Updating Object Detection Models with Probabilistic Programming

Martijn Oldenhof [1] **Adam Arany** [1] **Yves Moreau** [1] **Edward De Brouwer** [1]

Abstract

Training object detection models require detailed image annotations, such as the bounding boxes and labels of all objects present in the image. However, such rich annotations are often not available in practice and only high-level image information is provided. Nevertheless, knowledge built on existing richly annotated domains should intuitively be transferable to the weakly supervised domains. In this work, we propose ProbKT, a fine-tuning framework based on probabilistic reasoning to update object detection models with weak supervision, by transferring knowledge from a pre-trained architecture. Unlike previous transfer learning methods, our probabilistic reasoning approach allows for arbitrary types of supervision on the target domain, such as complex logic statements. We empirically show on different datasets and with different types of supervision that fine-tuning with ProbKT leads to significant improvement on target domain and better generalization compared to existing baselines.

1. Introduction

Object detection is a fundamental ability of numerous high-level machine learning pipelines such as autonomous driving (Behl et al., 2017), augmented reality (Tomei et al., 2019) or image retrieval (Hameed et al., 2021). However, training state-of-the-art object detection models generally requires detailed images annotations such as the box-coordinates locations and the labels of each object in the image. If several large benchmark datasets with detailed annotations are indeed available (Lin et al., 2014; Everingham et al., 2010), providing such detailed annotation on new specific dataset comes with a significant cost that is often not affordable for many applications. More frequently, datasets come with only limited annotation also referred to as *weak supervision*, such as the number of each type of objects in the image. A popular approach is to update a pre-trained model to a new target domain with weak annotations, also known as weakly supervised transfer learning (Deselaers et al., 2012; Zhong et al., 2020; Uijlings et al., 2018).

A recurring assumption across weakly supervised knowledge transfer approaches lies in the type of limited annotation that is available on the target domain. Indeed, all works assume the class counts, *i.e.* the counts of different classes of objects, are available for each image. This assumption, while well founded in many cases, results in hard-coded inductive biases in the proposed architectures that prevents other types of weak supervision. Examples of other types of weak-supervision include a mere Boolean indicator of the presence of a particular class (*e.g.* this image contains at least 1 dog), a mixture of class counts and Boolean class indicators (*e.g.* this image contains 2 cars and at least one bicycle), or any type of logical combination thereof.

In this work, we propose to generalize knowledge transfer in objects detection to arbitrary types of weak supervision by using neural probabilistic reasoning (Manhaeve et al., 2018). This paradigm allows to connect probabilistic outputs of neural networks with logical rules and to infer the resulting probability of particular queries. Based on the output probabilities of a pre-trained object detection model (*e.g.* that would provide probabilities of specific classes such as dogs, cat, cars, ...), one can use probabilistic reasoning to infer the probability of a query (*i.e.* "the image contains at least 2 animals"). Remarkably, one can differentiate through this process and thus update the neural network producing these probabilities. Our approach leverages this type knowledge representation to allow for arbitrary type of weak supervision on the new image domain and it thus, to the best of our knowledge, the first to allow for such versatility in terms of the available information on the new domain.

Key contributions: (1) We propose a novel transfer knowledge framework for objects detection relying on probabilistic programming that uniquely allows to use arbitrary types of supervision on the target domain. (2) We make our approach amenable to different levels of computational capabilities by proposing different approximations. (3) We provide an extensive experimental setup to study the capabilities of our framework for knowledge transfer and out-of-distribution generalization.

[1]ESAT-STADIUS,KU Leuven, 3001 Leuven, Belgium. Correspondence to: Martijn Oldenhof <martijn.oldenhof@esat.kuleuven.be>.

2. Methodology

2.1. Problem Statement

Using a object detection model trained on a richly annotated source domain, we aim at improving its performance on a poorly annotated target domain. Let $\mathcal{D}_s = \{(I_s^i, b_s^i, y_s^i) : i = 1, ..., N_s)\}$ be a dataset issued from the source domain and consisting of N_s images I_s along with their annotations. We write $b_s^i \in \mathbb{R}^{n_i \times 4}$ for the box coordinates of objects in image I_s^i, $y_s^i \in \{1, .., K\}$ for the objects class labels and n_i is the number of objects present in image I_s^i. The target dataset $\mathcal{D}_t = \{(I_t^i, q_t^i) : i = 1, ..., N_t)\}$ contains N_t image from the target domain along with image-level annotations q_t^i. These are logical statements about the content of the image. Examples include the counts of different classes in each image (*i.e.* the classical assumption in weakly supervised object detection) or a complex combination of counts of objects attributes (*e.g.* "two red objects, 1 green object, 1 car and 2 bikes"). Logical statements q_t^i can include classes not already present in the source domain.

