Supplementary Material for DUO

Xin Yao*, Yu Zhan*, Yimin Chen†§, Fengxiao Tang*, Ming Zhao*, Enlang Li*, Yanchao Zhang‡§
*School of Computer Science and Engineering, Central South University, Changsha, Hunan, 410082, China
†Miner School of Computer & Information Sciences, University of Massachusetts Lowell, Lowell, MA 01854, USA
‡School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, AZ 85287, USA
§Corresponding authors

I. More Details of Surrogate Model Building

The architecture of the surrogate model is also a typical DNN (denoted by $\mathcal{S}(\cdot)$) with model parameters $\boldsymbol{\rho}$. Similar to [1], $\mathcal{S}(\cdot)$ consists of a stacked convolution neural network and multiple fully connected layers to extract the spatial features of video frames and the temporal features of the video and outputs flattened and continuous video features. Note that $\mathcal{S}(\cdot)$ can be in other forms such as C3D [1] and our attack is applicable to them as well. Given $\mathcal{S}(\cdot)$, we move forward to denote the projected features of a given video $\boldsymbol{v} \in \mathbb{R}^{N\times W\times H\times C}$ by $\texttt{Fea}_{\boldsymbol{\rho}}(\boldsymbol{v})$ and consequently the distance between the features of $\boldsymbol{v}$ and $\boldsymbol{v}_i$ by $D(\boldsymbol{v},\boldsymbol{v}_i) = \|\texttt{Fea}_{\boldsymbol{\rho}}(\boldsymbol{v}) - \texttt{Fea}_{\boldsymbol{\rho}}(\boldsymbol{v}_i)\|_2^2$. The smaller $D(\boldsymbol{v},\boldsymbol{v}_i)$ is, the higher similarity $\boldsymbol{v}$ and $\boldsymbol{v}_i$ share.

The training process of the surrogate model is very straightforward. Utilizing $\mathbf{T} = \{\langle \boldsymbol{v},\boldsymbol{v}_i,\boldsymbol{v}_j\rangle | \boldsymbol{v}_i,\boldsymbol{v}_j \in \mathcal{R}^n(\boldsymbol{v}), i<j \in [1,n]\}$ as the training set, we use the following loss function to update $\boldsymbol{\rho}$, similar to [2]:

$$\arg\max_{\boldsymbol{\rho}} \sum_{j>i} [D(\boldsymbol{v},\boldsymbol{v}_j) - D(\boldsymbol{v},\boldsymbol{v}_i) + \gamma]_+ \ .$$

The intuition behind is to find the optimal $\boldsymbol{\rho}$ to maximize the feature distance between the query video $\boldsymbol{v}$ and the returned video $\boldsymbol{v}_j$ while minimizing that between the query video $\boldsymbol{v}$ and the returned video $\boldsymbol{v}_i$. The parameter γ is a margin constant to prevent a negative loss value. Apparently, we can use the popular adaptive moment estimation (ADAM) [3] optimizer to obtain the updated $\boldsymbol{\rho}$. In the end, we can obtain a trained surrogate model for the victim system.

II. More details of AE generation on $\mathcal{S}(\cdot)$

We formulate AE generation on $\mathcal{S}(\cdot)$ as the following:

$$\begin{aligned} &\arg\min_{\boldsymbol{\theta},\boldsymbol{\mathcal{I}},\boldsymbol{\mathcal{F}}} \mathcal{L}(\texttt{Fea}_{\boldsymbol{\rho}}(\boldsymbol{v}_{adv}),\texttt{Fea}_{\boldsymbol{\rho}}(\boldsymbol{v}_t)) + \lambda\|\boldsymbol{\theta}\odot\boldsymbol{\mathcal{I}}\odot\boldsymbol{\mathcal{F}}\|_2^2 \\ &\quad s.t. \quad \mathbf{1}^\top\boldsymbol{\mathcal{I}} = k, \|\boldsymbol{\mathcal{F}}\|_{2,0} = n, \|\boldsymbol{\theta}\|_\infty \le \tau \ . \end{aligned} \tag{1}$$

