \beta_b$),

where β_p is your confidence in your prior probabilistic belief $p(S)$, and β_b is the confidence of the soft constraint b (i.e., your "degree of faith").

Proposition 5. *Suppose $b(\mathtt{T}{=}\mathtt{t} \mid U{=}u) = k \cdot \exp(u)$ for some constant k.*

link to proof

1. *If $\beta_p = \infty$ and $\beta_b < \infty$, then the action(s) $a \in \mathcal{V}A$ that minimize the inconsistency are those that maximize expected utility. Formally, for all $\gamma \geq 0$,*

$$\arg\min_{a\in\mathcal{V}A} \langle\!\langle \mathcal{M} + A{=}a \rangle\!\rangle_\gamma = \arg\max_{a\in\mathcal{V}A} \mathop{\mathbb{E}}_{\substack{s\sim p\\ o\sim\tau|s,a}} \big[u(o)\big].$$

2. *If $\beta_p < \infty$ and $\beta_b = \infty$, then the action(s) $a \in \mathcal{V}A$ that minimize overall inconsistency are those that can lead to the best possible outcome, i.e.,*

$$\arg\min_{a\in\mathcal{V}A} \langle\!\langle \mathcal{M} + A{=}a \rangle\!\rangle_\gamma = \arg\max_{a\in\mathcal{V}A} \max_{s\in\mathcal{V}S} \mathop{\mathbb{E}}_{o\sim\tau|s,a} [u(o)].$$

We suspect that it may also be possible to implement other decision rules such as minimax or maximin. Like the GAN objective however, these decision rules look like two-player games, and so will likely require two different focuses of LIR, rather than just one. We leave a careful exploration of this avenue to future work.

We conclude our discussion of decision theory with a high-level observation. One's preferences, beliefs, and actions can be jointly modeled with a PDG. When that PDG is inconsistent (i.e., there is a conflict between your preferences, beliefs, and actions), there are, in principle, three possible resolutions. You can resolve the inconsistency by changing your action, which amounts to maximizing expected utility (Proposition 5.1); this is thought of as the rational approach. Alternatively, you can change your preferences so as to become content with your current situation, which is perhaps a more zen perspective. Observe that these two resolutions to the internal conflict are two halves of the famous Serenity Prayer (Niebuhr, 1933):

> *Oh, God, give us the courage to change what must be altered, serenity to accept what can not be helped, and insight to know the one from the other.*

That "insight" amounts to the choice of the control mask χ. There is, of course, a third way to resolve the inconsistency: change beliefs $p(S)$ about the state of the world (i.e., wishful thinking.) However, such a strategy may lead to more inconsistency in the future, making it undesirable and ineffective. This example clearly shows that there can be good reasons for restricting control χ beyond computational savings or the limits of one's power.

5 Synthetic Experiments

Let's start writing about experiments!

6 Conclusion

References

Biggio, B., Corona, I., Maiorca, D., Nelson, B., Š rndić, N., Laskov, P., Giacinto, G., and Roli, F. (2013). Evasion attacks against machine learning at test time. In *Advanced Information Systems Engineering*, pages 387–402. Springer Berlin Heidelberg.

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society: series B (methodological)*, 39(1):1–22.

Festinger, L. (1962). Cognitive dissonance. *Scientific American*, 207(4):93–106.

Friston, K. (2005). A theory of cortical responses. *Philosophical transactions of the Royal Society B: Biological sciences*, 360(1456):815–836.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2020). Generative adversarial networks. *Communications of the ACM*, 63(11):139–144.

Goodfellow, I. J., Shlens, J., and Szegedy, C. (2014). Explaining and harnessing adversarial examples.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. (1999). An introduction to variational methods for graphical models. *Machine learning*, 37:183–233.

Kingma, D. P. and Welling, M. (2014). Auto-encoding variational bayes. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Kishore, V., Chen, X., Wang, Y., Li, B., and Weinberger, K. Q. (2021). Fixed neural network steganography: Train the images, not the network. In *International Conference on Learning Representations*.

