Inference for Probabilistic Dependency Graphs (Supplementary material)

Oliver E. Richardson[1] **Joseph Y. Halpern**[1] **Christopher De Sa**[1]

[1]Deparment of Computer Science, Cornell University, Ithaca NY 14853

A PROOFS

Proposition 1. *If* $(\mu, \mathbf{u})$ *is a solution to* (5)*, then* $\mu \in [\![\mathcal{M}]\!]_0^*$*, and* $\sum_{(a,s,t)\in\mathcal{VA}} \beta_a u_{a,s,t} = \langle\!\langle \mathcal{M} \rangle\!\rangle_0$.

Proof. Suppose that $(\mu, \mathbf{u})$ is a solution to (5). The exponential cone constraints ensure that, for every $(a, s, t) \in \mathcal{VA}$,

$$u_{a,s,t} \geq \mu(s,t) \log \frac{\mu(s,t)}{\mathbb{P}_a(t|s)\mu(s)}$$

where $\mu(s,t)$ and $\mu(s)$, as usual, are shorthand for $\mu(S_a{=}s, T_a{=}t)$ and $\mu(S_a{=}s)$, respectively.

Suppose, for contradiction, that one of these inequalities is strict at some an index $(a', s', t') \in \mathcal{VA}$ for which $\beta_{a'} > 0$. Explicitly, this means

$$u_{a',s',t'} > \mu(s_0,t_0) \log \frac{\mu(s',t')}{\mathbb{P}_{a'}(t'|s')\mu(s')}.$$

In that case, we can define a vector $\mathbf{u}' = [u'_{a,s,t}]_{(a,s,t)\in\mathcal{VA}}$ which is identical to $\mathbf{u}$, except that at (a', s', t'), it is halfway between the two quantities described as different above. More precisely:

$$u'_{a',s',t'} = \frac{1}{2}u_{a',s',t'} + \frac{1}{2}\log\mu(s',t') \log \frac{\mu(s',t')}{\mathbb{P}_a(t'|s')\mu(s')}.$$

Note that $u'_{a',s',t'} < u_{a',s',t'}$, and also that, by construction, $(\mu, \mathbf{u}')$ also satisfies the constraints of (5). In more detail: for (a', s', t') it doesn't violate the associated exponential cone constraint, as

$$\left(\text{formally:} \quad u'_{a',s',t'} = \frac{1}{2}u_{a',s',t'} + \frac{1}{2}\log\mu(s',t') \log \frac{\mu(s',t')}{\mathbb{P}_{a'}(t'|s')\mu(s')} > \mu(s',t') \log \frac{\mu(s',t')}{\mathbb{P}_{a'}(t'|s')\mu(s')}\right),$$

and $\mathbf{u}'$ remains unchanged at the other indices, and so satisfies the constraints at those indices, becasuse $\mathbf{u}$ does. But now, because $u'_{a',s',t'} < u_{a',s',t'}$, and $\beta_{a'} > 0$, we also have

$$\sum_{(a,s,t)\in\mathcal{VA}} \beta_a u'_{a,s,t} > \sum_{(a,s,t)\in\mathcal{VA}} \beta_a u'_{a,s,t}.$$

Thus the objective value at $(\mu, \mathbf{u}')$ is strictly smaller than the one at $(\mu, \mathbf{u})$, both of which are feasible points. This contradicts the assumption that $(\mu, \mathbf{u})$ is optimal. We therefore conclude that none of these inequalities can be strict at points where $\beta_a > 0$. This can be compactly written as:

$$\forall (a,s,t) \in \mathcal{VA}. \quad \beta_a u_{a,s,t} = \beta_a \mu(s,t) \log \frac{\mu(s,t)}{\mathbb{P}_a(t|s)\mu(s)}$$

$$\implies \sum_{(a,s,t)\in\mathcal{VA}} \beta_a u_{a,s,t} = \sum_{(a,s,t)\in\mathcal{VA}} \beta_a \mu(s,t) \log \frac{\mu(s,t)}{\mathbb{P}_a(t|s)\mu(s)} = \mathit{OInc}_{\mathcal{M}}(\mu).$$

Accepted for the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023).

In other words, the objective of problem (5) at $(\mu, \mathbf{u})$ is equal to the observational incompatibility $OInc_{\mathcal{M}}(\mu)$ of μ with $\mathcal{M}$. And, because $(\mu, \mathbf{u})$ minimizes this value among all joint distributions, μ must be a minimum of $OInc_{\mathcal{M}}$.

More formally: assume for contradiciton that μ is not a minimizer of $OInc_{\mathcal{M}}$. Then there would be some other distribution μ' for which $OInc_{\mathcal{M}}(\mu') < OInc_{\mathcal{M}}(\mu)$. Let $\mathbf{u}'' := [\mu'(s,t) \log \frac{\mu'(s,t)}{\mathbb{P}_a(t|s)\mu'(s)}]_{(a,s,t)\in\mathcal{VA}}$. Clearly $(\mu', \mathbf{u}'')$ satisfies the constraints of the problem, and moreover,

$$\sum_{(a,s,t)\in\mathcal{VA}} \beta_a u_{a,s,t} = OInc_{\mathcal{M}}(\mu) > OInc_{\mathcal{M}}(\mu') = \sum_{(a,s,t)\in\mathcal{VA}} \beta_a u'_{a,s,t},$$

contradicting the assumption that the $(\mu, \mathbf{u})$ is optimal for problem (5). Thus, μ is a minimizer of $OInc_{\mathcal{M}}$, and the objective value is $\inf_\mu OInc_{\mathcal{M}}(\mu) = \langle\!\langle \mathcal{M} \rangle\!\rangle_0$, as desired. □

Proposition 2. *If $(\mu, \mathbf{u}, \mathbf{v})$ is a solution to (7), and $\boldsymbol{\beta} \geq \gamma\boldsymbol{\alpha}$, then μ is the unique element of $[\![\mathcal{M}]\!]^*_\gamma$, and $\langle\!\langle \mathcal{M} \rangle\!\rangle_\gamma$ equals the objective of (7) evaluated at $(\mu, \mathbf{u}, \mathbf{v})$.*

For convenience, we repeat problem (7) (left) and an equivalent variant of it that we implement (right) below.

$$\begin{aligned}
\underset{\mu,\mathbf{u},\mathbf{v}}{\text{minimize}} \quad & \sum_{(a,s,t)\in\mathcal{VA}} (\beta_a - \alpha_a\gamma) u_{a,s,t} + \gamma \sum_{w\in\mathcal{VX}} v_w \\
& - \sum_{(a,s,t)\in\mathcal{VA}^+} \alpha_a\gamma\, \mu(S_a{=}s, T_a{=}t) \log \mathbb{P}_a(t|s) \\
\text{subject to} \quad & \mu \in \Delta\mathcal{VX}, \quad (-\mathbf{v}, \mu, \mathbf{1}) \in K^{\mathcal{VX}}_{\exp}, \\
& \forall a \in \mathcal{A}.\ \big(-\mathbf{u}_a, \mu(T_a, S_a), \mathbb{P}_a(T_a|S_a)\mu(S_a)\big) \in K^{\mathcal{V}a}_{\exp}, \\
& \forall (a,s,t) \in \mathcal{VA}^0.\ \mu(S_a{=}s, T_a{=}t) = 0;
\end{aligned} \tag{7}$$

$$\begin{aligned}
\underset{\mu,\mathbf{u},\mathbf{v}}{\text{minimize}} \quad & \sum_{(a,s,t)\in\mathcal{VA}} (\beta_a - \alpha_a\gamma) u_{a,s,t} + \gamma \sum_{w\in\mathcal{VX}} v_w \\
& - \sum_{(a,s,t)\in\mathcal{VA}^+} \beta_a\, \mu(S_a{=}s, T_a{=}t) \log \mathbb{P}_a(t|s) \\
\text{subject to} \quad & \mu \in \Delta\mathcal{VX}, \quad (-\mathbf{v}, \mu, \mathbf{1}) \in K^{\mathcal{VX}}_{\exp}, \\
& \forall a \in \mathcal{A}.\ \big(-\mathbf{u}_a, \mu(T_a, S_a), \big[\mu(S_a = s)\big]_{(s,t)\in\mathcal{V}a}\big) \in K^{\mathcal{V}a}_{\exp}, \\
& \forall (a,s,t) \in \mathcal{VA}^0.\ \mu(S_a{=}s, T_a{=}t) = 0.
\end{aligned} \tag{7b}$$

Proof. We start with the problem on the left, which is (7) from the main text. Suppose that $(\mu, \mathbf{u}, \mathbf{v})$ is a solution to (7). The exponential constraints ensure that

$$\forall (a,s,t) \in \mathcal{VA}.\ u_{a,s,t} \geq \mu(s,t) \log \frac{\mu(t|s)}{\mathbb{P}_a(t|s)} \qquad \text{and} \qquad \forall w \in \mathcal{VX}.\ v_w \geq \mu(w) \log \mu(w).$$

As in the previous proof, we claim that these must hold with equality (except possibly for $u_{a,s,t}$ at indices satisfying $\beta_a = \gamma\alpha_a$, when it doesn't matter). This is because otherwise one could reduce the value of a component of u or v while still satisfying all of the constraints, to obtain a strictly smaller objective, contradicing the assumption that $(\mu, \mathbf{u}, \mathbf{v})$ minimizes it.

Thus, $\mathbf{v}$ is a function of μ, as is every value of $\mathbf{u}$ that affects the objective value of (7), meaning that this objective

value can be written as a function of μ alone:

$$\begin{aligned}
&\sum_{(a,s,t)\in\mathcal{VA}}(\beta_a-\alpha_a\gamma)u_{a,s,t} \;+\gamma\sum_{w\in\mathcal{VX}}v_w \;-\sum_{(a,s,t)\in\mathcal{VA}^+}\alpha_a\gamma\,\mu(s,t)\log\mathbb{P}_a(t|s)\\
&=\sum_{(a,s,t)\in\mathcal{VA}}(\beta_a-\alpha_a\gamma)\Big(\mu(s,t)\log\frac{\mu(t|s)}{\mathbb{P}_a(t|s)}\Big)\;+\;\gamma\sum_{w\in\mathcal{VX}}\mu(w)\log\mu(w)\;-\;\sum_{(a,s,t)\in\mathcal{VA}^+}\alpha_a\gamma\,\mu(s,t)\log\mathbb{P}_a(t|s)\\
&=\sum_{a\in\mathcal{A}}(\beta_a-\alpha_a\gamma)\sum_{(s,t)\in\mathcal{V}a}\Big(\mu(s,t)\log\frac{\mu(t|s)}{\mathbb{P}_a(t|s)}\Big)-\gamma\,\mathrm{H}(\mu)-\sum_{a\in\mathcal{A}}\alpha_a\gamma\sum_{(s,t)\in\mathcal{VA}}\mu(s,t)\log\mathbb{P}_a(t|s)\\
&=\sum_{a\in\mathcal{A}}(\beta_a-\alpha_a\gamma)\sum_{(s,t)\in\mathcal{V}a}\mu(s,t)\Big(\log\frac{1}{\mathbb{P}_a(t|s)}-\log\frac{1}{\mu(t|s)}\Big)-\gamma\,\mathrm{H}(\mu)-\sum_{a\in\mathcal{A}}\alpha_a\gamma\;\mathop{\mathbb{E}}_{\mu}[\log\mathbb{P}_a(T_a|S_a)]\\
&=\sum_{a\in\mathcal{A}}(\beta_a-\alpha_a\gamma)\,\mathop{\mathbb{E}}_{\mu}[-\log\mathbb{P}_a(T_a|S_a)]-\sum_{a\in\mathcal{A}}(\beta_a-\alpha_a\gamma)\,\mathrm{H}_\mu(T_a|S_a)-\gamma\,\mathrm{H}(\mu)-\sum_{a\in\mathcal{A}}\alpha_a\gamma\;\mathop{\mathbb{E}}_{\mu}[\log\mathbb{P}_a(T_a|S_a)]\\
&=\sum_{a\in\mathcal{A}}\Big(-\alpha_a\gamma-(\beta_a-\alpha_a\gamma)\Big)\mathop{\mathbb{E}}_{\mu}[\log\mathbb{P}_a(T_a|S_a)]+\sum_{a\in\mathcal{A}}(\alpha_a\gamma-\beta_a)\,\mathrm{H}_\mu(T_a|S_a)-\gamma\,\mathrm{H}(\mu)\\
&=-\sum_{a\in\mathcal{A}}\beta_a\,\mathop{\mathbb{E}}_{\mu}[\log\mathbb{P}_a(T_a|S_a)]+\sum_{a\in\mathcal{A}}(\alpha_a\gamma-\beta_a)\,\mathrm{H}_\mu(T_a|S_a)-\gamma\,\mathrm{H}(\mu).
\end{aligned}$$

(In the third step, we were able to convert $\mathcal{VA}^+$ to $\mathcal{VA}$ because, as usual in when dealing with information-therotic quantities, we interpret $0\log\frac{1}{0}$ as equal to zero, which is its limit.)

The algebra, for the right side variant (7b) is slightly simpler. In this case the middle conic constraint is almost the same, except for that $\mathbb{P}_a(t|s)$ has been replaced with 1, and so it ensures that $u_{a,s,t}=\mu(s,t)\log\mu(t\mid s)$ (i.e., the same as before, but without the probability in the denomiator). So,

$$\begin{aligned}
&\sum_{(a,s,t)\in\mathcal{VA}}(\beta_a-\alpha_a\gamma)u_{a,s,t} \;+\gamma\sum_{w\in\mathcal{VX}}v_w \;-\sum_{(a,s,t)\in\mathcal{VA}^+}\beta_a\,\mu(s,t)\log\mathbb{P}_a(t|s)\\
&=\sum_{(a,s,t)\in\mathcal{VA}}(\beta_a-\alpha_a\gamma)\mu(s,t)\log\mu(t|s)\;+\;\gamma\sum_{w\in\mathcal{VX}}\mu(w)\log\mu(w)\;-\;\sum_{(a,s,t)\in\mathcal{VA}^+}\beta_a\,\mu(s,t)\log\mathbb{P}_a(t|s)\\
&=\sum_{a\in\mathcal{A}}(\beta_a-\alpha_a\gamma)\sum_{(s,t)\in\mathcal{V}a}\mu(s,t)\log\mu(t|s)-\gamma\,\mathrm{H}(\mu)-\sum_{a\in\mathcal{A}}\beta_a\sum_{(s,t)\in\mathcal{VA}}\mu(s,t)\log\mathbb{P}_a(t|s)\\
&=\sum_{a\in\mathcal{A}}(\alpha_a\gamma-\beta_a)\,\mathrm{H}_\mu(T_a|S_a)-\gamma\,\mathrm{H}(\mu)-\sum_{a\in\mathcal{A}}\beta_a\,\mathop{\mathbb{E}}_{\mu}[\log\mathbb{P}_a(T_a|S_a)].
\end{aligned}$$

In either case, the objective value is equal to $[\![\mathcal{M}]\!]_\gamma(\mu)$, by (6). Because $(\mu,\mathbf{u},\mathbf{v})$ is optimal for this problem, we know that μ is a minimizer of $[\![\mathcal{M}]\!]_\gamma(\mu)$, and that the objective value equals $\langle\!\langle\mathcal{M}\rangle\!\rangle_\gamma$. □

Lemma 1. *The gradient and Hessian of the conditional relative entropy are given by*

$$\begin{aligned}
\Big[\nabla_\mu\boldsymbol{D}(\mu(X,Y)\parallel\mu(X)p(Y|X))\Big]_u&=\log\frac{\mu(Yu|Xu)}{p(Yu|Xu)}\\
\Big[\nabla^2_\mu\boldsymbol{D}(\mu(X,Y)\parallel\mu(X)p(Y|X))\Big]_{u,v}&=\frac{\mathbb{1}[Xu{=}Xv\wedge Yu{=}Yv]}{\mu(Yu,Xu)}-\frac{\mathbb{1}[Xv=Xu]}{\mu(Xu)}.
\end{aligned}$$

Proof. Represent μ as a vector $[\mu_w]_{w\in\mathcal{VX}}$. We will make repeated use of the following facts:

$$\frac{\partial}{\partial\mu_u}\Big[\mu(X{=}x)\Big]\;=\;\frac{\partial}{\partial\mu_u}\Big[\mu(x)\Big]=\sum_w\frac{\partial}{\partial\mu_u}\Big[\mu_w\Big]\,\mathbb{1}[Xw{=}x]\;=\;\mathbb{1}[Xu{=}x];\quad\text{and}$$

$$\begin{aligned}
\frac{\partial}{\partial \mu_u}\Big[\mu(y|x)\Big] &= \frac{\partial}{\partial \mu_u}\left[\frac{\mu(x,y)}{\mu(x)}\right] \\
&= -\mu(x,y)\,\frac{\partial}{\partial \mu_u}\left[\frac{1}{\mu(x)}\right] + \frac{1}{\mu(x)}\,\frac{\partial}{\partial \mu_u}\Big[\mu(x,y)\Big] \\
&= -\frac{\mu(x,y)}{\mu(x)^2}\,\mathbb{1}[Xu = x] + \frac{1}{\mu(x)}\,\mathbb{1}[XY(u){=}xy] \\
&= \frac{\mathbb{1}[Xu = x]}{\mu(x)}\Big(\mathbb{1}[Yu = y] - \mu(y|x)\Big).
\end{aligned}$$

We now apply this to the (conditional) relative entropy:

$$\begin{aligned}
&\frac{\partial}{\partial \mu_u}\Big[\boldsymbol{D}(\mu(X,Y) \parallel \mu(X)p(Y|X))\Big] \\
&= \frac{\partial}{\partial \mu_u}\left[\sum_w \mu_w \log \frac{\mu(Yw|Xw)}{p(Yw|Xw)}\right] \\
&= \sum_w \mathbb{1}[u{=}w] \log \frac{\mu(Yw|Xw)}{p(Yw|Xw)} + \sum_w \mu_w\,\frac{\partial}{\partial \mu_u}\left[\log \frac{\mu(Yw|Xw)}{p(Yw|Xw)}\right] \\
&= \log \frac{\mu(Yu|Xu)}{p(Yu|Xu)} + \sum_w \mu_w \frac{p(Yw|Xw)}{\mu(Yw|Xw)}\,\frac{\partial}{\partial \mu_u}\left[\frac{\mu(Yw|Xw)}{p(Yw|Xw)}\right] \\
&= \log \frac{\mu(Yu|Xu)}{p(Yu|Xu)} + \sum_w \mu_w \frac{1}{\mu(Yw|Xw)}\,\frac{\partial}{\partial \mu_u}\Big[\mu(Yw|Xw)\Big] \\
&= \log \frac{\mu(Yu|Xu)}{p(Yu|Xu)} + \sum_w \mu_w \frac{1}{\mu(Yw|Xw)}\,\frac{\mathbb{1}[Xu = Xw]}{\mu(Xw)}\Big(\mathbb{1}[Yu = Yw] - \mu(Yw|Xw)\Big) \\
&= \log \frac{\mu(Yu|Xu)}{p(Yu|Xu)} + \sum_w \mu_w \frac{\mathbb{1}[Xu{=}Xw \wedge Yu{=}Yw]}{\mu(Xw,Yw)} - \sum_w \mu_w \frac{\mathbb{1}[Xu = Xw]}{\mu(Xw)} \\
&= \log \frac{\mu(Yu|Xu)}{p(Yu|Xu)} + \frac{1}{\mu(Xu,Yu)} \sum_w \mu_w\,\mathbb{1}[Xu{=}Xw \wedge Yu{=}Yw] - \frac{1}{\mu(Xu)} \sum_w \mu_w\,\mathbb{1}[Xu = Xw] \\
&= \log \frac{\mu(Yu|Xu)}{p(Yu|Xu)} + \frac{\mu(Xu,Yu)}{\mu(Xu,Yu)} - \frac{\mu(Xu)}{\mu(Xu)} \\
&= \log \frac{\mu(Yu|Xu)}{p(Yu|Xu)}
\end{aligned}$$

This allows us to compute the Hessian of the conditional relative entropy, whose components are

$$\begin{aligned}
\frac{\partial^2}{\partial \mu_u \partial \mu_v}\Big[\boldsymbol{D}(\mu(XY) \parallel \mu(X)p(Y|X))\Big] &= \frac{\partial}{\partial \mu_v}\left[\log \frac{\mu(Yu|Xu)}{p(Yu|Xu)}\right] \\
&= \frac{p(Yu|Xu)}{\mu(Yu|Xu)}\,\frac{1}{p(Yu|Xu)}\,\frac{\partial}{\partial \mu_v}\Big[\mu(Yu|Xu)\Big] \\
&= \frac{1}{\mu(Yu|Xu)}\,\frac{\mathbb{1}[Xv{=}Xu]}{\mu(Xu)}\Big(\mathbb{1}[Yv{=}Yu] - \mu(Yu|Xu)\Big) \\
&= \frac{\mathbb{1}[Xu{=}Xv \wedge Yu{=}Yv]}{\mu(Yu,Xu)} - \frac{\mathbb{1}[Xv = Xu]}{\mu(Xu)}.
\end{aligned}$$

□

Lemma 2. *Let $p(Y|X)$ be a cpd, and suppose $\mu_0, \mu_1 \in \Delta\mathcal{V}(X,Y)$ are joint distributions that have different conditional marginals on Y given X; that is, that there exist $(x,y) \in \mathcal{V}(X,Y)$ such that $\mu_0(x,y)\mu_1(x) \neq \mu_1(x,y)\mu_0(x)$. Then the*

conditional relative entropy $\boldsymbol{D}\Big(\mu(X,Y)\;\Big\|\;\mu(X)p(Y|X)\Big)$ *is strictly convex in* μ *along the line segment from* μ_0 *to* μ_1. *More precisely, for* $t \in [0,1]$, *if we define* $\mu_t := (1-t)\mu_0 + t\,\mu_1$, *then the function*

$$t \;\mapsto\; \boldsymbol{D}\Big(\mu_t(X,Y)\;\Big\|\;\mu_t(X)p(Y|X)\Big) \qquad \textit{is strictly convex.}$$

Proof. We can only have non-strict convexity if the direction δ lies in the null-space of the Hessian matrix $\mathbf{H}(\mu)$ of the relative entropy. By Lemma 1,

$$\mathbf{H}_{(xy),(x'y')} = \frac{\mathbb{1}[x{=}x' \wedge y{=}y']}{\mu(x,y)} - \frac{\mathbb{1}[x{=}x']}{\mu(x)}.$$

Consider a function $\delta : \mathcal{V}(X,Y) \to \mathbb{R}$ that is not identically zero, which can be viewed as a vector $\boldsymbol{\delta} = [\delta(x,y)]_{(x,y)\in\mathcal{V}(X,Y)} \in \mathbb{R}^{\mathcal{V}(X,Y)}$. We can also view δ as a (signed) measure on $\mathcal{V}(X,Y)$, that has marginals in the usual sense. In particular, we use the analogous notation

$$\delta(x) := \sum_{y\in\mathcal{V}Y} \delta(x,y).$$

We then compute

$$\begin{aligned}\big(\,\mathbf{H}(\mu)\,\boldsymbol{\delta}\,\big)_{x,y} &= \sum_{x',y'} \delta(x',y') \left(\frac{\mathbb{1}[x{=}x' \wedge y{=}y']}{\mu(x,y)} - \frac{\mathbb{1}[x{=}x']}{\mu(x)}\right)\\ &= \frac{\delta(x,y)}{\mu(x,y)} - \frac{\delta(x)}{\mu(x)}.\end{aligned}$$

and also

$$\begin{aligned}\boldsymbol{\delta}^{\mathsf{T}}\mathbf{H}(\mu)\,\boldsymbol{\delta} &= \sum_{x,y} \delta(x,y)(\,\mathbf{H}(\mu)\,\boldsymbol{\delta}\,)_{x,y}\\ &= \sum_{x,y} \delta(x,y)\left(\frac{\delta(x,y)}{\mu(x,y)} - \frac{\delta(x)}{\mu(x)}\right)\\ &= \sum_{x,y} \frac{\delta(x,y)^2}{\mu(x,y)} - \sum_{x} \frac{\delta(x)}{\mu(x)} \sum_{y} \delta(x,y)\\ &= \sum_{x,y} \frac{\delta(x,y)^2}{\mu(x,y)} - \sum_{x} \frac{\delta(x)^2}{\mu(x)}\\ &= \sum_{x} \frac{\delta(x)^2}{\mu(x)} \left(\sum_{y} \frac{\delta(x,y)^2}{\delta(x)^2\mu(y|x)} - 1\right). \qquad (1)\end{aligned}$$

Now, consider another discrete measure $|\delta|$, whose value at each component is the absolute value of the value of δ at that component, i.e., $|\delta|(x,y) := |\delta(x,y)|$. By construction, $|\delta|$ is now an unnormalized probability measure: $|\delta| = kq(X,Y)$, where $k = \sum_{x,y} |\delta(x,y)| > 0$ and $q \in \Delta\mathcal{V}(X,Y)$.