Considering that $\boldsymbol{\theta}$ is a continuous variable while $\boldsymbol{\mathcal{I}}$ and $\boldsymbol{\mathcal{F}}$ are binary ones, the problem in Eq. (1) is mixed integer programming [4], making it difficult to find the binary pixel mask $\boldsymbol{\mathcal{I}}$ and the binary frame mask $\boldsymbol{\mathcal{F}}$. To tackle this problem, we follow [5] to loosen the binary constraint with two strong continuous constraints. For instance, the binary variable $\boldsymbol{\mathcal{I}}$ can be represented as $\boldsymbol{S}_b \cup \boldsymbol{S}_p$, where $\boldsymbol{S}_b = [0,1]^{N\times W\times H\times C}$ is a box function and $\boldsymbol{S}_p = \{\boldsymbol{\mathcal{I}} : \|\boldsymbol{\mathcal{I}} - \frac{1}{2}\|_2^2 = \frac{N\times W\times H\times C}{4}\}$ is a l_2-sphere constraint. Similarly, it is very challenging to directly obtain the binary frame mask $\boldsymbol{\mathcal{F}}$ with the $\ell_{2,0}$-norm constraint. To this end, we introduce a continuous variable $\boldsymbol{\mathcal{C}} \in \mathbb{R}^{N\times W\times H\times C}$ to replace $\boldsymbol{\mathcal{F}}$, obtain it with [6], and utilize an indicator function to select more sensitive frames. Eq. (1) is a non-convexity problem, so we cannot directly solve it (i.e., obtain the solution) but calculate the parameters $\boldsymbol{\mathcal{P}} = \{\boldsymbol{\theta},\boldsymbol{\mathcal{F}},\boldsymbol{\mathcal{I}}\}$ in an iterative fashion using the following steps until they converge.

Step 1: Given $\boldsymbol{\mathcal{F}}$ and $\boldsymbol{\mathcal{I}}$, the sub-problem of $\mathcal{P}_{\boldsymbol{\theta}}$ can be formulated as:

$$\begin{aligned} &\arg\min_{\boldsymbol{\theta}} \mathbb{L}(\mathcal{P}_{\boldsymbol{\theta}}) = \mathcal{L}(\texttt{Fea}_{\boldsymbol{\rho}}(\boldsymbol{v}_{adv}),\texttt{Fea}_{\boldsymbol{\rho}}(\boldsymbol{v}_t)) + \lambda\|\boldsymbol{\theta}\odot\boldsymbol{\mathcal{I}}\odot\boldsymbol{\mathcal{F}}\|_2^2 \\ &\quad s.t. \quad \|\boldsymbol{\theta}\|_\infty \le \tau \ . \end{aligned} \tag{2}$$

This step is similar to a typical adversarial example attack [7] and therefore can be readily solved by the gradient descent method as

$$\begin{aligned} \boldsymbol{\theta} &= \boldsymbol{\theta} - \mu_1 \cdot \frac{\mathbb{L}(\mathcal{P}_{\boldsymbol{\theta}})}{\partial\boldsymbol{\theta}} \\ &= \boldsymbol{\theta} - \mu_1 \cdot [\frac{\partial\mathcal{L}(\texttt{Fea}_{\boldsymbol{\rho}}(\boldsymbol{v}_{adv}),\texttt{Fea}_{\boldsymbol{\rho}}(\boldsymbol{v}_t))}{\partial\boldsymbol{\theta}} + 2\lambda\boldsymbol{\theta}\odot(\boldsymbol{\mathcal{I}}\odot\boldsymbol{\mathcal{F}})^2] \ , \end{aligned}$$

where μ_1 is the update step size. To satisfy the constraint of $\|\boldsymbol{\theta}\|_\infty \le \tau$, we clip each perturbation magnitude of $\boldsymbol{\theta}_{i,w,h,c}$ with the function of $\max(-\tau,\min(\boldsymbol{\theta}_{i,w,h,c},\tau))$ after each iteration.