Kschischang, F. R., Frey, B. J., and Loeliger, H.-A. (2001). Factor graphs and the sum-product algorithm. *IEEE Transactions on information theory*, 47(2):498–519.

McLachlan, G. J. and Krishnan, T. (2008). *The EM algorithm and extensions*. John Wiley & Sons.

Minka, T. (2005). Divergence measures and message passing. Technical Report MSR-TR-2005–173, Microsoft Research, Cambridge, U.K.

Müller, R., Kornblith, S., and Hinton, G. E. (2019). When does label smoothing help? *Advances in neural information processing systems*, 32.

Neal, R. M. and Hinton, G. E. (1998). A view of the em algorithm that justifies incremental, sparse, and other variants. In *Learning in Graphical Models*, pages 355–368. Springer.

Niebuhr, R. (1933). page 1.

Rao, R. P. and Ballard, D. H. (1999). Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature neuroscience*, 2(1):79–87.

Richardson, O. E. (2022). Loss as the inconsistency of a probabilistic dependency graph: Choose your model, not your loss function. *AISTATS '22*, 151.

Richardson, O. E. (2024). *A Unified Theory of Probabilistic Modeling, Dependence, and Inconsistency*. PhD thesis.

Richardson, O. E. and Halpern, J. Y. (2021). Probabilistic dependency graphs. *AAAI '21*.

Richardson, O. E., Halpern, J. Y., and Sa, C. D. (2023). Inference in probabilistic dependency graphs. *UAI '23*.

Richardson, O. E., Peters, S., and Halpern, J. Y. (2024). Qualitative mechanism independence. In Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., and Zhang, C., editors, *Advances in Neural Information Processing Systems*, volume 37, pages 69471–69504. Curran Associates, Inc.

Shafahi, A., Huang, W. R., Studer, C., Feizi, S., and Goldstein, T. (2018). Are adversarial examples inevitable? *arXiv preprint arXiv:1809.02104*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

A Details on Belief Propogation

We now define the views. Modulo a small subtlety, the following is essentially true: (6) selects $C_{X\to a} := \{m_{X\to a}\}$ so as to minimize 1-inconsistency in context $A_{X\to a} := \{m_{b\to X}\}_{b\in\partial X\setminus a} \cup \{m_{X\to a}\}$, while (7) selects $C_{a\to X} := \{m_{a\to X}\}$ so as to minimize the 1-inconsistency in context $A_{a\to X} := \{\phi_a, m_{a\to X}\} \cup \{m_{Y\to a}\}_{Y\in\mathbf{X}_a\setminus X}$.

The only wrinkle is that we do not attend to the *structural* aspect of the edges e that we're updating; i.e., we must select φ to effectively set $\alpha_e = 0$. Intuitively: although all of the input messages summarize causal information, we're trying to capture that information with a distribution. Thus, it's not appropriate to attend to the causal structure of the edges that we're modifying. Thus, for each $f \in \bigcup_{a\in\mathcal{A}, X\in\mathbf{X}_a}\{a\to X, X\to a, X\}$, define a focus $(\varphi_f, \chi_f) \in \mathbf{F}$ according to

$$\varphi_f(a) := \begin{cases} \binom{1}{1} & \text{if } a \in A_f \setminus C_f \\ \binom{1}{0} & \text{if } a \in C_f \\ \binom{0}{0} & \text{otherwise} \end{cases}, \qquad \chi_f(a) := \begin{cases} \infty & \text{if } a \in C_v \\ 0 & \text{otherwise.} \end{cases}$$

where $\binom{\phi_1}{\phi_2}$ scales β_a by ϕ_1 and α_a by ϕ_2. With these definitions, Proposition 3 follows easily.

B Proofs

First, some extra details for Proposition 1. By parameteriations $\mathbb{P}$ log-concave, we mean that, for every $a \in \mathcal{A}$, and $(s,t) \in \mathcal{V}(S_a, T_a)$, the function

$$\theta \mapsto -\log \mathbb{P}_a^{\theta}(T_a = t \mid S_a = a) \quad : \Theta_a \to [0, \infty]$$

is convex. This is true for many families of distributions of interest. For example, if S_a, T_a is discrete, and the cpd is parameterized by stochastic matrices $\mathbf{P} = [p_{s,t}] \in [0,1]^{\mathcal{V}(S_a,T_a)}$, then

$$-\log \mathbb{P}_a^{\mathbf{P}}(T_a = t | S_a = s) = -\log(p_{s,t})$$

which is clearly convex in $\mathbf{P}$.

