Supplemental Material

In the supplemental material we also present a plot that combines on-time and total-detection rates. We also present plots obtained when we optimize to total-accuracy rather than on-time performance. Finally, we present pseudo-code for all algorithms considered in the paper.

Bivariate KL fusion

If

$$p(x) \sim \mathcal{N}(\begin{bmatrix}\mu_1\\ \mu_2\end{bmatrix}, \begin{bmatrix}\sigma_1^2 & \rho\sigma_1\sigma_2\\ \rho\sigma_1\sigma_2 & \sigma_2^2\end{bmatrix}), \tag{21}$$

$$q(x) \sim \mathcal{N}(\begin{bmatrix}m_1\\ m_2\end{bmatrix}, \begin{bmatrix}s_1^2 & rs_1s_2\\ rs_1s_2 & s_2^2\end{bmatrix}) \tag{22}$$

Then, (9) can be written as

$$\text{KL }(P\|Q) = \log(\frac{s_1s_2\sqrt{1-r^2}}{\sigma_1\sigma_2\sqrt{1-\rho^2}}) + \frac{1}{2(1-r^2)}(\frac{(\mu_1-m_1)^2+(\sigma_1-s_1)^2}{s_1^2} + \frac{(\mu_2-m_2)^2+(\sigma_2-s_2)^2}{s_2^2} - 2r\frac{(\mu_1-m_1)(\mu_2-m_2)+\rho\sigma_1\sigma_2-rs_1s_2}{s_1s_2}) \tag{23}$$

Plots

To facilitate overall comparison We can consider a combined reliability score defined as $on\_time\_rate * total\_detection\_rate$. Fig. 5 demonstrate the score of proposed algorithms when each point has a different threshold to compare the best performance of each algorithm in different percentage of unknown. The score is computed as the maximum multiplication of on-times and total detected ratio over all possible thresholds. We can see that the algorithm performs well for distributions of either EVM data or SoftMax value for decisions.

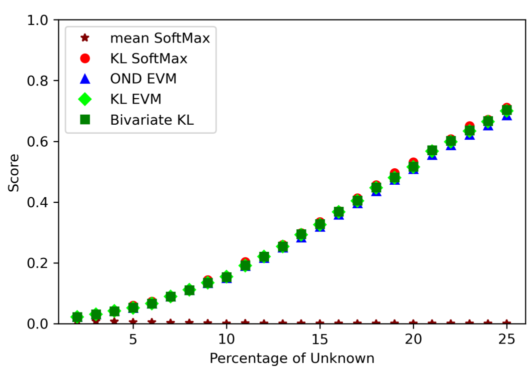
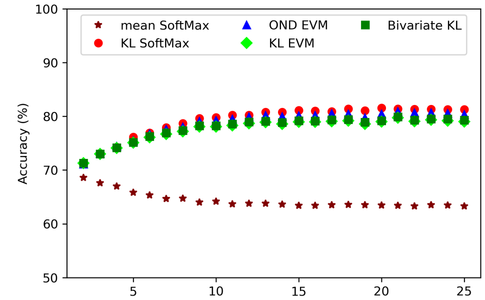
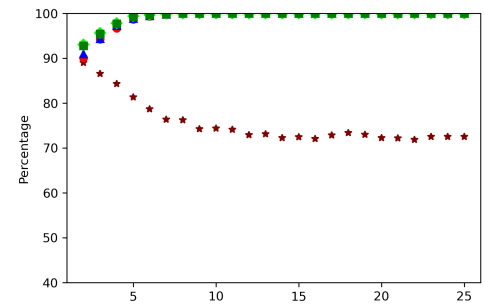


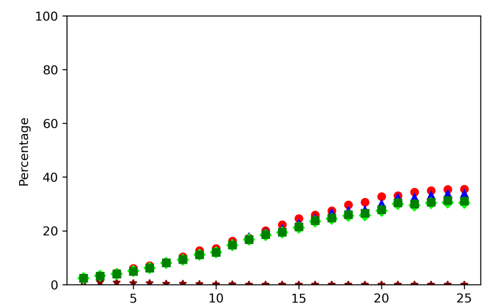
Fig 5: *Reliability Score of proposed algorithms when the best threshold for each point is selected.*