Step 2: Given $\boldsymbol{\theta}$ and $\boldsymbol{\mathcal{F}}$, the sub-problem on $\mathcal{P}_{\boldsymbol{\mathcal{I}}}$ can be formulated as:

$$\begin{aligned} &\arg\min_{\boldsymbol{\mathcal{I}}} \mathcal{L}(\texttt{Fea}_{\boldsymbol{\rho}}(\boldsymbol{v}_{adv}),\texttt{Fea}_{\boldsymbol{\rho}}(\boldsymbol{v}_t)) + \lambda\|\boldsymbol{\theta}\odot\boldsymbol{\mathcal{I}}\odot\boldsymbol{\mathcal{F}}\|_2^2 \\ &\quad s.t. \quad \boldsymbol{\mathcal{I}} \in \{1,0\}^{N\times W\times H\times C}, \mathbf{1}^\top\boldsymbol{\mathcal{I}} = k \ . \end{aligned} \tag{3}$$

Since $\boldsymbol{\theta}$ and $\boldsymbol{\mathcal{I}}$ are a continuous variable and a binary pixel mask, respectively, the optimization problem of Eq. (3) is also mixed integer programming. We follow [5] to replace the binary constraint $\boldsymbol{\mathcal{I}}$ with $\boldsymbol{\mathcal{I}} = \boldsymbol{Y}_b \in \boldsymbol{S}_b$ and $\boldsymbol{\mathcal{I}} = \boldsymbol{Y}_p \in \boldsymbol{S}_p$. As a result, the problem of Eq. (3) can be further formulated as

$$\begin{aligned} &\arg\min_{\boldsymbol{\mathcal{I}}} \mathcal{L}(\texttt{Fea}_{\boldsymbol{\rho}}(\boldsymbol{v}_{adv}),\texttt{Fea}_{\boldsymbol{\rho}}(\boldsymbol{v}_t)) + \lambda\|\boldsymbol{\theta}\odot\boldsymbol{\mathcal{I}}\odot\boldsymbol{\mathcal{F}}\|_2^2 \\ &\quad s.t. \quad \boldsymbol{\mathcal{I}} = \boldsymbol{Y}_b, \boldsymbol{\mathcal{I}} = \boldsymbol{Y}_p, \mathbf{1}^\top\boldsymbol{\mathcal{I}} = k \ . \end{aligned} \tag{4}$$

The augmented Lagrangian function for the objective function in Eq. (4) is given by

$$\begin{aligned}\mathbb{L}(\mathcal{P}_{\boldsymbol{\mathcal{I}}}) &= \mathcal{L}(\texttt{Fea}_{\boldsymbol{\rho}}(\boldsymbol{v}_{adv}), \texttt{Fea}_{\boldsymbol{\rho}}(\boldsymbol{v}_t)) + \lambda\|\boldsymbol{\theta}\odot\boldsymbol{\mathcal{I}}\odot\boldsymbol{\mathcal{F}}\|_2^2 \\ &+ \boldsymbol{w}_1^\top(\boldsymbol{\mathcal{I}}-\boldsymbol{Y}_b) + \boldsymbol{w}_2^\top(\boldsymbol{\mathcal{I}}-\boldsymbol{Y}_p) + \boldsymbol{w}_3^\top(\mathbf{1}^\top\boldsymbol{\mathcal{I}}-k) \\ &+ \frac{\rho_1}{2}\|\boldsymbol{\mathcal{I}}-\boldsymbol{Y}_b\|_2^2 + \frac{\rho_2}{2}\|\boldsymbol{\mathcal{I}}-\boldsymbol{Y}_p\|_2^2 + \frac{\rho_3}{2}(\mathbf{1}^\top\boldsymbol{\mathcal{I}}-k)^2 \\ &+ h_1(\boldsymbol{Y}_b) + h_2(\boldsymbol{Y}_p)\ ,\end{aligned} \tag{5}$$

where $\{\boldsymbol{w}_1, \boldsymbol{w}_2\} \in \mathbb{R}^{N\times W\times H\times C}$, $\boldsymbol{w}_3 \in \mathbb{R}^N$ are dual variables, and $\{\rho_1, \rho_2, \rho_3\}$ are penalty parameters. $h_1(\boldsymbol{Y}_b) = \mathbb{I}_{\{\boldsymbol{Y}_b\in\boldsymbol{S}_b\}}$ and $h_2(\boldsymbol{Y}_p) = \mathbb{I}_{\{\boldsymbol{Y}_p\in\boldsymbol{S}_p\}}$ are "indicator functions", meaning $\mathbb{I}_{\{a\}} = 0$ if a is true and $+\infty$ otherwise. We follow the general procedure of ADMM to update the above dual variables and primal variables iteratively as follows.