Note also that $|\delta|(x)^2 = (\sum_y |\delta(x,y)|)^2 \geq (\sum_y \delta(x,y))^2$, and strictly so if there are y, y' such that $\delta(x,y) < 0 < \delta(x,y')$. In other words, the vector $\boldsymbol{\delta}_x = [\delta(x,y)]_{y\in\mathcal{V}Y}$ is either non-negative or non-positive: $\boldsymbol{\delta}_x \geq 0$ or $\boldsymbol{\delta}_x \leq 0$ for

each x. Meanwhile, $|\delta|(x,y)^2 = \delta(x,y)^2$ is unchanged. Thus, for every $x \in \mathcal{V}X$, we have:

$$\begin{aligned}\sum_y \frac{\delta(x,y)^2}{\delta(x)^2\mu(y|x)} - 1 &\geq \sum_y \frac{|\delta|(x,y)^2}{|\delta|(x)^2\mu(y|x)} - 1 \\ &= \sum_y \frac{k^2 q(x,y)^2}{k^2 q(x)^2 \mu(y|x)} - 1 \\ &= \sum_y \frac{q(y|x)^2}{\mu(y|x)} - 1 \\ &= \chi^2\Big(q(Y|x)\Big\|\mu(Y|x)\Big) \geq 0.\end{aligned}$$

The final line depicts the χ^2 divergence between the distributions $q(Y|x)$ and $\mu(Y|x)$, both distributions over Y. Since it is a divergence, this quantity is non-negative and equals zero if and only if $q(Y|x) = \mu(Y|x)$.

Picking up where we left off, we have:

$$\begin{aligned}\boldsymbol{\delta}^\mathsf{T}\mathbf{H}(\mu)\boldsymbol{\delta} &= \sum_x \frac{\delta(x)^2}{\mu(x)}\left(\sum_y \frac{\delta(x,y)^2}{\delta(x)^2\mu(y|x)} - 1\right) \\ &\geq \sum_x \frac{\delta(x)^2}{\mu(x)}\left(\sum_y \frac{|\delta|(x,y)^2}{|\delta|(x)^2\mu(y|x)} - 1\right) \\ &= \sum_x \frac{\delta(x)^2}{\mu(x)}\chi^2\Big(q(Y|x)\Big\|\mu(Y|x)\Big) \geq 0.\end{aligned}$$

As a non-negatively weighted sum of non-negative numbers, this final quantity is non-negative, and equals zero if and only if, for each $x \in \mathcal{V}X$, we have either $q(Y|x) = \mu(Y|x)$, or $\delta(x) = 0$. Furthermore, if $\boldsymbol{\delta}^\mathsf{T}\mathbf{H}(\mu)\boldsymbol{\delta} = 0$, then *both* inequalities hold with equality. Therefore, we know that if $\delta(x) \neq 0$, then $\boldsymbol{\delta}_x \geq \mathbf{0}$ or $\boldsymbol{\delta}_x \leq \mathbf{0}$. These two conditions are also sufficient to show that $\boldsymbol{\delta}^\mathsf{T}\mathbf{H}(\mu)\boldsymbol{\delta} = 0$. To summarize what we know so far:

$$\boldsymbol{\delta}^\mathsf{T}\mathbf{H}(\mu)\boldsymbol{\delta} = 0 \quad \iff \quad \forall x \in \mathcal{V}X. \quad \text{either } (\boldsymbol{\delta}_x \geq \mathbf{0} \text{ or } \boldsymbol{\delta}_y \leq \mathbf{0}) \text{ and } |\delta|(Y|x) = \mu(Y|x) \text{ or } \delta(x) = 0.$$

The second possibility, however, is somewhat of a fluke; we now return to the expression we had in (1) before considering $|\delta|$, We've already shown that the contribution to the sum at each value of x is non-negative, so if $\boldsymbol{\delta}^\mathsf{T}\mathbf{H}(\mu)\boldsymbol{\delta}$ is to equal zero, each summand which depends on x must be zero as well. So if x is a value of X for which $\delta(x) = 0$, then

$$0 = \frac{1}{\mu(x)}\left(\sum_y \frac{\delta(x,y)^2}{\mu(y|x)} - \delta(x)^2\right) = \frac{1}{\mu(x)}\sum_y \frac{\delta(x,y)^2}{\mu(y|x)} = \sum_y \frac{\delta(x,y)^2}{\mu(x,y)},$$

which is only possible if $\delta(x,y) = 0$ for all y. This allows us to compute, more simply, that

$$\boldsymbol{\delta}^\mathsf{T}\mathbf{H}(\mu)\boldsymbol{\delta} = 0 \quad \iff \quad (\forall x.\, \boldsymbol{\delta}_x \geq \mathbf{0} \text{ or } \boldsymbol{\delta}_x \leq \mathbf{0}) \qquad \text{and} \qquad \forall (x,y) \in \mathcal{V}(X,Y).\ \delta(x,y)\mu(x) = \delta(x)\mu(x,y)$$

Finally, we are in a position to prove the lemma. Suppose $\mu_0, \mu_1 \in \Delta\mathcal{V}(X,Y)$ and $(x^*, y^*) \in \mathcal{V}(X,Y)$ are such that $\mu_0(x^*,y^*)\mu_1(x^*) \neq \mu_1(x^*,y^*)\mu_0(x^*)$. So, the quantity

$$gap := \mu_1(x^*,y^*)\mu_0(x^*) - \mu_0(x^*,y^*)\mu_1(x^*) \quad \text{is nonzero.}$$

Then for all $t \in (0,1)$ the intermediate point $\mu_t = (1-t)\,\mu_0 + t\,\mu_1$ must have different conditional marginals from

both μ_0 and μ_1, as

$$\begin{aligned}
&\mu_t(x^*,y^*)\mu_0(x^*) - \mu_0(x^*,y^*)\mu_t(x^*)\\
&= (1-t)\mu_0(x^*,y^*)\mu_0(x^*) + t\mu_1(x^*,y^*)\mu_0(x^*) - (1-t)\mu_0(x^*,y^*)\mu_0(x^*) - t\mu_0(x^*,y^*)\mu_1(x^*)\\
&= t\big(\mu_1(x^*,y^*)\mu_0(x^*) - \mu_0(x^*,y^*)\mu_1(x^*)\big)\\
&= \ t\cdot gap \ \neq\ 0,
\end{aligned}$$

and analogously for μ_1,

$$\begin{aligned}
&\mu_t(x^*,y^*)\mu_1(x^*) - \mu_1(x^*,y^*)\mu_t(x^*)\\
&= (1-t)\mu_0(x^*,y^*)\mu_1(x^*) + t\mu_1(x^*,y^*)\mu_1(x^*) - (1-t)\mu_1(x^*,y^*)\mu_0(x^*) - t\mu_1(x^*,y^*)\mu_1(x^*)\\
&= (1-t)(\mu_0(x^*,y^*)\mu_1(x^*) - \mu_1(x^*,y^*)\mu_0(x^*))\\
&= \ -(1-t)\cdot gap \ \neq\ 0.
\end{aligned}$$

Then for any direction $\delta := k(\mu_0 - \mu_1)$ parallel to the segment between μ_0 and μ_1 (intuitively a tangent vector at μ_t, although this fact doesn't affect the computation), of nonzero length ($k \neq 0$), we have:

$$\begin{aligned}
&\mu_t(x^*,y^*)\delta(x^*) - \delta(x^*,y^*)\mu_t(x^*)\\
&= k\ \mu_t(x^*,y^*)\big(\mu_0(x^*) - \mu_1(x^*)\big) - k\ \big(\mu_0(x^*,y^*) - \mu_1(x^*,y^*)\big)\mu_t(x^*)\\
&= k\Big(\mu_t(x^*,y^*)\mu_0(x^*) - \mu_t(x^*,y^*)\mu_1(x^*) - \mu_0(x^*,y^*)\mu_t(x^*) + \mu_1(x^*,y^*)\mu_t(x^*)\Big)\\
&= k\Big(\big(\mu_t(x^*,y^*)\mu_0(x^*) - \mu_0(x^*,y^*)\mu_t(x^*)\big) + \big(\mu_1(x^*,y^*)\mu_t(x^*) - \mu_t(x^*,y^*)\mu_1(x^*)\big)\Big)\\
&= k\big(+t\ gap + (1-t)\ gap\big)\\
&= k\ gap \qquad \neq\ 0.
\end{aligned}$$

So at every t, directions parallel to the segment are not in the null space of $\mathbf{H}(\mu_t)$, meaning that $\boldsymbol{\delta}^\mathsf{T}\mathbf{H}(\mu_t)\boldsymbol{\delta} > 0$ and so our function is strictly convex along this segment. □

Proposition 3. *If $\mathcal{M}$ has arcs $\mathcal{A}$ and $\boldsymbol{\beta} \geq 0$, the minimizers of $OInc_{\mathcal{M}}$ all have the same conditional marginals along $\mathcal{A}$. That is, for all $\mu_1, \mu_2 \in [\![\mathcal{M}]\!]_0^*$ and all $S \xrightarrow{a} T \in \mathcal{A}$ with $\beta_a > 0$, we have $\mu_1(T,S)\mu_2(S) = \mu_2(T,S)\mu_1(S)$.*

Proof. For contradiction, suppose that $\mu_1, \mu_2 \in [\![\mathcal{M}]\!]_0^*$, but there is some $(\hat{a}, \hat{s}, \hat{t}) \in \mathcal{VA}$ such that $\beta_a > 0$ and

$$\mu_1(T_a{=}\hat{t}, S_a{=}\hat{s})\mu_2(S_a{=}\hat{s}) \neq \mu_2(T_a{=}\hat{t}, S_a{=}\hat{s})\mu_1(S_a\hat{s}).$$

For $t \in [0,1]$, let $\mu_t := (1-t)\mu_0 + t\,\mu_1$ as before. Then define

$$F(t) := \boldsymbol{D}\Big(\mu_t(S_a, T_a) \ \Big\|\ \mu_t(S_a)\mathbb{P}_a(T_a|S_a)\Big).$$

Since $\mu_0(S_a, T_a)$ and $\mu_1(S_a, T_a)$ are joint distributions voer two variables, with different conditional marginals, as above, Lemma 2 applies, and so $F(t)$ is strictly convex.

Let

$$OInc_{\mathcal{M}\setminus\hat{a}} := \sum_{a\neq\hat{a}} \beta_a \boldsymbol{D}(\mu(T_a, S_a) \parallel \mathbb{P}_a(T_a|S_a)\mu(S_a))$$

be the observational incompatibility loss, but without the term corresponding to edge $\hat{a}$. Since $OInc_{\mathcal{M}\setminus\hat{a}}$ is convex in its argument, it is in particular convex along the segment from μ_0 to μ_1; that is, for $t \in [0,1]$, the function $t \mapsto OInc_{\mathcal{M}\setminus\hat{a}}(\mu_t)$ is convex. Therefore, we know that the function

$$G(t) := OInc_{\mathcal{M}}(\mu_t) = OInc_{\mathcal{M}\setminus a}(\mu_t) + \beta_a\, F(t),$$

is *strictly* convex. But then this means $\mu_{1/2}$ satisfies

$$OInc_{\mathcal{M}}(\mu_{1/2}) < OInc_{\mathcal{M}}(\mu_0),$$

contradicting the premise that μ_0 minimizes $OInc_{\mathcal{M}}$ (i.e., $\mu_0 \in [\![\mathcal{M}]\!]_0^*$). Therefore, it must be the case that all distributions in $[\![\mathcal{M}]\!]_0^*$ have the same conditional marginals, as promised. □

Proposition 4. *If $\mu \in [\![\mathcal{M}]\!]_0^*$, then*

$$\underset{\mathcal{M}}{SInc}(\mu) = \sum_{w \in \mathcal{VX}} \mu(w) \log \left(\mu(w) \Big/ \prod_{a \in \mathcal{A}} \nu\big(T_a(w) \big| S_a(w)\big)^{\alpha_a} \right), \tag{2}$$

where $\{\nu(T_a|S_a)\}_{a\in\mathcal{A}}$ are the marginals along the arcs $\mathcal{A}$ shared by all distributions in $[\![\mathcal{M}]\!]_0^*$ (per Proposition 3), and $S_a(w), T_a(w)$ are the values of variables S_a and T_a in w.

Proof. This is mostly a simple algebraic manipulation. By definition:

$$\begin{aligned}
SInc_{\mathcal{M}}(\mu) &= -\,\mathrm{H}(\mu) + \sum_{a\in\mathcal{A}} \alpha_a \,\mathrm{H}_\mu(T_a|S_a) \\
&= \mathop{\mathbb{E}}_{\mu} \left[-\log \frac{1}{\mu} + \sum_{a\in\mathcal{A}} \alpha_a \log \frac{1}{\mu(T_a|S_a)} \right] \\
&= \sum_{w\in\mathcal{VX}} \mu(w) \left[\log \mu(w) + \sum_{a\in\mathcal{A}} \log \frac{1}{\mu(T_a(w)|S_a(w))^{\alpha_a}} \right] \\
&= \sum_{w\in\mathcal{VX}} \mu(w) \log \left(\mu(w) \Big/ \prod_{a\in\mathcal{A}} \mu(T_a(w)|S_a(w))^{\alpha_a} \right)
\end{aligned}$$

But, by Proposition 3, if we restrict $\mu \in [\![\mathcal{M}]\!]_0^*$, then the conditoinal marginals in the denominator do not depend on the particular choice of μ; they're shared among all $\nu \in [\![\mathcal{M}]\!]_0^*$. □

Proposition 5. *If $\nu \in [\![\mathcal{M}]\!]_0^*$ and $(\mu, \mathbf{u})$ solves the problem*

$$\begin{aligned}
\underset{\mu,\mathbf{u}}{\text{minimize}} \quad & \mathbf{1}^\mathsf{T}\mathbf{u} \\
\text{subject to} \quad & (-\mathbf{u}, \mu, \mathbf{k}) \in K_{\exp}^{\mathcal{VX}}, \qquad \mu \in \Delta\mathcal{VX}, \\
& \forall S \xrightarrow{a} T \in \mathcal{A}.\ \ \mu(S,T)\,\nu(S) = \mu(S)\,\nu(S,T),
\end{aligned} \tag{3}$$

then $[\![\mathcal{M}]\!]_{0+}^* = \{\mu\}$ and $\mathbf{1}^\mathsf{T}\mathbf{u} = SInc_{\mathcal{M}}(\mu)$.

Proof. Suppose $(-\mathbf{u}, \mu, \mathbf{k})$ is a solution to problem (3). The second constraint, by Proposition 3, ensures that $\mu \in [\![\mathcal{M}]\!]_0^*$. Then,

$$\begin{aligned}
(-\mathbf{u}, \mu, \mathbf{k}) \in K^{\mathcal{VX}} \quad \Longrightarrow \quad & \forall w \in \mathcal{VX}.\ u_w \geq \mu(w) \log \frac{\mu(w)}{k_w} \\
& = \mu(w) \log \left(\mu(w) \Big/ \prod_{a\in\mathcal{A}} \mu(T_a(w)|S_a(w))^{\alpha_a} \right).
\end{aligned}$$

The same logic as in the proofs of Propositions 1 and 2 shows that this inequality must be tight, or else $(-\mathbf{u}, \mu, \mathbf{k})$ would not be optimal for (3). So, $\mathbf{u}$ is a function of μ. Also, by Proposition 4, the problem objective satisfies

$$\mathbf{1}^\mathsf{T}\mathbf{u} = \sum_{w\in\mathcal{V}X} u_w = SInc_{\mathcal{M}}(\mu).$$

Finally, because μ is optimal, it must be the unique distribution $[\![\mathcal{M}]\!]^*$, which among those distributions that minimize $OInc_{\mathcal{M}}$, also minimizes $SInc_{\mathcal{M}}$, meaning $\mu = [\![\mathcal{M}]\!]^*$. □

Theorem 6. *If $\mathcal{M}_1$ and $\mathcal{M}_2$ are PDGs over the sets of variables $\mathcal{X}_1$ and $\mathcal{X}_2$, respectively, then $\mathcal{X}_1$ and $\mathcal{X}_2$ are conditionally independent given $\mathcal{X}_1 \cap \mathcal{X}_2$ in every $\mu \in [\![\mathcal{M}_1 + \mathcal{M}_2]\!]^*_\gamma$, for all $\gamma > 0$ and $\gamma = 0^+$.*

Or symbolically: $\qquad \mathcal{M}_1 + \mathcal{M}_2 \models \mathcal{X}_1 \perp\!\!\!\perp \mathcal{X}_2 \mid \mathcal{X}_1 \cap \mathcal{X}_2.$

Proof. Note that, save for the joint entropy, every summand the scoring function $[\![\mathcal{M}_1 + \mathcal{M}_2]\!]_\gamma : \Delta(\mathcal{VX}_1 \times \mathcal{VX}_2)$, is a function of the conditional marginal of μ along some edge. In particular, those terms that correspond to edges of $\mathcal{M}_1$ can be computed from the marginal $\mu(\mathcal{X}_1)$, while those that correspond to edges of $\mathcal{M}_2$ can be computed from the marginal $\mu(\mathcal{X}_2)$. Therefore, there are functions f and g such that:

$$[\![\mathcal{M}_1 + \mathcal{M}_2]\!]_\gamma(\mu) = f(\mu(\mathcal{X}_1)) + g(\mu(\mathcal{X}_2)) - \gamma\,\mathrm{H}(\mu).$$

To make this next step extra clear, let $\mathbf{X} := \mathcal{X}_1 \setminus \mathcal{X}_2$ and $\mathbf{Z} := \mathcal{X}_2 \setminus \mathcal{X}_1$, be the variables unique to each PDG, and $\mathbf{S} := \mathcal{X}_1 \cap \mathcal{X}_2$ be the set of variables they have in common, so that $(\mathbf{X}, \mathbf{S}, \mathbf{Z})$ is a partition of all variables $\mathbf{X}_1 \cup \mathbf{X}_2$. Now, define a new distribution $\mu' \in \Delta(\mathcal{VX}_1 \times \mathcal{VX}_2)$ by

$$\mu'(\mathbf{X}, \mathbf{S}, \mathbf{Z}) := \mu(\mathbf{S})\mu(\mathbf{Z} \mid \mathbf{S})\mu(\mathbf{X} \mid \mathbf{S}) \qquad \Big(= \mu(\mathbf{X}, \mathbf{S})\mu(\mathbf{Z} \mid \mathbf{S}) = \mu(\mathbf{Z}, \mathbf{S})\mu(\mathbf{X} \mid \mathbf{S}) \Big).$$

One can easily verify that $\mathbf{X}$ and $\mathbf{Z}$ are independent given $\mathbf{S}$ in μ' (by construction), and the alternate forms on the right make it easy to see that $\mu(\mathcal{X}_1) = \mu'(\mathcal{X}_1)$ and $\mu(\mathcal{X}_2) = \mu'(\mathcal{X}_2)$. Furthermore, for any $\nu'(\mathbf{X}, \mathbf{S}, \mathbf{Z})$, we can write

$$\begin{aligned}
\mathrm{H}(\nu) &= \mathrm{H}_\nu(\mathbf{X}, \mathbf{S}, \mathbf{Z}) = \mathrm{H}_\nu(\mathbf{X}, \mathbf{S}) + \mathrm{H}_\nu(\mathbf{Z} \mid \mathbf{X}, \mathbf{S}) \\
&= \mathrm{H}_\nu(\mathbf{X}, \mathbf{S}) + \mathrm{H}_\nu(\mathbf{Z} \mid \mathbf{X}, \mathbf{S}) - \mathrm{H}_\nu(\mathbf{Z} \mid \mathbf{S}) + \mathrm{H}_\nu(\mathbf{Z} \mid \mathbf{S}) \\
&= \mathrm{H}_\nu(\mathbf{X}, \mathbf{S}) + \mathrm{H}_\nu(\mathbf{Z} \mid \mathbf{S}) - \mathrm{I}_\nu(\mathbf{Z}; \mathbf{X} | \mathbf{S}),
\end{aligned}$$

where $\mathrm{I}_\nu(\mathbf{X}; \mathbf{Z}|\mathbf{S})$, the conditional mutual information between $\mathbf{X}$ and $\mathbf{Z}$ given $\mathbf{S}$ (in ν), is non-negative, and equal to zero if and only if $\mathbf{X}$ and $\mathbf{Z}$ are conditionally independent given $\mathbf{S}$ [see, for instance, MacKay, 2003, §1]. So $\mathrm{I}_{\mu'}(\mathbf{X}; \mathbf{Z}|\mathbf{S}) = 0$, and $\mathrm{H}_{\mu'} = \mathrm{H}_{\mu'}(\mathbf{X}, \mathbf{S}) + \mathrm{H}_{\mu'}(\mathbf{Z}|\mathbf{S})$. Because μ and μ' share marginals on $\mathcal{X}_1$ and $\mathcal{X}_2$, while the terms $\mathrm{H}(\mathbf{X}, \mathbf{S})$ and $\mathrm{H}(\mathbf{Z}|\mathbf{S})$ depend only on these marginals, respectively, we also know that $\mathrm{H}_\mu(\mathbf{X}, \mathbf{S}) = \mathrm{H}_{\mu'}(\mathbf{X}, \mathbf{S})$ and $\mathrm{H}_\mu(\mathbf{Z}|\mathbf{S}) = \mathrm{H}_{\mu'}(\mathbf{Z}|\mathbf{S})$; thus we have

$$\begin{aligned}
\mathrm{H}(\mu) &= \mathrm{H}_\mu(\mathbf{X}, \mathbf{S}) + \mathrm{H}_\mu(\mathbf{Z} \mid \mathbf{S}) - \mathrm{I}_\mu(\mathbf{Z}; \mathbf{X}|\mathbf{S}) \\
&= \mathrm{H}(\mu') - \mathrm{I}_\mu(\mathbf{Z}; \mathbf{X}|\mathbf{S}).
\end{aligned}$$

Therefore,

$$\begin{aligned}
[\![\mathcal{M}_1 + \mathcal{M}_2]\!]_\gamma(\mu) &= f(\mu(\mathcal{X}_1)) + g(\mu(\mathcal{X}_2)) - \gamma\,\mathrm{H}(\mu) \\
&= f(\mu'(\mathcal{X}_1)) + g(\mu'(\mathcal{X}_2)) - \gamma\,\mathrm{H}(\mu') + \gamma\,\mathrm{I}_\mu(\mathbf{Z}; \mathbf{X}|\mathbf{S}) \\
&= [\![\mathcal{M}_1 + \mathcal{M}_2]\!]_\gamma(\mu') + \gamma\,\mathrm{I}_\mu(\mathbf{Z}; \mathbf{X}|\mathbf{S}).
\end{aligned}$$

But conditional mutual information is non-negative, and by assumption, $[\![\mathcal{M} + \mathcal{M}_2]\!]_\gamma(\mu)$ is minimal. Therefore, it must be the case that

$$\mathrm{I}_\mu(\mathbf{Z}; \mathbf{X}|\mathbf{S}) = \mathrm{I}_\mu(\mathcal{X}_1; \mathcal{X}_2 \mid \mathcal{X}_1 \cap \mathcal{X}_2) = 0,$$

showing that $\mathcal{X}_1$ and $\mathcal{X}_2$ are conditionally independent given the varaibles that they have in common.
(The fact that $\mathrm{I}_\mu(\mathbf{Z}; \mathbf{X}|\mathbf{S}) = \mathrm{I}_\mu(\mathcal{X}_1; \mathcal{X}_2 \mid \mathcal{X}_1 \cap \mathcal{X}_2)$ is both easy to show and an instance of a well-known identity; see CIRV2 in Theorem 4.4.4 of Halpern [2017], for instance.) □

Corollary 6.1. *If $\mathcal{M}$ is a PDG with arcs $\mathcal{A}$, $(\mathcal{C}, \mathcal{T})$ is a tree decomposition of $\mathcal{A}$, $\gamma > 0$, and $\mu \in [\![\mathcal{M}]\!]^*_\gamma$, then there exists a clique tree $\boldsymbol{\mu}$ over $(\mathcal{C}, \mathcal{T})$ such that $\Pr_{\boldsymbol{\mu}} = \mu$.*

Proof. The set of distributions that can be represented by a calibrated clique tree over $(\mathcal{C}, \mathcal{T})$ is the same as the set of distributions that can represeted by a factor graph for which $(\mathcal{C}, \mathcal{T})$ is a tree decomposition. One direction holds because any such product of factors "calibrated", via message passing algorithms such as belief propogation, to form a clique tree. The other direction holds because $\Pr_{\boldsymbol{\mu}}$ itself is a product of factors that decomposes over $(\mathcal{C}, \mathcal{T})$.