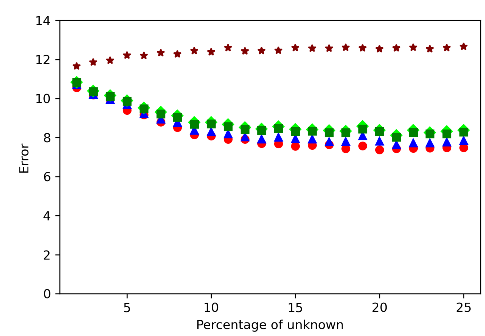
(a) *Total Accuracy*



(b) *Total Detection Percentage*

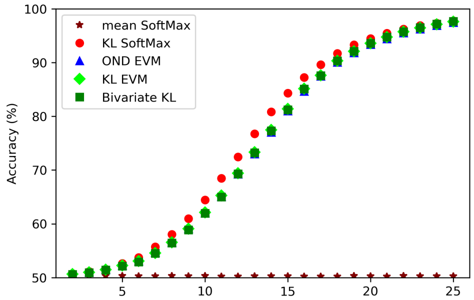


(c) *Percentage of on-time*

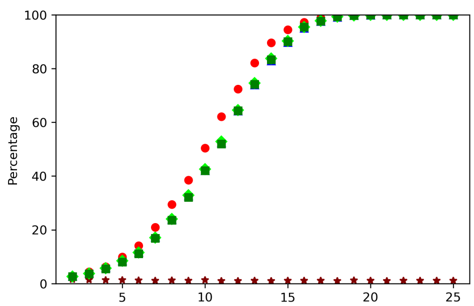


(d) *Mean Absolute Error*

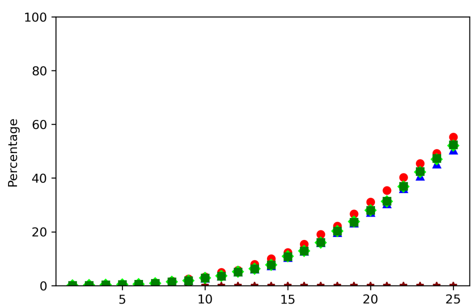
Fig 6: *Performance of proposed policies when the threshold is selected to maximize the total accuracy validation test with 2% unknown. Compare with Fig 2 in main paper.*



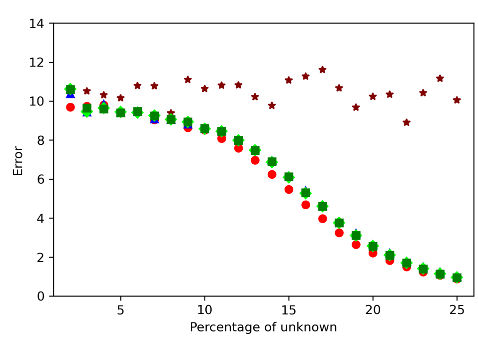
(a) *Total Accuracy*



(b) *Total Detection Percentage*

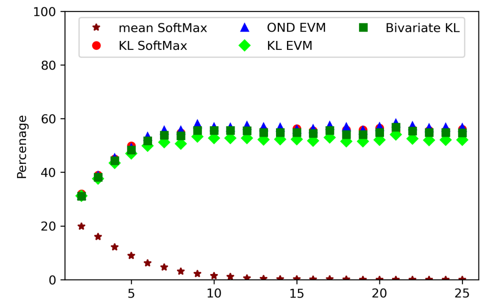


(c) *Percentage of on-time*

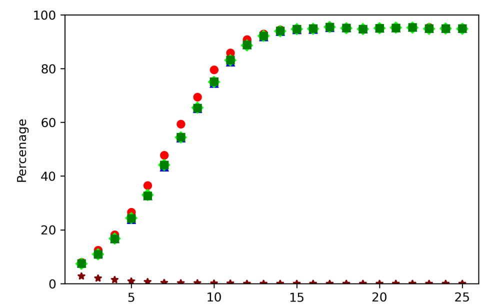


(d) *Mean Absolute Error*

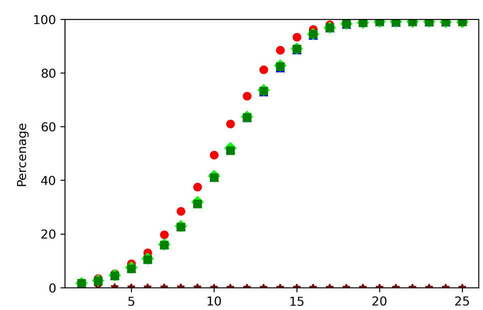
Fig 7: *Performance of proposed policies when the threshold is selected to have less than 1% early detection on validation test with 2% unknown. Compare with Fig 2 in main paper.*



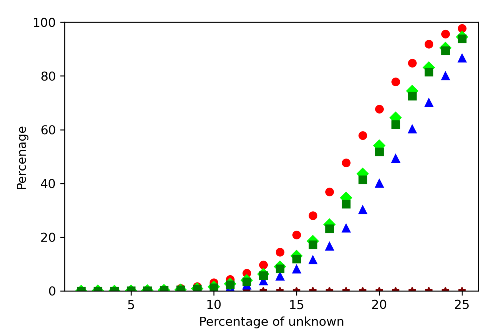
(a)



(b)



(c)



(d)

Fig 8: *True detection percentage of proposed policies when the threshold is selected to (a) maximize true detection, (b) have 5% early detection, (c) have 1% early detection, (d) have not early detection on validation test with 2% unknown.*

Algorithm 1: Simplest (baseline) automatic reliability assessment of open-set image classifiers using mean of SoftMax

