Supplementary Material: Semantic-Aware Multi-Label Adversarial Attacks

Hassan Mahmood
Northeastern University
mahmood.h@northeastern.edu

Ehsan Elhamifar
Northeastern University
e.elhamifar@northeastern.edu

1. Generalized Multi-label Attack Generation

In the main paper, we showed that adding a new constraint on non-targeted labels leads to the following optimization:

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \ & -\boldsymbol{g}_{\boldsymbol{x},\Psi_{\boldsymbol{x}}}^{\top}(\boldsymbol{P}_{\boldsymbol{x},\bar{\Psi}_{\boldsymbol{x}}}\boldsymbol{\alpha}), \\ \text{s.t.} \ & \|\boldsymbol{P}_{\boldsymbol{x},\bar{\Psi}_{\boldsymbol{x}}}\boldsymbol{\alpha}\|_p \leq \epsilon, \ \ \Psi_{\boldsymbol{x}} = h\big(\Omega_{\boldsymbol{x}}, \mathcal{G}\big). \end{aligned} \tag{1}$$

where, $\boldsymbol{g}_{\boldsymbol{x},\Psi_{\boldsymbol{x}}}$ is the gradient of the target loss w.r.t. x. For $p=\infty$, we can solve (1) to get the closed form solution by setting:

$$\begin{aligned} \boldsymbol{P}_{\boldsymbol{x},\bar{\Psi}_{\boldsymbol{x}}}\boldsymbol{\alpha} \ &= \ \nu, \\ \nu \ &= \ sgn\left(\boldsymbol{g}_{\boldsymbol{x},\Psi_{\boldsymbol{x}}}\right), \end{aligned} \tag{2}$$

However, since ν might not lie in the span of $\boldsymbol{P}_{\boldsymbol{x},\bar{\Psi}_{\boldsymbol{x}}}$, we can get its closest projection onto $\boldsymbol{P}_{\boldsymbol{x},\bar{\Psi}_{\boldsymbol{x}}}$ using $(\boldsymbol{P}_{\boldsymbol{x},\bar{\Psi}_{\boldsymbol{x}}}\nu)$. Alternatively, we can use (2) to derive the projection:

$$\begin{aligned} \boldsymbol{P}_{\boldsymbol{x},\bar{\Psi}_{\boldsymbol{x}}}^2\boldsymbol{\alpha} &= \boldsymbol{P}_{\boldsymbol{x},\bar{\Psi}_{\boldsymbol{x}}}\nu, \\ \boldsymbol{P}_{\boldsymbol{x},\bar{\Psi}_{\boldsymbol{x}}}\boldsymbol{\alpha} &= \boldsymbol{P}_{\boldsymbol{x},\bar{\Psi}_{\boldsymbol{x}}}\nu, \end{aligned} \tag{3}$$

where $\boldsymbol{P}_{\boldsymbol{x},\bar{\Psi}_{\boldsymbol{x}}}$ is the projection matrix. From the main paper, $\boldsymbol{e} = \boldsymbol{P}_{\boldsymbol{x},\bar{\Psi}_{\boldsymbol{x}}}\boldsymbol{\alpha}$. Therefore, the update step for $\boldsymbol{e}$ for a fixed ϵ at each iteration is given as:

$$\boldsymbol{e} \ = \ \epsilon \, \frac{\boldsymbol{P}_{\boldsymbol{x},\bar{\Psi}_{\boldsymbol{x}}}\,\nu}{\|\boldsymbol{P}_{\boldsymbol{x},\bar{\Psi}_{\boldsymbol{x}}}\,\nu\|_{\infty}}, \tag{4}$$

