A Hardness of General Distribution

We first recall the following definitions:

Counting. Given input polynomial-time computable weight function $w(\cdot)$ and query function $Q(\cdot)$, parameters α, a real number $\epsilon > 0$, a COUNTING oracle outputs a real number Z such that

$$1 - \epsilon \leq \frac{Z}{\mathbb{E}[\sigma \sim \pi_\alpha]Q(\sigma)} \leq 1 + \epsilon.$$

Robustness. Given input polynomial-time computable weight function $w(\cdot)$ and query function $Q(\cdot)$, parameters α, two real numbers $\epsilon > 0$ and $\delta > 0$, a ROBUSTNESS oracle decides, for any $\alpha' \in P^{[m]}$ such that $\|\alpha - \alpha'\|_\infty \leq \epsilon$, whether the following is true:

$$|\mathbb{E}[\sigma \sim \pi_\alpha]Q(\sigma) - \mathbb{E}[\sigma \sim \pi_{\alpha'}]Q(\sigma)| < \delta.$$

Proof of Theorem 1

Theorem 1 (COUNTING $\leq_{\mathrm{t}}$ ROBUSTNESS)**.** Given polynomial-time computable weight function $w(\cdot)$ and query function $Q(\cdot)$, parameters α and real number $\epsilon > 0$, the instance of COUNTING, (w, Q, α, ϵ) can be determined by up to $O(1/\varepsilon_c^2)$ queries of the ROBUSTNESS oracle with input perturbation $\epsilon = O(\varepsilon_c)$.

Proof. Let (w, Q, α, ϵ) be an instance of COUNTING. Define a new distribution τ_β over X with a single parameter $\beta \in \mathbb{R}$ such that $\tau_\beta(\sigma) \propto t(\sigma; \beta)$, where $t(\sigma; \beta) = w(\sigma; \alpha)\exp(\beta Q(\sigma))$. Since Q is polynomial-time computable, τ_β is accessible for any β. We will choose β later. For $i \in \{0, 1\}$, define $Z_i := \sum_{\sigma: Q(\sigma)=i} w(\sigma; \alpha)$. Then we have

$$\mathbb{E}[\sigma \sim \pi_\alpha]Q(\sigma) = \frac{Z_1}{Z_0 + Z_1}, \quad \mathbb{E}[\sigma \sim \tau_\beta]Q(\sigma) = \frac{e^\beta Z_1}{Z_0 + e^\beta Z_1}.$$

We further define

$$\begin{aligned} Y^+(\beta, x) &:= \mathbb{E}[\sigma \sim \tau_{\beta+x}]Q(\sigma) - \mathbb{E}[\sigma \sim \tau_\beta]Q(\sigma) \\ &= \frac{e^x e^\beta Z_1}{Z_0 + e^x e^\beta Z_1} - \frac{e^\beta Z_1}{Z_0 + e^\beta Z_1} \\ &= \frac{(e^x - 1)e^\beta Z_0 Z_1}{(Z_0 + e^x e^\beta Z_1)(Z_0 + e^\beta Z_1)} = \frac{(e^x - 1)e^\beta}{R + (e^x + 1)e^\beta + \frac{e^x e^{2\beta}}{R}}, \end{aligned}$$

where $R := \frac{Z_0}{Z_1}$, and similarly

$$\begin{aligned} Y^-(\beta, x) &:= \mathbb{E}[\sigma \sim \tau_\beta]Q(\sigma) - \mathbb{E}[\sigma \sim \tau_{\beta-x}]Q(\sigma) \\ &= \frac{e^\beta Z_1}{Z_0 + e^\beta Z_1} - \frac{e^{-x} e^\beta Z_1}{Z_0 + e^{-x} e^\beta Z_1} \\ &= \frac{(1 - e^{-x})e^\beta Z_0 Z_1}{(Z_0 + e^{-x} e^\beta Z_1)(Z_0 + e^\beta Z_1)} = \frac{(e^x - 1)e^\beta}{e^x R + (e^x + 1)e^\beta + \frac{e^{2\beta}}{R}}. \end{aligned}$$

Easy calculation implies that for $x > 0$, $Y^+(\beta, x) > Y^-(\beta, x)$ if and only if $R > e^\beta$. Note that

$$\begin{aligned} Y^+(\beta, x) &= \frac{e^x - 1}{Re^{-\beta} + e^x + 1 + \frac{e^{x+\beta}}{R}} \leq \frac{e^{x/2} - 1}{e^{x/2} + 1}; \\ Y^-(\beta, x) &= \frac{e^x - 1}{Re^{x-\beta} + e^x + 1 + \frac{e^\beta}{R}} \leq \frac{e^{x/2} - 1}{e^{x/2} + 1}. \end{aligned}$$

The two maximum are achieved when $R = e^{\beta \pm x/2}$. We will choose $x = O(\epsilon)$. Define

$$\begin{aligned} Y(\beta) &:= \max\{Y^+(\beta, x), Y^-(\beta, x)\} \\ &= \begin{cases} \frac{e^x - 1}{Re^{-\beta} + e^x + 1 + \frac{e^{x+\beta}}{R}} & \text{if } e^\beta < R; \\ \frac{e^x - 1}{Re^{x-\beta} + e^x + 1 + \frac{e^\beta}{R}} & \text{if } e^\beta \geq R. \end{cases} \end{aligned}$$

This function Y is increasing in $[0, \log R - x/2]$, decreasing in $[\log R - x/2, \log R]$, increasing in $[\log R, \log R + x/2]$ again, and decreasing in $[\log R + x/2, \infty)$ once again.

Our goal is to estimate R. For any fixed β, we will query the ROBUSTNESS oracle with parameters (t, Q, β, x, δ). Using binary search in δ, we can estimate the function $Y(\beta)$ above efficiently with additive error ϵ' with at most $O(\log \frac{1}{\epsilon'})$ oracle calls. We use binary search once again in β so that it stops only if $Y(\beta_0) \geq \frac{e^{x/2}-1}{e^{x/2}+1} - \varepsilon_0$ for some β_0 and the accuracy $\varepsilon_0 \leq \frac{e^{x/2}-1}{2(e^{x/2}+1)}$ is to be fixed later. In particular, $Y(\beta_0) \geq \frac{e^{x/2}-1}{2(e^{x/2}+1)}$. Note that here ε_0 is the accumulated error from binary searching twice.

We claim that β_0 is a good estimator for $\log R$. First assume that $e^{\beta_0} < R$, which implies that

$$\begin{aligned}
\frac{e^{x/2}-1}{e^{x/2}+1} - Y(\beta_0) &= \frac{e^{x/2}-1}{e^{x/2}+1} - \frac{e^x - 1}{Re^{-\beta_0} + e^x + 1 + \frac{e^{x+\beta_0}}{R}} \\
&= \frac{(e^x - 1)}{(e^{x/2}+1)^2(Re^{-\beta_0} + e^x + 1 + \frac{e^{x+\beta_0}}{R})} \times \\
&\quad \left(\sqrt{Re^{-\beta_0}} - \sqrt{\frac{e^{x+\beta_0}}{R}}\right)^2 \\
&= \frac{Y(\beta_0)}{(e^{x/2}+1)^2} \left(\sqrt{Re^{-\beta_0}} - \sqrt{\frac{e^{x+\beta_0}}{R}}\right)^2 \leq \varepsilon_0.
\end{aligned}$$

Thus,

$$\left|\sqrt{Re^{-\beta_0}} - \sqrt{\frac{e^{x+\beta_0}}{R}}\right| \leq \sqrt{\frac{2\varepsilon_0(e^{x/2}+1)^3}{e^{x/2}-1}}.$$

Let $\rho := Re^{-\beta_0}$. Note that $\rho > 1$. We choose $\varepsilon_0 := \frac{1}{2}\left(\frac{e^{x/2}-1}{e^{x/2}+1}\right)^3$. Then $\left|\sqrt{\rho} - \sqrt{e^x/\rho}\right| < e^{x/2} - 1$. If $\rho \geq e^x$, then $\left|\sqrt{\rho} - \sqrt{e^x/\rho}\right| \geq e^{x/2} - 1$, a contradiction. Thus, $\rho < e^x$. It implies that $1 < \frac{R}{e^{\beta_0}} < e^x$. Similarly for the case of $e^{\beta_0} > R$, we have that $e^{-x} < \frac{R}{e^{\beta_0}} < 1$. Thus in both cases, we have our estimator $e^{-x} < \frac{R}{e^{\beta_0}} < e^x$.