2.2. Background

2.2.1. Object Detection

Object detection aims at predicting the location and labels of objects in images via a parametric function $f_\theta : \mathcal{I} \rightarrow \{\mathcal{B} \times \mathbb{R}^K\}^{\mathbb{Z}}$ with $f_\theta(I) = \{(\hat{b}, \hat{p}_y)\}^{\hat{n}} = \{(\hat{b}_i, \hat{p}_{y,i}) : i = 1, ..., \hat{b}\}$ such that the distance between predicted and true boxes and labels, $d(\{(\hat{b}, \hat{p}_y)\}^{\hat{n}}, \{(b, y)\}^n)$, is minimum. Objects detection architecture output box features proposals $\{h_i : i = 1, ..., \hat{n}\}$ conditioned on which they predict the probability vector of class labels $\hat{p}_{y,i} = g_p(h_i)$ and the box locations predictions $\hat{b}_i = g_b(h_i)$ using shared parametric functions $g_p(\cdot)$ and $g_b(\cdot)$. For an object n, $\hat{p}_{y,n}^k$ is the predicted probability of the object belonging to class k.

2.2.2. Probabilistic Reasoning

Probabilistic reasoning is a type of knowledge representation relying on probabilities and allowing to encode uncertainty in knowledge. A probabilistic program $\mathcal{P}$ is represented as a set of N probabilistic facts $U = \{U_1, ..., U_N\}$ and M rules $F = \{f_1, ...f_M\}$ connecting them. A simple example is "Alice and Bob will each pass their exam with probability 0.5" and "if both Alice and Bob pass their exam, they will host a party". The first statement is a probabilistic fact and the hypothetical syllogism about the party is a rule. Combining probabilistic facts and rules, one can then construct complex probabilistic knowledge representation.

Probabilistic logic programming allows to perform inference on a knowledge graphs $\mathcal{P}$, *i.e.* computing the probability of a query such as probability that Alice and Bob will host a party. Under the hood, this proceeds by cleverly adding up the probabilities of occurrence of the different *worlds* $w = \{u_1, ..., u_N\}$ (*i.e.* individual realization of the set of probabilistic facts) compatible with the query q. The probability of a query q in a program $\mathcal{P}$ can then be computed as $P_\mathcal{P}(q) = \sum_w P(w) \cdot \mathbb{I}[F(w) \equiv q]$, where $F(w) \equiv q$ stands for the fact that the realization w is compatible with q being true.

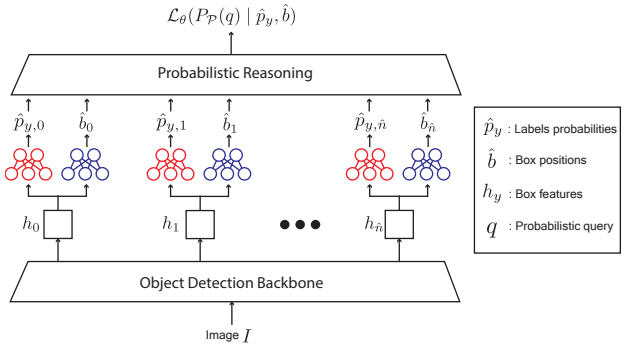

Figure 1. The pretrained object detection backbone outputs the box features h for the detected objects and the corresponding labels predictions $\hat{p}_y$ and box positions predictions $\hat{b}$ that are fed to the probabilistic reasoning layer. This layer computes the probability of the query along with the gradients with respect to $\hat{p}_y$ and $\hat{b}$ that can be backpropagated through the entire network.

Remarkably, recent advances in probabilistic programming have lead to *learnable* probabilistic facts (Manhaeve et al., 2018). One can train their parameters by minimizing a loss that depends on the probability of a querry q: $\hat{\theta} = \arg\min_\theta \mathcal{L}(P(q \mid \theta))$.

2.3. ProbKT: weakly supervised knowledge transfer with probabilistic reasoning

Fig. 1 shows a graphical description of our approach. It starts form a pre-trained object detection model f_θ on the source domain. We extract its backbone and insert it into a new model f_θ^* with new target box position predictors and box label classifiers. The box positions and labels predictions of this new model are used as neural predicates and fed to a probabilistic logic module. This module evaluates the probability of queries q_t, the loss and the corresponding gradient that can be backpropagated to the classifier and the backbone. As we want to maximize the probability of the queries being true, we use the following loss function:

$$\mathcal{L}_\theta = \sum_{(I_t, q_t) \in \mathcal{D}_t} -log P_\mathcal{P}(q_t \mid f_\theta^*(I_t))$$

If the backbone can in theory be trained end to end with this procedure, our experiments showed that only updating the box features classifiers resulted in more stability as also

shown in previous works (Zhong et al., 2020).