As shown in Fig. 8 in the main paper, non-targeted loss gradients $\boldsymbol{g}_{\boldsymbol{x},\bar{\Psi}_{\boldsymbol{x}}}$ are negatively correlated with targeted loss gradients $\boldsymbol{g}_{\boldsymbol{x},\Psi_{\boldsymbol{x}}}$. Eq. (4) addresses the negative correlation by finding the optimal perturbation in the orthogonal complement of non-targeted gradients. For the infrequent scenario when the two gradients $(\boldsymbol{g}_{\boldsymbol{x},\Psi_{\boldsymbol{x}}}, \boldsymbol{g}_{\boldsymbol{x},\bar{\Psi}_{\boldsymbol{x}}})$ are positively correlated (quantified by a value τ), we solve the following:

$$\begin{aligned} \min_{\boldsymbol{e}} & -\mathcal{L}_{bce}(\boldsymbol{x}+\boldsymbol{e}, \Psi_{\boldsymbol{x}}) + \mathcal{L}_{bce}(\boldsymbol{x}+\boldsymbol{e}, \bar{\Psi}_{\boldsymbol{x}}), \\ \text{s.t.} \ & \|\boldsymbol{e}\|_p \leq \epsilon, \end{aligned} \tag{5}$$

We can linearize (5) around x for small $\boldsymbol{e}$ as:

$$\begin{aligned} \min_{\boldsymbol{e}} \ & \boldsymbol{e}^{\top}(-\boldsymbol{g}_{\boldsymbol{x},\Psi_{\boldsymbol{x}}} + \boldsymbol{g}_{\boldsymbol{x},\bar{\Psi}_{\boldsymbol{x}}}) \\ \text{s.t.} \ & \|\boldsymbol{e}\|_p \leq \epsilon \\ \text{where,} \ & \boldsymbol{g}_{\boldsymbol{x},\Psi_{\boldsymbol{x}}} \triangleq \frac{\partial \mathcal{L}_{bce}(\boldsymbol{x}, \Psi_{\boldsymbol{x}})}{\partial \boldsymbol{x}}, \\ & \boldsymbol{g}_{\boldsymbol{x},\bar{\Psi}_{\boldsymbol{x}}} \triangleq \frac{\partial \mathcal{L}_{bce}(\boldsymbol{x}, \bar{\Psi}_{\boldsymbol{x}})}{\partial \boldsymbol{x}}. \end{aligned} \tag{6}$$

As we mentioned in the main paper, the gradient of non-targeted classes $(\boldsymbol{g}_{\boldsymbol{x},\bar{\Psi}_{\boldsymbol{x}}})$ can dominate the optimization for large number of labels in (6). To avoid that, we normalize the gradients to the same scale:

$$\begin{aligned} \min_{\boldsymbol{e}} \ & \boldsymbol{e}^{\top}\left(-\frac{\boldsymbol{g}_{\boldsymbol{x},\Psi_{\boldsymbol{x}}}}{\|\boldsymbol{g}_{\boldsymbol{x},\Psi_{\boldsymbol{x}}}\|_2} + \frac{\boldsymbol{g}_{\boldsymbol{x},\bar{\Psi}_{\boldsymbol{x}}}}{\|\boldsymbol{g}_{\boldsymbol{x},\bar{\Psi}_{\boldsymbol{x}}}\|_2}\right) \\ \text{s.t.} \ & \|\boldsymbol{e}\|_p \leq \epsilon \end{aligned} \tag{7}$$

Therefore, we have the following perturbation update step based on the gradients correlation:

$$e \ = \ \epsilon \begin{cases} \dfrac{\boldsymbol{P}_{\boldsymbol{x},\bar{\Psi}_{\boldsymbol{x}}}\,\nu}{\|\boldsymbol{P}_{\boldsymbol{x},\bar{\Psi}_{\boldsymbol{x}}}\,\nu\|_{\infty}}, & \text{if } \dfrac{\boldsymbol{g}_{\boldsymbol{x},\bar{\Psi}_{\boldsymbol{x}}}^{\top}\boldsymbol{g}_{\boldsymbol{x},\Psi_{\boldsymbol{x}}}}{\|\boldsymbol{g}_{\boldsymbol{x},\bar{\Psi}_{\boldsymbol{x}}}\|_2\|\boldsymbol{g}_{\boldsymbol{x},\Psi_{\boldsymbol{x}}}\|_2} \leq \tau \\ -\,sgn\left(-\dfrac{\boldsymbol{g}_{\boldsymbol{x},\Psi_{\boldsymbol{x}}}}{\|\boldsymbol{g}_{\boldsymbol{x},\Psi_{\boldsymbol{x}}}\|_2} + \dfrac{\boldsymbol{g}_{\boldsymbol{x},\bar{\Psi}_{\boldsymbol{x}}}}{\|\boldsymbol{g}_{\boldsymbol{x},\bar{\Psi}_{\boldsymbol{x}}}\|_2}\right), & otherwise \end{cases} \tag{8}$$