To take another example: if $\mathbb{P}_a$ is linear Gaussian, i.e., $\mathbb{P}_a(T|S) = \mathcal{N}(T|\mathbf{A}s + b, \sigma^2)$, parameterized by $(\mathbf{A}, b, 1/\sigma^2)$, then

$$-\log \mathbb{P}_a^{(\mathbf{A},b,\sigma^2)}(t|s) = -\frac{1}{2}\log\frac{2\pi}{\sigma^2} + \frac{1}{2}\left(\frac{t - \mathbf{A}s + b}{\sigma}\right)^2$$

which is convex in $(\mathbf{A}, b, \frac{1}{\sigma^2})$. Now, for the proof.

Proposition 1. *If $\mathcal{M}(\Theta)$ is an unweighted parametric PDG whose parameterizations $\mathbb{P}$ are either constant or unconditional and log-concave and $\boldsymbol{\beta} \geq \gamma\boldsymbol{\alpha}$, the map $\boldsymbol{\theta} \mapsto \langle\!\langle \mathcal{M}(\boldsymbol{\theta}), \boldsymbol{\alpha}, \boldsymbol{\beta} \rangle\!\rangle_\gamma$ is convex.*

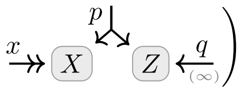
Proof. By definition,

$$\langle\!\langle \varphi \odot (\mathit{Ctx} + \mathcal{M}(\theta)) \rangle\!\rangle_\gamma = \inf_\mu \Big\{ \mathit{SDef}_{\mathit{Ctx}} + \mathit{OInc}_{\mathit{Ctx}}(\mu) + \mathit{SDef}_{\mathcal{M}(\theta)}(\mu) + \mathit{OInc}_{\mathcal{M}(\theta)}(\mu) \Big\}.$$

Only the final term actually depends on θ, though. Let $F(\mu)$ capture the first three terms. For all of our examples, and indeed, if γ is chosen small enough, it will be convex in μ (Richardson and Halpern, 2021, from the proof of Proposition 3.2). Then we have

$$\begin{aligned}\langle\!\langle \varphi \odot (\mathit{Ctx} + \mathcal{M}(\theta)) \rangle\!\rangle_\gamma &= \inf_\mu \left(F(\mu) + \mathop{\mathbb{E}}_\mu \left[\sum_{S \xrightarrow{a} T} \beta_a \log \frac{\mu(T|S)}{\mathbb{P}_a(T|S)} \right] \right) \\ &= \inf_\mu \left(F(\mu) + \mathop{\mathbb{E}}_\mu \left[\sum_{S \xrightarrow{a} T} \beta_a \log \frac{\mu(T|S)}{\lambda(T|S)} \right] + \mathop{\mathbb{E}}_\mu \left[\sum_{S \xrightarrow{a} T} \beta_a \log \frac{\lambda(T|S)}{\mathbb{P}_a(T|S)} \right] \right)\end{aligned}$$

The second term is then entropy (relative to the base distribution), which is convex in μ. The first term, $F(\mu)$, is convex in μ as well, and neither depend on θ. The final term is linear in μ. Since $\mathbb{P}$ is log-convex in θ, the $\log \frac{\lambda(t|s)}{\mathbb{P}_a(t|s)}$ convex in θ, and so that third term is a conic combination of expectations that are all convex, and hence itself convex in θ. Thus, the sum of all three terms in the infemum is jointly convex in θ and in μ. Taking an infemum over μ pointwise, the result is still convex in θ. □

Proposition 2. LIR 

in which Refocus *re-samples x and alternates between full control of p and q implements EM, in that $\theta_{EM}^{(t)} = \theta_{LIR}^{(2t)}$.*

Proposition 3. *If* Refocus *selects a focus non-deterministically from $\{a \to X, X \to a, X\}_{X \in \mathcal{X}, a \in \partial X}$ (see illustration above; details in Appendix A), then the possible runs of* $\mathrm{LIR}(\mathcal{M}_\Phi, \mathcal{M}_{\mathit{sg}} + \mathcal{B})$ *are precisely those of BP for different message schedules.*

Proof. When $\gamma = 1$, and $\alpha, \beta = 1$ for all of the input factors, then the optimal distribution μ^* that realizes the infemum is just the product of factors. It follows that any distribution that has those marginals will minimize the observational inconsistency.