Finally, to estimate $\mathbb{E}[\sigma \sim \pi_\alpha]Q(\sigma) = \frac{1}{1+R}$ with multiplicative error ϵ, we only need to pick $x := \log(1+\epsilon) = O(\epsilon)$. □

B Robustness of MLN

Lagrange multipliers

Before proving the robustness result of MLN, we first briefly review the technique of Lagrange multipliers for constrained optimization: Consider following problem P,

$$\texttt{P:} \quad \max_{x_1, x_2} f(x_1) + g(x_2), \quad s.t., x_1 = x_2, \quad h(x_1), k(x_2) \geq 0.$$

Introducing another real variable λ, we define following problem P',

$$\texttt{P':} \quad \max_{x_1, x_2} f(x_1) + g(x_2) + \lambda(x_1 - x_2), \quad s.t., h(x_1), k(x_2) \geq 0.$$

For all λ, let (x_1^*, x_2^*) be the solution of P and let $(\bar{x_1}, \bar{x_2})$ be the solution of P', we have

$$f(x_1^*) + g(x_2^*) \leq f(x_1^*) + g(x_2^*) + \lambda(x_1^* - x_2^*) \leq f(\bar{x_1}) + g(\bar{x_2}) + \lambda(\bar{x_1} - \bar{x_2})$$

Proof of Lemma 4.1

Lemma 4.1 (MLN Robustness)**.** Given access to partition functions $Z_1(\{p_i(X)\}_{i\in[n]})$ and $Z_2(\{p_i(X)\}_{i\in[n]})$, and a maximum perturbations $\{C_i\}_{i\in[n]}$, $\forall \epsilon_1, ..., \epsilon_n$, if $\forall i.\ |\epsilon_i| < C_i$, we have

that $\forall \lambda_1, ..., \lambda_n \in \mathbb{R}$,

$$\max_{\{|\epsilon_i<C_i|\}} \ln \mathbb{E}[R_{MLN}(\{p_i(X)+\epsilon_i\}_{i\in[n]})] \leq \max_{\{|\epsilon_i|<C_i\}} \widetilde{Z_1}(\{\epsilon_i\}_{i\in[n]}) - \min_{\{|\epsilon_i'|<C_i\}} \widetilde{Z_2}(\{\epsilon_i'\}_{i\in[n]})$$

$$\min_{\{|\epsilon_i<C_i|\}} \ln \mathbb{E}[R_{MLN}(\{p_i(X)+\epsilon_i\}_{i\in[n]})] \geq \min_{\{|\epsilon_i|<C_i\}} \widetilde{Z_1}(\{\epsilon_i\}_{i\in[n]}) - \max_{\{|\epsilon_i'|<C_i\}} \widetilde{Z_2}(\{\epsilon_i'\}_{i\in[n]})$$

where

$$\widetilde{Z_r}(\{\epsilon_i\}_{i\in[n]}) = \ln Z_r(\{p_i(X)+\epsilon_i\}_{i\in[n]}) + \sum_i \lambda_i \epsilon_i.$$

Proof. Consider the upper bound, we have

$$\begin{aligned}\max_{\{|\epsilon_i<C_i|\}} \ln \mathbb{E}[R_{MLN}(\{p_i(X)+\epsilon_i\}_{i\in[n]})] &= \max_{\{|\epsilon_i<C_i|\}} \ln\left(\frac{Z_1(\{p_i(X)+\epsilon_i\}_{i\in[n]})}{Z_2(\{p_i(X)+\epsilon_i\}_{i\in[n]})}\right) \\ &= \max_{\{\epsilon_i\},\{\epsilon_i'\}} \ln Z_1(\{p_i(X)+\epsilon_i\}_{i\in[n]}) \\ &\quad - \ln Z_2(\{p_i(X)+\epsilon_i'\}_{i\in[n]}) \\ &\quad s.t., \epsilon_i = \epsilon_i', \quad |\epsilon_i|, |\epsilon_i'| \leq C_i.\end{aligned}$$

Introducing Lagrange multipliers $\{\lambda_i\}$. Note that any choice of $\{\lambda_i\}$ corresponds to a valid upper bound. Thus $\forall \lambda_1, ..., \lambda_n \in \mathbb{R}$, we can reformulate the above into

$$\begin{aligned}\max_{\{|\epsilon_i<C_i|\}} \ln \mathbb{E}[R_{MLN}(\{p_i(X)+\epsilon_i\}_{i\in[n]})] \leq \max_{\{\epsilon_i\},\{\epsilon_i'\}} &\ln Z_1(\{p_i(X)+\epsilon_i\}_{i\in[n]}) \\ &- \ln Z_2(\{p_i(X)+\epsilon_i'\}_{i\in[n]}) \\ &+ \sum_i \lambda_i(\epsilon_i - \epsilon_i'), \\ &s.t., |\epsilon_i|, |\epsilon_i'| \leq C_i.\end{aligned}$$

Define

$$\widetilde{Z_1}(\{\epsilon_i\}_{i\in[n]}) = \ln Z_1(\{p_i(X)+\epsilon_i\}_{i\in[n]}) + \sum_i \lambda_i \epsilon_i;$$

$$\widetilde{Z_2}(\{\epsilon_i'\}_{i\in[n]}) = \ln Z_1(\{p_i(X)+\epsilon_i'\}_{i\in[n]}) + \sum_i \lambda_i \epsilon_i',$$

We have the claimed upper-bound,

$$\max_{\{|\epsilon_i<C_i|\}} \ln \mathbb{E}[R_{MLN}(\{p_i(X)+\epsilon_i\}_{i\in[n]})] \leq \max_{\{|\epsilon_i|<C_i\}} \widetilde{Z_1}(\{\epsilon_i\}_{i\in[n]}) - \min_{\{|\epsilon_i'|<C_i\}} \widetilde{Z_2}(\{\epsilon_i'\}_{i\in[n]}).$$

Similarly, the lower-bound can be written in terms of Lagrange multipliers, and $\forall \lambda_1, ..., \lambda_n \in \mathbb{R}$, we have

$$\begin{aligned}\min_{\{|\epsilon_i<C_i|\}} \ln \mathbb{E}[R_{MLN}(\{p_i(X)+\epsilon_i\}_{i\in[n]})] \geq \min_{\{\epsilon_i\},\{\epsilon_i'\}} &\ln Z_1(\{p_i(X)+\epsilon_i\}_{i\in[n]}) \\ &- \ln Z_2(\{p_i(X)+\epsilon_i'\}_{i\in[n]}) \\ &+ \sum_i \lambda_i(\epsilon_i - \epsilon_i'), \\ &s.t., |\epsilon_i|, |\epsilon_i'| \leq C_i.\end{aligned}$$

Hence we have the claimed lower-bound,

$$\min_{\{|\epsilon_i<C_i|\}} \ln \mathbb{E}[R_{MLN}(\{p_i(X)+\epsilon_i\}_{i\in[n]})] \geq \min_{\{|\epsilon_i|<C_i\}} \widetilde{Z_1}(\{\epsilon_i\}_{i\in[n]}) - \max_{\{|\epsilon_i'|<C_i\}} \widetilde{Z_2}(\{\epsilon_i'\}_{i\in[n]}).$$

□

C Supplementary Results for Algorithm 1

Proposition (Monotonicity)**.** When $\lambda_i \geq 0$, $\widetilde{Z_r}(\{\epsilon_i\}_{i\in[n]})$ monotonically increases w.r.t. ϵ_i; When $\lambda_i \leq -1$, $\widetilde{Z_r}(\{\epsilon_i\}_{i\in[n]})$ monotonically decreases w.r.t. ϵ_i.

Proof. Recall that by definition we have

$$Z_r(\{p_i(X)+\epsilon_i\}_{i\in[n]}) = \sum_{\sigma\in\mathcal{I}_r} \exp\left\{\sum_{G_i\in\mathcal{G}} w_{G_i}(p_i(X)+\epsilon_i)\sigma(x_i) + \sum_{H\in\mathcal{H}} w_H f_H(\sigma(\bar{\mathbf{v}}_H))\right\}$$

where $w_{G_i}(p_i(x)) = \log[p_i(X)/(1-p_i(X))]$ and $\mathcal{I}_1 = \Sigma \wedge \{\sigma(v)=1\}$ and $\mathcal{I}_2 = \Sigma$. We can rewrite the perturbation on $p_i(X)$ as a perturbation on w_{G_i}: $w_{G_i}(p_i(X)+\epsilon_i) = w_{G_i} + \tilde{\epsilon}_i$,

where

$$\tilde{\epsilon}_i = \log\left[\frac{(1-p_i(X))(p_i(X)+\epsilon_i)}{p_i(X)(1-p_i(X)-\epsilon_i)}\right].$$

Note that $\tilde{\epsilon}_i$ is monatomic in ϵ_i. We also have

$$\ln\mathbb{E}[R_{MLN}(\{p_i(X)+\epsilon_i\}_{i\in[n]})] = \ln\mathbb{E}[R_{MLN}(\{w_{G_i}(X)+\tilde{\epsilon}_i\}_{i\in[n]})]$$