We follow Auto-PGD[2] to iteratively solve (8). Auto-PGD is an improved version of PGD-attack [3, 4] that overcomes the failures due to fixed step size and PGD objective. Mainly, it adds a momentum term to the original algorithm and modifies the step size at run-time based on the improvement (objective value) in attack's success in previous iterations. Hence, the update step is given as:

$$z^{(k+1)} = \Pi_{\epsilon}\left(x^{(k)} + \eta^{(k)}\boldsymbol{e}^{(k)}\right) \tag{9}$$

$$\begin{aligned} x^{(k+1)} = \Pi_{\epsilon}\Big(x^{(k)} &+ \alpha \cdot \left(z^{(k+1)} - x^{(k)}\right) \\ &+ (1-\alpha)\cdot\left(x^{(k)} - x^{(k-1)}\right)\Big) \end{aligned} \tag{10}$$

Where, Π_ϵ is the projection function onto the ϵ norm ball, $\eta^{(k)}$ is the step size at k^{th} iteration, and α is the momentum term to regulate the effect of previous step on current step. In our experiments, we set $\eta = 0.006$, $\alpha = 0.75$. We perform ablation experiments for τ in Section 3.

2. Three-Node Attacks

In the main paper, we showed results of one-node and two-node attacks. In this section, we show the results of attacking three classes on PASCAL-VOC and NUS-WIDE. Tab. 1 shows the results of attacking three classes using different models. We can see that three-node attacks have similar trend as two-node attacks, presented in the main paper. Also, the naive fooling rates on NUS-WIDE are generally lower than PASCAL-VOC as the former dataset has more number of labels. We further make the following observations:

1. Given large enough perturbation budget, the MLA-U, MLA-C, and GMLA can achieve high naive fooling rates, yet once we filter out the semantically inconsistent predictions, MLA-U and MLA-C perform significantly worse.
2. Note that MLA-C and GMLA have the additional non-target classes objective, so the space of perturbations is smaller than that of MLA-U, which is why the naive fooling rates of MLA-C and GMLA are lower than MLA-U.
3. All attacks generate imperceptible perturbations as the structural similarity metric (SSIM) value is close to 1.
4. MLA-LP achieves lowest NT_R values across datasets but it consistently performs worse (using fooling rates) as we increase the target set size and as we move to larger datasets. This shows that overall, generating attacks using LP-based solvers is not effective.
5. MLA-U affects the most percentage of non-targeted labels. This is because MLA-U does not impose any constraint on the non-targeted labels.
6. GMLA is the most successful at generating semantically consistent attacks (FR_S) while achieving low non-targeted flip rate (NT_R).

3. Ablation Study

In the main paper, we showed the effectiveness of using knowledge graphs for semantically consistent attacks by comparing MLA attacks with GMLA attack. In this section, we perform ablation experiments to show the effectiveness of GMLA optimization proposed in Section 4.1 in the main paper. We use OpenImages dataset and pretrained TResNet model from [6]. We study two GMLA variants, both using the knowledge graph to attack semantically related labels without the proposed optimization. We define the following variants of GMLA:

1. GMLA_α: Find the perturbation to attack semantically related classes without constraining the non-targeted classes:

$$\text{GMLA}_\alpha: \quad \min_{\boldsymbol{e}} -\mathcal{L}_{bce}(\boldsymbol{x}+\boldsymbol{e}, \Psi_{\boldsymbol{x}}), \quad \text{s.t.} \quad \|\boldsymbol{e}\|_p \leq \epsilon, \;\; \Psi_{\boldsymbol{x}} = h\big(\Omega_{\boldsymbol{x}}, \mathcal{G}\big), \tag{11}$$