Alternatively, this same set of distributions that satisfy the independencies of the Markov Network obtained by connecting every pair of variables that share a cluster. More formally, this network is the graph $G := (\mathcal{X}, E := \{(X{-}Y) : \exists C \in \mathcal{C}.\ \{X, Y\} \subseteq C\})$. Also, G happens to chordal as well, which we prove at the end.

Using only the PDG Markov property (Theorem 6), we now show that every independence described by G also holds in every distribution $\mu \in [\![\mathcal{M}]\!]^*_\gamma$. Suppose that, for sets of variables $\mathbf{X}, \mathbf{Y}, \mathbf{Z} \subseteq \mathcal{X}$, $\mathrm{I}(\mathbf{X}; \mathbf{Y} | \mathbf{Z})$ is an independence described by G. This means [Koller and Friedman, 2009, Defn 4.8] that if $X \in \mathbf{X}$, $Y \in \mathbf{Y}$, and π is a path in G between them, then some node along π lies in $\mathbf{Z}$.

Let $\mathcal{T}'$ be the graph that results from removing each edge $(C{-}D) \in \mathcal{T}$ that satisfies $C \cap D \subseteq \mathbf{Z}$, which is a disjoint union $\mathcal{T}' = \mathcal{T}_1 \sqcup \ldots \sqcup \mathcal{T}_n$ of subtrees that have no clusters in common. To parallel this notation, let $\mathcal{C}_1, \ldots, \mathcal{C}_n$ be their respective vertex sets. Note that for every edge $e = (C{-}D) \in \mathcal{T}'$, there must by definiton be some variable $U_e \in (C \cap D) \setminus \mathbf{Z}$.

We claim that no subtree $\mathcal{T}_i$ can have both a cluster D_X containing a variable $X \in \mathbf{X} \setminus \mathbf{Z}$ and also a cluster D_Y containing a variable $Y \in \mathbf{Y} \setminus \mathbf{Z}$. Suppose that it did. Then the (unique) path in $\mathcal{T}$ between D_X and D_Y, which we label

$$D_X = \quad D_0 \overset{e_1}{\text{---}} D_1 \overset{e_2}{\text{---}} \cdots \overset{e_{m-1}}{\text{---}} D_{m-1} \overset{e_m}{\text{---}} D_m \quad = D_Y \ ,$$

would lie entirely within $\mathcal{T}_\rangle \subseteq \mathcal{T}'$. This gives rise to a corresponding path in G:

$$\begin{array}{ccccccccccc} X & \text{---} & U_{e_1} & \text{---} & U_{e_2} & \text{---} & \cdots & \text{---} & U_{e_{n-1}} & \text{---} & U_{e_n} & \text{---} & Y \\ \cap & & \cap & & \cap & & & & \cap & & \cap & & \cap \\ D_0 & & D_0 \cap D_1 & & D_1 \cap D_2 & & & & D_{n-2} \cap D_{n-1} & & D_{n-1} \cap D_n & & D_n \end{array} \ ,$$

and moreover, this path is disjoint from $\mathbf{Z}$. This contradicts our assumption that every path in G between a member of $\mathbf{X}$ and a member of $\mathbf{Y}$ must intersect with $\mathbf{Z}$, and so no subtree can have both a cluster containing a variable $X \in \mathbf{X} \setminus \mathbf{Z}$ and also one containing $Y \in \mathbf{Y} \setminus \mathbf{Z}$.

We can now partition the clusters as $\mathcal{C} = \mathcal{C}_{\mathbf{X}} \sqcup \mathcal{C}^+_{\mathbf{Y}}$, where $\mathcal{C}_{\mathbf{X}}$ is the set of the clusters that belong to subtrees $\mathcal{T}_i$ with a cluster containing some $X \in \mathbf{X} \setminus \mathbf{Z}$, and its $\mathcal{C}^+_{\mathbf{Y}}$ is its complement, which in particular contains those subrees have some $Y \in \mathbf{Y} \setminus \mathbf{Z}$. Or, more formally, we define

$$\mathcal{C}_{\mathbf{X}} := \bigcup_{\substack{i \in \{1,\ldots,n\} \\ (\cup \mathcal{C}_i) \cap (\mathbf{X} \setminus \mathbf{Z}) \neq \emptyset}} \mathcal{C}_i \qquad \text{and} \qquad \mathcal{C}^+_{\mathbf{Y}} := \bigcup_{\substack{i \in \{1,\ldots,n\} \\ (\cup \mathcal{C}_i) \cap (\mathbf{X} \setminus \mathbf{Z}) = \emptyset}} \mathcal{C}_i \quad .$$

Let $\mathcal{X}_{\mathbf{X}} := \cup \mathcal{C}_{\mathbf{X}}$ set of all variables appearing in the clusters $\mathcal{C}_{\mathbf{X}}$; symmetrically, define $\mathcal{X}^+_{\mathbf{Y}} := \cup \mathcal{C}^+_{\mathbf{Y}}$.

We claim that $\mathcal{X}_{\mathbf{X}} \cap \mathcal{X}^+_{\mathbf{Y}} \subset \mathbf{Z}$. Choose any variable $U \in \mathcal{X}_{\mathbf{X}} \cap \mathcal{X}^+_{\mathbf{Y}}$. From the definitions of $\mathcal{X}_{\mathbf{X}}$ and $\mathcal{X}^+_{\mathbf{Y}}$, this means U is a member of some cluster $C \in \mathcal{C}_{\mathbf{X}}$, and also a member of a cluster $D \in \mathcal{C}^+_{\mathbf{Y}}$. Recall that the clusters of each disjoint subtree $\mathcal{T}_i$ either fall entirely within $\mathcal{C}_{\mathbf{X}}$ or entirely within $\mathcal{C}^+_{\mathbf{Y}}$ by construction. This means that C and D, which are on opposite sides of the partition, must have come from distinct subtrees. So, some edge $e = (C'{-}D') \in \mathcal{T}$ along the (unique) path from C to D must have been removed when forming $\mathcal{T}'$, which by the definition of $\mathcal{T}'$, means that $(C' \cap D') \subset Z$. But by the running intersection property (clique tree property 2), every cluster along the path from C to D must contain $C \cap D$—in particular, this must be true of both C' and D'. Therefore,

$$U \in C \cap D \subset C' \cap D' \subset \mathbf{Z}.$$

So $\mathcal{X}_{\mathbf{X}} \cap \mathcal{X}^+_{\mathbf{Y}} \subset \mathbf{Z}$, as promised. We will rather use it in the equivalent form $(\mathcal{X}_{\mathbf{X}} \cap \mathcal{X}^+_{\mathbf{Y}}) \cup \mathbf{Z} = \mathbf{Z}$.

Next, since $(\mathcal{C}, \mathcal{T})$ is a tree decomposition of $\mathcal{A}$, each hyperarc $a \in \mathcal{A}$ can be assigned to some cluster C_a that contains all of its variables; this allows us to lift the cluster partition $\mathcal{C} = \mathcal{C}_{\mathbf{X}} \sqcup \mathcal{C}^+_{\mathbf{Y}}$ to a partition $\mathcal{A} = \mathcal{A}_{\mathbf{X}} \sqcup \mathcal{A}^+_{\mathbf{Y}}$ of

edges, and consequently, a partition of PDGs $\mathcal{M} = \mathcal{M}_{\mathbf{X}} + \mathcal{M}^{+}_{\mathbf{Y}}$. Concretely: let $\mathcal{M}_{\mathbf{X}}$ be the sub-PDG of $\mathcal{M}$ induced by restricting to the variables $\mathcal{X}_{\mathbf{X}} \subseteq \mathcal{X}$ arcs $\mathcal{A}_{\mathbf{X}} = \{a \in \mathcal{A} : C_a \in \mathcal{C}_{\mathbf{X}}\} \subseteq \mathcal{A}$; define $\mathcal{M}^{+}_{\mathbf{Y}}$ symmetrically. (To be explicit: the other data of $\mathcal{M}_{\mathbf{X}}$ and $\mathcal{M}^{+}_{\mathbf{Y}}$ are given by restricting each of $\{\mathbb{P}, \boldsymbol{\alpha}, \boldsymbol{\beta}\}$ to $\mathcal{A}_{\mathbf{X}}$ and $\mathcal{A}^{+}_{\mathbf{Y}}$, respectively.)

This partition of $\mathcal{M}$ allows us to use the PDG Markov property. Suppose for some $\gamma > 0$ that $\mu \in [\![\mathcal{M}]\!]^{*}_{\gamma} = [\![\mathcal{M}_{\mathbf{X}} + \mathcal{M}_{2}]\!]^{*}_{\gamma}$. We can then apply Theorem 6, to find that $\mathcal{X}_{\mathbf{X}}$ and $\mathcal{X}^{+}_{\mathbf{Y}}$ are independent given $\mathcal{X}_{\mathbf{X}} \cap \mathcal{X}^{+}_{\mathbf{Y}}$. Then, we use standard standard properties of random variable independence [CIRV1-5 of Halpern, 2017, Theorem 4.4.4] to find that μ must satisfy:

$$
\begin{aligned}
& \mathcal{X}_{\mathbf{X}} \perp\!\!\!\perp \mathcal{X}^{+}_{\mathbf{Y}} \mid \mathcal{X}_{\mathbf{X}} \cap \mathcal{X}^{+}_{\mathbf{Y}} \\
\implies \quad & (\mathcal{X}_{\mathbf{X}} \setminus \mathbf{Z}) \perp\!\!\!\perp (\mathcal{X}^{+}_{\mathbf{Y}} \setminus \mathbf{Z}) \mid (\mathcal{X}_{\mathbf{X}} \cap \mathcal{X}^{+}_{\mathbf{Y}}) \cup \mathbf{Z} && \big[\text{CIRV3}\big] \\
\implies \quad & (\mathbf{X} \setminus \mathbf{Z}) \perp\!\!\!\perp (\mathbf{Y} \setminus \mathbf{Z}) \mid (\mathcal{X}_{\mathbf{X}} \cap \mathcal{X}^{+}_{\mathbf{Y}}) \cup \mathbf{Z} && \big[\text{by CIRV2, as } \mathbf{X} \subseteq \mathcal{X}_{\mathbf{X}} \text{ and } \mathbf{Y} \subseteq \mathcal{X}^{+}_{\mathbf{Y}}\big] \\
\implies \quad & (\mathbf{X} \setminus \mathbf{Z}) \perp\!\!\!\perp (\mathbf{Y} \setminus \mathbf{Z}) \mid \mathbf{Z} && \big[\text{since } (\mathcal{X}_{\mathbf{X}} \cap \mathcal{X}^{+}_{\mathbf{Y}}) \cup \mathbf{Z} = \mathbf{Z}\big] \\
\iff \quad & \mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z} && \big[\text{standard; e.g., Exercise 4.18 of Halpern [2017]}\big]
\end{aligned}
$$

Using only the PDG Markov property, we have now shown that every independence modeled by the Markov Network G also holds in every distribution $\mu \in [\![\mathcal{M}]\!]^{*}_{\gamma}$. Moreover, G is chordal (as we will prove momentarily), and is well-known that distributions that have the independencies of a chordal graph can be can be represented by clique trees [Koller and Friedman, 2009, Theorem 4.12]. Therefore, there is a clique tree $\boldsymbol{\mu}$ representing every $\mu \in [\![\mathcal{M}]\!]^{*}_{\gamma}$.

Claim 2.1. *G is chordal.*

Proof. Suppose that G contains a loop $X - Y - Z - W - X$. Suppose further, for contradiction, that neither X and Z nor Y and W share a cluster. Given a variable V, it is easy to see that property (2) of the tree decomposition ensures that the subtree $\mathcal{T}(V) \subseteq \mathcal{T}$ induced by the clusters $C \in \mathcal{C}$ that contain V, is connected. By assumption, $\mathcal{T}(Y)$ and $\mathcal{T}(W)$ must be disjoint. There is an edge between Y and Z, so some cluster must contain both variables, meaning $\mathcal{T}(Y) \cap \mathcal{T}(Z)$ is non-empty. Similarly, $\mathcal{T}(Z) \cap \mathcal{T}(W)$ is non-empty because of the edge between Z and W. This creates an (indirect) connection in $\mathcal{T}$ between $\mathcal{T}(Y)$ and $\mathcal{T}(W)$. Because $\mathcal{T}$ is a tree, and $\mathcal{T}(Y) \cap \mathcal{T}(W) = \emptyset$, every path from a cluster $C_1 \in \mathcal{T}(Y)$ to a cluster $C_2 \in \mathcal{T}(W)$ must pass through $\mathcal{T}(Z)$, which is not part of $\mathcal{T}(Y)$ or $\mathcal{T}(W)$. Now, $\mathcal{T}(X)$ and $\mathcal{T}(Y)$ intersect as well, meaning that, for any $C \in \mathcal{T}(X)$, there is a (unique) path from C to that point of intersection, then across edges of $\mathcal{T}(Y)$, then edges of $\mathcal{T}(Z)$, and finally connects to the clusters of $\mathcal{T}(W)$. And also, since $\mathcal{T}$ is a tree, that path must be unique. The problem is that there is also an edge between X and W, so there's some cluster that contains X and W; let's call it C_0. It's distinct from the cluster D_0 that contains Z and W, since no cluster contains both X and Z by assumption. The unique path from C_0 to D_0 intersects with $\mathcal{T}(Y)$. But now $W \in C_0 \cap D_0$, and by the running intersection property, every node along this unique path must contain W as well. But this contradicts our assumption that W is disjoint from Y! So G is chordal. □

□

Proposition 7. *If $(\boldsymbol{\mu}, \mathbf{u})$ is a solution to (11), then*

(a) $\boldsymbol{\mu}$ is a calibrated, with $\Pr_{\boldsymbol{\mu}} \in [\![\mathcal{M}]\!]^{*}_{0}$, and
(b) the objective of (11) evaluated at $\mathbf{u}$ equals $\langle\!\langle \mathcal{M} \rangle\!\rangle_{0}$.

Proof. The final constraints alone are enough to ensure that $\boldsymbol{\mu}$ is calibrated. Much like before, the exponential conic constraints tell us that

$$\forall (a, s, t) \in \mathcal{V}\mathcal{A}. \quad u_{a,s,t} \geq \mu_{C_a}(s, t) \log \frac{\mu_{C_a}(s, t)}{\mu_{C_a}(s) \mathbb{P}_a(t|s)}$$

and they hold with equality (at least at those indices where $\beta_a > 0$) because $\mathbf{u}$ is optimal. So

$$\begin{aligned}\sum_{(a,s,t)\in\mathcal{VA}} \beta_a u_{a,s,t} &= \sum_{(a,s,t)\in\mathcal{VA}} \beta_a \mu_{C_a}(s,t) \log \frac{\mu_{C_a}(s,t)}{\mu_{C_a}(s)\mathbb{P}_a(t|s)} \\ &= \sum_a \beta_a \sum_{(s,t)\in\mathcal{V}a} \mu_{C_a}(s,t) \log \frac{\mu_{C_a}(s,t)}{\mu_{C_a}(s)\mathbb{P}_a(t|s)} \\ &= \mathit{OInc}_{\mathcal{M}}(\Pr_{\boldsymbol{\mu}}).\end{aligned}$$

Because $\boldsymbol{\mu}$ is optimal, it is the choice of calibrated clique tree that minimizes this quantity. By Corollary 6.1, the distribution $[\![\mathcal{M}]\!]^*$ can be represented by such a clique tree, and by Richardson and Halpern [2021, Prop. 3.4], this distribution minimizes $\mathit{OInc}_{\mathcal{M}}$. All this is to say that there exist clique trees of this form whose corresonding distributions attain the minimum value $\mathit{OInc}_{\mathcal{M}}(\Pr_{\boldsymbol{\mu}}) = \langle\!\langle \mathcal{M} \rangle\!\rangle_0$. So $\boldsymbol{\mu}$ must be one of them, as it minimizes $\mathit{OInc}(\Pr_{\boldsymbol{\mu}})$ among such clique trees by assumption. Thus $\Pr_{\boldsymbol{\mu}} \in [\![\mathcal{M}]\!]^*_0$ and the objective value of (11) equals $\langle\!\langle \mathcal{M} \rangle\!\rangle_0$. □

Proposition 8. *If $(\boldsymbol{\mu}, \mathbf{u}, \mathbf{v})$ is a solution to (14), and $\boldsymbol{\beta} \geq \gamma\boldsymbol{\alpha}$, then $\Pr_{\boldsymbol{\mu}}$ is the unique element of $[\![\mathcal{M}]\!]^*_\gamma$, and the objective of (14) at $(\boldsymbol{\mu}, \mathbf{u}, \mathbf{v})$ equals $\langle\!\langle \mathcal{M} \rangle\!\rangle_\gamma$.*

Proof. Suppose that $(\boldsymbol{\mu}, \mathbf{u}, \mathbf{v})$ is a solution to (14). The first and fourth lines of constraints ensures that $\boldsymbol{\mu}$ is indeed a calibrated clique tree. The second line of constraints, plays exactly the same role that it did in the previous problems, most directly in the variant (11) for $\gamma = 0$. In particular, it tells says

$$\forall (a,s,t) \in \mathcal{VA}. \quad u_{a,s,t} \geq \mu_{C_a}(s,t) \log \frac{\mu_{C_a}(s,t)}{\mu_{C_a}(s)\mathbb{P}_a(t|s)}$$

as before, this holds with equality (at least at those indices where $\beta_a > \alpha_a\gamma$) because $\mathbf{u}$ is optimal. Because $\boldsymbol{\beta} \geq \gamma\alpha$ by assumption, either $\beta_a > \gamma\alpha_a$ or the two are equal, for every $a \in \mathcal{A}$. Either way, the argument used at this point in the proof of Proposition 7 goes through, giving us:

$$\begin{aligned}\sum_{(a,s,t)\in\mathcal{VA}} (\beta_a - \alpha_a\gamma) u_{a,s,t} &= \sum_{(a,s,t)\in\mathcal{VA}} ((\beta_a - \alpha_a\gamma) \mu_{C_a}(s,t) \log \frac{\mu_{C_a}(s,t)}{\mu_{C_a}(s)\mathbb{P}_a(t|s)} \\ &= \sum_a (\beta_a - \alpha_a\gamma) \sum_{(s,t)\in\mathcal{V}a} \mu_{C_a}(s,t) \log \frac{\mu_{C_a}(s,t)}{\mu_{C_a}(s)\mathbb{P}_a(t|s)} \\ &= \sum_a (\beta_a - \alpha_a\gamma)\, \boldsymbol{D}\Big(\mu_{C_a}(S_a,T_a) \,\Big\|\, \mu_{C_a}(S_a)\,\mathbb{P}_a(T_a|S_a)\Big)\end{aligned}$$

This time, though, that's not the problem objective. In this regard, our problem (14) is more closely related to (14).