We can hence apply the same Lagrange multiplier procedure as in the above proof of Lemma 6 and conclude that

$$\begin{aligned}\widetilde{Z_r}(\{\epsilon_i\}_{i\in[n]}) &:= \ln Z_r(\{p_i(X)+\epsilon_i\}_{i\in[n]}) + \sum_i \lambda_i\epsilon_i \\ &= \ln Z_r(\{w_{G_i}(X)+\tilde{\epsilon}_i\}_{i\in[n]}) + \sum_i \lambda_i\tilde{\epsilon}_i,\end{aligned}$$

where $\epsilon_i \in [-C_i, C_i]$ $\tilde{\epsilon}_i \in [-C_i', C_i']$ with $C_i' = \log\left[\frac{(1-p_i(X))(p_i(X)+C_i)}{p_i(X)(1-p_i(X)-C_i)}\right]$. We are now in the position to rewrite $\widetilde{Z_r}$ as a function of $\tilde{\epsilon}_i$ and obtain

$$\begin{aligned}&\widetilde{Z_r}(\{\epsilon_i'\}_{i\in[n]}) \\ =&\ln\sum_{\sigma\in\mathcal{I}_r}\exp\left\{\sum_{G_i\in\mathcal{G}}(w_{G_i}+\tilde{\epsilon}_i)\sigma(x_i) + \sum_{H\in\mathcal{H}} w_H f_H(\sigma(\bar{\mathbf{v}}_H))\right\} + \sum_i \lambda_i\tilde{\epsilon}_i \\ =&\ln\sum_{\sigma\in\mathcal{I}_r}\exp\left\{\sum_{G_i\in\mathcal{G}} w_{G_i}\sigma(x_i) + \sum_i(\sigma(x_i)+\lambda_i)\tilde{\epsilon}_i + \sum_{H\in\mathcal{H}} w_H f_H(\sigma(\bar{\mathbf{v}}_H))\right\}\end{aligned}$$

Since $\sigma(x_i)\in\{0,1\}$, when $\lambda_i \geq 0$, $\sigma(x_i)+\lambda_i \geq 0$ and $\widetilde{Z_r}$ monotonically increases in $\tilde{\epsilon}_i$ and hence in ϵ_i. When $\lambda_i \leq -1$, $\sigma(x_i)+\lambda_i \leq 0$ and $\widetilde{Z_r}$ monotonically decreases in $\tilde{\epsilon}_i$ and hence in ϵ_i. □

Proposition (Convexity)**.** $\widetilde{Z_r}(\{\tilde{\epsilon}_i\}_{i\in[n]})$ is a convex function in $\tilde{\epsilon}_i, \forall i$ with

$$\tilde{\epsilon}_i = \log\left[\frac{(1-p_i(X))(p_i(X)+\epsilon_i)}{p_i(X)(1-p_i(X)-\epsilon_i)}\right].$$

Proof. We take the second derivative of $\widetilde{Z_r}$ with respect to $\tilde{\epsilon}_i$,

$$\frac{\partial^2\widetilde{Z_r}}{\partial\epsilon_1^2} = \frac{\sum_{\sigma\in\mathcal{I}_r}(\sigma(x_i)+\lambda_i)^2\exp\left\{\sum_{G_j\in\mathcal{G}} w_{G_j}\sigma(x_j) + \sum_j(\sigma(x_j)+\lambda_j)\tilde{\epsilon}_j + \sum_{H\in\mathcal{H}} w_H f_H(\sigma(\bar{\mathbf{v}}_H))\right\}}{\sum_{\sigma\in\mathcal{I}_r}\exp\left\{\sum_{G_j\in\mathcal{G}} w_{G_j}\sigma(x_j) + \sum_j(\sigma(x_j)+\lambda_j)\tilde{\epsilon}_j + \sum_{H\in\mathcal{H}} w_H f_H(\sigma(\bar{\mathbf{v}}_H))\right\}}$$

$$-\left[\frac{\sum_{\sigma\in\mathcal{I}_r}(\sigma(x_i)+\lambda_i)\exp\left\{\sum_{G_j\in\mathcal{G}} w_{G_j}\sigma(x_j) + \sum_j(\sigma(x_j)+\lambda_j)\tilde{\epsilon}_j + \sum_{H\in\mathcal{H}} w_H f_H(\sigma(\bar{\mathbf{v}}_H))\right\}}{\sum_{\sigma\in\mathcal{I}_r}\exp\left\{\sum_{G_j\in\mathcal{G}} w_{G_j}\sigma(x_j) + \sum_j(\sigma(x_j)+\lambda_j)\tilde{\epsilon}_j + \sum_{H\in\mathcal{H}} w_H f_H(\sigma(\bar{\mathbf{v}}_H))\right\}}\right]^2.$$

The above is simply the variance of $\sigma(x_i)+\lambda_i$, namely $\mathbb{E}\left[(\sigma(x_i)+\lambda_i)^2\right] - \mathbb{E}\left[\sigma(x_i)+\lambda_i\right]^2 \geq 0$. The convexity of $\widetilde{Z_r}$ in $\tilde{\epsilon}_i$ follows. □

D Image Classification on Road Sign Dataset

All the experiments shown in Appendix D - I are run on 4 RTX 2080 Ti GPUs.

Task and Dataset. For road sign classification task, the whole dataset can be viewed as a subset of GTSRB dataset [44], which contains 12 types of German road signs {"Stop", "Priority Road", "Yield", "Construction Area", "Keep Right", "Turn Left", "Do not Enter", "No Vihicles", "Speed Limit 20", "Speed Limit 50", "Speed Limit 120", "End of Previous Limitation"}, with 14880 training samples, 972 validation samples and 3888 testing samples in total. Besides the road sign classes, we construct 13 attribute classes as follows:

- Border shape classes: "Octagon", "Square", "Triangle", "Circle".
- Border color classes: "Red", "Blue", "Black".
- Digit classes: "Digit 20", "Digit 50", "Digit 120".
- Content classes: "Left", "Right", "Blank".

Based on the indication direction from road sign classes to attribute classes, and the exclusive relationship between attribute classes with the same type, we develop the following two types of knowledge rules as follows:

- Indication rules (u, v): Road sign class u indicates attribute v.
- Exclusion rules (u, v): Attribute classes u and v with the same type ("Shape", "Color", "Digit" or "Content") are naturally exclusive. (e.g., One road sign can not have "Octagon" shape and "Triangle" shape at the same time.)

Knowledge. We construct our first-order logical rules based on our predefined indication and exclusion knowledge as follows:

- Indication edge $u \implies v$: if one object belongs to road sign class u, it should have attribute v:

$$x_u \wedge \neg x_v = \texttt{False} \tag{1}$$

- Exclusion edge $u \oplus v$: On object can not have attribute u and v at the same time:

$$x_u \wedge x_v = \texttt{False} \tag{2}$$

Intuitive Example. Following the HEX graph-based knowledge structure and rules, we will show several adversary scenarios which could be mitigated through the inference reasoning phase. For instance, if the "Construction Area" object is attacked to be "Stop Sign" while other sensing nodes remain unaffected, like the border shape is still detected as the "Triangle" shape. Then the indication knowledge rule (The "Stop Sign" object should have the "Octagon" border shape) and the exclusive knowledge rule (No class can have the "Triangle" border shape and "Octagon" shape at the same time) would be violated. Such violation of the knowledge rules would discourage our pipeline to predict "Stop Sign" as what the attacker wants. However, the sensing-reasoning pipeline may not distinguish the "Yield", and "Construction Area" classes if the attacker fooled the "Construction Area" sensing completely, which shows the limitation of such structural knowledge, and more knowledge would be required in this case to help improve the robustness.

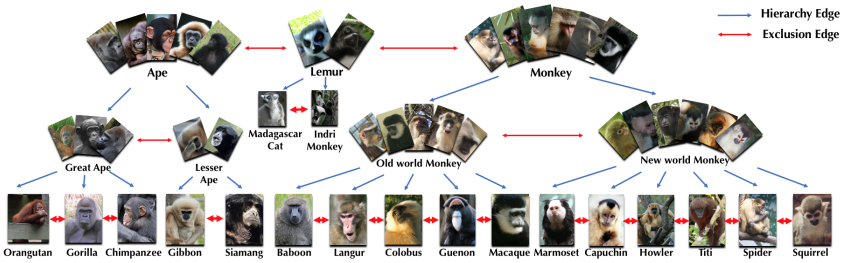

Figure 4: **PrimateNet**. The knowledge structure of PrimateNet dataset. The **Blue** arrows represent the Hierarchical rules between different classes, and the **Red** arrows represent the Exclusive rules. Some exclusive rules are omitted due to the space limit.

E Information Extraction on Stock News

To further evaluate the certified robustness of the reasoning component, in this section we will focus on the perturbation directly added to the reasoning component (e.g. C_S in Figure 1).