2. GMLA_β: Find the perturbation to attack semantically related classes while fixing non-targeted classes i.e., we solve the following:

$$\text{GMLA}_\beta: \quad \min_{\boldsymbol{e}} -\mathcal{L}_{bce}(\boldsymbol{x}+\boldsymbol{e}, \Psi_{\boldsymbol{x}}) + \mathcal{L}_{bce}(\boldsymbol{x}+\boldsymbol{e}, \bar{\Psi}_{\boldsymbol{x}}), \quad \text{s.t.} \quad \|\boldsymbol{e}\|_p \leq \epsilon, \;\; \Psi_{\boldsymbol{x}} = h\big(\Omega_{\boldsymbol{x}}, \mathcal{G}\big), \tag{12}$$

We perform experiments for different sizes of the target set. The results are shown in Table 2. We make the following observations:

1. For each target set size, GMLA_α achieves the highest naive fooling rate. Since it does not optimize the non-targeted labels, optimizing for target classes is easier. Note that GMLA_α is different from MLA-U because GMLA_α uses knowledge graph to attack semantically related labels. This is the reason that it achieves almost similar semantic-based fooling rate(FR_S) as GMLA for all target set sizes. However, the drawback is the high non-targeted flip rate (NT_R). This shows the significance of fixing non-targeted labels.
2. GMLA_β provides empirical evidence of our claim that fixing non-targeted classes leads to negatively correlated targeted and non-targeted gradients and hence, suboptimal perturbations. GMLA_β starts off at $|\Omega| = 1$ with comparable naive and semantic-based fooling rate but quickly declines as the number of target labels increase. Unlike GMLA_α, GMLA_β fixes non-targeted labels but faces the issue of opposite gradients. This leads to suboptimal results and low fooling rates. Also, note that the low NT_R of GMLA_β with increasing target set size is because of low success rate and the smaller number of images to evaluate the metric on.
3. Note that GMLA achieves highest naive and semantic-based fooling rates while maintaining low NT_R for all target set sizes.

Now, we investigate the effect of different values of τ on the performance of GMLA attack. For this experiment, we fix the target set size $|\Omega| = 5$ and evaluate GMLA using pretrained T-ResNet model from ASL[6] on OpenImages. From Fig. 1, we make the following observations:

1. As the value of τ decreases, both the semantic-based fooling rate (FR_S) and the non-target flip rate (NT_R) decreases simultaneously.
2. Note that the percentage of affected non-targeted labels is very small for all thresholds. This shows the effec-