The different orders that the (6), and (7) can be ordered for different adjacent pairs (a, X) correspond to both the message passing schedules, and to the possible view selections of LIR. □

Proposition 5. *Suppose $b(\mathtt{T}{=}\mathtt{t} \mid U{=}u) = k \cdot \exp(u)$ for some constant k.*

1. *If $\beta_p = \infty$ and $\beta_b < \infty$, then the action(s) $a \in \mathcal{V}A$ that minimize the inconsistency are those that maximize expected utility. Formally, for all $\gamma \geq 0$,*

$$\operatorname*{arg\,min}_{a \in \mathcal{V}A} \langle\!\langle \mathcal{M} + A{=}a \rangle\!\rangle_\gamma = \operatorname*{arg\,max}_{a \in \mathcal{V}A} \mathop{\mathbb{E}}_{\substack{s \sim p \\ o \sim \tau|s,a}} \big[u(o)\big].$$

2. *If $\beta_p < \infty$ and $\beta_b = \infty$, then the action(s) $a \in \mathcal{V}A$ that minimize overall inconsistency are those that can lead to the best possible outcome, i.e.,*

$$\operatorname*{arg\,min}_{a \in \mathcal{V}A} \langle\!\langle \mathcal{M} + A{=}a \rangle\!\rangle_\gamma = \operatorname*{arg\,max}_{a \in \mathcal{V}A} \max_{s \in \mathcal{V}S} \mathop{\mathbb{E}}_{o \sim \tau|s,a} [u(o)].$$

Proof. Since the choice a is deterministic, the value of A is determined; likewise the value of $\mathtt{T}$ is also fixed. Similarly, as u is a deterministic function, the value of U is determined according to u. Thus we need only consider distributions $\mu(S, O)$ in our minimization; the other variables can be found as a function of these. However, we also know that $\mu(O \mid S) = \tau(O|S, A{=}a)$ since it is given with high confidence. Therefore it suffices to restrict our search to distributions over the variable S.

To simplify notation, let $EU(s, a) := \mathbb{E}_{o \sim \tau(O|s,a)}[u(o)] + \log k$ be the expected utility of an action, shifted by the constant $\log k$. Note that

$$\log \frac{1}{b(\mathtt{T}{=}\mathtt{t} \mid U{=}u(o))} = -\log(k \cdot \exp(u(o))) = -u(o) - \log k,$$

which in expectation over $\tau(O|s, a)$, is $-EU(s, a)$. With this in mind, we calculate:

$$\begin{aligned}
\langle\!\langle \mathcal{M}_{p,\tau,u,b} + A{=}a \rangle\!\rangle_\gamma &= \inf_{\mu(S)} \beta_p \mathop{\mathbb{E}}_{s \sim \mu} \Big[\log \frac{\mu(s)}{p(s)} + \frac{\beta_b}{\beta_p} \mathop{\mathbb{E}}_{o \sim \tau|a,s} \Big[\log \frac{1}{b(\mathtt{T}{=}\mathtt{t} \mid U{=}u(o))} \Big] \Big] \\
&= \inf_{\mu(S)} \beta_p \mathop{\mathbb{E}}_{s \sim \mu} \Big[\log \frac{\mu(s)}{p(s)} + \log \circ \exp \Big(-\frac{\beta_b}{\beta_p} EU(s, a) \Big) \\
&= \inf_{\mu(S)} \beta_p \mathop{\mathbb{E}}_{s \sim \mu} \Big[\log \frac{\mu(s)}{1} \cdot \frac{\exp(-\frac{\beta_b}{\beta_p} EU(s, a))}{p(s)} \cdot \frac{Z}{Z} \Big] \\
&= \inf_{\mu(S)} \beta_p \mathop{\mathbb{E}}_{s \sim \mu} \Big[\log \frac{\mu(s)}{1} \cdot \frac{Z}{p(s) \cdot \exp(\frac{\beta_b}{\beta_p} EU(s, a))} \cdot \frac{1}{Z} \Big] \qquad (8)
\end{aligned}$$