Input: A batch of images, μ_{old} state from past epoch (init 1.0)
Config: M tolerance to say image classifier is unreliable
Output: Reliability, μ state

```
// N: number of images in the batch
```
x $\leftarrow$ normalize each images to range $[-1, +1]$
s $\leftarrow$ CNN(x) `// SoftMax, N x M`
p $\leftarrow$ max (s) `// Max over row, N x 1`
$\mu \leftarrow$ mean (p)
$\mu \leftarrow \min\{\mu_{\text{old}}, \mu\}$
if $\mu > M$ **then**
    return (Reliable , μ)
else
    return (Unreliable , μ)

Algorithm 2: Information theory method for automatic reliability assessment of open-set image classifiers using Kullback–Leibler divergence of SoftMax

Input: A batch of images, D_{old} state from past epoch (init 0.0)
Config: (m,s) mean and standard deviation of SoftMax of training data set, κ tolerance to say image classifier is unreliable
Output: Reliability, D state

```
// N: number of images in the batch
```
x $\leftarrow$ normalize each images to range $[-1, +1]$
s $\leftarrow$ CNN(x) `// SoftMax, N x M`
p $\leftarrow$ max (s) `// Max over row, N x 1`
$\mu \leftarrow$ mean (p)
$\sigma \leftarrow$ std (p)
D $\leftarrow$ KL(μ, σ, m , s) `// Equation 10`
D $\leftarrow \max\{D_{\text{old}}$, D $\}$
if D $< \kappa$ **then**
    return (Reliable , D)
else
    return (Unreliable , D)

Algorithm 3: Proposed OND automatic reliability assessment using EVM open-set image classifier

Input: A batch of images, ε_{old} state from past epoch (init 0)
Config: Δ lower bound limit of probability of EVM for image to be considered as known classes, $\hat{\rho}$ estimation of OOD class ratio, Ξ tolerance to say image classifier is unreliable
Output: Reliability, ε state

```
// N: number of images in the batch
// M: feature size of CNN
// L: number of known classes
```
x $\leftarrow$ normalize each images to range $[-1, +1]$
f $\leftarrow$ CNN(x) `// Deep features, N x M`
P $\leftarrow$ EVM(f) `// Equation 4, N x L`
p $\leftarrow$ max (P) `// Max over row, N x 1`
$\upsilon \leftarrow$ 1 - Δ - p `// N x 1`
$\nu \leftarrow \max\{0, \upsilon\}$ `// element wise maximum`
$\mu \leftarrow$ mean (ν)
$\zeta \leftarrow \mu - \hat{\rho}(1 - \Delta)$
$\eta \leftarrow \max\{0, \zeta\}$
$\varepsilon \leftarrow \max\{\varepsilon_{\text{old}}, \eta\}$
if $\varepsilon < \Xi$ **then**
    return (Reliable , ε)
else
    return (Unreliable , ε)

Algorithm 4: Proposed automatic reliability assessment of open-set image classifiers using Kullback–Leibler divergence of EVM

Input: A batch of images, D_{old} state from past epoch (init 0.0)
Config: (m,s) mean and standard deviation of maximum class probability of EVM on training data set, κ tolerance to say image classifier is unreliable
Output: Reliability, D state

```
// N: number of images in the batch
// M: feature size of CNN
// L: number of known classes
x ← normalize each images to range [−1, +1]
f ← CNN(x)      // Deep features, N x M
P ← EVM(f)         // Equation 4, N x L
p ← max (P)      // Max over row, N x 1
μ ← mean (p)
σ ← std (p)
D ← KL(μ, σ, m , s)            // Equation 10
D ← max{D_old , D }
if D < κ then
    return (Reliable , D)
else
    return (Unreliable , D)
```

Algorithm 5: Proposed automatic reliability assessment of open-set image classifiers using bivariate Kullback–Leibler divergence of SoftMax and EVM

Input: A batch of images, D_{old} state from past epoch (init 0.0)
Config: (m_1, m_2, s_1, s_2) mean and standard deviation of maximum SoftMax and maximum class probability of EVM on training data set, κ tolerance to say image classifier is unreliable
Output: Reliability, D state

```
// N: number of images in the batch
// M: feature size of CNN
// L: number of known classes
x ← normalize each images to range [−1, +1]
f, s ← CNN(x)       // Deep features and
 SoftMax
P ← EVM(f)          // Equation 4, N x L
p_1 ← max (s)      // Max over row, N x 1
p_2 ← max (P)      // Max over row, N x 1
μ_1 ← mean (p_1)
μ_2 ← mean (p_2)
σ_1 ← std (p_1)
σ_2 ← std (p_2)
ρ ← correlation (p_1, p_2)
// KL from equation 23
D ← KL(μ_1, μ_2, σ_1, σ_2, ρ, m_1, m_2, s_1, s_2, r)
D ← max{D_old , D }
if D < κ then
    return (Reliable , D)
else
    return (Unreliable , D)
```