Step 2.1: We independently update $\boldsymbol{Y}_b$ and $\boldsymbol{Y}_p$ in parallel as

$$\begin{cases}\boldsymbol{Y}_b^{t+1} = \underset{\boldsymbol{Y}_b\in\boldsymbol{S}_b}{\arg\min}\ \boldsymbol{w}_1^\top(\boldsymbol{\mathcal{I}}-\boldsymbol{Y}_b) + \frac{\rho_1}{2}\|\boldsymbol{\mathcal{I}}-\boldsymbol{Y}_b\|_2^2 \\ \qquad\quad = \mathcal{P}_b(\boldsymbol{\mathcal{I}} + \frac{1}{\rho_1}\boldsymbol{w}_1)\ , \\ \boldsymbol{Y}_p^{t+1} = \underset{\boldsymbol{Y}_p\in\boldsymbol{S}_p}{\arg\min}\ \boldsymbol{w}_2^\top(\boldsymbol{\mathcal{I}}-\boldsymbol{Y}_p) + \frac{\rho_2}{2}\|\boldsymbol{\mathcal{I}}-\boldsymbol{Y}_p\|_2^2 \\ \qquad\quad = \mathcal{P}_p(\boldsymbol{\mathcal{I}} + \frac{1}{\rho_2}\boldsymbol{w}_2)\ ,\end{cases} \tag{6}$$

where $\mathcal{P}_b(\boldsymbol{a}) = \min(\mathbf{1}, \max(\mathbf{0}, \boldsymbol{a}))$ with $\boldsymbol{a}\in\mathbb{R}^{N\times W\times H\times C}$ denotes the projection onto the box constraint $\boldsymbol{S}_b$, and $\mathcal{P}_p(\boldsymbol{a}) = \frac{\sqrt{N\times W\times H\times C}}{2}\frac{\bar{\boldsymbol{a}}}{\|\bar{\boldsymbol{a}}\|} + \frac{1}{2}\mathbf{1}$ with $\bar{\boldsymbol{a}} = \boldsymbol{a} - \frac{1}{2}\mathbf{1}$ and $\boldsymbol{a}\in\mathbb{R}^{N\times W\times H\times C}$ denotes the projection onto the ℓ_2-sphere constraint. Both of the constraints have been proved in [5].

Step 2.2: Given $\boldsymbol{Y}_b, \boldsymbol{Y}_p, \boldsymbol{w}_1, \boldsymbol{w}_2$, and $\boldsymbol{w}_3$, we update $\boldsymbol{\mathcal{I}}$ by gradient descent as below:

$$\begin{aligned}\boldsymbol{\mathcal{I}} &= \boldsymbol{\mathcal{I}} - \mu_2\cdot\frac{\mathbb{L}(\mathcal{P}_{\boldsymbol{\mathcal{I}}})}{\partial\boldsymbol{\mathcal{I}}} \\ &= \boldsymbol{\mathcal{I}} - \mu_2\cdot[\frac{\partial\mathcal{L}(\texttt{Fea}_{\boldsymbol{\rho}}(\boldsymbol{v}_{adv}), \texttt{Fea}_{\boldsymbol{\rho}}(\boldsymbol{v}_t))}{\partial\boldsymbol{\mathcal{I}}} + 2\lambda\boldsymbol{I}\odot(\boldsymbol{\theta}\odot\boldsymbol{\mathcal{F}})^2 \\ &+ (\boldsymbol{w}_1 + \rho_1(\boldsymbol{\mathcal{I}}-\boldsymbol{Y}_b)) + (\boldsymbol{w}_2 + \rho_2(\boldsymbol{\mathcal{I}}-\boldsymbol{Y}_p)) \\ &+ (\boldsymbol{w}_3 + \rho_3(\mathbf{1}^T\boldsymbol{\mathcal{I}} - k))]\ ,\end{aligned} \tag{7}$$

where μ_2 is the update step size.