Tasks and Dataset. We consider information extraction tasks in NLP based on a stock news dataset — HighTech dataset which consists of both daily closing asset price and financial news from *2006* to *2013* [12]. We choose 9 companies with the most news, resulting in 4810 articles related to 9 stocks filtered by company name. We split the dataset into training and testing days chronologically. We define three information extraction tasks as our sensing models: `StockPrice(Day, Company, Price)`, `StockPriceChange(Day, Company, Percent)`, `StockPriceGain(Day, Company)`. The domain knowledge that we integrate depicts the relationships between these relations. We describe the three information extraction tasks in details below:

- **StockPrice(day, company, price)** In this task, we aim to extract the daily closing price of the stock from the article. We first extract numbers in every sentence from the article as candidate relations. Then we label every relation by the given daily closing asset price: label the relation whose number starts with "$" and has the minimum difference with the given closing price as positive and label others as negative. We train a BERT-based classifier [11] as the sensing model to judge the relation of whether the number was the closing price of the stock on that day and output the confidence.
- **StockPriceChange(day, company, percent)** In this task, we want to extract the percentage that the closing price of the stock changed from the collected news articles. We first extract numbers in every sentence from the articles as candidate relations. Then we label every relation via yesterday's and today's closing asset price. We train a BERT-based classifier as the sensing model to judge the relations of whether the number was the change rate of the closing price of the stock on that day and output the confidence.
- **StockPriceGain(day, company, gain)** In this task, we want to extract information about whether the closing price of the stock rose or fell on the day based on the news article. We treat each sentence with the stock name and the numbers which start with "$" as a candidate. Then we judge each relationship by whether it indicates the stock price rose or fell by counting the positive and negative words in the sentence. We label the relation as positive: when *Count(positive word)* > *Count(negative words)*; and negative: when *Count(positive word)* < *Count(negative words)*. We train a BERT-based classifier as the sensing model and output the confidence.
- **StockPriceGain(day, company, gain)** In this task, we want to extract information about whether the closing price of the stock rose or fell on the day based on the news article. We treat each sentence with the stock name and the numbers which start with "$" as a candidate. Then we judge each relationship by whether it indicates the stock price rose or fell by counting the positive and negative words in the sentence. We label the relation as positive: when *Count(positive word)* > *Count(negative words)*; and negative: when *Count(positive word)* < *Count(negative words)*. We train a BERT-base classifier as the sensing model and output the confidence.

Implementation Details. To train BERT classifiers, we use the final hidden state of the first token [CLS] from BERT as the representation of the whole input and we apply dropout with probability $p = 0.5$ on this final hidden state. A fully connected layer is added to the top of BERT for classification. To fine-tune the BERT classifiers for three information tasks, we use Adam

optimization with a learning rate of 10^{-5} and weight decay of 10^{-4}. We train our classifiers for 30 epochs with the batch size of 32.

Knowledge. We construct a new test set for the above three tasks. Specifically, for each news article, given the current date d and company name, we extract stock price p_1 on the current date, and stock price p_0 on the date before the current date. We also predict whether the stock price goes up or down y ($y = 0$ if the prediction is "down" otherwise $y = 1$) and extract the percentage of stock price change β. The extracted information forms a 4-tuple (p_0, p_1, y, β) that satisfies the following rules (knowledge):

- **Rule 1**: The extracted stock price p_0 and p_1 (sensing model `StockPrice`) should be consistent with the stock price change prediction (sensing model `StockPriceGain`).

$$y = \mathbb{I}[p_1 - p_0 > 0] \tag{3}$$

- **Rule 2**: The extracted stock price p_0 and p_1 (sensing model `StockPrice`) should be consistent with the percentage change of stock price prediction (sensing model `StockPriceChange`).

$$p_1 \approx p_0 \times [1 + (-1)^{\mathbb{I}[p_1 - p_0 > 0]} \times \beta] \tag{4}$$

Threat Model. We attack sensing models by adding perturbations on a sensing group's top-1 confidence value P without change other choices' confidence value on the perturbed sensing position: $P' = \text{clip}(P - C_S, 10^{-5}, 1 - 10^{-5})$, where C_S is the perturbation scale on the confidence output of sensing models. In our attack setting, we add perturbations to all sensing groups.

Intuitive Example. Here we show an intuitive example of how our knowledge can help improve the ML robustness under adversarial attacks. Assume our sensors extract the correct stock price information $(p_0^*, p_1^*, y^*, \beta^*)$, where price $p_0^* > p_1^*$ and the stock price change is "down" ($y = 0$) by $\beta\%$. Now if the first stock price extraction sensor is attacked to output an incorrect prediction p_0' such that $p_0' < p_1^*$ while other sensors remain intact; p_0' will violate our knowledge rules 1 and 2. Specifically, the stock price change $p_0' - p_1^* < 0$ is inconsistent with stock price change prediction $y = 0$, i.e., $p_0' - p_1^* > 0$. As a result, our reasoning component will reduce the confidence of the wrong prediction p_0' and increase the confidence of the ground truth p_0^* as it is consistent with knowledge rules, therefore potentially recovering the correct prediction of p_0^*.

F Image Classification on PrimateNet Dataset

Task and Dataset. We aim to evaluate the certified robustness of our sensing-reasoning pipeline on large-scale dataset such as ImageNet ILSVRC2012 [9]. In particular, to obtain domain knowledge for the images, we select 18 Primate animal categories to form a PrimateNet dataset, containing {Orangutan, Gorilla, Chimpanzee, Gibbon, Siamang, Madagascar cat, Woolly indris, Guenon, Baboon, Macaque, Langur, Colobus, Marmosets, Capuchin monkey, Howler monkey, Titi monkey, Spider monkey, Squirrel monkey}. Moreover, we create 7 internal classes {Greater ape, Lesser ape, Ape, Lemur, Old-world monkey, New-world monkey, Monkey} to construct the hierarchical structure according to the WordNet [14]. With such a hierarchical structure, we can build the Primate-class Hierarchy and Exclusion(HEX) graph based on the concepts from [10] as shown in Fig 4. Within the HEX graph, we develop two types of knowledge rules described as follows:

- Hierarchy rules (u, v): class u subsumes class v (e.g. Great Ape subsumes Gorilla);
- Exclusion edge (u, v): class u and class v are naturally exclusive (e.g. Gorilla cannot belong to Great Ape and Lesser Ape at the same time).

We consider each class in the HEX graph as the prediction of one sensing model in the sensing-reasoning pipeline, and we construct 25 sensing models as the leaf and internal nodes in the HEX graph. Here we use the MLN as our reasoning component connecting to these sensing models.

Implementation details. For each leaf sensing model, we utilize 1300 images from the ILSVRC2012 training set and 50 images from the ILSVRC2012 dev set. We split the 1300 images into 1000 images for training and 300 for testing. For each internal node, we uniformly sample the training images from all its children nodes' training images to form its training set with the same size 1300, since there are no specific instances belonging to internal nodes' categories in PrimateNet.

During training, we utilize the sensing DNN model for each node in the knowledge hierarchy to output the probability value given the input images. The models consist of a pre-trained ResNet18 feature extractor concatenated by two Fully-Connected layers with ReLU activation. In order to provide the certified robustness of the end-to-end sensing-reasoning pipeline, we adapt the randomized smoothing strategy mentioned in [8] to certify the robustness of sensing models, and then compose it with the certified robustness of the reasoning component. Specifically, we smoothed our sensing models by adding the isotropic Gaussian noise $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$ to the training images during training. We train each sensing model for 80 epochs with the Adam optimizer (initial learning rate is set to 2×10^{-4}) and evaluate the sensing models' performance on the validation set containing 50 images after every training epoch to avoid over-fitting. During testing, we certify the robustness of trained sensing models with the same smoothing parameter σ used during model training.

Knowledge. The knowledge used in this task includes the *hierarchical* and *exclusive* relationships between different categories of the sensing predictions. For instance, the category "Ape" would include all the instances classified as "Greater ape, Lesser ape" (hierarchical); and there should not be any intersection for instances predicted as "Monkey" or "Lemur" (exclusive). Thus, we build our knowledge rules based on the structural relationships such as hierarchy and exclusion knowledge:

- **Hierarchy edge** $u \implies v$: If one object belongs to class u, it should belong to class v as well:

$$x_u \wedge \neg x_v = \texttt{False} \tag{5}$$

- **Exclusion edge** $u \oplus v$: One object should not belong to class u and class v at the same time:

$$x_u \wedge x_v = \texttt{False} \tag{6}$$

Threat Model. In this paper, we consider a strong attacker who has access to perturbing several sensing models' input instances during inference time. To perform the attack, the attacker will add perturbation δ, bounded by C_I under the ℓ_2 norm, onto the test instance against the victim sensing models: $||\delta||_2 < C_I$. In particular, we consider the attacker to attack α percent of the total sensing models.