Model	Attack	PASCAL-VOC ↑FR_N	PASCAL-VOC ↑FR_S	PASCAL-VOC ↓NT_R	PASCAL-VOC ↑SSIM	NUS-WIDE ↑FR_N	NUS-WIDE ↑FR_S	NUS-WIDE ↓NT_R	NUS-WIDE ↑SSIM
ML-GCN[1]	**MLA-U**[8]	100.0 ± 0.0	71.4 ± 18.1	4.3 ± 2.8	0.98	99.8 ± 0.7	31.2 ± 24.8	1.8 ± 1.3	0.97
	MLA-C[8]	99.8 ± 1.0	62.8 ± 21.8	2.7 ± 1.8	0.97	87.7 ± 20.5	19.2 ± 24.6	0.5 ± 0.4	0.97
	MLA-LP[7]	15.4 ± 11.5	4.70 ± 9.6	1.5 ± 0.8	0.98	13.5 ± 12.4	1.30 ± 4.7	0.5 ± 0.2	0.98
	GMLA (Ours)	99.9 ± 0.7	97.9 ± 2.1	2.1 ± 1.0	0.98	94.3 ± 13.7	87.8 ± 14.7	0.5 ± 0.5	0.97
ASL[6]	**MLA-U**[8]	100.0 ± 0.0	50.6 ± 22.4	4.7 ± 1.9	0.97	100.0 ± 0.0	40.9 ± 25.3	2.4 ± 1.4	0.97
	MLA-C[8]	100.0 ± 0.0	36.2 ± 24.1	2.0 ± 1.0	0.97	100.0 ± 0.0	26.3 ± 24.0	0.8 ± 0.5	0.97
	MLA-LP[7]	11.2 ± 10.9	1.70 ± 3.9	1.2 ± 0.7	0.98	12.2 ± 8.9	1.10 ± 4.1	0.4 ± 0.2	0.98
	GMLA (Ours)	98.4 ± 2.7	98.4 ± 2.7	1.8 ± 0.5	0.97	98.1 ± 4.9	94.8 ± 5.8	0.6 ± 0.4	0.98
ML-Decoder[7]	**MLA-U**[8]	99.6 ± 0.8	64.0 ± 17.1	5.4 ± 2.0	0.97	98.1 ± 1.4	53.2 ± 30.6	5.3 ± 2.5	0.98
	MLA-C[8]	97.8 ± 5.5	45.9 ± 23.6	2.3 ± 1.1	0.97	63.0 ± 30.6	25.5 ± 26.5	1.0 ± 0.6	0.96
	MLA-LP[7]	13.7 ± 11.5	1.60 ± 3.2	0.5 ± 0.4	0.98	5.20 ± 8.6	0.20 ± 1.2	0.1 ± 0.1	0.98
	GMLA (Ours)	98.4 ± 13.4	93.9 ± 9.7	1.9 ± 0.7	0.97	92.1 ± 17.6	81.5 ± 19.1	1.0 ± 1.1	0.97

Table 1. Experimental evaluation of the four attack methods on three models for $\epsilon = 0.01$. The values represent the mean and standard deviations computed using the attack performance across all the combinations of target classes of size $|\Omega| = 3$.

	$\|\Omega\| = 1$			$\|\Omega\| = 2$			$\|\Omega\| = 3$			$\|\Omega\| = 4$			$\|\Omega\| = 5$		
	$GMLA_\alpha$	**$GMLA_\beta$**	**GMLA**	**$GMLA_\alpha$**	**$GMLA_\beta$**	**GMLA**	**$GMLA_\alpha$**	**$GMLA_\beta$**	**GMLA**	**$GMLA_\alpha$**	**$GMLA_\beta$**	**GMLA**	**$GMLA_\alpha$**	**$GMLA_\beta$**	**GMLA**
FR_N	$\mathbf{100.0 \pm 0.0}$	91.4 ± 8.8	$\mathbf{100.0 \pm 0.0}$	99.0 ± 0.2	52.7 ± 19.3	99.0 ± 0.4	100.0 ± 0.0	22.5 ± 19.6	99.0 ± 1.1	98.0 ± 0.3	15.1 ± 11.5	98.2 ± 3.5	98.1 ± 0.1	10.1 ± 13.2	$\mathbf{96.5 \pm 9.1}$
FR_S	$\mathbf{99.7 \pm 2.1}$	90.9 ± 9.4	99.6 ± 7.9	77.9 ± 13.1	46.6 ± 18.2	93.8 ± 11.2	87.6 ± 11.9	19.5 ± 17.8	87.6 ± 20.8	80.1 ± 14.0	13.1 ± 9.1	80.1 ± 23.5	82.1 ± 9.8	8.8 ± 12.7	$\mathbf{82.0 \pm 25.9}$
NT_R	0.53 ± 0.17	0.3 ± 0.1	$\mathbf{0.32 \pm 0.14}$	0.65 ± 0.16	0.21 ± 0.08	0.16 ± 0.12	0.77 ± 0.12	0.05 ± 0.05	0.21 ± 0.13	0.78 ± 0.10	0.03 ± 0.02	0.11 ± 0.07	0.88 ± 0.09	0.02 ± 0.03	$\mathbf{0.06 \pm 0.04}$

Table 2. Ablation Study: Performance of the proposed optimization of GMLA at $\epsilon = 0.05$ for Asymmetric Loss [6] model on OpenImages.