Before we get to that, we have to first bring in the final collection of exponential constraints, which show that

$$\forall C \in \mathcal{C}.\, \forall c \in \mathcal{V}(C). \quad v_{C,c} \geq \mu_C(c) \log \frac{\mu_C(c)}{\mathit{VCP}_C(c)},$$

and yet again these constraints hold with equality, for otherwise $\mathbf{v}$ would not be optimal (since we assumed $\gamma > 0$). Therefore,

$$\sum_{(C,c)\in\mathcal{VC}} v_{C,c} = \sum_{(C,c)\in\mathcal{VC}} \mu_C(c) \log \frac{\mu_C(c)}{\mathit{VCP}_C(c)} = -\,\mathrm{H}(\Pr_{\boldsymbol{\mu}}) \quad \text{by Equation (13).}$$

Now, the objective of our problem (14) is essentially the same as that of (7), so the analysis in the proof of Proposition 2 applies with only a handful of superficial modifications. Using that proof to take a shortcut, the

objective of (14) must equal

$$\begin{aligned}
&\sum_{(a,s,t)\in\mathcal{VA}}(\beta_a-\alpha_a\gamma)u_{a,s,t} \;+\gamma\sum_{(C,c)\in\mathcal{VC}} v_{C,c} \;-\sum_{(a,s,t)\in\mathcal{VA}^+}\alpha_a\gamma\,\mu_{C_a}(s,t)\log\mathbb{P}_a(t|s)\\
&=\sum_{(a,s,t)\in\mathcal{VA}}(\beta_a-\alpha_a\gamma)\mu_{C_a}(s,t)\log\frac{\mu_{C_a}(s,t)}{\mu_{C_a}(s)\mathbb{P}_a(t|s)} \;-\;\gamma\,\mathrm{H}(\mathrm{Pr}_{\boldsymbol{\mu}}) \;-\sum_{(a,s,t)\in\mathcal{VA}^+}\alpha_a\gamma\,\mu_{C_a}(s,t)\log\mathbb{P}_a(t|s)\\
&=\sum_{a\in\mathcal{A}}\beta_a\mathop{\mathbb{E}}_{\mu_{C_a}}[\log\mathbb{P}_a(T_a|S_a)]+\sum_{a\in\mathcal{A}}(\alpha_a\gamma-\beta_a)\,\mathrm{H}_{\mathrm{Pr}_{\boldsymbol{\mu}}}(T_a|S_a)-\gamma\,\mathrm{H}(\mathop{\mathrm{Pr}}_{\boldsymbol{\mu}})\\
&=[\![\mathcal{M}]\!]_\gamma(\mathrm{Pr}_{\boldsymbol{\mu}}),\quad .
\end{aligned}$$

Finally, since $\boldsymbol{\mu}$ is such that this quantity is minimized, and because its unique minimizer can be represented as a cluster tree (per Corollary 6.1), we conclude that $\boldsymbol{\mu}$ must be the cluster tree representation of it. Therefore, $\mathrm{Pr}_{\boldsymbol{\mu}}$ is the unique element of $[\![\mathcal{M}]\!]^*_\gamma$, and the objective at $(\boldsymbol{\mu},\mathbf{u},\mathbf{v})$ equals $\langle\!\langle\mathcal{M}\rangle\!\rangle_\gamma$, as promised. □

Proposition 9. *If $(\boldsymbol{\mu},\mathbf{u})$ is a solution to (15), then $\boldsymbol{\mu}$ is a calibrated clique tree and $[\![\mathcal{M}]\!]^*_{0^+}=\{\mathrm{Pr}_{\boldsymbol{\mu}}\}$.*

Proof. Suppose that $(\boldsymbol{\mu},\mathbf{u})$ is a solution to (15). The exponential cone constraints state that

$$\begin{aligned}
\forall C\in\mathcal{C}.\,\forall c\in\mathcal{V}(C).\quad u_{C,c}&\geq\mu_C(c)\log\frac{\mu_C(c)}{k_{C,c}\,VCP_C(c)}\\
&=\mu_C(c)\log\frac{\mu_C(c)}{VCP_C(c)}-\mu_C(c)\log\prod_{a\in\mathcal{A}_C}\nu_C(T_a(c)|S_a(c))^{\alpha_a}\\
&=\mu_C(c)\log\frac{\mu_C(c)}{VCP_C(c)}-\mu_C(c)\sum_{a\in\mathcal{A}_C}\alpha_a\log\nu_C(T_a(c)|S_a(c)),
\end{aligned}$$

and once again this holds with equality, as each $u_{C,c}$ is minimal with this property. The third line of constraints

$$\forall a\in\mathcal{A}.\;\;\mu_{C_a}(S_a,T_a)\nu_{C_a}(S_a)=\mu_{C_a}(S_a)\nu_{C_a}(S_a,T_a)$$

and the assumption that $\mathrm{Pr}_{\boldsymbol{\nu}}\in[\![\mathcal{M}]\!]^*_0$, suffice to ensure that $\mathrm{Pr}_{\boldsymbol{\mu}}\in[\![\mathcal{M}]\!]^*_0$ by Proposition 3. They also allow us to replace each $\nu_{C_a}(T_a(c)|S_a(c))$ with $\nu_{C_a}(T_a(c)|S_a(c))$, in cases where $S_a(c)\neq 0$. Therefore, we calculate the objective to be:

$$\begin{aligned}
\mathbf{1}^\mathsf{T}\mathbf{u}&=\sum_{C\in\mathcal{C}}\sum_{c\in\mathcal{V}(C)}\left(\mu_C(c)\log\frac{\mu_C(c)}{VCP_C(c)}-\mu_C(c)\sum_{a\in\mathcal{A}_C}\alpha_a\log\nu_C(T_a(c)|S_a(c))\right)\\
&=\sum_{C\in\mathcal{C}}\sum_{c\in\mathcal{V}(C)}\mu_C(c)\log\frac{\mu_C(c)}{VCP_C(c)}-\sum_{C\in\mathcal{C}}\sum_{c\in\mathcal{V}(C)}\mu_C(c)\sum_{a\in\mathcal{A}}\mathbb{1}[C=C_a]\alpha_a\log\nu_C(T_a(c)|S_a(c))\\
&=-\,\mathrm{H}(\mathrm{Pr}_{\boldsymbol{\mu}})-\sum_{a\in\mathcal{A}}\alpha_a\sum_{C\in\mathcal{C}}\mathbb{1}[C=C_a]\sum_{c\in\mathcal{V}(C)}\mu_C(c)\log\nu_C(T_a(c)|S_a(c))\qquad[\text{by (13)}]\\
&=-\,\mathrm{H}(\mathrm{Pr}_{\boldsymbol{\mu}})-\sum_{a\in\mathcal{A}}\alpha_a\sum_{c\in\mathcal{V}(C)_a}\mu_{C_a}(c)\log\nu_{C_a}(T_a(c)|S_a(c))\\
&=-\,\mathrm{H}(\mathrm{Pr}_{\boldsymbol{\mu}})-\sum_{a\in\mathcal{A}}\alpha_a\sum_{c\in\mathcal{V}(C)_a}\mu_{C_a}(c)\log\mu_{C_a}(T_a(c)|S_a(c))\qquad\left[\text{since }\mu_{C_a}(S_a(c))>0\text{ whenever }\mu_{C_a}(c)>0\right]\\
&=-\,\mathrm{H}(\mathrm{Pr}_{\boldsymbol{\mu}})+\sum_{a\in\mathcal{A}}\alpha_a\,\mathrm{H}_{\mathrm{Pr}_{\boldsymbol{\mu}}}(T_a|S_a)\\
&=SInc_{\mathcal{M}}(\mathrm{Pr}_{\boldsymbol{\mu}}).
\end{aligned}$$

To summarize: $\mathrm{Pr}_{\boldsymbol{\mu}}$ minimizes $SInc_{\mathcal{M}}(\mathrm{Pr}_{\boldsymbol{\mu}})$ among calibrated clique trees with condtional marginals matching those of $\boldsymbol{\nu}$. Since we know that there is a unique distribution that minimizes $SInc_{\mathcal{M}}$ among the elements $[\![\mathcal{M}]\!]^*_0$, and also

that this distribution can be represented by a clique tree (by Corollary 6.1), we conclude that $\boldsymbol{\mu}$ must represent this distribution. Thus, $\Pr_{\boldsymbol{\mu}} = [\![\mathcal{M}]\!]^*$ as desired. □

Lemma 3. *Fix integers $n_\mathsf{o}, n_\mathsf{e} \in \mathbb{N}$, and let $n := 3n_\mathsf{e} + n_\mathsf{o}$. Suppose that $K = \mathbb{R}^{n_\mathsf{o}}_{\geq 0} \times K^{n_\mathsf{e}}_{\text{exp}} \subset \mathbb{R}^n$ is a product cone, consisting of n_o copies of the non-negative orthant and n_e copies of the exponential cone. If, for $c \in [-1,1]^n$, $b \in [-1,1]^m$, and $A \in [-1,1]^{m\times n}$, the exponential conic program*

$$\underset{\mathbf{x}\in K}{\text{minimize}} \quad \mathbf{c}^\mathsf{T}\mathbf{x} \quad \text{subject to} \quad A\mathbf{x} = \mathbf{b}, \tag{3}$$

is strictly feasible (i.e., if there exists $\mathbf{x} \in \operatorname{int} K$ such that $A\mathbf{x} = \mathbf{b}$) as is its dual problem

$$\underset{\mathbf{s}\in K^*,\, \mathbf{y}\in\mathbb{R}^m}{\text{maximize}} \quad \mathbf{b}^\mathsf{T}\mathbf{y} \quad \text{subject to} \quad A^\mathsf{T}\mathbf{y} + \mathbf{s} = \mathbf{c},$$

(i.e, if there exists $\mathbf{s} \in \operatorname{int} K_*$ such that $A^\mathsf{T}\mathbf{y} + \mathbf{s} = \mathbf{c}$), then both can be simultaneously solved to precision ϵ in $O(n(m+n)^\omega \log \frac{n+m}{\epsilon})$ time, where ω is the smallest exponent such that a linear system of k variables and equations can be solved in $O(k^\omega)$ time. Furthermore, MOSEK solves this problem in $O(n(m+n)^3 \log \frac{n+m}{\epsilon})$ time.

Proof. For this, we begin by appealing to the algorithm and analysis of Badenbroek and Dahl [2021], threading details through for this specific choice of cone K. To finish the proof, however, we will also need to supplement that analysis with some other well-established results of Nesterov et al. [1999] that the authors were no doubt familiar with, but did not bother referencing.

First, we'll need some background material from convex optimization. A *logarithmically homogeneous self-concordant barrier* with parameter ν (ν-LHSCB) for a cone K is a thrice differentiable strictly convex function $F : \operatorname{int} K \to \mathbb{R}$ satisfying $F(tx) = F(x) - \nu \log t$ for all $t > 0$ and $x \in \operatorname{int} K$. In some sense, the point of such a barrier function is to augment the optimization objective so that we remain within the cone during the optimization process.

For the positive orthant cone $\mathbb{R}_{\geq 0}$, the function $x \mapsto -\log x$ is a 1-LHSCB. We now fill in some background facts about exponential cones. The *dual* the exponential cone is

$$\begin{aligned} K^*_{\text{exp}} &:= \big\{(s_1,s_2,s_3) \in \mathbb{R}^3 \;:\; \forall (x_1,x_2,x_3) \in K_{\text{exp}}.\; x_1s_1 + x_2s_2 + x_3s_3 \geq 0\big\} \\ &= \big\{(s_1,s_2,s_3) \;:\; -s_1 \log(-s_1/s_3) + s_1 - s_2 \leq 0,\; s_1 \leq 0,\; s_3 \geq 0\big\}. \end{aligned}$$

Consider points $x = (x_1,x_2,x_3) \in K_{\text{exp}}$. The function

$$F_{\text{exp}}(x) := -\log\left(x_2 \log \frac{x_1}{x_2} - x_3\right) - \log x_1 x_2 \tag{4}$$

is a 3-LHSCB for K_{exp}, since

$$\begin{aligned} F_{\text{exp}}(tx) &= -\log\left(tx_2 \log \frac{tx_1}{tx_2} - tx_3\right) - \log(t^2 x_1 x_2) \\ &= -\log\left(t\big(\log \frac{x_1}{x_2} - x_3\big)\right) - \log(x_1x_2) - 2\log t \\ &= F_{\text{exp}}(x) - 3\log t \end{aligned}$$

Such barrier functions can be combined to act on product cones by summation. Concretely, suppose that for each $i \in \{1,\dots,k\}$, we have a ν_i-LHSCB $F_i : \operatorname{int} K_i \to \bar{\mathbb{R}}$. Then, for $x = (x_i)_{i=1}^k \in \prod_i K_i$, the function $F(x) := \sum_{i=1}^k F_i(x_i)$ is a $(\sum_i \nu_i)$-LHSCB for $\prod_i K_i$, since

$$F(tx) = \sum_{i=1}^k F_i(tx_i) = \sum_{i=1}^k (F(x_i) - \nu_i \log t) = F(x) - \sum_{i=1}^k \nu_i.$$

In this way, our product cone $K = \mathbb{R}^{n_\mathsf{o}}_{\geq 0} \times K^{n_\mathsf{e}}_{\text{exp}}$ admits a LHSCB F with parameter $\nu = n_\mathsf{o} + 3n_\mathsf{e} = n$. Furthermore can be evaluated in $O(n)$ time, as can each component of its gradient $F'(x)$ and Hessian $F''(x) \in \mathbb{R}^{n\times n}$ at x, all of which can be expressed analytically. In addition, the convex conjugate of F also has a known analytic form.

Generally speaking, the idea behind primal-dual interior point methods [Nesterov and Nemirovskii, 1994] such as the one behind MOSEK, is to maintain both a point $x \in K$ and a dual point $s \in K_*$ (as well as $y \in \mathbb{R}^m$) and iteratively update them, as we slowly relax the barrier and approach a point on the boundary of the cone. The quantity $\mu(z) := {}^{\langle s,x\rangle}/_{\nu} \geq 0$, called the complementarity gap, is a measure of how close the process is to converging.

Because the initial points may not satisfy the constraints, instead the standard algorithms work with "extended points" $\bar{x} = (x, \tau)$ and $\bar{s} = (s, \kappa)$, for which the analogous complementarity gap is $\mu^{\mathsf{e}}(\bar{x}, \bar{s}) := (\langle x, s\rangle + \kappa\tau)/(\nu + 1)$. Altogether, the data at each iteration may be summarized as a point $z = (y, x, \tau, s, \kappa) \in \mathbb{R} \times (K \times \mathbb{R}_{\geq 0}) \times (K_* \times \mathbb{R}_{\geq 0})$. The primary object of interest is then something called the *homogenous self-dual* model. Originally due to Nesterov et al. [1999] and also used by others [Skajaa and Ye, 2015], it can be defined as a linear operator:

$$G : \bar{\mathbb{R}}^{m+2n+2} \to \bar{\mathbb{R}}^{n+m+1}$$
$$G(y, x, \tau, s, \kappa) := \begin{bmatrix} 0 & A & -b \\ -A^{\mathsf{T}} & 0 & c \\ b^{\mathsf{T}} & -c^{\mathsf{T}} & 0 \end{bmatrix} \begin{bmatrix} y \\ x \\ \tau \end{bmatrix} - \begin{bmatrix} 0 \\ s \\ k \end{bmatrix}.$$

The reason for our interest is that if z is such that $G(z) = 0$ and $\tau > 0$, then ${}^{(x)}/_{\tau}$ is a solution to the primal problem, and ${}^{(y,s)}/_{\tau}$ is a solution to the dual problem [Skajaa and Ye, 2015, Lemma 1], while if $G(z) = 0$ and $\kappa > 0$, then at least one of the two problems is infeasible.

We now are in a better position to describe the algorithm. According to the MOSEK documentation [Dahl and Andersen, 2022], for the exponential cone, begins with an initial point

$$\mathbf{v} := (1.291, 0.805, -0.828) \in (K_{\text{exp}} \cap K^*_{\text{exp}})$$

for this particular cone K, the algorithm begins at the initial point

$$z_0 := (y_0, x_0, \tau_0, s_0, \kappa_0) \qquad \text{where} \quad x_0 = s_0 = (\overbrace{1, \ldots, 1}^{n_{\mathsf{o}} \text{ copies}}, \overbrace{\mathbf{v}, \ldots, \mathbf{v}}^{n_{\mathsf{e}} \text{ copies}}) \in (\mathbb{R}_{\geq 0})^{n_{\mathsf{o}}} \times (K_{\text{exp}} \cap K^*_{\text{exp}})^{n_{\mathsf{e}}},$$
$$y_0 = \mathbf{0} \in \mathbb{R}^m, \quad \tau_0 = \kappa_0 = 1.$$

At each iteration, the first step is to predict a direction for which Badenbroek and Dahl [2021] compute a scaling matrix W. To describe it, we first need to define *shadow iterates*

$$\tilde{x} := -F'_*(s) \qquad \text{and} \qquad \tilde{s} := -F'(x).$$

which are in a sense reflections of s and x across their barrier functions, and can be computed in in $O(n)$ time. The analogous notion of complementarity can then be defined as $\tilde{\mu}(z) := {}^{\langle \tilde{x}, \tilde{s}\rangle}/_{\nu}$. The scaling matrix, which we do not interpret here, can then be calculated as:

$$W := \mu F''(x) + \frac{ss^{\mathsf{T}}}{\nu\mu} - \frac{\mu\tilde{s}\tilde{s}^{\mathsf{T}}}{\nu} + \frac{(s - \mu\tilde{s})(s - \mu\tilde{s})^{\mathsf{T}}}{(s - \mu\tilde{s})^{\mathsf{T}}(x - \mu\tilde{x})} - \frac{\mu[F''(x)\tilde{x} - \tilde{\mu}\tilde{s}][F''(x)\tilde{x} - \tilde{\mu}\tilde{s}]^{\mathsf{T}}}{\tilde{x}^{\mathsf{T}}F''(x)\tilde{x} - \nu\tilde{\mu}^2} \tag{5}$$

Doing so requires $O(n^2)$ steps (although it may be parallelized). The first four terms clearly require $O(n^2)$ steps, since each one is an outer product resulting in a $n \times n$ matrix. The last term computes a matrix-vector product (which requires $O(n^2)$ steps), and computes an outer product with the resulting vector, which takes $O(n^2)$ steps as well.

The next step involves finding a solution $\Delta z^{\text{aff}} = (\cdots)$ to the system of equations

$$G(\Delta z^{\text{aff}}) = -G(z) \tag{6a}$$
$$\tau\Delta\kappa^{\text{aff}} + \Delta\tau^{\text{aff}} = -\tau\kappa \tag{6b}$$
$$W\Delta x^{\text{aff}} + \Delta s^{\text{aff}} = -s \tag{6c}$$

(6a-c) describe a system of $(n + m + 1) + 1 + (n) = 2n + m + 2$ equations and equally many unknowns, and solved in $O((n + m)^\omega)$ steps. It may be possible to exploit the sparsity of G to do better.

The next step is to center that search direction so that it lies on the central path. This is done by finding a solution Δz^{cen} to

$$G(\Delta z^{\text{cen}}) = G(z) \tag{7a}$$
$$\tau\Delta\kappa^{\text{cen}} + \kappa\Delta\tau^{\text{cen}} = \mu^e \tag{7b}$$
$$W\Delta x^{\text{cen}} + \Delta s^{\text{cen}} = \mu^e \tilde{s} \tag{7c}$$

which again can be done in $O((n+m)^3)$ steps with Gaussian elimination, or with a fancier solver in $O((n+m)^2.332)$ steps. The two updates are then applied to the current point z to obtain

$$z_+ = (y_+, x_+, \tau_+, s_+, \kappa_+) := z + \alpha(\Delta z^{\text{aff}} + \gamma\Delta z^{\text{cen}}).$$

Finally, a "correction step", which is the primary innovation of Badenbroek and Dahl [2021] and used in MOSEK's algorithm, is a third direction Δz_+^{cor}, which is found by solving the system of equations

$$G(\Delta z^{\text{cor}}) = 0 \tag{8a}$$
$$\tau_+\Delta\kappa^{\text{cor}} + \kappa_+\Delta\tau^{\text{cor}} = 0 \tag{8b}$$
$$W_+\Delta {x_+}^{\text{cor}} + \Delta s^{\text{cen}} = \mu^e \tilde{s} \tag{8c}$$

where W_+ is defined the same way that W is, except that it uses the components of z_+ instead of z. After adding the correction step Δz_+^{cor} to z, we repeat the entire process. The full algorithm, then, is summarized as follows:

$z \leftarrow (y_0, x_0, \tau_0, s_0, \kappa_0)$;
while **do**
  Compute scaling matrix W as in (5);
  Find the solution Δz^{aff} to (6a-c), and the solution Δz^{cen} to (7a-c);
  $z_+ \leftarrow z + \alpha(\Delta z^{\text{aff}} + \gamma\Delta z^{\text{cen}})$;
  Compute the saling matrix W_+;
  Find the solution Δz_+^{cor} to (8a-c);
  $z \leftarrow z_+ + \Delta z_+^{\text{cor}}$;
end while

We have verified that each iteration of this process can be done in $O((n+m)^\omega))$ time. Their main result [Badenbroek and Dahl, 2021, Theorem 3], states that for every $\epsilon \in (0, 1)$, the algorithm results in a solution z satisfying

$$\mu^{\text{e}}(z) \le \epsilon \qquad \text{and} \qquad \|G(z)\| \le \epsilon\|G(z_0)\|$$

in $O(n\log(1/\epsilon))$ iterations, for a total cost of $O(n(m+n)^3\log(1/\epsilon))$ time with Gaussian elimination, or $O(n(m+n)^{2.332}\log(1/\epsilon))$ time using the linear solver with best known asymptotatic complexity as of 2022 Duan et al. [2022].

Verifying that the solution is approximately optimal. What we have at this point is not quite enough: simply because the residual quantity $G(z)$ is approximately zero (so that we have approximately solved the homogenous model), does not mean that we've approximately solved the original problem. Specifically, it's entirely possible on the surface that the parameter τ goes to zero at the same rate as everything else, and the quantity (x/τ) does not converge to a solution to the primal problem. To address this issue, we must also trace the analysis of the seminal work of Nesterov et al. [1999], who use slightly different quantities, conflicting with the notation we have been using thus far.

Following Nesterov et al. [1999, pg. 231], fix an initial point z_0, and let *shifted feasible set* $\mathcal{F} := \{z \in \mathbb{R} \times K \times \mathbb{R}_{\ge 0} \times K^* \times \mathbb{R}_{\ge 0} : G(z) = G(z_0)\}$ be the collection of all points that have the same residual as z_0. Nesterov, Todd, and Ye also refer to a complementary gap by $\mu(z)$ and define it identically, but the meaning of this parameter is different, because the set $\mathcal{F}$ on which it's defined is quite distinct from (if closely related to) the iterates of Badenbroek and Dahl's algorithm. In the service of clarity, will call this quantity $\mu^{\mathsf{N}}(z^{\mathsf{N}})$, for $z^{\mathsf{N}} = (y^{\mathsf{N}}, x^{\mathsf{N}}, \tau^{\mathsf{N}}, s^{\mathsf{N}}, \kappa^{\mathsf{N}}) \in \mathcal{F}$.

Although we made a point of emphasizing that the two are distinct, the actual relationship between them is straightforward. Let $z = (y, x, \tau, s, \kappa)$ be the final output of Badenbroek and Dahl [2021]. In proving their main

theorem, they also prove that $G(z) = \epsilon G(z_0)$, and $\mu^{\mathsf{e}} = \epsilon$; because G is linear, we know that $G(z/\epsilon) = G(z_0)$. This means that $z^{\mathsf{N}} := z/\epsilon \in \mathcal{F}$. Therefore,

$$\mu^{\mathsf{N}}(z^{\mathsf{N}}) = \frac{1}{\nu+1}\left(\left\langle \frac{s}{\epsilon}, \frac{x}{\epsilon}\right\rangle + \frac{\tau}{\epsilon}\frac{\kappa}{\epsilon}\right) = \frac{1}{\epsilon^2}\mu^{\mathsf{e}}(z) = \frac{1}{\epsilon}.$$

So, roughly speaking, μ^{N} and μ^{e} are reciprocals. Badenbroek and Dahl also prove that, every iterate z satisfies their assumption (A2): for a fixed constant β (equal to 0.9 in their analysis), $\beta\mu^{\mathsf{e}}(z) \le \tau\kappa$. Consequently, it happens that the same inequality holds with Nesterov's notation:

$$\tau^{\mathsf{N}}\kappa^{\mathsf{N}} = \frac{\tau}{\epsilon}\frac{\kappa}{\epsilon} = \frac{\tau\kappa}{\epsilon^2} \ge \frac{\beta\epsilon}{\epsilon^2} = \frac{\beta}{\epsilon} = \beta\mu^{\mathsf{N}}(z^{\mathsf{N}}).$$

This witnesses that $z^{\mathsf{N}} = \frac{z}{\epsilon}$ satisfies equation (81) of Nesterov et al., which allows us to apply one of their main theorems, which addresses these issues. Supposing that the orignal problem is solvable, let (x^*, s^*) be any solution to the primal and dual problems, and define the value $\psi := 1 + \langle s_0, x^*\rangle + \langle s^*, x_0\rangle \ge 1$, which depends only on the problem and the choice of initialization. Then Theroem 1, part 1 of Nesterov, Todd, and Ye, allows us to conclude that

$$\frac{\kappa}{\epsilon} \le \psi \quad \text{and} \quad \frac{\tau}{\epsilon} \ge \frac{\beta}{\epsilon\psi} \qquad \Longleftrightarrow \qquad \kappa \le \epsilon\psi \quad \text{and} \quad \tau \ge \frac{\beta}{\psi}.$$

Finally, the original theorem guarantees that $\|G(x)\| \le \epsilon\|G(z_0)\|$, meaning that

$$\left\|A\left(\frac{x}{\tau}\right) - b\right\|\tau \;+\; \left\|A^{\mathsf{T}}\left(\frac{y}{\tau}\right) - \frac{s}{\tau} - c\right\|\tau \;+\; \left\|b^{\mathsf{T}}\left(\frac{y}{\tau}\right) - c^{\mathsf{T}}\left(\frac{x}{\tau}\right) - \frac{\kappa}{\tau}\right\|\tau \le \epsilon\|G(z_0)\|$$

Since the euclidean norm is an upper bound on the deviation in any component ($\|v\| := \sqrt{\sum_i v_i^2} \ge \sqrt{\max_i v_i^2} = \max_i v_i =: \|v\|_\infty$), this means that in light of our bound on τ above, we have

$$\left\|A\left(\frac{x}{\tau}\right) - b\right\|_\infty \;+\; \left\|A^{\mathsf{T}}\left(\frac{y}{\tau}\right) + \frac{s}{\tau} - c\right\|_\infty \;+\; \left\|b^{\mathsf{T}}\left(\frac{y}{\tau}\right) - c^{\mathsf{T}}\left(\frac{x}{\tau}\right) - \frac{\kappa}{\tau}\right\|_\infty \le \epsilon\frac{\beta\|G(z_0)\|}{\psi}.$$

The first two components show that the total constraint violation (in the primal and dual problems, respectively) is at most $\epsilon\beta/\psi\|G(z_0)\|$. Meanwhile, the final component shows that the duality gap $gap = b^{\mathsf{T}}(\frac{y}{\tau}) - c^{\mathsf{T}}(\frac{x}{\tau})$, which is positive and an upper bound on the difference between the objective at x/τ and the optimal objective value, satisfies

$$gap \le gap + \frac{\kappa}{\tau} \le \frac{\epsilon\beta\|G(z_0)\|}{\psi}.$$

Thus x/τ is an $(\epsilon\|G(z_0)\|)$-approximate solution to the original exponential conic problem. Since also $\psi \ge 1$, we may freely drop it to get a looser bound. All that remains is to investigate $\|G(z_0)\|$, the residual norm of the initial point chosen by the MOSEK solver, which equals:

$$\|G(z_0)\| = \|Ax_0 - b\| + \|A^{\mathsf{T}}y_0 + s_0 - c\| + |c^{\mathsf{T}}x - b^{\mathsf{T}}y + 1|.$$

Making use of our assumption that every component of A, b, and c is at most one, we find that

$$\begin{aligned}
\|Ax_0 - b\|^2 &= \sum_j (\sum_i A_{j,i}(1.3) - b_j)^2 \le m(1.3n+1)^2 \in O(mn^2) &&\subset O((m+n)^3)\\
\|A^{\mathsf{T}}y_0 + s_0 - c\|^2 &= \sum_i (\sum_j (A_{j,i}) \le n(m+2)^2 \in O(nm^2) &&\subset O((m+n)^3)\\
|c^{\mathsf{T}}x - b^{\mathsf{T}}y + 1|^2 &\le (1.3n + m + 1)^2 \in O((n+m)^2) &&\subset O((n+m)^3).
\end{aligned}$$

Therefore, the residual of the initial point is $G(z_0) \in O((n+m)^{3/2})$.