Since we apply randomized smoothing to sensing models during training, for each sensing model, we can certify the output probability p' as a function of the original confidence p, the bound of the perturbation C_I, and smoothing parameter σ according to Corollary 2 as below:

$$p' \in [\Phi(\Phi^{-1}(p) - C_I/\sigma), \Phi(\Phi^{-1}(p) + C_I/\sigma)].$$

Evaluation Metrics. To evaluate the certified robustness of sensing-reasoning pipeline, we focus on the standard *certified accuracy* on a given test set, and the *certified ratio* measuring the percentage of instances that could be certified within a certain perturbation magnitude/radius.

Based on the previous analysis, given the ℓ_2 based perturbation bound C_I, we can certify the output probability of the sensing-reasoning pipeline as $[\mathcal{L}, \mathcal{U}]$. In order to evaluate the certified robustness of sensing-reasoning pipeline, we define the **Certified Robustness**, measuring the percentage of instances that could be certified to make correct prediction within a perturbation radius, to evaluate the certified robustness following existing work [8], which is formally defined as:

$$\frac{\sum_{i=1}^{N} \mathbb{I}(([\mathcal{U}_i < 0.5] \wedge [y_i = 0]) \vee ([\mathcal{L}_i > 0.5] \wedge [y_i = 1]))}{N},$$

where N refers to the number instances and y_i the ground truth label of the given instance i. $\mathbb{I}(\cdot)$ is an indicator function which outputs 1 when its argument takes value true and 0 otherwise.

Moreover, we report the **Certified Ratio** to measure the percentage of instances that could be certified as a consistent prediction within a perturbation radius (even the consistent prediction might be wrong). The Certified Ratio is defined as:

$$\frac{\sum_{i=1}^{N} \mathbb{I}([\mathcal{U}_i < 0.5] \vee [\mathcal{L}_i > 0.5])}{N}.$$

Here the lower and upper bounds of the output probability $\mathcal{L}_i$ and $\mathcal{U}_i$ indicate the binary prediction of each sensing model. We assume when the output probability is less than 0.5, it outputs 0.

Table 3: *Benign* accuracy (i.e. $C_I = 0, \alpha = 0$) of models with and without knowledge under different smoothing parameters σ evaluated on PrimateNet.

σ	**With knowledge**	**Without knowledge**
0.12	**0.9670**	0.9638
0.25	**0.9612**	0.9554
0.50	**0.9435**	0.9371

Table 4: Certified Robustness and Certified Ratio under different perturbation magnitude C_I and sensing model attack ratio α on PrimateNet. The sensing models are smoothed with Gaussian noise $\epsilon \sim \mathcal{N}(0, \hat{\sigma}^2 I_d)$ with different smoothing parameter σ.

(a) $\hat{\sigma} = 0.12$

		With knowledge		**Without knowledge**	
C_I	α	Cert. Robustness	Cert. Ratio	Cert. Robustness	Cert. Ratio
0.12	10%	**0.8849**	**0.9419**	0.5724	0.5724
	20%	**0.8078**	**0.8609**	0.5717	0.5717
	30%	**0.7508**	**0.7988**	0.5706	0.5706
	50%	**0.6236**	**0.6647**	0.5706	0.5706
0.25	10%	**0.7888**	**0.8428**	0.2342	0.2342
	20%	**0.6226**	**0.6657**	0.2320	0.2320
	30%	**0.5225**	**0.5596**	0.2309	0.2309
	50%	**0.3594**	**0.3824**	0.2268	0.2268

(b) $\hat{\sigma} = 0.25$

		With knowledge		**Without knowledge**	
C_I	α	Cert. Robustness	Cert. Ratio	Cert. Robustness	Cert. Ratio
0.25	10%	**0.8498**	**0.9499**	0.5314	0.5314
	20%	**0.7608**	**0.8952**	0.5302	0.5302
	30%	**0.7217**	**0.8048**	0.5294	0.5294
	50%	**0.6026**	**0.6747**	0.5235	0.5235
0.50	10%	**0.7622**	**0.8489**	0.2024	0.2024
	20%	**0.5988**	**0.6467**	0.2024	0.2024
	30%	**0.5324**	**0.5541**	0.2010	0.2010
	50%	**0.3417**	**0.3615**	0.2000	0.2000

(c) $\hat{\sigma} = 0.50$

		With knowledge		**Without knowledge**	
C_I	α	Cert. Robustness	Cert. Ratio	Cert. Robustness	Cert. Ratio
0.50	10%	**0.8288**	**0.9449**	0.4762	0.4762
	20%	**0.7407**	**0.8488**	0.4749	0.4749
	30%	**0.6907**	**0.7968**	0.4736	0.4736
	50%	**0.5581**	**0.6395**	0.4635	0.4635
1.00	10%	**0.7307**	**0.8448**	0.1679	0.1679
	20%	**0.5285**	**0.6336**	0.1615	0.1615
	30%	**0.4347**	**0.5375**	0.1612	0.1612
	50%	**0.2624**	**0.3318**	0.1584	0.1584

Intuitive Example. Following the HEX graph-based knowledge structure and rules, we will show several adversary scenarios which could be mitigated through the inference reasoning phase. For instance, based on Figure 4, if one “Gorilla" object is attacked to be “Siamang" while other sensing nodes remain unaffected, the hierarchical knowledge rule (An object belongs to “Great Ape" class cannot belong to “Siamang" class) and the exclusive knowledge rule (No object could belong to “Great Ape" and “Siamang" classes at the same time) would be violated. Such violation of the knowledge rules would discourage our pipeline to predicting “Siamang" as what the attacker wants. However, the sensing-reasoning pipeline may not distinguish the “Orangutan", “Gorilla", and “Chimpanzee" classes if the attacker fooled the “Gorilla" sensing completely, which shows the limitation of such structural knowledge, and more knowledge would be required in this case to help improve the robustness.

Evaluation Results. We evaluate the robustness of the sensing-reasoning pipeline compared with the baseline which is consist of 25 randomized smoothed sensing models for each Primate categories (without knowledge). We evaluate the average certified robustness of both under benign and adversarial scenarios with different smoothing parameter $\hat{\sigma} \in \{0.12, 0.25, 0.50\}$ and ℓ_2 perturbation bound $C_I = \{\hat{\sigma}, 2\hat{\sigma}\}$. The evaluation results are shown in Table 4 and Table 3.

First, we evaluate both the sensing-reasoning pipeline and the smoothed ML model with benign test data as shown in Table 3. It is interesting that the sensing-reasoning pipeline with knowledge even outperforms the single model without knowledge about 0.7% over different randomized smoothing parameter σ. It shows that even without attacks, the knowledge could help to improve the classification

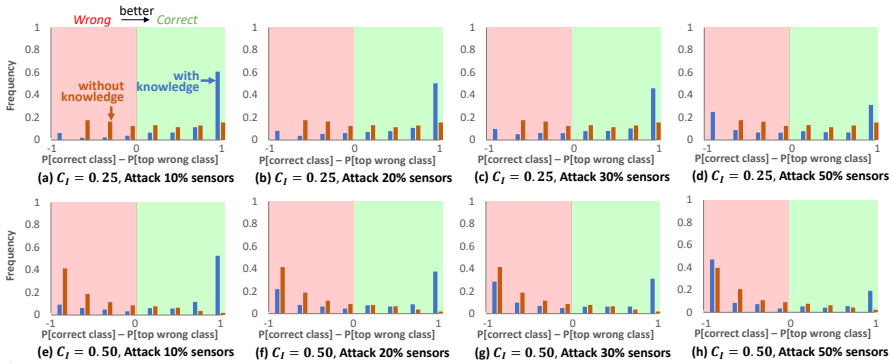

Figure 5: (**PrimateNet**) Histogram of the **robustness margin** (difference between the probability of the correct class (lower bound) and the top wrong class (upper bound)) under perturbation. If such a difference is positive, it means that the classifier makes the right prediction under perturbation. Evaluation is made under smoothing parameter $\sigma = 0.25$ with ℓ_2 perturbation scale $C_I = \{\sigma, 2\sigma\}$. The ratio of the attacked sensors α equals to $10\%, 20\%, 30\%, 50\%$.

accuracy slightly, indicating that the domain knowledge integration can help relax the tradeoff between benign accuracy and robustness.

Next, we evaluate the certified robustness of sensing-reasoning pipeline and the smoothed ML model considering different smoothing parameters $\hat{\sigma} = \{0.12, 0.25, 0.50\}$ and the input perturbation bound $C_I = \{\hat{\sigma}, 2\hat{\sigma}\}$ in Table 4. We can see that when the attack ratio of sensing models α is small, both the Certified Robustness and Certified Ratio of sensing-reasoning pipeline are significantly higher than that of the baseline smoothed ML model. In the meantime, when the sensing attack ratio α is large (e.g. 50%) both the sensing-reasoning pipeline and baseline smoothed ML model obtain low Certified Robustness and Certified Ratio, and their performance gap becomes small.

This is interesting and intuitive, since if a large percent of sensing models are attacked, such structure-based knowledge, for which the solution to a given regular expression is not unique, would have higher confidence to prefer the other (wrong) side of the prediction. As a result, it is interesting for future work to identify more "robust" knowledge which is resilient against the large attack ratio of sensing models, in addition to the hierarchical structure knowledge.