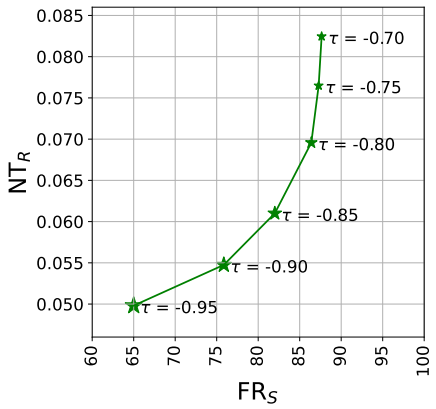

Figure 1. Ablation Study: Investigating the effect of τ on GMLA attack performance using OpenImages.

.

tiveness of our proposed optimization to keep the non-targeted labels fixed.

3. The fooling rate (FR_S) plateues as we increase the threshold from $\tau = -0.8$, which shows that most of the targeted and non-targeted gradients are negatively correlated i.e., they have dot product less than or equal to -0.8. This is also shown in Fig. 8 in the main paper.
4. While increasing the value of τ increases the fooling rate, it also amplifies the non-targeted flip rate (NT_R). Since we want to achieve high fooling rate and low NT_R, we find a trade-off and set $\tau = -0.85$ for our experiments in OpenImages.

4. Multi-Label DeepFool Attack (ML-DP)

To evaluate the proposed attack, we use ML-DP, the greedy algorithm proposed in [8], to compute perturbations. We follow the original formulation to compute the perturbations and report the success rate for different perturbation norms (RMSD).

$$RMSD = \sqrt{\frac{\|x - x'\|_2^2}{N}} \tag{13}$$

where, x is the original image and x' is the adversarial image. For each target image, we run the algorithm for 100 iterations and select the minimum perturbation that achieves the target label. Since the algorithm has no constraint on the norm of the perturbation, it can compute arbitrarily large perturbations. To compare the performance of our proposed approach, we restrict the perturbations to maximum l_2 norm of 5000. We group the norms in bins and plot the frequency of adversarial examples within each bin for all attacks. We show the results in Figure 2. It is important to note that this algorithm computes the perceptible perturbations (large norm) as compared to the ones com-

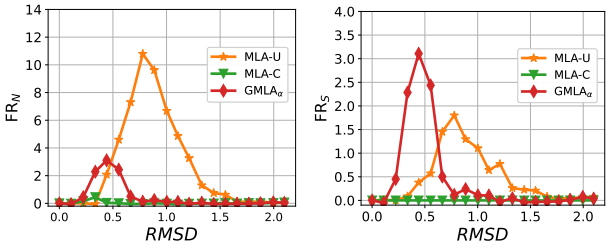

Figure 2. Performance of ML-DP[8] based multi-label attacks using RMSD on PASCAL-VOC

.

puted using our approach, which are imperceptible with a maximum norm of $l_\infty = 0.01$.

4.1. Results

We perform experiments on PASCAL-VOC because ML-DP requires computing gradient w.r.t. each class and does not scale well to very large datasets. Moreover, the algorithm does not converge for most of the images (the convergence issue was also mentioned in the original paper). Figure 2 shows results of ML-DP on the attacks explained in the main paper. To follow the same formulation proposed in [8], we implement our proposed GMLA attack without orthogonal projection (same as GMLA$_\alpha$ from Eq. (11)) in Figure 2. In our experiments, ML-DP algorithm did not converge for most of the images, mainly for MLA-C and GMLA$_\alpha$. This is because these attacks put constraints on all classes and the algorithm cannot find the optimal perturbation satisfying all constraints. In some success cases, it finds the perturbation with a very large norm, which we filter out. We make the following observations:

1. MLA-U attack achieves the highest naive fooling rate FR$_N$ as it targets to modify only a small subset of classes (small number of constraints). However, the fooling rate drops significantly once we evaluate the semantic consistency (using FR$_S$).
2. GMLA achieves success within smaller perturbation norm as compared to other attacks but has lower naive fooling rate than MLA-U. This is because adding a large number of label constraints makes the optimization difficult and usually requires a larger perturbation norm to be successful. However, it achieves the highest semantic-based fooling rate which shows the significance of using knowledge-graph (notice the change of y-axis scale in Figure 2).
3. MLA-C has the lowest naive and semantic-based fooling rates and it does not converge for more than 90% of images.