To obtain a solution at most ϵ_0 away from the true solution in any coordinate, we need to select ϵ small enough that the final output of the algorithm z satisfies

$$\epsilon\|G(z_0)\| \le \epsilon_0 \qquad \Longleftrightarrow \qquad \frac{1}{\epsilon} \ge \frac{1}{\epsilon_0}\|G(z_0)\|$$

It therefore suffices to choose $\frac{1}{\epsilon} \in O(\frac{1}{\epsilon_0}(n+m)^{3/2})$, leading to $\log \frac{1}{\epsilon} = O(\log \frac{n+m}{\epsilon_0}))$ iterations.

Thus, we arrive at our total advertised asymptotic complexity of time

$$O\Big(n(n+m)^{\omega} \log \frac{n+m}{\epsilon_0}\Big).$$

In particular, to attain machine precision, we can fix ϵ_0 to be the smallest gap between numbers representable (say with 64-bit floats, leading to $\epsilon_0 = 10^{-78}$ in the worst case), and omit the dependance on ϵ_0 for the price of relatively small constant (78, for 64-bit floats). □

Having combed through all of the details of the analysis of Badenbroek and Dahl [2021] and Nesterov et al. [1999] for exponential conic programs as we have defined them, we are ready to show that this algorithm solves the problems presented in Section 4 within polynomial time.

Lemma 4. *Problem (11) can be solved to ϵ precision in time*

$$O\Big((\mathcal{VA}+\mathcal{VC})^{1+\omega}(\log \frac{|\mathcal{VA}|+|\mathcal{VC}|}{\epsilon} + \log \frac{\beta^{\max}}{\beta^{\min}}\Big) \quad \subset \quad \tilde{O}\Big((\mathcal{VA}+\mathcal{VC})^4\Big),$$

where $\beta^{\max} := \max_{a\in\mathcal{A}} \beta_a$ is the largest value of β, and $\beta^{\min} := \min_{a\in\mathcal{A}}\{\beta_a : \beta_a > 0\}$ is the smallest positive one.

Proof. Problem (11) can translated via the DCP framework to the following exponential conic program, which has:

- variables $x = (\mathbf{u}, \mathbf{v}, \mathbf{w}, \boldsymbol{\mu}) \in K_{\text{exp}}^{\mathcal{VA}} \times \mathbb{R}_{\geq 0}^{\mathcal{VC}}$, where
  - $\mathbf{u}, \mathbf{v}, \mathbf{w} \in \bar{\mathbb{R}}^{\mathcal{VA}}$ are all vectors over $\mathcal{VA}$, that at index $\iota = (a,s,t) \in \mathcal{VA}$, have components u_ι, v_ι, and w_ι, respectively;
  - $\boldsymbol{\mu} = [\mu_C(C{=}c)]_{C\in\mathcal{C},\, c\in\mathcal{V}(C)} \in \bar{\mathbb{R}}^{\mathcal{VC}}$ is a vector representation of a clique tree over clusters $\mathcal{C}$;
- constraints as follows:
  - two linear constraints for every $(a,s,t) \in \mathcal{VA}$ to ensure that

$$v_{a,s,t} = \mu_{C_a}(s,t) \qquad \Big(= \sum_{\bar{c}\in\mathcal{V}(C_a\setminus\{S_a,T_a\})} \mu_{C_a}(\bar{c},s,t)\Big)$$

$$\text{and} \qquad w_{a,s,t} = \mu_{C_a}(S_a{=}s)\,\mathbb{P}_a(T_a{=}t \mid S_a{=}s) \qquad \Big(= \mathbb{P}_a(T_a{=}t \mid S_a{=}s) \sum_{\bar{c}\in\mathcal{V}(C_a\setminus\{S_a\})} \mu_{C_a}(\bar{c},s)\Big)$$

  - for every edge $(C\text{–}D) \in \mathcal{T}$, and every value $\omega \in \mathcal{V}(C\cap D)$ of the variables that clusters C and D have in common, a linear constraint

$$\sum_{\bar{c}\in\mathcal{V}(C\setminus D)} \mu_C(\bar{c},\omega) = \sum_{\bar{d}\in\mathcal{V}(D\setminus C)} \mu_D(\bar{d},\omega)$$

  - and one constraint for each cluster $C \in \mathcal{C}$ to ensure that μ_C lies on the probability simplex, i.e.,

$$\sum_{c\in\mathcal{V}(C)} \mu_C(c) = 1.$$

Altogether this means that we have an exponential conic program in the form of Lemma 3, with $n = 3|\mathcal{VA}| + |\mathcal{VC}|$ variables, and $m = 2|\mathcal{VA}| + |\mathcal{VT}| + |\mathcal{C}|$ constraints, where $\mathcal{VT} = \{(C\text{–}D,\omega) : C\text{–}D \in \mathcal{T}, \omega \in \mathcal{V}(C\cap D)\}$. Since we can simply disregard variables whose value sets are singletons, we can assume $\mathcal{V}(C) > 1$; summing over all clusters yields $\mathcal{VC} > |\mathcal{C}|$. At the same time, since $\mathcal{VT} \leq \mathcal{VC}$, we have

$$m, n, (m+n) \in O(\mathcal{VA}, +\mathcal{VC}).$$

We now give the explicit construction of the data (A, b, c) of the exponential conic program that (11) compiles to. The variables are indexed by tuples of the form $i = (\ell, a, s, t)$ for $(a, s, t) \in \mathcal{VA}$ and $\ell \in \{u, v, w\}$, or by tuples of the form (C, c), for $c \in \mathcal{V}(C)$ and $C \in \mathcal{C}$, while the constraints are indexed by tuples of the form $j = (\ell, a, s, t)$ for $(a, s, t) \in \mathcal{VA}$ and $\ell \in \{v, w\}$, of the form $(C\text{–}D, \omega)$, for an edge $(C\text{–}D) \in \mathcal{T}$ and $\omega \in \mathcal{V}(C \cap D)$, or simply by (C), the name of a cluster $C \in \mathcal{C}$. The problem data $A = [A_{j,i}], b = [b_j], c = [c_i]$ of this program are zero, except (possibly) for the components:

$$\begin{aligned}
c_{(u,a,s,t)} &= \beta_a \\
A_{(v,a,s,t),(C,c)} &= \mathbb{1}[C{=}C_a \ \wedge \ S_a(c){=}s \ \wedge \ T_a(c){=}t] \\
A_{(w,a,s,t),(C,c)} &= \mathbb{P}_a(T_a{=}t \mid S_a{=}s)\,\mathbb{1}[C{=}C_a \ \wedge \ S_a(c){=}s] \\
A_{(w,a,s,t),(w,a,s,t)} &= -1 \\
A_{(v,a,s,t),(v,a,s,t)} &= -1 \\
A_{(C\text{–}D,\omega),(C',c)} &= \mathbb{1}[C{=}C'] - \mathbb{1}[C'{=}D] \\
A_{(C),(C,c)} &= 1 \\
b_{(C)} &= 1,
\end{aligned}$$

where $\mathbb{1}[\varphi]$ is equal to 1 if φ is true, and zero if φ is false. We note that we can equivalently divide each β_a by $\max_a \beta_a$ without affecting the problem, although this could affect the approximation accuracy by the same factor. Thus, we get another factor of

$$\log(\max\{1\} \cup \{\beta_a : a \in \mathcal{A}\}) \ \subseteq \ O(\log(1 + \max_a \beta_a)).$$

Finally, to find a point that is ϵ-close (say, in 2-norm) to the limiting point μ^* on the central path, as opposed to simply one that for which the suboptimality gap is at most ϵ, we can appeal to strong concavity of the objective function. Now, (conditional) relative entropy is 1-strongly convex, and each relative entropy term is scaled by β_a. Furthermore, we're only considering marginal conditional entropy, so this convexity may not hold in all directions. Still, if the next step direction δ is not far from the gradient, as is the case if the interior point method has nearly converged, then in that direction, the objective will be at least $(\min_a\{\beta_a : \beta_a > 0\})$-strongly convex. Therefore, by requiring an precision to an additional factor of $\min_a\{\beta_a : \beta_a > 0\}$, we can guarantee that our point is ϵ-close to μ^*, and not just in complementarity gap.

To summarize, applying Lemma 3, we find that we can solve problem (11) in time

$$O\Big(\big(|\mathcal{VA}| + |\mathcal{VC}|\big)^{1+\omega}\Big(\log\frac{|\mathcal{VA}| + |\mathcal{VC}|}{\epsilon} + \log\frac{\beta_{\max}}{\beta_{\min}}\Big)\Big) \quad \subset \tilde{O}\Big(\big(|\mathcal{VA}| + |\mathcal{VC}|\big)^4\Big).$$

□

We now quickly step through the analogous construction for problems (14) and (15), which solve the $\hat{\gamma}$-inference problem, and 0^+-inference, respectively.

Lemma 5. *Problem* (14) *is solved to precision* ϵ *in time*

$$O\Big(|\mathcal{VA} + \mathcal{VC}|^{1+\omega}\Big(\log\frac{|\mathcal{VA}| + |\mathcal{VC}|}{\epsilon} + \log(1 + \|\boldsymbol{\beta}\|_\infty) + \log\log\frac{1}{p}\Big)\Big) \quad \subset \quad \tilde{O}\Big((\mathcal{VA} + \mathcal{VC})^4\Big)$$

where p *is the smallest nonzero probability in the PDG.*

Proof. Problem (14) has

- variables $x = (\mathbf{u}, \mathbf{y}, \mathbf{w}, \ \mathbf{v}, \boldsymbol{\mu}, \mathbf{z}) \in K_{\exp}^{\mathcal{VA}} \times K_{\exp}^{\mathcal{VC}}$ where
  - $\mathbf{u}, \mathbf{y}, \mathbf{w} \in \overline{\mathbb{R}}^{\mathcal{VA}}$ are all vectors over $\mathcal{VA}$ that at index $\iota = (a, s, t) \in \mathcal{VA}$, have components u_ι, v_ι, and w_ι, respectively;

- Meanwhile, $\mathbf{v}, \boldsymbol{\mu}, \mathbf{z} \in \bar{\mathbb{R}}^{\mathcal{VC}}$ are all vectors over $\mathcal{VC}$ which at index (C, c), have components $v_{C,c}$, $\mu_C(c)$, and $z_{C,c}$, respectively. Once again, $\boldsymbol{\mu} = [\mu_C(C{=}c)]_{C \in \mathcal{C},\, c \in \mathcal{V}(C)} \in \bar{\mathbb{R}}^{\mathcal{VC}}$ is intended to be a vector representation of a clique tree.

► constraints as follows:

- two linear constraints for each $(a, s, t) \in \mathcal{VA}$, to ensure that

$$y_{a,s,t} = \mu_{C_a}(s, t) \qquad \text{and} \qquad w_{a,s,t} = \mu_{C_a}(S_a{=}s)\, \mathbb{P}_a(T_a{=}t \mid S_a{=}s),$$

- for every edge $(C\text{–}D) \in \mathcal{T}$, and every value $\omega \in \mathcal{V}(C \cap D)$ of the variables that clusters C and D have in common, a linear constraint

$$\sum_{\bar{c} \in \mathcal{V}(C \setminus D)} \mu_C(\bar{c}, \omega) = \sum_{\bar{d} \in \mathcal{V}(D \setminus C)} \mu_D(\bar{d}, \omega)$$

- for every $(a, s, t) \in \mathcal{VA}^0$, a linear constraint that ensures

$$0 = \mu_{C_a}(S_a{=}s, T_a{=}t) \qquad \Big(\;= \sum_{\bar{c} \in \mathcal{V}(C \setminus \{S_a, T_a\})} \mu_{C_a}(\bar{c}, s, t) \Big)$$

- a linear constraint for every value $c \in \mathcal{V}(C)$ of every cluster $C \in \mathcal{C}$, to ensure that

$$z_{C,c} = \mu_C(\mathit{VCP}_C(c)) \qquad \Big(\;= \sum_{\bar{c} \in \mathcal{V}(C \setminus \mathit{VCP}_C)} \mu_C(\bar{c},\, \mathit{VCP}_C(c)) \;\Big)$$

- and one constraint for each cluster $C \in \mathcal{C}$ to ensure that μ_C lies on the probability simplex, i.e.,

$$\sum_{c \in \mathcal{V}(C)} \mu_C(c) = 1.$$

So in total, there are $n = |3\mathcal{VA} + 3\mathcal{VC}|$ variables, and $m = 2|\mathcal{VA}| + |\mathcal{VT}| + |\mathcal{VA}^0| + |\mathcal{VC}| + |\mathcal{C}|$ constraints. The same arguments made in Lemma 4 show that both $n, m \in O(|\mathcal{VA} + \mathcal{VC}|)$.

Also like before, it is easy to see that the components of A and b are all at most 1. However, we will need to rescale the objective c in order for each of its components to be most 1. We can do this by dividing it by $\max\{-\beta_a \log p_a(t|s)\}_{(a,s,t) \in \mathcal{VA}} \cup \{1\}$.

Finally, to ensure that we have a solution that is ϵ-close to the end of the central path, as opposed to one that is merely ϵ-close in compelementarity gap, we must appeal to convexity. As in the proof of Lemma 4, this amounts to reducing the target accuracy by a factor of the smallest possible coefficient of strong convexity, along the next step direction. In this case, the bound is simpler: because negative entropy is (unconditionally) 1-strongly convex, and since $\boldsymbol{\beta} \geq \boldsymbol{\alpha}\gamma$, the remaining terms are convex, this could be, at worst, $\frac{1}{\gamma}$.

This gives rise to our result: problem (14) can be solved in

$$O\left(|\mathcal{VA} + \mathcal{VC}|^{1+\omega} \left(\log \frac{|\mathcal{VA} + \mathcal{VC}|}{\epsilon} + \log \frac{1}{\gamma} \left(1 + \max_{(a,s,t) \in \mathcal{VA}} \beta_a \log \frac{1}{\mathbb{P}_a(t|s)}\right)\right)\right)$$
$$\subset O\left(|\mathcal{VA} + \mathcal{VC}|^{1+\omega} \left\{\log \frac{|\mathcal{VA} + \mathcal{VC}|}{\epsilon} + \log \frac{\beta^{\max}}{\gamma} + \log \log \frac{1}{p}\right\}\right)$$

operations, where p is the smallest nonzero probability in the PDG, and $\beta^{\max}$ is the largest confidence in the PDG larger than 1. □

Lemma 6. *Problem (15) is solved to precision ϵ in*

$$O\Big(|\mathcal{VC}||\mathcal{VA} + \mathcal{VC}|^{\omega} \log \frac{|\mathcal{VA} + \mathcal{VC}|}{\epsilon}\Big) \quad \subset \quad \tilde{O}\Big(|\mathcal{VC} + \mathcal{VA}|^4\Big) \;\; \textit{time.}$$

Proof. Problem (15) is slightly more straightforward; having done Lemmas 4 and 5 in depth, we do this one more quickly. In the standard form, problem (15), has variables $x = (\mathbf{u}, \boldsymbol{\mu}, \mathbf{w}) \in K_{\exp}^{\mathcal{VC}}$. The constraints are:

- one linear constraint for each $(C, c) \in \mathcal{VC}$, to ensure that

$$w_{C,c} = k_{(C,c)}\mu_C(\mathit{VCP}_C(c)) \qquad \Big(\; = \sum_{\bar{c} \in \mathcal{V}(C \setminus \mathit{VCP}_C)} \mu_C(\bar{c}, \mathit{VCP}_C(c)) \Big)$$

- for every edge $(C\text{–}D) \in \mathcal{T}$, and every value $\omega \in \mathcal{V}(C \cap D)$ of the variables that clusters C and D have in common, a linear constraint

$$\sum_{\bar{c} \in \mathcal{V}(C \setminus D)} \mu_C(\bar{c}, \omega) = \sum_{\bar{d} \in \mathcal{V}(D \setminus C)} \mu_D(\bar{d}, \omega)$$

- for every $(a, s, t) \in \mathcal{VA}$, a linear constraint that ensures

$$\mu_{C_a}(S_a{=}s, T_a{=}t)\, \nu_{C_a}(S_a{=}s) = \nu_{C_a}(S_a{=}s, T_a{=}t)\, \mu_{C_a}(S_a{=}s).$$

  This is linear, because recall that ν is a constant in this optimization problem, found by having previously solved (11).

- and one constraint for each cluster $C \in \mathcal{C}$ to ensure that μ_C lies on the probability simplex.

So in total, there are $n = 3|\mathcal{VC}|$ variables, and $m = |\mathcal{VC}| + |\mathcal{VT}| + |\mathcal{VA}| + |\mathcal{C}|$ constraints. Once again the components of A and b are all at most one, and now the components of the cost function $c = \mathbf{1}$ are identically one. Furthermore, our objective is 1-strongly convex, so no additional multaplicative terms are required to convert an ϵ-close solution in the sense of suboptimality, to an ϵ-close solution in the sense of proximity to the true solution.

Therefore (15) can be solved in

$$O\Big(|\mathcal{VC}||\mathcal{VA} + \mathcal{VC}|^{\omega} \log \frac{|\mathcal{VA} + \mathcal{VC}|}{\epsilon}\Big) \quad \subset \quad \tilde{O}(|\mathcal{VC} + \mathcal{VA}|^4)$$

operations. □

Theorem 10. *Let $\mathcal{M} = (\mathcal{X}, \mathcal{A}, \mathbb{P}, \boldsymbol{\alpha}, \boldsymbol{\beta})$ be a proper discrete PDG with $N = |\mathcal{X}|$ variables each taking at most V values, and $A = |\mathcal{A}|$ arcs forming a hypergraph of treewidth T. Then for all $\gamma \in \{0^+\} \cup (0, \min_{a \in \mathcal{A}} \frac{\beta_a}{\alpha_a}]$ and $\epsilon > 0$, we can do $\hat{\gamma}$-inference to precision ϵ in*

$$O\Big((N{+}A)^4 V^{4T}\Big(T \log V + \log \frac{N{+}A}{\epsilon} + \log \frac{\beta^{\max}}{\beta^{\min}}\Big)\Big)$$

time,[†] *where* $\beta^{\max} := \max_{a \in \mathcal{A}} \beta_a$ *and*

$$\beta^{\min} := \begin{cases} \min\limits_{a \in \mathcal{A}} \{\beta_a : \beta_a > 0\} & \text{if } \gamma = 0^+ \\ \gamma & \text{if } \gamma > 0 \; . \end{cases}$$

Proof. Suppose the PDG has N variables (each of which can take at most V distinct values), and A hyperarcs, which together form a structure has tree-width T.

Then each cluster (of which there are at most N) can have at most T variables, and so can take at most V^T values. Therefore, $|\mathcal{VC}| \leq NV^T$. Since each arc must be entirely contained within some cluster, $|\mathcal{VA}| \leq AV^T$. So, $|\mathcal{VA} + \mathcal{VC}| \leq (N + A)V^T$.

[†]At the cost of substantial overhead and engineering effort, the exponent 4 can be reduced to 2.872, by appeal to Skajaa and Ye [2015] and the current best matrix multiplication algorithm [Duan et al., 2022, $O(n^{2.372})$] to invert $n \times n$ linear systems.

Applying Lemmas 4 to 6, we conclude that, for $\gamma \in (0, \min_a \frac{\beta_a}{\alpha_a}]$, $\hat{\gamma}$-inference can be done in time

$$O\Big((N+A)^4 V^{4T} \log\Big(\frac{N+A}{\epsilon} V^{4T} \frac{\beta^{\max}}{\gamma}) + \log\frac{1}{p}\Big)\Big)$$

while 0^+ inference can be done in time

$$O\Big((N+A)^4 V^{4T} \log\Big(\frac{N+A}{\epsilon} V^{4T}\Big) \log\frac{\beta^{\max}}{\beta^{\min}}\Big).$$

Finally, we suppress the factor of $\log\log\frac{1}{p}$ in the first case. We argue this is justified because it is small even for the smallest number representable with 64 bits, and because this notion of precision is dwarfed by the one captured by ϵ. There is also no problem when $p = 0$, because then it simply becomes a constraint and is handled gracefully. □

A.1 HARDNESS RESULTS AND REDUCTIONS

We now prove Theorem 11 parts (a) and (b) directly by reduction to 3-SAT. They also follow from part (c) combined with the hardness of inference in BNs, which PDGs generalize, but the direct proof is nicer.

Theorem 11.

(a) *Determining whether or not there is a distribution that shares all cpds with $\mathcal{M}$ is NP-hard.*
(b) *Computing $\langle\!\langle \mathcal{M} \rangle\!\rangle_\gamma$ is #P-hard, for all $\gamma \geq 0$.*
(c) *For $\gamma \in \{0^+\} \cup (0, \min_a \frac{\beta_a}{\alpha_a})$, there is an $O(\text{size of query result})$ reduction from $\hat{\gamma}$-inference to the problem of calculating γ-inconsistency. Under bounded tree-width, there is also an $O(|\mathcal{VC}|)$ reduction in the other direction, making the problems essentially equivalent.*

Proof. **(a).** We can directly encode SAT problems in PDGs. Specifically, let

$$\varphi := \bigwedge_{j \in \mathcal{J}} \bigvee_{i \in \mathcal{I}(j)} (X_{j,i})$$

be a CNF formula over binary variables $\mathbf{X} := \bigcup_{j,i} X_{j,i}$. Let $\mathcal{M}_\varphi$ be the PDG containing every variable $X \in \mathbf{X}$ and a binary variable C_j (taking the value 0 or 1) for each clause $j \in \mathcal{J}$, as well as the following edges, for each $j \in \mathcal{J}$:

- a hyperedge $\{X_{j,i} : i \in \mathcal{I}(j)\} \twoheadrightarrow C_j$, together with a degenerate cpd encoding the boolean OR function (i.e., the truth of C_j given $\{X_{j,i}\}$);
- an edge $\mathbb{1} \twoheadrightarrow C_j$, together with a cpd asserting C_j be equal to 1.