Step 2.3: Given $\boldsymbol{\mathcal{I}}, \boldsymbol{Y}_b$, and $\boldsymbol{Y}_p$, we update the dual variables $\boldsymbol{w}_1, \boldsymbol{w}_2$, and $\boldsymbol{w}_3$ using the general ascent method as follows:

$$\begin{cases}\boldsymbol{w}_1^{t+1} = \boldsymbol{w}_1^t + \rho_1(\boldsymbol{\mathcal{I}}^{t+1} - \boldsymbol{Y}_b^{t+1}) \\ \boldsymbol{w}_2^{t+1} = \boldsymbol{w}_2^t + \rho_2(\boldsymbol{\mathcal{I}}^{t+1} - \boldsymbol{Y}_p^{t+1}) \\ \boldsymbol{w}_3^{t+1} = \boldsymbol{w}_3^t + \rho_3(\mathbf{1}^\top\boldsymbol{\mathcal{I}}^{t+1} - k)\end{cases} \tag{8}$$

Step 3: Given $\boldsymbol{\theta}$ and $\boldsymbol{\mathcal{I}}$, the sub-problem of $\mathcal{P}_{\boldsymbol{\mathcal{F}}}$ can be formulated below:

$$\begin{aligned}&\underset{\boldsymbol{\mathcal{F}}}{\arg\min}\ \mathcal{L}(\texttt{Fea}_{\boldsymbol{\rho}}(\boldsymbol{v}_{adv}), \texttt{Fea}_{\boldsymbol{\rho}}(\boldsymbol{v}_t)) + \lambda\|\boldsymbol{\theta}\odot\boldsymbol{\mathcal{I}}\odot\boldsymbol{\mathcal{F}}\|_2^2 \\ &\textit{s.t.}\quad \boldsymbol{\mathcal{F}}\in\{0,1\}^{N\times W\times H\times C}, \|\boldsymbol{\mathcal{F}}\|_{2,0} = n\ .\end{aligned} \tag{9}$$

Similarly, it is very challenging to obtain the binary frame mask $\boldsymbol{\mathcal{F}}$. Thus, we introduce a continuous variable $\boldsymbol{\mathcal{C}}\in\mathbb{R}^{N\times W\times H\times C}$ to replace $\boldsymbol{\mathcal{F}}$ and formulate Eq. (9) as

$$\begin{aligned}&\underset{\boldsymbol{\mathcal{C}}}{\arg\min}\ \mathcal{L}(\texttt{Fea}_{\boldsymbol{\rho}}(\boldsymbol{v}_{adv}), \texttt{Fea}_{\boldsymbol{\rho}}(\boldsymbol{v}_t)) + \lambda\|\boldsymbol{\theta}\odot\boldsymbol{\mathcal{I}}\odot\boldsymbol{\mathcal{C}}\|_2^2 \\ &\textit{s.t.}\quad \boldsymbol{\mathcal{C}}\in\mathbb{R}^{N\times W\times H\times C}, \|\boldsymbol{\mathcal{C}}\|_{2,0} = n\ .\end{aligned}$$

$\boldsymbol{\mathcal{C}}$ can be easily solved following [6]. Subsequently, we update $\boldsymbol{\mathcal{F}}$ as

$$\boldsymbol{\mathcal{F}}_{\pi(i)} = \begin{cases}\mathbf{1}, i\le n\ , \\ \mathbf{0}, i > n\ ,\end{cases}$$

where $\boldsymbol{\mathcal{F}}_{\pi(i)}$ is the $\pi(i)$-th row of $\boldsymbol{\mathcal{F}}$ and $\pi(\cdot)$ is an indicator function such that $\|\boldsymbol{\mathcal{C}}_{\pi(1)}\|_2 \ge \|\boldsymbol{\mathcal{C}}_{\pi(2)}\|_2 \ge \cdots \ge \|\boldsymbol{\mathcal{C}}_{\pi(N)}\|_2$. Note that $\|\boldsymbol{\mathcal{C}}_{\pi(i)}\| \ge \|\boldsymbol{\mathcal{C}}_{\pi(j)}\|$ means that the $\pi(i)$-th frame is more sensitive than the $\pi(j)$-th frame for the same amount of perturbations and thus should be selected with a higher probability.