We also find that when $C_I/\hat{\sigma}$ is small ($C_I = \hat{\sigma}$), the model with knowledge can perform consistently better than the baseline ML models. When $C_I/\hat{\sigma}$ is large ($C_I = 2\hat{\sigma}$), the performance gap becomes even larger. This phenomenon indicates that sensing-reasoning pipeline could demonstrate its strength of robustness compared to the traditional smoothed DNN against an adversary with stronger ability.

To further evaluate the strength of our certified robustness, we calculate the *robustness margin* — the difference between the lower bound of the true class probability and the upper bound of the top wrong class probability under different perturbation scales — to inspect the robustness certification (larger difference infer stronger certification). Figure 5 shows the histogram of the robustness margin for the model with and without knowledge under smoothing parameter $\hat{\sigma} = 0.25$ and different perturbation scale C_I. We leave histogram figures under other σ settings in Appendix.

From Figure 5, we can see that under different adversary scenarios, more instances could receive the positive margin (i.e correct prediction) with sensing-reasoning pipeline, which indicates its robustness. Moreover, we find that the sensing-reasoning pipeline could output a large margin value with high frequency under various attacks. That means, it can certify the robustness of the ground truth class with high confidence, which is challenging for current certified robustness approaches for single ML models.

In addition, to evaluate the utility of different knowledge, we also develop sensing-reasoning pipeline by using only one type of knowledge (hierarchical or exclusive relationship only) and the results are shown in Appendix I. We observe that using partial knowledge, the robustness of sensing-reasoning pipeline would decrease compared with that using the full knowledge.

G Image Classification on Word50 Dataset

Task and Dataset. In addition, we also conduct experiments on Word50 dataset [6], which is created by randomly selecting 50 words and each consisting of five characters. Here we only pick 10 words from it to reduce the computation complexity, and the goal is to classify these 10 words. All the character images are of size 28×28 and perturbed by scaling, rotation, and translation. The background of the characters is blurry by inserting different patches, which makes it a quite challenging task. For reference, Some word images sampled from the dataset are shown in Figure 6. The interesting property of this dataset is that the character combination is given as the prior knowledge, which can be integrated into our sensing-reasoning pipeline. The training, validation, and test sets contain $2,049$, 408, and 423 variations of word styles respectively.

Similar to the classification task on Road Sign dataset, we develop the following two types of knowledge rules as follows:

- Deduction rules (u, v_i): word u contains character v_i on the ith position of the word.
- Exclusion rules (u_i, v_i): character u_i and character v_i are naturally exclusive on the ith position of the word.

Implementation details. Multi-layer perceptrons (MLPs) are used as the main model architecture for the main task that the classification of the 10 words, which is the same to [6], and the input is the concatenation of the images of 5 characters which consist of a full word. As for the extra knowledge, we train another five MLP models for the classification of the character on each position of the input word, then the corresponding output dimensions for each such character classifier is 26. While during the inference, we will only pick the top2 of the output from each character classifier, so the final input dimension to the MLN is $10 + 5 \times 2 = 20$ dimensions. Thus, to keep the certification probability the same as the baseline, the ζ_0 here will be set to $1 - (1 - 0.001)^{(1/20)} = 5.002 \times 10^{-5}$.

For these sensing models, we adapt the randomized smoothing strategy [8] to give the certified robustness guarantee of their output confidence under the ℓ_2-norm bounded perturbation. The w_H is set to 2 for the deduction rules, and the corresponding f_H is the identity function; while for the exclusion rules, the w_H is set to $-\infty$, and the f_H here is the negation function, namely, $f_H(v) = 1 - v$.

Knowledge. We construct our first-order logical rules based on our predefined Deduction and Exclusion knowledge rules:

- Deduction edge $u \implies v_i$:

$$x_u \wedge \neg x_{v_i} = \texttt{False} \tag{7}$$

- Exclusion edge $u_i \oplus v_i$:

$$x_{u_i} \wedge x_{v_i} = \texttt{False} \tag{8}$$

Threat Model. Same to the setting of the experiments on the Stop Sign dataset, here we consider a stronger attack scenario where the attacker can attack the main task model and all the attribute sensors with **different** ℓ_2-norm bounded perturbation $\delta : ||\delta||_2 < C_I$ at the same time. Later on, we can see our sensing-reasoning pipeline could still achieve higher end-to-end certified robustness under even harder cases.

Given the ℓ_2-norm bound C_I, for each sensing model, we can bound its output probability p' under such perturbation, given the original probability p and the certification smoothing parameter σ according to Corollary 2 as below:

$$p' \in [\Phi(\Phi^{-1}(p) - C_I/\sigma), \Phi(\Phi^{-1}(p) + C_I/\sigma)].$$

Evaluation Metrics. We adopt the standard *certified accuracy* as our evaluation metric, defined by the percentage of instances that can be certified under any ℓ_2-norm bounded perturbation $\delta : ||\delta||_2 < C_I$. Specifically, given the input x with ground-truth label y, we can certify the bound of confidence on predicting label y as $[\mathcal{L}, \mathcal{U}]$ for either a vanilla randomize smoothing-based model or our sensing-reasoning pipeline. After that, the certified accuracy can be defined by: $\frac{1}{N}\sum_{i=1}^{N} \mathbb{I}([\mathcal{L}_i > 0.5])$ where $\mathbb{I}(\cdot)$ denotes the indicator function.

Table 5: Certified accuracy under different perturbation magnitude C_I on Word10 dataset. The sensing models are smoothed with Gaussian noise $\epsilon \sim \mathcal{N}(0, \hat{\sigma}^2 I_d)$ with different smoothing parameter $\hat{\sigma}$. Rows with $*$ denote the best certified accuracy among all the $\hat{\sigma} \in \{0.12, 0.25, 0.50\}$. (All certificates holds with 99.9% confidence)

Methods	σ	$C_I = 0.12$	$C_I = 0.25$	$C_I = 0.50$	$C_I = 1.00$
Vanilla Smoothing (w/o knowledge)	0.12	58.2	49.2	0.0	0.0
	0.25	51.8	42.3	25.3	0.0
	0.50	42.6	33.1	19.1	2.6
	$*$	58.2	49.2	25.3	2.6
Sensing-Reasoning Pipeline (w/ knowledge)	0.12	**88.7**	**77.8**	**30.7**	**0.0**
	0.25	**95.0**	**90.8**	**52.5**	**2.8**
	0.50	**91.5**	**86.8**	**69.3**	**6.4**
	$*$	**95.0**	**90.8**	**69.3**	**6.4**

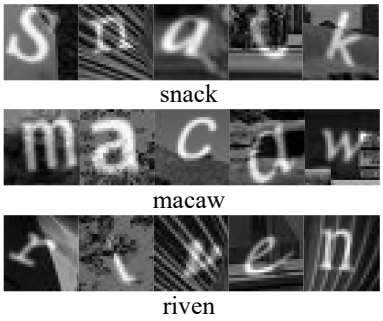

Figure 6: Several word images sampled from Word50 dataset.

Intuitive Example. To make the example more clear, here we use *pos('a'*, i) to represent that the character *'a'* is in the i*th* position of the word. Then during the inference, given an input word image, we assume the top2 characters returned from the character classifiers for each position is *'s,m'*, *'n,b'*, *'a,o'*, *'q,a'*, *'k,c'*, which are shown in the order of the position. Now, for word *'snack'*, the corresponding first-order logical form of its deduction rules would be *'snack'* $\implies$ *pos('s'*, 1), *'snack'* $\implies$ *pos('n'*, 2), *'snack'* $\implies$ *pos('a'*, 3) and *'snack'* $\implies$ *pos('k'*, 5); while for other words like *'macaw'*, the corresponding rules would be *'macaw'* $\implies$ *pos('m'*, 1) and *'macaw'* $\implies$ *pos('a'*, 4). Notice, if the character of the specific word is not shown in the top2 returned characters of its corresponding position, then there will be no deduction rule built for this word and this character. At the meantime, when we consider the possible worlds that satisfy $\sigma(x_{snack}) \wedge \sigma(v_{pos('q',4)}) = 1$, we will still consider it as a violation of the exclusive rules. In other words, even if the character *'c'* is not shown in the top2 characters returned from the knowledge classifier in fourth position and thus we do not build the deduction rule *'snack'* $\implies$ *pos('c'*, 4) explicitly at this time as said above, this rule is still assumed to be true underlyingly.

Evaluation Results. We evaluate the robustness of the sensing-reasoning pipeline and compare it to the baseline as a vanilla randomized smoothed main task model (without knowledge). We train our models under different smoothing parameters $\hat{\sigma} = \{0.12, 0.25, 0.50\}$ and evaluate our sensing-reasoning pipeline under various ℓ_2 perturbation magnitude $C_I = \{0.12, 0.25, 0.50, 1.00\}$. Results are show in Table 5, and as we can see, with extra knowledge, the performance is improved tremendously which strongly demonstrates the potential of the sensing-reasoning pipeline.