First, note that the number of nodes, edges, and non-zero entries in the cpds are polynomial in the $|\mathcal{J}|$, $|\mathbf{X}|$, and the total number of parameters in a simple matrix representation of the cpds is also polynomial if $\mathcal{I}$ is bounded (e.g., if φ is a 3-CNF formula). A satisfying assignment $\mathbf{x} \models \varphi$ of the variables $\mathbf{X}$ can be regarded as a degenerate joint distribution $\delta_{\mathbf{X}=\mathbf{x}}$ on $\mathbf{X}$, and extends uniquely to a full joint distribution $\mu_\mathbf{x} \in \Delta\mathcal{V}(\mathcal{M}_\varphi)$ consistent with all of the edges, by

$$\mu_\mathbf{x} = \delta_\mathbf{x} \otimes \delta_{\{C_j = \vee_i x_{j,i}\}}$$

Conversely, if μ is a joint distribution consistent with the edges above, then any point $\mathbf{x}$ in the support of $\mu(\mathbf{X})$ must be a satisfying assignment, since the two classes of edges respectively ensure that $1 = \mu(C_j = 1 \mid \mathbf{X} = \mathbf{x}) = \bigvee_{i \in \mathcal{I}(j)} \mathbf{x}_{j,i}$ for all $j \in \mathcal{J}$, and so $\mathbf{x} \models \varphi$.

Thus, $\{\!\{\mathcal{M}_\varphi\}\!\} \neq \emptyset$ if and only if φ is satisfiable, so an algorithm for determining if a PDG is consistent can also be adapted (in polynomial space and time) for use as a SAT solver, and so the problem of determining if a PDG consistent is NP-hard.

(b). We prove this by reduction to #SAT. Again, let φ be some CNF formula over $\mathbf{X}$, and construct $\mathcal{M}_\varphi$ as in the proof of Theorem 11. Furthemore, let $[\![\varphi]\!] := \{\mathbf{x} : \mathbf{x} \models \varphi\}$ be the set of assingments to $\mathbf{X}$ satisfying φ, and $\#_\varphi := |[\![\mathcal{M}]\!]|$

denote the number such assignments. We now claim that

$$\#_\varphi = \exp\left[-\frac{1}{\gamma}\langle\!\langle \mathcal{M}_\varphi \rangle\!\rangle_\gamma\right]. \tag{9}$$

If true, we would have a reduced the #P-hard problem of computing $\#_\varphi$ to the problem of computing $\langle\!\langle \mathcal{M} \rangle\!\rangle_\gamma$ for fixed γ. We now proceed with proof (9). By definition, we have

$$\langle\!\langle \mathcal{M}_\varphi \rangle\!\rangle_\gamma = \inf_\mu \Big[\mathit{OInc}_{\mathcal{M}_\varphi}(\mu) + \gamma \mathit{SInc}_{\mathcal{M}_\varphi}(\mu) \Big].$$

We start with a claim about first term.

Claim 6.1. $\mathit{OInc}_{\mathcal{M}_\varphi}(\mu) = \begin{cases} 0 & \textit{if } \operatorname{supp}\mu \subseteq [\![\varphi]\!] \times \{\mathbf{1}\} \\ \infty & \textit{otherwise} \end{cases}$.

Proof. Writing out the definition explicitly, the first can be written as

$$\mathit{OInc}_{\mathcal{M}_\varphi}(\mu) = \sum_j \left[\boldsymbol{D}\Big(\mu(C_j) \,\Big\|\, \delta_1\Big) + \mathop{\mathbb{E}}_{\mathbf{x}\sim\mu(\mathbf{X}_j)} \boldsymbol{D}\Big(\mu(C_j \mid \mathbf{X}_j = \mathbf{x}) \,\Big\|\, \delta_{\vee_i \mathbf{x}_{j,i}}\Big)\right], \tag{10}$$

where $\mathbf{X}_j = \{X_{ij} : j \in \mathcal{I}(j)\}$ is the set of variables that appear in clause j, and $\delta_{(-)}$ is the probability distribution placing all mass on the point indicated by its subscript. As a reminder, the relative entropy is given by

$$\boldsymbol{D}\Big(\mu(\Omega) \,\Big\|\, \nu(\Omega)\Big) := \mathop{\mathbb{E}}_{\omega\sim\mu} \log \frac{\mu(\omega)}{\nu(\omega)}, \qquad \begin{array}{c}\text{and in particular,}\\ \text{if } \Omega \text{ is binary,}\end{array} \qquad \boldsymbol{D}\big(\mu(\Omega) \,\big\|\, \delta_\omega\big) = \begin{cases} 0 & \text{if } \mu(\omega) = 1; \\ \infty & \text{otherwise.} \end{cases}$$

Applying this to (10), we find that either:

1. Every term of (10) is finite (and zero) so $\mathit{OInc}_{\mathcal{M}_\varphi}(\mu) = 0$, which happens when $\mu(C_j = 1) = 1$ and $\mu(C_j = \vee_i x_{j,i}) = 1$ for all j. In this case, $\mathbf{c} = \mathbf{1} = \{\vee_i x_{j,i}\}_j$ so $\mathbf{x} \models \varphi$ for every $(\mathbf{c}, \mathbf{x}) \in \operatorname{supp}\mu$;
2. Some term of (10) is infinite, so that $\mathit{OInc}_{\mathcal{M}_\varphi}(\mu) = \infty$, which happens if some j, either
   (a) $\mu(C_j \neq 1) > 0$ — in which case there is some $(\mathbf{x}, \mathbf{c}) \in \operatorname{supp}\mu$ with $\mathbf{c} \neq 1$, or
   (b) $\operatorname{supp}\mu(\mathbf{C}) = \{\mathbf{1}\}$, but $\mu(C_j \neq \vee_i x_{j,i}) > 0$ — in which case there is some $(\mathbf{x}, \mathbf{1}) \in \operatorname{supp}\mu$ for which $1 = c_j \neq \vee_i x_{j,i}$, and so $\mathbf{x} \not\models \varphi$.

Condensing and rearranging slightly, we have shown that

$$\mathit{OInc}_{\mathcal{M}_\varphi}(\mu) = \begin{cases} 0 & \text{if } \mathbf{x} \models \varphi \text{ and } \mathbf{c} = \mathbf{1} \text{ for all } (\mathbf{x}, \mathbf{c}) \in \operatorname{supp}\mu \\ \infty & \text{otherwise} \end{cases}.$$

□

Because SInc is bounded, it follows immediately that $\langle\!\langle \mathcal{M}_\varphi \rangle\!\rangle_\gamma$, is finite if and only if there is some distribution $\mu \in \Delta\mathcal{V}(\mathbf{X}, \mathbf{C})$ for which $\mathit{OInc}_{\mathcal{M}_\varphi}(\mu)$ is finite, or equivalently, by Claim 6.1, iff there exists some $\mu(\mathbf{X}) \in \Delta\mathcal{V}(\mathbf{X})$ for which $\operatorname{supp}\mu(\mathbf{X}) \subseteq [\![\varphi]\!]$, which in turn is true if and only if φ is satisfiable.

In particular, if φ is not satisfiable (i.e., $\#_\varphi = 0$), then $\langle\!\langle \mathcal{M}_\varphi \rangle\!\rangle_\gamma = +\infty$, and

$$\exp\left[-\frac{1}{\gamma}\langle\!\langle \mathcal{M}_\varphi \rangle\!\rangle_\gamma\right] = \exp[-\infty] = 0 = \#_\varphi,$$

so in this case (9) holds as promised. On the other hand, if φ *is* satisfiable, then, again by Claim 6.1, every μ minimizing $[\![\mathcal{M}_\varphi]\!]_\gamma$, (i.e., every $\mu \in [\![\mathcal{M}_\varphi]\!]^*_\gamma$) must be supported entirely on $[\![\varphi]\!]$ and have $\mathit{OInc}_{\mathcal{M}_\varphi}(\mu) = 0$. As a result, we have

$$\langle\!\langle \mathcal{M}_\varphi \rangle\!\rangle_\gamma = \inf_{\mu \in \Delta\left[[\![\varphi]\!] \times \{\mathbf{1}\}\right]} \gamma\ \mathit{SInc}_{\mathcal{M}_\varphi}(\mu).$$

A priori, by the definition of $SInc_{\mathcal{M}_\varphi}$, we have

$$SInc_{\mathcal{M}_\varphi}(\mu) = -\operatorname{H}(\mu) + \sum_j \Big[\alpha_{j,1} \operatorname{H}_\mu(C_j \mid \mathbf{X}_j) + \alpha_{j,0} \operatorname{H}_\mu(C_j)\Big],$$

where $\alpha_{j,0}$ and $\alpha_{j,1}$ are values of α for the edges of $\mathcal{M}_\varphi$, which we have not specified because they are rendered irrelevant by the fact that their corresponding cpds are deterministic. We now show how this plays out in the present case. Any $\mu \in \Delta\big[[\![\varphi]\!] \times \{\mathbf{1}\}\big]$ we consider has a degenerate marginal on $\mathbf{C}$. Specifcally, for every j, we have $\mu(C_j) = \delta_1$, and since entropy is non-negative and never increased by conditioning,

$$0 \leq \operatorname{H}_\mu(C_j \mid \mathbf{X}_j) \leq \operatorname{H}_\mu(C_j) = 0.$$

Therefore, $SInc_{\mathcal{M}_\varphi}(\mu)$ reduces to the negative entropy of μ. Finally, making use of the fact that the maximum entropy distribution μ^* supported on a finite set S is the uniform distribution on S, and has $\operatorname{H}(\mu^*) = \log|S|$, we have

$$\begin{aligned}
\langle\!\langle \mathcal{M}_\varphi \rangle\!\rangle_\gamma &= \inf_{\mu \in \Delta\left([\![\varphi]\!] \times \{\mathbf{1}\}\right)} \gamma\ SInc_{\mathcal{M}_\varphi}(\mu) \\
&= \inf_{\mu \in \Delta\left([\![\varphi]\!] \times \{\mathbf{1}\}\right)} -\gamma\ \operatorname{H}(\mu) \\
&= -\gamma\ \sup_{\mu \in \Delta\left([\![\varphi]\!] \times \{\mathbf{1}\}\right)} \operatorname{H}(\mu) \\
&= -\gamma\ \log(\#_\varphi),
\end{aligned}$$

giving us

$$\#_\varphi = \exp\left[-\frac{1}{\gamma} \langle\!\langle \mathcal{M}_\varphi \rangle\!\rangle_\gamma\right],$$

as desired. We have now reduced #SAT to computing $\langle\!\langle \mathcal{M} \rangle\!\rangle_\gamma$, for $\gamma > 0$ and an arbitrary PDG $\mathcal{M}$, which is therefore #P-hard.

To show the same for $\gamma = 0$, it suffices to add an additional hyperedge pointing to all variables, and associate it with a joint uniform distribution, and confidence 1, resulting in a new PDG $\mathcal{M}'_\varphi$. Then, because this new edge's contribution to $OInc_{\mathcal{M}}$ equals $\boldsymbol{D}(\mu \parallel \mathsf{Unif}(\mathcal{X})) = \log|\mathcal{VX}| - \operatorname{H}(\mu)$, we have

$$[\![\mathcal{M}'_\varphi]\!]_0(\mu) = OInc_{\mathcal{M}'_\varphi}(\mu) = [\![\mathcal{M}_\varphi]\!](\mu) + \log|\mathcal{VX}| - \operatorname{H}(\mu) = [\![\mathcal{M}_\varphi]\!]_1(\mu) - \log|\mathcal{VX}|.$$

Since this is true for all μ, we conclude that

$$\langle\!\langle \mathcal{M}'_\varphi \rangle\!\rangle = \langle\!\langle \mathcal{M}_\varphi \rangle\!\rangle_1 - \log|\mathcal{VX}| = -\log\left(|\mathcal{VX}| \cdot \#_\varphi\right)$$

so the two differ by a constant, and both compute the number of satisfying assignments to φ. So in general, computing $\langle\!\langle \mathcal{M} \rangle\!\rangle$ is #P-hard as well. □

We prove part (c) in the next section.

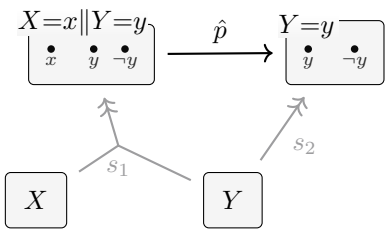
A.2 INFERENCE VIA INCONSISTENCY MINIMIZATION

We now address part (c) of Theorem 11, which is closely related to Richardson and Halpern [2021]'s original idea for an inference algorithm. While that idea does not yield an efficient inference algorithm, it does yield a very efficient reduction from inconsistency minimization to inference. In order to prove this, we first need another construction with PDGs. A cpd between discrete variables can be represented by a stochastic matrix (i.e., a matirx whose rows sum to one). It turns out that it is possible to use the machinery of PDGs to, effectively, give only one value of that matrix. That is, for any $p \in [0, 1]$, we can construct a PDG that represents the belief that $\Pr(Y{=}y|X = x) = p$, but say nothing about how the probability splits between other values of y, and also says nothing about the probability of Y if $X \neq x$. We now describe that construction.

First, we introduce two new auxiliary variables. The first variable, which we might like to call "$Y{=}y$", but mostly refer to as Y_y to prevent confusion with the synonomous event, is a binary variable, with $\mathcal{V}(Y_y) = \{y, \neg y\}$, and takes the value y if $Y = y$, and $\neg y$ if $Y \neq y$. The second variable, which we would like to call "$X{=}x \| Y{=}y$", but instead mostly refer to as

X_xY_y to prevent notational confusion, can take three values: $\mathcal{V}(X_xY_y) := \{x, y, \neg y\}$. The value x is meant to correspond exactly to the event $X{=}x$, much like before, so that $X_xY_y = x$ if and only if $X = x$. The values y and $\neg y$ also correspond to their respective events, but more loosely; the variable X_xY_y only takes one of these values when $X \neq x$. Note that both variables can be determined from X and Y (although we will need to enforce this with additional arcs), and therefore there is a unique way to extend a distribution over X and Y to also include the variables Y_y and X_xY_y.

With these definitions in place, there is now an obvious way to add an arc from the variable (X_xY_y) to the variable Y_y, together with a cpd asserting that $\Pr(Y{=}y|X{=}x) = p$. This cpd is written as a stochastic matrix $\hat{p}$ on the right of the figure below. The PDG we have just constructed is illustrated on the left of the figure below. In addition to $\hat{p}$ and the new variables, this PDG includes the structural constraints s_1 and s_2 needed to define the variables X_xY_y and Y_y in terms of X and Y; they are deterministic functions, drawn in double-headed gray arrows.



$$s_1(X_xY_y|X,Y) := \begin{cases} x & \text{if } X = x \\ y & \text{if } X \neq x \text{ and } Y = y \\ \neg y & \text{if } X \neq x \text{ and } Y \neq y \end{cases}$$

$$s_2(Y_y|Y) := \begin{cases} y & \text{if } Y = y \\ \neg y & \text{if } Y \neq y \end{cases}$$

$$\hat{p}(Y_y|X_xY_y) = \begin{array}{c} \\ x \\ y \\ \neg y \end{array} \begin{array}{c} \begin{array}{cc} y & \neg y \end{array} \\ \begin{bmatrix} p & 1-p \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \end{array}$$

So, when we add $\Pr(Y = y|X = x) = p$ to a PDG, we really first implicitly convert this information to a PDG as above. The first order of business is to prove that this works as we should expect, semantically, in the case we're interested in.

Lemma 7. *Suppose $\mathcal{M}$ is a PDG with variables $\mathcal{X}$ and $\boldsymbol{\beta} \geq \mathbf{0}$. Then, for all $X, Y \subseteq \mathcal{X}$, $x \in \mathcal{V}X$, $y \in \mathcal{V}Y$, $p \in [0,1]$ and $\gamma \geq 0$, we have that:*

$$\Big\langle\!\Big\langle \mathcal{M} + \Pr(Y{=}y|X{=}x) = p \Big\rangle\!\Big\rangle_\gamma \geq \langle\!\langle \mathcal{M} \rangle\!\rangle_\gamma,$$

with equality if and only if there exists $\mu \in [\![\mathcal{M}]\!]^*_\gamma$ such that $\mu(Y{=}y|X{=}x) = p$.

Proof. The inequality is immediate; it is an instance of monotonicity of inconsistency [Richardson, 2022, Lemma 1]. Intuitively: believing more cannot make you any less inconsistent. We now prove that equality holds iff there is a minimizer with the appropriate conditional probability.

($\Longleftarrow$). Suppose there is some $\mu \in [\![\mathcal{M}]\!]^*_\gamma$ with $\mu(Y{=}y|X{=}x) = p$. Because $\mu \in [\![\mathcal{M}]\!]^*_\gamma$, we know that $[\![\mathcal{M}]\!]_\gamma(\mu) = \langle\!\langle \mathcal{M} \rangle\!\rangle$. Let $\hat{\mu}$ be the extension of μ to the new variables "$X{=}x\|Y{=}y$" and "$Y{=}y$", whose values are functions of X and Y according to s_1 and s_2. Then,

$$\begin{aligned}
\Big\langle\!\Big\langle \mathcal{M} + \Pr(Y{=}y|X{=}x) = p \Big\rangle\!\Big\rangle_\gamma &\leq \Big[\!\Big[\mathcal{M} + \Pr(Y{=}y|X{=}x) = p \Big]\!\Big]_\gamma(\hat{\mu}) \\
&= [\![\mathcal{M}]\!]_\gamma(\mu) + \mathop{\mathbb{E}}_{\mu}\left[\log \frac{\hat{\mu}(Y_y|X_xY_y)}{\hat{p}(Y_y|X_xY_y)}\right] \\
&= [\![\mathcal{M}]\!]_\gamma(\mu) + \mu(X{=}x, Y{=}y)\log\frac{\mu(Y{=}y|X{=}x)}{p} + \mu(X{=}x, Y{\neq}y)\log\frac{\mu(Y{\neq}y|X{=}x)}{1-p} \\
&= [\![\mathcal{M}]\!]_\gamma(\mu) + \mu(X{=}x, Y{=}y)\log(1) + \mu(X{=}x, Y{\neq}y)\log(1) \\
&= [\![\mathcal{M}]\!]_\gamma(\mu) = \langle\!\langle \mathcal{M} \rangle\!\rangle_\gamma.
\end{aligned}$$

The equality between the second and third lines is perhaps the trickiest to see, but follows because for joint settings in which $X{\neq}x$, one can easily see that $\hat{\mu}(Y_y|X_xY_y)$ equals 1 with probability 1, as does $\hat{p}(Y_y|X_xY_y)$. So, after dividing one by the other and taking a logarithm, these cases contribute nothing to the expectation. What remains are the two

possibilities where $X{=}x$, which are shown in the second line.

To complete this direction of the proof, it suffices to observe that we already knew the inequality held in the opposite direction (by monotonicity), so the two terms are equal.

($\implies$). Suppose the two inconsistencies are equal, i.e., $\Big\langle\!\Big\langle \mathcal{M} + \Pr(Y{=}y|X{=}x) = p \Big\rangle\!\Big\rangle_\gamma = \langle\!\langle \mathcal{M} \rangle\!\rangle_\gamma$.

This time, choose $\hat{\mu} \in [\![\mathcal{M} + \Pr(Y{=}y|X{=}x) = p]\!]^*_\gamma$, and define μ to be its marginal on the variables of $\mathcal{M}$ (which contains the same information as $\hat{\mu}$ itself). Let $q := \mu(Y{=}y|X{=}x)$. Then,

$$\begin{aligned}
\langle\!\langle \mathcal{M} \rangle\!\rangle_\gamma &= \Big\langle\!\Big\langle \mathcal{M} + \Pr(Y{=}y|X{=}x) = p \Big\rangle\!\Big\rangle_\gamma \\
&= \Big[\!\Big[\mathcal{M} + \Pr(Y{=}y|X{=}x) = p \Big]\!\Big]_\gamma (\hat{\mu}) \\
&= [\![\mathcal{M}]\!]_\gamma(\mu) + \mu(X{=}x, Y{=}y) \log \frac{\mu(Y{=}y|X{=}x)}{p} + \mu(X{=}x, Y{\neq}y) \log \frac{\mu(Y{\neq}y|X{=}x)}{1-p} \\
&= [\![\mathcal{M}]\!]_\gamma(\mu) + \mu(X{=}x) \left[q \log \frac{q}{p} + (1-q) \log \frac{1-q}{1-p} \right] \\
&= [\![\mathcal{M}]\!]_\gamma(\mu) + \mu(X{=}x) \boldsymbol{D}(q \parallel p) \\
&\geq \langle\!\langle \mathcal{M} \rangle\!\rangle_\gamma + \mu(X{=}x) \boldsymbol{D}(q \parallel p)
\end{aligned}$$

Therefore $0 \geq \mu(X{=}x)\boldsymbol{D}(q \parallel p)$. But relative entropy is non-negative, by Gibbs inequality. This shows $\mu(X{=}x)\boldsymbol{D}(q \parallel p) = 0$. So either $\mu(X{=}x)$, or $p = \mu(Y{=}y|X{=}x)$, and the first case is just a special case of the second one. In addition, the algebra above shows that $\mu \in [\![\mathcal{M}]\!]^*_\gamma$, as its score is $\langle\!\langle \mathcal{M} \rangle\!\rangle_\gamma$. Thus, we have found $\mu \in [\![\mathcal{M}]\!]^*_\gamma$ such that $\mu(Y{=}y|X{=}x) = p$, completing the proof. □

Next, we show that the overall inconsistency is convex in the parameter $p \in [0, 1]$. It's perhaps easier to give the more general result:

Lemma 8. *If h is a cpd, then the function $h \mapsto \langle\!\langle \mathcal{M} + h \rangle\!\rangle_\gamma$ and strictly convex for $\gamma \in (0, \min_a \frac{\beta_a}{\alpha_a})$.*

Proof. We start by expanding the definitions, obtaining

$$\begin{aligned}
\langle\!\langle \mathcal{M} + h \rangle\!\rangle_\gamma &= \inf_\mu [\![\mathcal{M} + h]\!]_\gamma(\mu) \\
&= \inf_\mu \left[[\![\mathcal{M}]\!]_\gamma(\mu) + \mathop{\mathbb{E}}_{x \sim \mu_X} \boldsymbol{D}\Big(\mu(Y \mid x) \,\Big\|\, h(Y \mid x)\Big) \right] \\
&= \inf_\mu \left[[\![\mathcal{M}]\!]_\gamma(\mu) + \boldsymbol{D}\Big(\mu(X,Y) \,\Big\|\, h(Y \mid X)\,\mu(X)\Big) \right].
\end{aligned}$$

Fix $\gamma \leq \min_a \frac{\beta_a}{\alpha_a}$. Then we know that $[\![\mathcal{M}]\!]_\gamma(\mu)$ is a γ-strongly convex function for every PDG $\mathcal{M}$, and hence there is a unique joint distribution which minimizes it. We now show that the inconsistency is strictly convex.