5. Knowledge Graph Extraction

As already discussed in the main paper, we use Wordnet [5] to build knowledge graphs for PASCAL-VOC and NUS-WIDE. Given a set of labels $\mathcal{C}$, we associate a synset to each label and extract abstract classes using its hypernym relationship in *Noun* synsets from Wordnet. Once we extract all related classes, we refine the hierarchies by removing nodes(labels) which only have one child and one parent node. Furthermore, we build a tree by using the maximum WUP similarity score between a child node and multiple parent nodes to select a single parent node. The extracted hiearchy for PASCAL-VOC is shown in Figure 3. Note that the attack is applicable to any Directed Acyclic Graph. For OpenImages, we use the officially provided semantic hierarchy for 600 boxable classes.

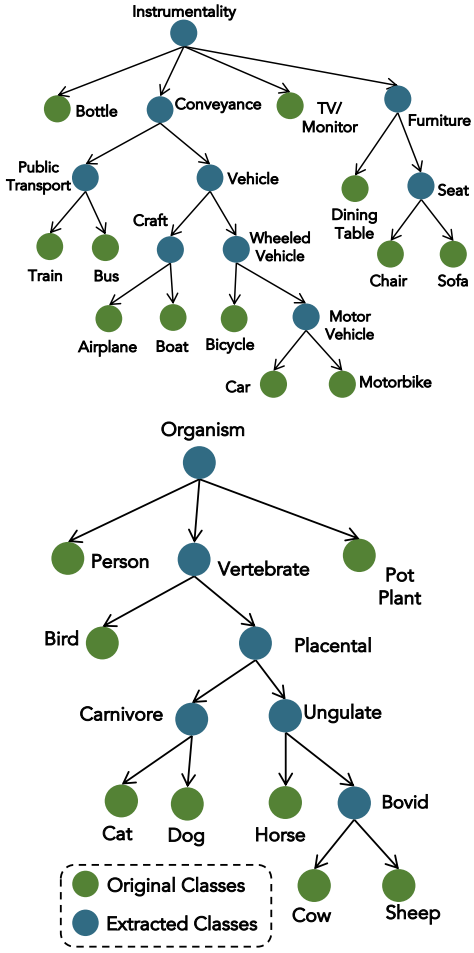

Figure 3. Extracted knowledge graph of Pascal-VOC dataset. We extract abstract classes from Wordnet for each of the original 20 classes using hypernym-hyponym relationships. The refined hierarchy is shown in this figure. Green nodes show the labels from PASCAL-VOC and blue nodes show the abstract classes extracted from Wordnet.

References

[1] Z. M. Chen, X. S. Wei, P. Wang, and Y. Guo. Multi-label image recognition with graph convolutional networks. *IEEE Conference on Computer Vision and Pattern Recognition*, abs/1904.03582, 2019.

[2] F. Croce and M. Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. *ArXiv*, 2020.

[3] A. Kurakin, I. Goodfellow, and S. Bengio. Adversarial examples in the physical world. *ArXiv preprint, arXiv:1607.02533*, 2016.

[4] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. *International Conference on Learning Representations*, 2018.

[5] G. A. Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11), 1995.

[6] Tal Ridnik, Emanuel Ben-Baruch, Nadav Zamir, Asaf Noy, Itamar Friedman, Matan Protter, and Lihi Zelnik-Manor. Asymmetric loss for multi-label classification. 2021.

[7] Tal Ridnik, Gilad Sharir, Avi Ben-Cohen, Emanuel Ben-Baruch, and Asaf Noy. Ml-decoder: Scalable and versatile classification head. 2023.

[8] Q. Song, H. Jin, X. Huang, and X. Hu. Multi-label adversarial perturbations. *IEEE International Conference on Data Mining*, 2018.