H Image Classification with Constructed Knowledge Rules

For natural image datasets with no apparent knowledge rules, we can still apply our sensing-reasoning pipeline based on some generated simple knowledge rules such as redundancy rules. For instance, we test on MNIST and CIFAR10 dataset by constructing basic rules as follows: for MNIST, we construct five pseudo attributes and randomly assign them to four different digits, so that each digit will exactly contain two pseudo attributes; for CIFAR10, we randomly generate ten pseudo attributes, and each pseudo attribute will be randomly assigned to 3 to 7 different categories. We build the indication rules between each pseudo attribute and its corresponding digits, and the exclusion rules between different digit classes.

During the training, we adopt the SOTA Consistency training [19] as our sensing model training method, and build our sensing-reasoning pipeline on top of these pretrained sensing models.

From the results shown in Table 6 and Table 7, we can see that the sensing-reasoning pipeline beats the SOTA baselines in terms of the certified robustness even with the simple and generated knowledge rules. Generally, we should expect higher certified robustness by integrating with natural and meaningful knowledge rules (e.g., road sign classification and information extraction tasks as shown in our paper).

Table 6: (**MNIST**) *Certified accuracy* under different input perturbation magnitudes (C_I).

Methods	$C_I = 0.00$	$C_I = 0.25$	$C_I = 0.50$	$C_I = 0.75$	$C_I = 1.00$	$C_I = 1.25$	$C_I = 1.50$	$C_I = 1.75$	$C_I = 2.00$
Consistency Training	99.5	98.9	**98.0**	96.0	93.0	87.8	78.5	60.5	41.7
Sensing-Reasoning Pipeline (Consistency)	**99.6**	**98.2**	97.6	**96.3**	**93.5**	**88.2**	**78.9**	**61.2**	**43.2**

Table 7: (**CIFAR10**) *Certified accuracy* under different input perturbation magnitudes (C_I).

Methods	$C_I = 0.00$	$C_I = 0.25$	$C_I = 0.50$	$C_I = 0.75$	$C_I = 1.00$	$C_I = 1.25$	$C_I = 1.50$	$C_I = 1.75$	$C_I = 2.00$
Consistency Training	77.8	68.8	**57.4**	43.8	36.2	29.5	22.9	19.7	16.6
Sensing-Reasoning Pipeline (Consistency)	**78.4**	**70.4**	56.2	**46.0**	**37.4**	**29.6**	**25.2**	**21.8**	**18.8**

I Ablation Study on Partial Knowledge Enrichment.

In PrimateNet experiments, we also investigate how Hierarchy knowledge and Exclusive knowledge would affect the *End-to-end* robustness of our sensing-reasoning pipeline individually. We compare the certified robustness and certified ratio of our sensing-reasoning pipeline enriched by {No knowledge; Hierarchy knowledge only; Exclusive knowledge only; Hierarchy + Exclusive knowledge} and the results are shown in Table 8 and 9.

From the results, we can see while partial knowledge enrichment would lead to fragile robustness under severe scenarios ($\alpha = 0.5$), complete knowledge enrichment could achieve much better robustness compared to sensing-reasoning pipeline without knowledge enrichment. This indicates that incomplete (or weak) knowledge, which is easy to break and hard to recover under severe adversarial scenarios, could even harm the robustness of our sensing-reasoning pipeline. How to explore good and robust knowledge to enrich our sensing-reasoning pipeline could be our interesting future direction.

Table 8: **Certified Robustness** with different perturbation magnitude C_I and sensing model attack ratio α on PrimateNet. The sensing models are smoothed with Gaussian noise $\epsilon \sim \mathcal{N}(0, \hat{\sigma}^2 I_d)$ with different smoothing parameter σ. Here **"Hierarchy."** refers to the sensing-reasoning pipeline enriched by hierarchy knowledge only while **"Exclusive."** the exclusive knowledge only. **"Combined."** shows the sensing-reasoning pipeline enriched by both domain knowledge.

(a) $\hat{\sigma} = 0.12$

C_I	α	**No knowledge**	**Hierarchy.**	**Exclusive.**	**Combined.**
0.12	10%	0.5724	0.7912	0.7020	**0.8849**
	20%	0.5717	0.6932	0.6236	**0.8078**
	30%	0.5706	0.6280	0.5624	**0.7508**
	50%	0.5706	0.4868	0.4320	**0.6236**
0.25	10%	0.2342	0.6670	0.5232	**0.7888**
	20%	0.2320	0.4704	0.3468	**0.6226**
	30%	0.2309	0.3632	0.3158	**0.5225**
	50%	0.2268	0.2122	0.2004	**0.3594**

(b) $\hat{\sigma} = 0.25$

C_I	α	**No knowledge**	**Hierarchy.**	**Exclusive.**	**Combined.**
0.25	10%	0.5314	0.7766	0.6998	**0.8498**
	20%	0.5302	0.6810	0.6002	**0.7608**
	30%	0.5294	0.6278	0.5464	**0.7217**
	50%	0.5235	0.4924	0.4126	**0.6026**
0.50	10%	0.2024	0.6754	0.5196	**0.7622**
	20%	0.2024	0.4636	0.3298	**0.5988**
	30%	0.2010	0.3680	0.2870	**0.5324**
	50%	0.2000	0.2204	0.1652	**0.3417**

(c) $\hat{\sigma} = 0.50$

C_I	α	**No knowledge**	**Hierarchy.**	**Exclusive.**	**Combined.**
0.50	10%	0.4762	0.7412	0.6952	**0.8288**
	20%	0.4749	0.6120	0.5884	**0.7407**
	30%	0.4736	0.5410	0.5002	**0.6907**
	50%	0.4635	0.4040	0.3862	**0.5581**
1.00	10%	0.1679	0.6000	0.4838	**0.7307**
	20%	0.1615	0.3834	0.3184	**0.5285**
	30%	0.1612	0.2920	0.2362	**0.4347**
	50%	0.1584	0.1498	0.1404	**0.2624**

Table 9: **Certified Ratio** with different perturbation magnitude C_I and sensing model attack ratio α on PrimateNet. The sensing models are smoothed with Gaussian noise $\epsilon \sim \mathcal{N}(0, \hat{\sigma}^2 I_d)$ with different smoothing parameter σ. Here **"Hierarchy."** refers to the sensing-reasoning pipeline enriched by hierarchy knowledge only while **"Exclusive."** the exclusive knowledge only. **"Combined."** shows the sensing-reasoning pipeline enriched by both domain knowledge.

(a) $\hat{\sigma} = 0.12$

C_I	α	No knowledge	Hierarchy.	Exclusive.	Combined.
0.12	10%	0.5724	0.8714	0.7320	**0.9419**
	20%	0.5717	0.7586	0.6442	**0.8609**
	30%	0.5706	0.6850	0.5928	**0.7988**
	50%	0.5706	0.5270	0.4642	**0.6647**
0.25	10%	0.2342	0.7330	0.5482	**0.8428**
	20%	0.2320	0.5150	0.3842	**0.6657**
	30%	0.2309	0.4011	0.3422	**0.5596**
	50%	0.2268	0.2322	0.2262	**0.3824**

(b) $\hat{\sigma} = 0.25$

C_I	α	No knowledge	Hierarchy.	Exclusive.	Combined.
0.25	10%	0.5314	0.9102	0.7254	**0.9499**
	20%	0.5302	0.7910	0.6226	**0.8952**
	30%	0.5294	0.7322	0.5878	**0.8048**
	50%	0.5235	0.5670	0.4302	**0.6747**
0.50	10%	0.2024	0.7998	0.5322	**0.8489**
	20%	0.2024	0.5512	0.3490	**0.6467**
	30%	0.2010	0.4440	0.3266	**0.5541**
	50%	0.2000	0.2632	0.1734	**0.3635**

(c) $\hat{\sigma} = 0.50$

C_I	α	No knowledge	Hierarchy.	Exclusive.	Combined.
0.50	10%	0.4762	0.8924	0.7128	**0.9449**
	20%	0.4749	0.7370	0.6144	**0.8488**
	30%	0.4736	0.6552	0.5462	**0.7968**
	50%	0.4635	0.4938	0.4324	**0.6395**
1.00	10%	0.1679	0.7374	0.5204	**0.8448**
	20%	0.1615	0.4906	0.3398	**0.6336**
	30%	0.1612	0.3850	0.2926	**0.5375**
	50%	0.1584	0.1996	0.1628	**0.3318**

J Reasoning Component as Bayesian Networks

A Bayesian network (BN) is a probabilistic graphical model that represents a set of variables and their conditional dependencies with a directed acyclic graph. Let us first consider a Bayesian Network with tree structures, the probability of a random variable being 1 is given by

$$\mathbf{Pr}\left[X = 1, \{p_i\}\right] = \sum_{x_1,...x_n} P(1|x_1, ..., x_n) \prod_i p_i^{x_i} (1 - p_i)^{1-x_i}.$$

In the following subsections, we will prove a hardness result of checking robustness in general MLN and BNs and use the above definition to construct an efficient procedure to certify robustness for binary tree BNs.