Suppose $h_1(Y \mid X)$ and $h_2(Y \mid X)$ are two cpds on Y given X. Fix $\lambda \in [0, 1]$, and define $h_\lambda := (1-\lambda)h_1 + \lambda h_2$. Let μ_1, μ_2 and μ_λ be the joint distributions that minimze $[\![\mathcal{M} + h_1]\!]_\gamma$, $[\![\mathcal{M} + h_2]\!]_\gamma$ and $[\![\mathcal{M} + h_\lambda]\!]_\gamma$, respectively. Then we have

$$\langle\!\langle \mathcal{M} + h_\lambda \rangle\!\rangle_\gamma = [\![\mathcal{M}]\!]_\gamma(\mu_\lambda) + \boldsymbol{D}\Big(\mu_\lambda(X,Y) \,\Big\|\, h_\lambda(Y \mid X)\mu_\lambda(X)\Big).$$

By convexity of $[\![\mathcal{M}]\!]$ and $\boldsymbol{D}$, we have

$$[\![\mathcal{M}]\!]_\gamma(\mu_\lambda) \leq (\lambda - 1)[\![\mathcal{M}]\!]_\gamma(\mu_1) + \lambda [\![\mathcal{M}]\!]_\gamma(\mu_2) \tag{11}$$

and

$$\begin{aligned}
\boldsymbol{D}\Big(\mu_\lambda(XY) \,\Big\|\, h_\lambda(Y|X)\mu_\lambda(X)\Big) \leq (1-\lambda)\boldsymbol{D}\Big(\mu_1(XY) \,\Big\|\, h_1(Y|X)\mu_1(X)\Big) \\
+ \lambda\ \boldsymbol{D}\Big(\mu_2(XY) \,\Big\|\, h_2(Y|X)\mu_2(X)\Big).
\end{aligned} \tag{12}$$

If $\mu_1 \neq \mu_2$ then since $[\![\mathcal{M}]\!]$ is strictly convex, (11) must be a strict inequality. On the other hand, if $\mu_1 = \mu_2$, then since $\mu_\lambda = \mu_1 = \mu_2$ and $\boldsymbol{D}$ is stricly convex in its second argument when its first argument is fixed, (12) must be a strict inequality. In either case, the sum of the two inequalities must be strict, giving us

$$\begin{aligned}
\langle\!\langle \mathcal{M} + h_\lambda \rangle\!\rangle_\gamma &= [\![\mathcal{M}]\!]_\gamma(\mu_\lambda) + \boldsymbol{D}\Big(\mu_\lambda(XY) \,\Big\|\, h_\lambda(Y|X)\mu_\lambda(X)\Big) \\
&< (\lambda - 1)\left[[\![\mathcal{M}]\!]_\gamma(\mu_1) + \boldsymbol{D}\Big(\mu_1(XY) \,\Big\|\, h_1(Y|X)\mu_1(X)\Big)\right] \\
&\qquad + \lambda\left[[\![\mathcal{M}]\!]_\gamma(\mu_2) + \boldsymbol{D}\Big(\mu_2(XY) \,\Big\|\, h_2(Y|X)\mu_2(X)\Big)\right] \\
&= (\lambda - 1)\langle\!\langle \mathcal{M} + h_1 \rangle\!\rangle + \lambda\,\langle\!\langle \mathcal{M} + h_2 \rangle\!\rangle,
\end{aligned}$$

which shows that $\langle\!\langle \mathcal{M} + h \rangle\!\rangle$ is *strictly* convex in h, as desired. □

Corollary 8.1. *As before, let $\mathcal{M}$ is a PDG with $\boldsymbol{\beta} \geq \mathbf{0}$ and variables $\mathcal{X}$, and fix $X, Y \subseteq \mathcal{X}$, $x \in \mathcal{V}X$, $y \in \mathcal{V}Y$, and $\gamma > 0$. Then, for $p \in [0, 1]$ the map*

$$p \mapsto \Big\langle\!\!\Big\langle \mathcal{M} + \Pr(Y{=}y|X{=}x) = p \Big\rangle\!\!\Big\rangle_\gamma$$

is strictly convex.

Proof. Simply take h to be the cpd $\hat{p}$, and absorb the the other components of (the PDG representation of) $\Pr(Y{=}y|X{=}x) = p$ into $\mathcal{M}$, and then apply Lemma 8. □

We are now ready to tackle the theorem itself.

Theorem 11 (c). *For $\gamma \in \{0^+\} \cup (0, \min_a \frac{\beta_a}{\alpha_a})$, there is an $O(\text{size of query result})$ reduction from $\hat{\gamma}$-inference to the problem of calculating γ-inconsistency. Under bounded tree-width, there is also an $O(|\mathcal{V}\mathcal{C}|)$ reduction in the other direction, making the problems essentially equivalent.*

Proof. The claim is that, with an inconsistency oracle (in particular, with the ability to query the inconsistency of $\mathcal{M} + \Pr(Y{=}y|X{=}x) = p$ for free), we can find $[\![\mathcal{M}]\!]^*_\gamma(Y{=}y|X{=}x)$ in constant time. Thus, if we are looking to find the entire matrix of conditional probabilities $[\![\mathcal{M}]\!]^*_\gamma(Y|X)$, it will take time linear in the number of entries in that matrix (i.e., $|\mathcal{V}(X, Y)|$). The reduction is also linear in another sense. We will prove that we can approximate $[\![\mathcal{M}]\!]^*_\gamma(Y{=}y|X{=}x)$ to within precision ϵ in time $O(\log 1/\epsilon)$. Thus the result is the strongest reduction one can hope for: it is linear in the number of bits we output.

We proceed by describing an algorithm that uses an our inconsistency oracle to answer porbabilistic queries with a variant of binary search. The state of the algorith consists of three points in an interval $[a, b, c] \in [0, 1]$, labeled with the values $f(a)$, $f(b)$, and $f(c)$, where f is the function

$$\begin{aligned}
f &: [0, 1] \to \overline{\mathbb{R}} \\
p &\mapsto \langle\!\langle \mathcal{M} + (\Pr(Y{=}y|X{=}x) = p) \rangle\!\rangle_\gamma
\end{aligned}$$

that we assumed we have oracle access to. We can then find the minimizer x^* of f with the following algorithm.

Initialize $(a, b, c) \leftarrow (0, \frac{1}{2}, 1)$;
for $i = 1, 2, \ldots, \log(1/\epsilon)/(\log 4/3)$ **do**
  if $b - a \geq c - b$ **then**
    Let $x := \frac{b+a}{2}$, and evaluate $f(x)$;
    if $f(x) < f(b)$ **then**
      $(a, b, c) \leftarrow (a, x, b)$;
    else
      $(a, b, c) \leftarrow (x, b, c)$;
    end if

else if $c - b > b - a$ **then**
Let $x := \frac{b+c}{2}$, and evaluate $f(x)$;
if $f(x) < f(b)$ **then**
$(a, b, c) \leftarrow (b, x, c)$;
else
$(a, b, c) \leftarrow (a, b, x)$;
end if
end if
end for
return b;

We begin by proving that algorithm does, in fact, find x^*. Because f is convex, this procedure satisfies an important invariant: both b and the minimizer of f always lie in the interval $[a, c]$.

Proof. We proceed by induction. This is obviously true at first, because $[a, b] = [0, 1]$ contains all points in the domain of f. Let x^* be the minimizer of f, and suppose inductively that $x^* \in [a, b]$.

(case 1) If $b - a \geq c - b$, then $x \in [a, b]$.

- Suppose $f(x) < f(b)$. Then for all $y > b$, it must be the case that $f(y) > f(b)$ by convexity of f. (For if $f(y) < f(b)$, then segment between $(x, f(x))$ and $(y, f(y))$ would lie entirely below $(b, f(b))$, which contradicts convexity). Thus, we can rule out all such y as possible minimizers of f, so we can restrict our attention to $[a, b]$, which contains x.
- On the other hand, if $f(x) > f(b)$, then it must be the case that no $y < x$ can be a minimizer of f by convexity, with the same reasing as above. (Namely, if $f(y) < f(x)$ then the segment between $(y, f(y))$ and $(b, f(b))$ lies below $(x, f(x))$, contradicting convexity). Thus the true minimizer lies in $[x, c]$, which contains the point b.

(case 2) The other case is symmetric; we include it for completeness. Suppose $c - b > b - a$, and so $x = \frac{b+c}{2}$.

- Suppose $f(x) < f(b)$. Then $f(y) > f(b)$ for all $y < b$ (because if $f(y) < f(b)$, then segment between $(y, f(y))$ and $(x, f(x))$ would lie below $(b, f(b))$). So $x^*, x \in [b, c]$.
- On the other hand, if $f(x) > f(b)$, then $f(y) > f(x)$ for all $y > x$ (because, if $f(y) < f(x)$ then the segment between $(y, f(y))$ and $(b, f(b))$ lies below $(x, f(x))$, contradicting convexity). So $x^*, b \in [a, x]$. □

The other important aspect of the algorithm, is that at each each iteration reduces the size of the interval by a factor of at least 3/4 (and possibly much more). This is because in each case we focus on the larger half of the interval, and ultimately discard either half or all of it—so we reduce the size of the interval by at least one quarter. Thus, after n iterations, the size of the interval is at most $(3/4)^n$. At this point it should be clear that the number of iterations was selected to ensure that $|c - a| \leq \epsilon$. Of course, $[a, c]$ contains both b and x^*. Therefore the output of the algorithm (b) must be within ϵ of the true minimizer (x^*).

It is easy to see that this algorithm takes constant space, and $O(\log 1/\epsilon)$ time. And, fixing $\epsilon = 10^{-78}$ we get a constant-time algorithm that outputs the closest 64-bit number to x^*.

Reducing inconsistency calculation to inference. This reduction is much simpler, shares more techniques with the primary thrust of the paper. First find a tree decomposition $(\mathcal{C}, \mathcal{T})$ of the PDG's structure, and then query the marginals of each clique. Because of the work we've already done, we know this information is enough information to simply evaluate the scoring function, including the joint entropy, by (13). □

B THE CONVEX-CONCAVE PROCEDURE, AND IMPLEMENTATION DETAILS

Optimization problems (7) and (14) can be extended to apply slightly more broadly. There are some cases where there is a unique optimal distribution but γ is large enough that $\boldsymbol{\beta} \not\geq \gamma\boldsymbol{\alpha}$. In these cases, our convex program will fail to satisfy the

dcp requirements, and so we cannot compile it to an exponential conic program—but it turns out to still be a useful building block. We now describe how we can still do inference in some of these cases with the convex-concave procedure, or CCCP [Yuille and Rangarajan, 2003]. This will give us a local minimum of the PDG scoring function $[\![\mathcal{M}]\!]_\gamma$, without requiring us to write this scoring function in a way that proves its convexity, (as is necessary in order to specify a disciplined convex program). At this point, if we happen to know that the problem is convex (or even just pseudo-convex) for other reasons, then finding this distribution suffices for inference. We now describe how this can be done in more detail.

Suppose $\beta_a < \gamma\alpha_a$ some $a \in \mathcal{A}$. In this case $[\![\mathcal{M}]\!]_\gamma$ may not be convex, in general.[1] However, we do know how to decompose $[\![\mathcal{M}]\!]_\gamma$ into a sum of a convex function $f(\mu)$ and a concave one $g(\mu)$. Concretely: each term on the second line of (6) is either convex or concave, depending on the sign of the quantity $\gamma\alpha_a - \beta_a$. Once we sort the terms into convex terms $f(\mu)$ and strictly concave terms $g(\mu)$, the CCCP tells us to repeatedly solve f plus a linear approximation to g. In more detail, the algorithm proceeds as follows. First, choose an initial guess μ_0, and iteratively use the convex solver as in the main paper to compute

$$\mu_{t+1} := \arg\min_{\mu} f(\mu) + (\mu - \mu_t)^\mathsf{T} \nabla g(\mu_t).$$

This can be slow because each iteration of the solver is expensive. Still, it is guaranteed to make progress, since

$$\begin{aligned} f(\mu_{t+1}) + g(\mu_{t+1}) &< f(\mu_{t+1}) + (\mu_{t+1} - \mu_t)^\mathsf{T} \nabla g(\mu_t) + g(\mu_t) \\ &\leq f(\mu_t) + (\mu_t - \mu_t)^\mathsf{T} \nabla g(\mu_t) + g(\mu_t) \\ &= f(\mu_t) + g(\mu_t). \end{aligned}$$

Furthermore, because in our case g is bounded, the process eventually converges a local minimum of $[\![\mathcal{M}]\!]_\gamma$. This alone, however, is not sufficient for inference, because we may not be able to use this local minimum to answer queries in a way that is true of *all* minimizing distributions. But, if it happens there is a unique local minimum, then the CCCP will find it, leading to an inference procedure.

Notice that if $\boldsymbol{\beta} \geq \gamma\boldsymbol{\alpha}$, then the concave part g is identically zero, and CCCP converges after making just one call to the convex solver. Therefore, in the cases we could already handle, this extension reduces to the algorithm we described before. For this reason, all of our code that handles problems (7) and (14) is augmented with the CCCP.

Compared to the black-box optimization baselines (Adam and LBFGS), which also only find one minimum, the CCCP still has some advantages. One can see in Figure 4, for example, that when $\gamma = 2 > 1 = \max_a (\beta_a/\alpha_a)$, CCCP performs better than the baselines. In fact, the CCCP-augmented could probably even higher accuracy, if were we not limiting it to a maximum of only five iterations.

C DETAILS ON THE EMPIRICAL EVALUATION

Imagine a very steep V-shaped canyon, and inside a small slow-moving stream at a gentle incline. The end of the river may be very far away, and the whole landscape may be smooth and strongly convex, but the gradient will still almost always point perpendicular to it, and rather towards the center of the river. This intuition may help explain why, even though $[\![\mathcal{M}]\!]_\gamma$ is infinitely differentiable in μ and γ-strongly convex, it can still be challenging to optimize, especially when the β's are very different, or when γ is small. For example, a solution to (11) finds a minimizer of $OInc$, but such minimizers may be very far away from $[\![\mathcal{M}]\!]^*_{0+}$, despite sharing an objective value.

We now see how this is true even when working with very small PDGs and joint distributions.

C.1 SYNTHETIC EXPERIMENT: COMPARISON WITH BLACK-BOX OPTIMIZERS, ON JOINT DISTRIBUTIONS.

Here is a more precise description of our first synthetic experiment, on joint distributions, which contrasts the convex optimization approaches of Section 3 with black-box optimizers.

[1] Consider the PDG $(\rightarrow X,\, Y\leftarrow)$ for instance, which has arcs to X and Y, both with $\alpha = 1$ and $\beta = 0$. The minimizers of $[\![\rightarrow X,\, Y\leftarrow]\!]_\gamma$ are the distributions that make X and Y is independent. It is easily seen that this set is not convex: X and Y are independent if either variable is deterministic, and every distribution is a convex combination of deterministic distributions.

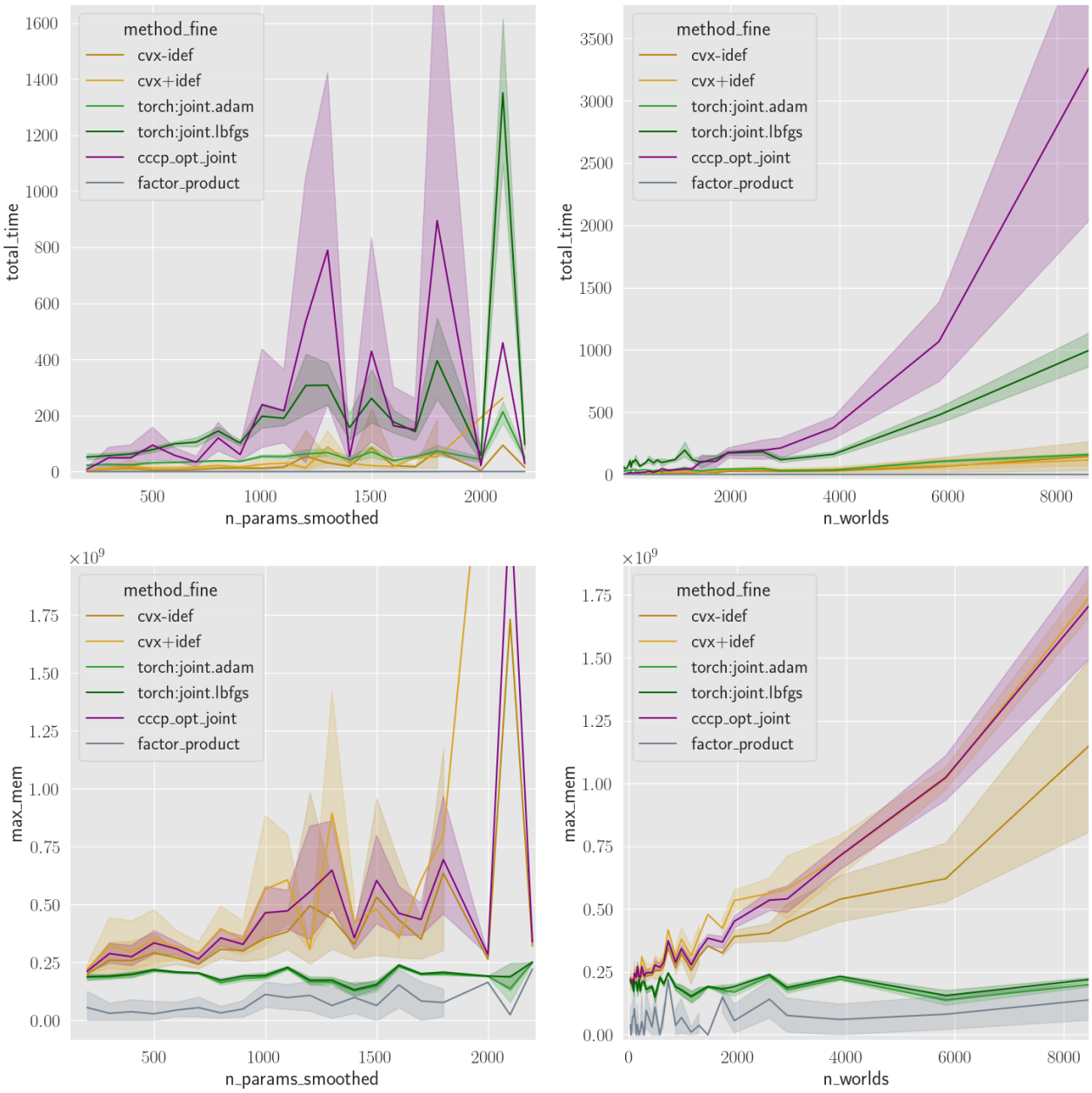


Figure 1: Resource costs for the joint-distribution optimization setting of Section 3. We measure computation time (`total_time`, top) and maximum memory usage (`max_mem`, bottom) for the various optimization methods (by color), as the size of the PDG increases, as measured by the number of parameters in the PDG (`n_params` $= \mathcal{VA}$, left), and the size of a joint distribution over its variables (`n_worlds` $= \mathcal{VX}$, right). Note that the convex solvers for the 0 and 0^+ semantics are significantly faster than LBFGS, and on par with Adam. However, all three convex-solver based approaches require significantly more memory than the black-box optimizers.

- generate 300 PDGs, each of which has the following quantities, to each of which we choose the following natural numbers uniformly at random:
  - $N \in \{5, \dots, 9\}$ of variables (so that $\mathcal{X} := \{1, \dots, N\}$),
  - $V_X \in \{2, 3\}$ values per variable (so that $|\mathcal{V}X| = V_X$ for each $X \in \mathcal{X}$)
  - $A \in \{7, \dots, 14\}$ hyperarcs, each $a \in \{1, \dots, A\} =: \mathcal{A}$ of which has
  - $N_a^S \in \{0, 1, 2, 3\}$ sources, and
  - $N_a^T \in \{1, 2\}$ targets.
- For each arc $a \in \mathcal{A}$, N_a^S of the N variables are choosen without replacement to be sources $S_a \subseteq N$, and N_a^T of remaining variables are chosen to be targets. Finally, to each value of S_a and T_a, a number $p_{a,s,t} \in [0, 1]$ is chosen uniformly at random, and the cpd

  $$\mathbb{P}_a(T_a{=}t \mid S_a{=}s) = \frac{p_{a,s,t}}{\sum_{t' \in \mathcal{V}(T)} p_{a,s,t'}} \qquad \text{is given by normalizing appropriately.}$$

  This defines a PDG $\mathcal{M} = (\mathcal{X}, \mathcal{A}, \mathbb{P}, \mathbf{1}, \mathbf{1})$, that has $\boldsymbol{\alpha} = \boldsymbol{\beta} = \mathbf{1}$, which will allow us to comapre against belief propogation and other graphical models at $\gamma = 1$. The complexity of this PDG is summarized by two numbers:
  - `n_params` $:= \mathcal{V}\mathcal{A}$, the total number of parameters in all cpds of $\mathcal{M}$, and
  - `n_worlds` $:= \mathcal{V}\mathcal{X}$, the dimension of joint distributions over $\mathcal{M}$'s variables.
- Run MOSEK on (5) to find a distribution that minimizes *OInc*; we refer to this method as `cvx-idef`
- Use the result to run MOSEK on (3) to find the special distribution $[\![\mathcal{M}]\!]^*_{0^+}$; we refer to this method as `cvx+idef`. These names are due to the fact that *SInc* is called *IDef* in previous work [Richardson and Halpern, 2021, Richardson, 2022]; thus, this refers to using the convex solver to compute minimizers of *OInc* with and without considering *IDef*.
- Now for the torch baselines. Let $\theta = [\theta_{\mathbf{x}}]_{\mathbf{x} \in \mathcal{V}\mathcal{X}} \in \mathbb{R}^{\mathcal{V}\mathcal{X}}$, be a vector of optimization variables, and choose a representation of the joint distribution, either by

  $$\begin{pmatrix} \text{renormalized} \\ \text{simplex} \end{pmatrix} \quad \mu_\theta(\mathbf{x}) = \frac{\max\{\theta_{\mathbf{x}}, 0\}}{\sum_{\mathbf{y} \in \mathcal{V}\mathcal{X}} \max\{\theta_{\mathbf{y}}, 0\}} \qquad \text{or} \qquad \mu_\theta(\mathbf{x}) = \frac{\exp(\theta_{\mathbf{x}})}{\sum_{\mathbf{y} \in \mathcal{V}\mathcal{X}} \exp(\theta_{\mathbf{y}})} \quad \text{(Gibbs)} \qquad \text{(see Figure 2)}$$

- Then, for each value of gamma: $\gamma \in \{0, 10^{-8}, 10^{-4}, 10^{-2}, 1\}$, and each learning rate `lr` $\in 1E{-}3, 1E{-}2, 1E{-}1, 1E0$, and each optimizer $opt \in \{\texttt{adam}, \texttt{L-BFGS}\}$, run opt over the parameters θ to minimize $[\![\mathcal{M}]\!]_\gamma(\mu_\theta)$ until convergence (or a maximum of 1500 iterations)
- We collect the following data about the resulting distribution and the process of computing it:
  - the total time taken to arrive at μ;
  - the maximmum memory taken by the process computing μ;
  - the objective and its component values:

    $$\texttt{inc} := \mathit{SInc}_{\mathcal{M}}(\mu), \qquad \texttt{idef} := \mathit{SInc}_{\mathcal{M}}(\mu), \qquad \texttt{obj} := \mathit{OInc}_{\mathcal{M}}(\mu) + \gamma \mathit{SInc}_{\mathcal{M}}(\mu) = [\![\mathcal{M}]\!]_\gamma(\mu)$$

The numbers can then be recreated by running our experimental script as follows:

```
python random_expts.py -N 300 -n 5 9 -e 7 14 -v 2 3
    --ozrs lbfgs adam
    --learning-rates 1E0 1E-1 1E-2 1E-3
    --gammas 0 1E-8 1E-4 1E-2 1E0
    --num-cores 20
    --data-dir random-joint-data
```

which creates a folder called `random-joint-data`, and fills it with `.mpt` files corresponding to each distribution and the method / parameters that gave rise to it.