At this point, we can use the same trick used repeatedly in the proving results of Richardson (2022): take Z to be the normalization constant needed to regard the middle fraction as the inverse of a probability distribution ν. Once we do so, we are left with an infimum over a KL divergence $\boldsymbol{D}(\mu \parallel \nu)$ plus the expectation of a constant:

$$\langle\!\langle \mathcal{M}_{p,\tau,u,b} + A{=}a \rangle\!\rangle_\gamma = \beta_p \log \frac{1}{Z} = -\log \sum_s p(s) \exp \Big(+ \frac{\beta_b}{\beta_p} EU(s, a) \Big).$$

We now look at the two extreme cases.

When $\beta_p \to \infty$, then the ratio $\frac{\beta_b}{\beta_p}$ becomes small, and we can use the fact that $\exp(\epsilon) \approx 1 + \epsilon$ for small ϵ to find that the inconsistency of interest is approximately

$$\approx -\beta_p \log \sum_s p(s) \Big[1 + \frac{\beta_b}{\beta_p} EU(s, a) \Big] = -\beta_p \log(1 + \frac{\beta_b}{\beta_p} EU(s, a)) \approx -\beta_b EU(s, a)$$

Alternatively, more directly, when $\beta_p = \infty$, the optimal distribution must be $\mu = p$, and so the inconsistency is immediately $-\mathbb{E}_{s\sim p}[\beta_b EU(s,a)]$. Either way, minimizing this quantity amounts to maximizing espected utility, as the two differ by a negative affine transformation.

At the other extreme, when $\beta_b \to \infty$, we can write our expression in terms of $\mathrm{LSE}\{x_1,\ldots,x_n\} := \log\sum_{i=1}^n \exp(x_i)$ (LogSumExp) which is a smooth approximation to a max. (In a moment, we will negate its arguments and the final output, using it as an approximation to a min, instead.) Picking back up from (8) and letting $t := \frac{\beta_b}{\beta_p}$, we find

$$\langle\!\langle \boldsymbol{m}_{p,\tau,u,b} + A{=}a \rangle\!\rangle_\gamma = -\beta_b \cdot \frac{-1}{t} \operatorname*{LSE}_{s\in\mathcal{V}S}\Big[-t\cdot\Big(\frac{1}{t}\log\frac{1}{p(s)} - EU(s,a)\Big)\Big].$$

Using the standard fact[4] that

$$\min_{i\in[n]} x_i - \frac{1}{t}\log n \le \frac{-1}{t}\operatorname*{LSE}_{i\in[n]}(-tx_i) < \min_{i\in[n]} x_i,$$

we find that, in our case,

$$M - \frac{1}{t}\log|\mathcal{V}S| \quad\le\quad \frac{1}{\beta_b}\langle\!\langle \boldsymbol{m}_{p,\tau,u,b} + A{=}a \rangle\!\rangle_\gamma \quad\le\quad M$$

where $M := \min_{s\in\mathcal{V}S}(-EU(s,a) + \frac{1}{t}\log\frac{1}{p(s)})$. In particular, when $\beta_b \to \infty$, meaning $t\to\infty$, the gap between the upper and lower bounds shrinks to zero, and the resulting inconsistency becomes $-\min_{s\in\mathcal{V}S}(-EU(s,a)) = \max_{s\in\mathcal{V}S)} EU(s,a)$, proving the result. □

[4] Letting $m := \min_i x_i$, observe that, for all $t > 0$, we have $\exp(-tm) \le \sum_i \exp(-tx_i) \le n\exp(-tm)$. Apply a logarithm and multiply by $-\frac{1}{t}$ to get the promised result.