J.1 Hardness of Certifying Bayesian Networks

Analogously with the above reasoning, we can also state the general hardness result for deciding the robustness of BNs:

Theorem 3 (BN hardness)**.** *Given a Bayesian network with a set of parameters $\{p_i\}$, a set of perturbation parameters $\{\epsilon_i\}$ and threshold δ, deciding whether*

$$|\mathbf{Pr}\left[X = 1; \{p_i\}\right] - \mathbf{Pr}\left[X = 1; \{p_i + \epsilon_i\}\right]| < \delta$$

is at least as hard as estimating $\mathbf{Pr}\left[X = 1; \{p_i\}\right]$ up to ε_c multiplicative error, with $\epsilon_i = O(\varepsilon_c)$.

Proof. Let $\alpha = [p_i]$, $Q(\sigma) = X$ and π_α defined by the the probability distribution of a target random variable. Since $X \in \{0, 1\}$, we have $\mathbb{E}[\sigma \sim \pi_\alpha]Q(\sigma) = \mathbf{Pr}\left[X = 1; \{p_i\}\right]$. The proof then follows analogously from Theorem 1. □

Based on the hardness analysis of the reasoning robustness, we can see that it is challenging to directly certify the robustness of the reasoning component. However, just as we can approximately certify the robustness of single ML models [25], in the next section, we will present and discuss how to approximately certify the robustness of the reasoning component, and we show that for some structures such as BN trees, the certification could even be tight.

J.2 Certifying Bayesian Networks

Apart from MLNs, we also aim to reason about the robustness for Bayesian networks with binary tree structures, and derive an efficient algorithm to provide the *tight* upper and lower bounds of reasoning robustness. Concretely, we introduce the set of perturbation $\{\epsilon_i\}$ on $\{p_i\}$ and consider the maximum resultant probability:

$$
\begin{aligned}
&\max_{\epsilon_1 \ldots \epsilon_n} \sum_{x_1, \ldots x_n} P(1|x_1, ..., x_n) \prod_i (p_i + \epsilon_i)^{x_i} (1 - p_i - \epsilon_i)^{1 - x_i} \\
&= \max_{\epsilon_1 \ldots \epsilon_n} \sum_{x_1, \ldots x_{n-1}} \left(\prod_{i<n} (p_i + \epsilon_i)^{x_i} (1 - p_i - \epsilon_i)^{1 - x_i} \right) \times \\
&\qquad \left(P(1|x_1, ..., x_{n-1}, 0)(1 - p_n - \epsilon_n) + P(1|x_1, ..., x_{n-1}, 1)(p_n + \epsilon_n) \right) \\
&= \max_{\epsilon_1 \ldots \epsilon_n} \sum_{x_1, \ldots x_{n-1}} \left(\prod_{i<n} (p_i + \epsilon_i)^{x_i} (1 - p_i - \epsilon_i)^{1 - x_i} \right) \times \\
&\qquad \left(P(1|x_1, ..., x_{n-1}, 0) + \Big(P(1|x_1, ..., x_{n-1}, 1) - P(1|x_1, ..., x_{n-1}, 0) \Big)(p_n + \epsilon_n) \right).
\end{aligned}
$$

In the above we have isolated the last variable in the expression. Without additional structure, the above optimisation over perturbation is hard as stated in Theorem 3. However, if additionally we require the Bayesian network to be binary trees, we show that the optimisation over perturbation and the checking of robustness of the model is trackable. We summarise the procedure for checking robustness of binary tree structured BNs in the following theorem with the proof.

Lemma J.1 (Binary BN Robustness)**.** *Given a Bayesian network with binary tree structure, and the set of parameters $\{p_i\}$, the probability of a variable $X = 1$,*

$$
\mathbf{Pr}\left[X = 1, \{p_i\}\right] = \sum_{x_1, x_2} P(1|x_1, x_2) \prod_i p_i^{x_i} (1 - p_i)^{1 - x_i}
$$

is δ_b-robust, where

$$
\begin{aligned}
\delta_b &= \max \left\{ \Big| \mathbf{Pr}\left[X = 1, \{p_i\}\right] - F_{max} \Big|, \Big| \mathbf{Pr}\left[X = 1, \{p_i\}\right] - F_{min} \Big| \right\}, \\
with \quad F_{max} &= \max_{y_1, y_2} A_0 + A_1(y_1 + y_2) + (A_2 - A_1) y_1 y_2, \\
F_{min} &= \min_{y_1, y_2} A_0 + A_1(y_1 + y_2) + (A_2 - A_1) y_1 y_2, \\
s.t., y_i &\in [p_i - C_i, p_i + C_i],
\end{aligned}
$$

Where $A_0 = P(1|0, 0)$, $A_1 = P(1|0, 1) - P(1|0, 0)$ and $A_2 = P(1|1, 1) - P(1|0, 1)$ are all pre-computable constants given the parameters of the Bayesian network.

Proof of Lemma J.1

Proof. We explicitly write out the probability subject to perturbation,

$$
\begin{aligned}
&\mathbf{Pr}\left[X=1,\{p_i+\epsilon_i\}\right]\\
=&\sum_{x_1,x_2} P(1|x_1,x_2)\prod_i (p_i+\epsilon_i)^{x_i}(1-p_i-\epsilon_i)^{1-x_i}\\
=&(p_1+\epsilon_1)\sum_{x_2} P(1|1,x_2)(p_2+\epsilon_2)^{x_2}(1-p_2-\epsilon_2)^{1-x_2}\\
&+(1-p_1-\epsilon_1)\sum_{x_2} P(1|0,x_2)(p_2+\epsilon_2)^{x_2}(1-p_2-\epsilon_2)^{1-x_2}\\
=&\sum_{x_2} P(1|0,x_2)(p_2+\epsilon_2)^{x_2}(1-p_2-\epsilon_2)^{1-x_2}\\
&+(p_1+\epsilon_1)\left(\sum_{x_2}\Big(P(1|1,x_2)-P(1|0,x_2)\Big)(p_2+\epsilon_2)^{x_2}(1-p_2-\epsilon_2)^{1-x_2}\right)\\
=&\sum_{x_2} P(1|0,x_2)(p_2+\epsilon_2)^{x_2}(1-p_2-\epsilon_2)^{1-x_2}\\
&+(p_1+\epsilon_1)\Big(\Big(P(1|1,1)-P(1|0,1)\Big)(p_2+\epsilon_2)+\Big(P(1|1,0)-P(1|0,0)\Big)(1-p_2-\epsilon_2)\Big)\\
=&P(1|0,0)+\Big(P(1|0,1)-P(1|0,0)\Big)(p_2+\epsilon_2)+(p_1+\epsilon_1)\times\\
&\Big(P(1|1,0)-P(1|0,0)+\Big(P(1|1,1)-P(1|0,1)-P(1|1,0)+P(1|0,0)\Big)(p_2+\epsilon_2)\Big).
\end{aligned}
$$

It follows that the robustness problem boils down to finding the maximum and minimum of $F = A_0 + A_1y_2 + A_1y_1 + (A_2 - A_1)y_1y_2$, with $y_i = p_i + \epsilon_i$. □

Specifically, in order to compute F_{max} and F_{min}, we take partial derivatives of F:

$$
\begin{aligned}
\frac{\partial F}{\partial y_1} &= A_1 + (A_2 - A_1)y_2,\\
\frac{\partial F}{\partial y_2} &= A_1 + (A_2 - A_1)y_1.
\end{aligned}
$$

Setting $\frac{\partial F}{\partial y_1} = \frac{\partial F}{\partial y_2} = 0$ leads to $y_1^* = y_2^* = \frac{A_1}{A_1 - A_2}$. In order to check if y_i^* correspond to maximum or minimum. evaluate $\frac{\partial^2 F}{\partial y_i^2} = A_2 - A_1$. We have the following scenarios:

- If $y_i^* \in [p_i - C_i, p_i + C_i]$ and $A_2 - A_1 > 0$, then y_i^* correspond to a minimum.
- If $y_i^* \in [p_i - C_i, p_i + C_i]$ and $A_2 - A_1 < 0$, then y_i^* correspond to a maximum.
- If $y_i^* \notin [p_i - C_i, p_i + C_i]$, then y_i is monotonic in the range of $[p_i - C_i, p_i + C_i]$ and the maximum or minimum are found at $p_i \pm C_i$.

Having shown the robustness of probability of one node in the Bayesian network, the robustness of the whole network can be computed recursively from the bottom to the top.