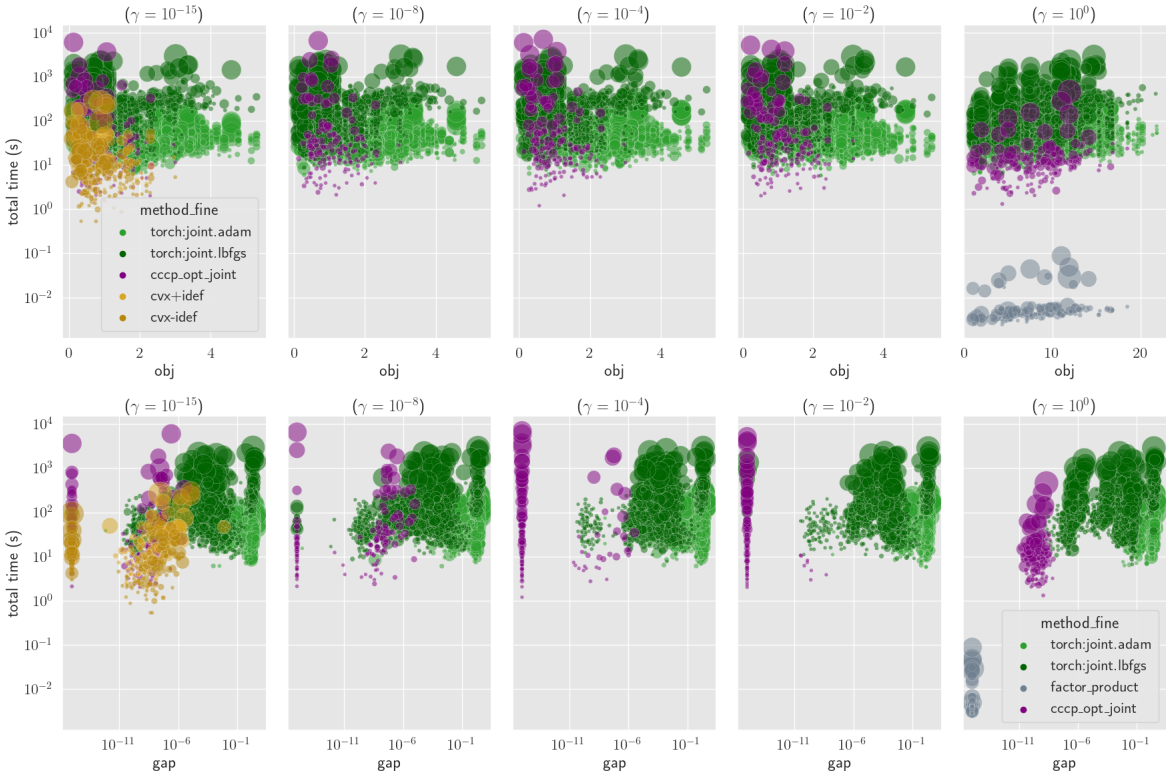


Figure 3: An un-compressed version of the information in Figure 1, that groups by the value of γ, and also gives the absolute values of the objectives (top row) in addition to the relative gaps (bottom row).

Analyzing the Results. Look at Figure 1. Our theoretical analysis, and in particular the proof of Lemma 4, suggest that the magnitudes of $\mathcal{VX}$ and $\mathcal{VA}$ play similar roles in the asymptotic complexity of PDG inference. Our experiments reveal that, at least for random PDGs, the number of worlds is the far more important of the two; observe how much more variation there is on the left side of the figure than the right—and now note that the left side has been smoothed, while the right side has not. The black-box py-torch based approaches clearly have an edge in that they can handle larger models, as evidenced by the cut-offs on the right-hand side of Figure 8, when with 5GB memory.

Note that the exponential-cone-based methods for the observational limit (gold) are actually faster than L-BFGS (the black-box optimizer with the lowest gap), and also seem to be growing at a slower rate. However, they use significantly more memory, and cannot handle large models. In addition to being faster, our techniques also seem to be more precise; they achieve objective values that are consistently much better than the black-box methods.

Now look at Figure 3, which contains a break-down of the information in Figure 1. The bottom half of the figure is just the same information, but with each value of γ separated out, so that the special cases of the factor product and 0^+ inference become clear, while the top half shows why it's more important to look at the gap than the actual objective value for these random PDGs. Figure 3 also makes it clearer how larger problems take longer, and especially so for `cccp` (violet), which solves the most complex version of the problem (7).

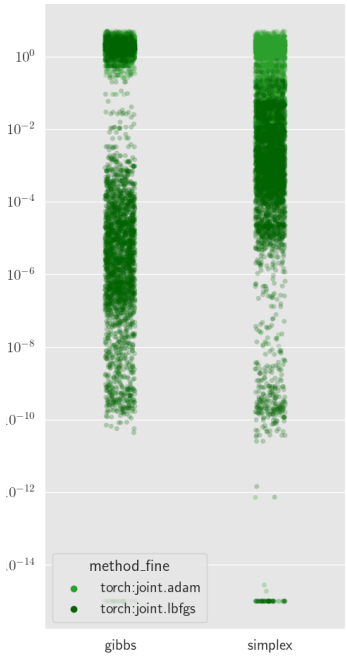


Figure 2: differences in performance between the Gibbs and simplex parameterizations of probabilities.

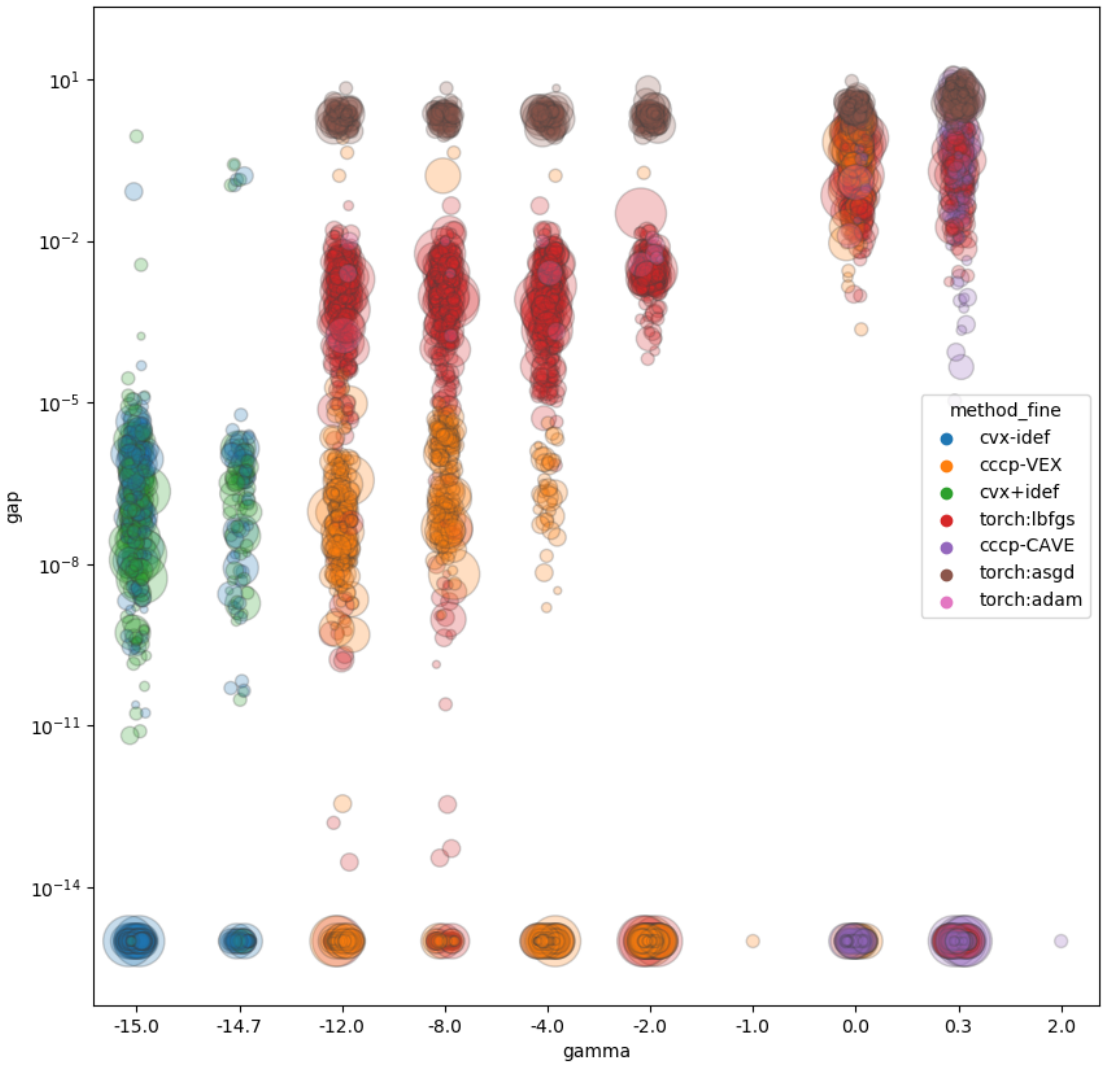


Figure 4: A graph of the gap (the difference between the attained objective value, and the best objective value obtained across all methods for that value of γ), as γ varies. The x-axis is $\log_{10}(\gamma + 10^{-15})$. As before, colors indicate the optimization method, and the size of the circle illustrates the number of optimization variables (i.e., the number of possible worlds). `cvx-idef` corresponds to just solving (5), and `cvx+idef` corresponds to then solving problem (3) afterwards. The CCCP runs are split into regimes where the entire problem is convex ($\gamma \leq 1$, labeled `cccp-VEX`), and the entire problem is concave ($\gamma > 1$, labeled `cccp-CAVE`). The optimization approaches `opt_dist` are split into three different optimizers: LBFGS, Adam, and also a third one that performs relatively poorly: accelerated gradient descent. Note that for small γ, the exponential-cone based methods significantly outperform the gradient-based ones.

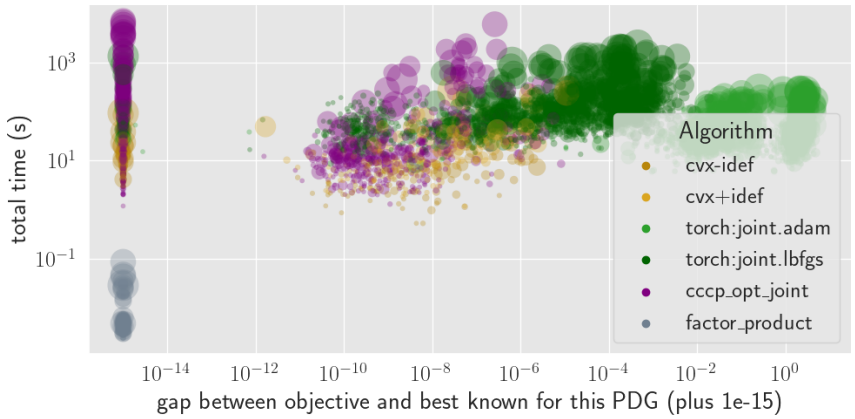


Figure 5: An analogue of Figure 1, for the cluster setting. Note that there is even more separation between the exponential-cone based approaches, and the black-box optimization based ones. The new grey points on the bottom correspond to belief propogation, which is both faster and typically the most accurate.

C.2 SYNTHETIC EXPERIMENT: COMPARING WITH BLACK-BOX OPTIMIZERS, ON CLIQUE TREES

1. Choose a number of variables $N \in \{8, \dots, 32\}$, and a treewidth $k \in \{1, \dots, 4\}$ uniformly at random. Then, draw a random k-tree and corresponding tree of clusters $(\mathcal{C}, \mathcal{T})$, as follows:
   (a) initialize $G \leftarrow K_{k+1}$ to a complete graph on $k+1$ vertices, and $\mathcal{C} \leftarrow \{K_{k+1}\}$ to be set containing a single cluster, and $\mathcal{T} \leftarrow \emptyset$.
   (b) until there are N vertices: add a new vertex v to G, then randomly select a size k-clique (fully-connected subgraph) $U \subset G$, and add edges between v and every vertex $u \in U$. Then, add $U \cup \{v\}$ to $\mathcal{C}$, and add edges to every other cluster $C \in \mathcal{C}$ such that $U \subset C$.
2. Then, draw the same parameters $V_X \in \{2, 3\}$, $A \in \{8, \cdots, 120\}$, $N_a^S \in \{0, 1, 2, 3\}$, and $N^T \in \{1, 2\}$ as in Appendix C.1 uniformly at random. While $N_a^S + N_a^T > k + 1$, for any a, resample N_a^S and N_a^T.
3. Form a PDG whose structure $\mathcal{A}$ can be decomposed by $(\mathcal{C}, \mathcal{T})$, as follows: for each edge $a \in \mathcal{A}$, sample a cluster $C \in \mathcal{C}$ uniformly at random; then select N_a^S nodes from that cluster without replacement as sources, and N_a^T nodes as targets; this is possible because each cluster has $k+1$ nodes, and $N_a^S + N_a^T \leq k + 1$ by construction.
4. Fill in the probabilities by drawing uniform random numbers and re-normalizing, just as before, to form a PDG $\mathcal{M}$
5. The black-box optimization baselines work in much the same way also, although now the optimization variables include not one distribution μ but a collection $\boldsymbol{\mu}$ of them; this time, we use only the simplex representation of $\boldsymbol{\mu}_\theta$. More importantly, we want these clusters to share appropriate marginals; to encourage this, we add a terms to the loss function, so overall, it is

$$\ell(\theta) := [\![\mathcal{M}]\!]_\gamma(\boldsymbol{\mu}_\theta) + \sum_{C-D \in \mathcal{T}} \exp\left(\sum_{w \in \mathcal{V}(C \cap D)} \Big(\mu_C(C \cap D{=}w) - \mu_D(C \cap D{=}w) \Big)^2 \right) - 1.$$

   This is admittedly pretty ad-hoc; the point is just that it is zero and does not contribute to the gradient if $\boldsymbol{\mu}_\theta$ is calibrated, and otherwise quickly becomes overwhelmingly important.

Analyzing the Results. Observe in Figure 2 that the separation between the clique tree convex solver and the black-box algorithms is even more distinct. This is because, in this case, the penalty for violating constraints was too small, and the optimization effort was largely wiped out by the calibration before evalution.

This illustrates another general advantage that the convex solver has over black-box optimizers: it is much less brittle and reliant and exactly tuning parameters correctly. Note that even in this minimal example, there were many hyper-parameters that require tuning: the regularization strengths that enforce soft constraints (clique tree calibration, normalization), as well as learning rate, not to mention various other structural choices: the optimizer, the representation of the distribution, and the

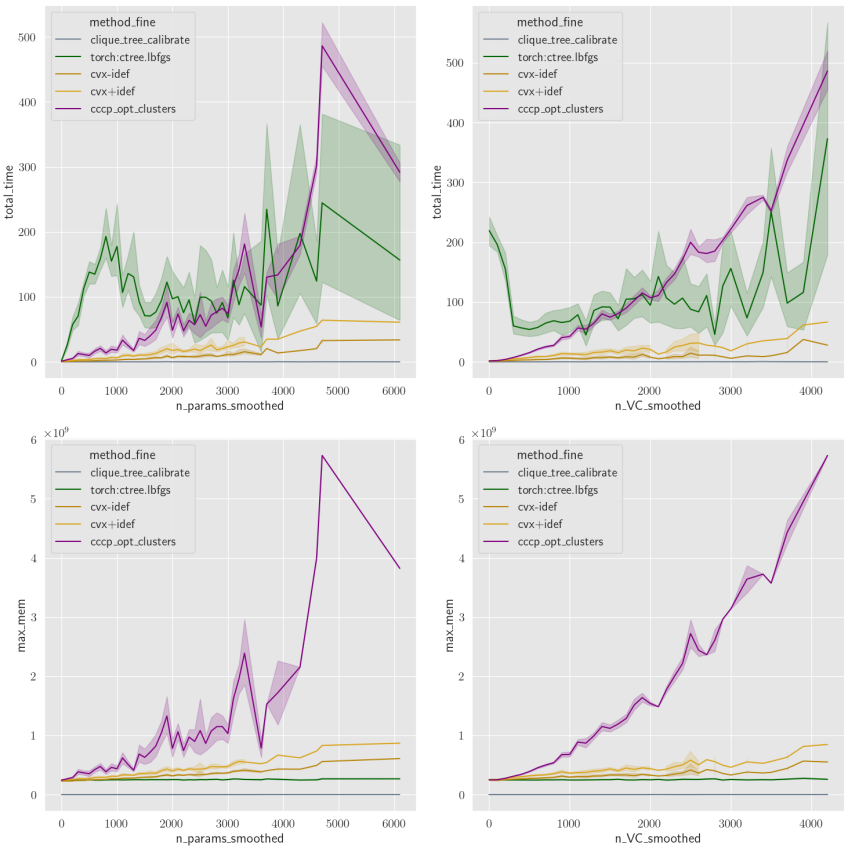


Figure 6: Resource costs for the cluster setting. Once again, the $OInc$-optimimzing exponential cone methods are in gold, the small-gamma and CCCP is in violet, and the baselines are in green. The bottom line is belief propogation, which is significantly faster and requires very little memory, but also only gives the correct answer under very specific cirucumstances.

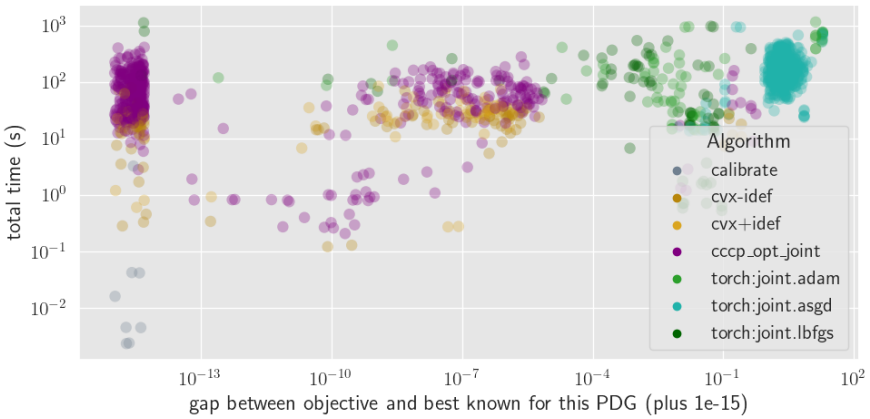


Figure 7: Gap vs inference time for the small PDGs in the `bnlearn` repository

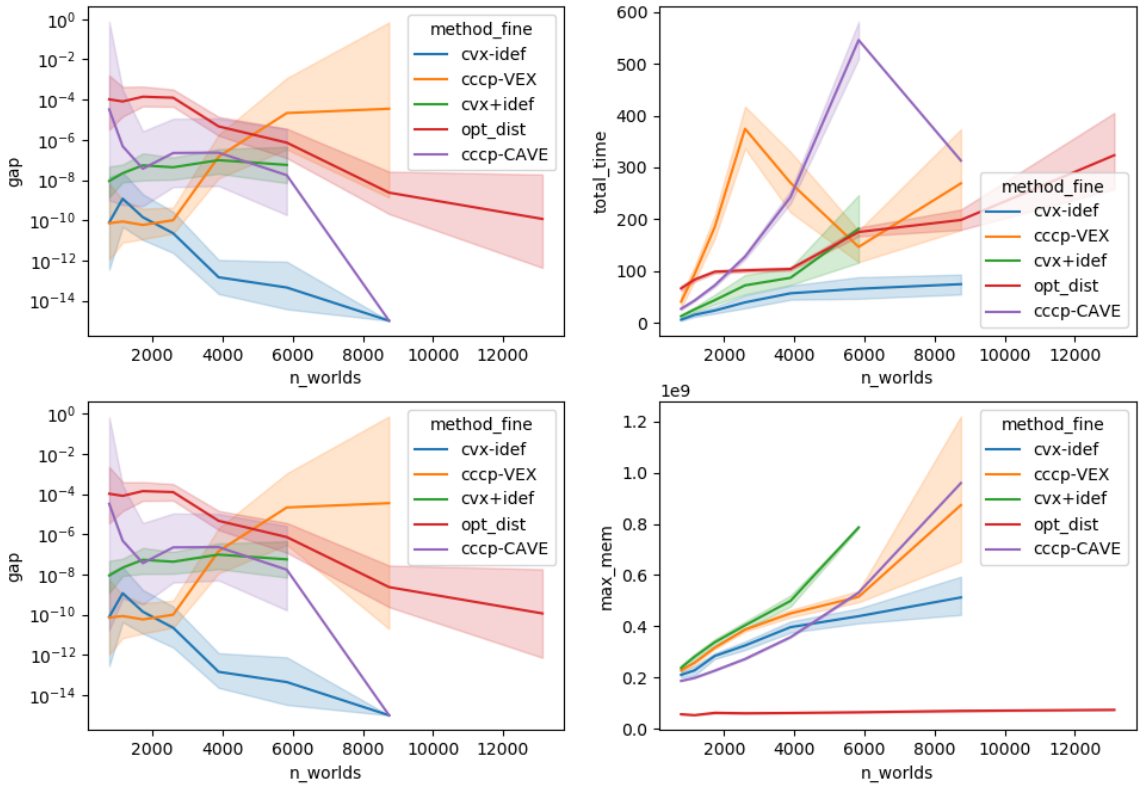


Figure 8: A variant of Figure 1, with with gap (accuracy) information on the left, and slightly different parameter settings.

maximum number of iterations, none of which are clear-cut choices, but rather require first being tuned to the data. While the convex solver does have internal parameters (tolerences and such) these do not need to be tuned to the problem under normal circumstances.

C.3 COMPARING TO BELIEF PROPAGATION, ON CLIQUE TREES.

Since PDGs generalize other graphical models, one might wonder how our method stacks up against algorithms tailored to the more traditional models. In brief: our algorithm is much slower, and only handle much smaller networks. Concretely, our methods can handle all of the "small" networks, and some of the "medium" ones, from the `bnlearn` repository. In these cases, we have verified that the two methods yield the same results. Figure 7 contains the analogue of Figures 1 and 2 for the Bayesian Nets. This graph looks qualitatively quite similar to the other graphs we've seen, suggesting that the results in our synthetic experiments hold more broadly for small real-world models as well.

References

Riley Badenbroek and Joachim Dahl. An algorithm for nonsymmetric conic optimization inspired by mosek. *Optimization Methods and Software*, pages 1–38, 2021.

Joachim Dahl and Erling D Andersen. A primal-dual interior-point algorithm for nonsymmetric exponential-cone optimization. *Mathematical Programming*, 194(1):341–370, 2022.

Ran Duan, Hongxun Wu, and Renfei Zhou. Faster matrix multiplication via asymmetric hashing. *arXiv preprint*, 2022. doi: 10.48550/ARXIV.2210.10173. URL https://arxiv.org/abs/2210.10173.

Joseph Y Halpern. *Reasoning About Uncertainty*. MIT press, 2017.

Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT press, 2009.

David J. C. MacKay. *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, 2003.

Yu E Nesterov, Michael J Todd, and Yinyu Ye. Infeasible-start primal-dual methods and infeasibility detectors for nonlinear programming problems. Technical report, 1999.

Yurii Nesterov and Arkadii Nemirovskii. *Interior-Point Polynomial Algorithms in Convex Programming*. SIAM, 1994.

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

Oliver E Richardson and Joseph Y Halpern. Probabilistic dependency graphs. *AAAI '21*, 2021.

Anders Skajaa and Yinyu Ye. A homogeneous interior-point algorithm for nonsymmetric convex conic optimization. *Mathematical Programming*, 150(2):391–422, 2015.

Alan L Yuille and Anand Rangarajan. The concave-convex procedure. *Neural computation*, 15(4):915–936, 2003.