III. More Details of $\mathbb{H}(\cdot,\cdot)$ and $\mathbf{Q}$

To evaluate the similarities of two retrieval lists, e.g., $\mathcal{R}^m(\boldsymbol{v}) = \{\boldsymbol{v}_1, \cdots, \boldsymbol{v}_i, \cdots, \boldsymbol{v}_m\}$ and $\mathcal{R}^m(\boldsymbol{v}') = \{\boldsymbol{v}'_1, \cdots, \boldsymbol{v}'_j, \cdots, \boldsymbol{v}'_m\}$, we introduce the NDCG-based function [2] $\mathbb{H}(\mathcal{R}^m(\boldsymbol{v}), \mathcal{R}^m(\boldsymbol{v}'))$. In effect, for each retrieval video $\boldsymbol{v}_i\in\mathcal{R}^m(\boldsymbol{v})$, we first calculate the prior sampling probability ω_i that $\boldsymbol{v}_i$ appears in $\mathcal{R}^m(\boldsymbol{v})$ as:

$$\omega_i = \frac{\log r_i - 1}{\sum_{i=1}^m(\log r_i - 1)}\ ,$$

where r_i represents the relevance between $\boldsymbol{v}_i$ and $\boldsymbol{v}$ and is equivalent to $m-i+1$. Note that the smaller i, the higher the similarity between $\boldsymbol{v}_i$ and $\boldsymbol{v}$. Subsequently, for each retrieval video $\boldsymbol{v}_i\in\mathcal{R}^m(\boldsymbol{v})$, we calculate the conditioned attack failure probability ϑ_i as:

$$\vartheta_i = \begin{cases}\omega_j, & \boldsymbol{v}_i\in\mathcal{R}^m(\boldsymbol{v}')\ and\ \boldsymbol{v}_i = \boldsymbol{v}'_j\ ; \\ 0, & \boldsymbol{v}_i\notin\mathcal{R}^m(\boldsymbol{v}')\ .\end{cases}$$

Therefore, the similarity function $\mathbb{H}(\mathcal{R}^m(\boldsymbol{v}), \mathcal{R}^m(\boldsymbol{v}'))$ can be defined as:

$$\sum_{i=1}^m \omega_i\cdot\vartheta_i\ .$$

Note that the more $\mathcal{R}^m(\boldsymbol{v})$ and $\mathcal{R}^m(\boldsymbol{v}')$ intersect, the larger $\mathbb{H}(\mathcal{R}^m(\boldsymbol{v}), \mathcal{R}^m(\boldsymbol{v}'))$, and vice versa.

In addition, for the κ-th iteration, the random matrix $\boldsymbol{q}^{(\kappa)}\in\mathbb{R}^{N,W,H,C}$ is selected from an orthonormal candidate vector set (e.g., the Cartesian basis) without replacement to ensure that the vector $\boldsymbol{q}^{(\kappa)}$ will not be cancelled out across different iterations.

References

[1] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in ICCV'15, Santiago, Chile, Dec. 2015.

[2] X. Li, J. Li, Y. Chen, S. Ye, Y. He, S. Wang, H. Su, and H. Xue, "Qair: Practical query-efficient black-box attacks for image retrieval," in CVPR'21, Nashville, TN, June 2021.

[3] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in ICLR'15, San Diego, CA, May 2015.

[4] B. Fan, Y. Wu, H. Li, Y. Zhang, Y. Li, F. Li, and J. Yang, "Sparse adversarial attack via perturbation factorization," in ECCV'20, Glasgow, UK, Aug. 2020.
[5] B. Wu and B. Ghanem, "ℓ_p-box ADMM: A Versatile Framework for Integer Programming," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 41, no. 7, pp. 1695–1708, Jan. 2018.
[6] W. Gu, X. Gu, J. Gu, B. Li, Z. Xiong, and W. Wang, "Dependence guided unsupervised feature selection," in AAAI'18, New Orleans, LA, Feb. 2018.
[7] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in S&P'17, San Jose, CA, Jun. 2017.