2.4. ProbKT*: the most probable world and connection to Hungarian matching

The probabilistic step inference described above requires a smart aggregation of all worlds compatible with the query q. Yet, in certain cases, one can reduce the computational cost by only considering the most probable world. Indeed, in the case when the query consists of the list of different class labels in the images, only accounting for the most probable world is equivalent to the Hungarian matching algorithm as used in *e.g.* Carion et al. (2020). More details are given in the Appendix A.

3. Experiments

3.1. Datasets

We evaluate our approach on two different datasets: a Molecules dataset with images of chemical compounds and a MNIST-based object detection dataset. For each dataset three subsets, corresponding to different domains, are used: (1) a source domain, (2) a target domain and a (3) out-of-distribution domain (OOD). The source domain is the richly annotated domain that was used to pre-train the object detection model. The target domain is the domain of interest but with poor annotations. Lastly, the OOD domain contains images from a different distribution than the source and target domains and is used to study the generalizability of the models. Source and target domains are split in 5 folds of train and validation sets and a independent test set. Sizes of the different splits per dataset are summarized in Table 2 in Appendix B.

3.1.1. Molecules Dataset

The Molecules dataset contains images depicting chemical compounds. For the richly annotated source domain a procedure was used as described in Oldenhof et al. (2020; 2021) for generating the bounding box labels for the individual atoms present in the chemical compound depiction in the image. As classes in source domain we have several atom types: carbon (C), hydrogen (H), oxygen (O) and nitrogen (N). In the poorly annotated target domain we have as labels only the counts of the atoms present in image which translates to the chemical formula of the molecule in the image (for example: $C_6H_{12}O_6$), so no information is available about the locations of the individual atoms in the image. The same classes from source domain (C,H, O and N) are also present in target domain and also an extra atom type: sulfur (S). As OOD test dataset we randomly select 1000 images from the external UoB dataset (Sadawi et al., 2012) in order to obtain images from depictions of chemical compounds containing only the atom types present in target domain (C,H,O,N and S). Some example images from the Molecules dataset are visualised in Figure 2 in Appendix B.

3.1.2. MNIST Object Detection Dataset

The MNIST object detection dataset is generated[1] using the original MNIST dataset (Deng, 2012) where on each image three MNIST digits appear. The MNIST object detection dataset allows to experiment with a more arbitrary type of weak supervision. Each object in this dataset represents a digit which can be aggregated and so this allows to label an image with only the sum of all digits in the image instead of the class counts of the objects. For the richly annotated source domain not all 10 digit classes (0-9) appear, the classes 7, 8 and 9 are left out. The poorly annotated target domain has all possible classes (0-9) and as labels only the sum of all digits in the images is provided. For the OOD test dataset, images are used that contain maximum 4 MNIST digits, instead of 3 digits as in target (and source) domain. Some example images from the MNIST object detection dataset are visualised in Figure 3 in Appendix B.

3.2. Models

In our experiments three (3) object detection models are used: (1) a pretrained object detection model on source domain, (2) a finetuned (using ProbKT) object detection model on target domain and lastly (3) a retrained object detection model using the fine-tuned model box predictions as pseudo labels in target domain. So the retrained model is trained end-to-end in a fully supervised fashion where transfer learning can occur through the pseudo labels on target domain. Besides the three (3) object detection models we also compare performance against a baseline model described in Section 3.2.2.

3.2.1. Object detection model

The object detection model used in our experiments is the widely used FasterRCNN (Ren et al., 2015) model. The initial FasterRCNN used for the experiments is pretrained on the COCO dataset (Lin et al., 2014) before pretraining it on source domain of our datasets.

3.2.2. Baseline

We compare our approach against a Resnet50 (He et al., 2016) backbone pretrained on ImageNet (Deng et al., 2009). Fine-tuning is performed by adding an extra multitask regression layer that is trained to predict the individual counts of the objects in the image as in Xue et al. (2016). This architecture naturally relies only on labels counts in the target images for fine-tuning. However, this comes at the cost of no direct box positions predictions. Rather, we predict box

[1] https://github.com/hukkelas/MNIST-ObjectDetection

Model	Type	MNIST count acc.	MNIST sum acc.	MNIST mAP (mAP@IoU=0.5)	Mol. count acc.	Mol. mAP (mAP@IoU=0.5)
Resnet50_count_(sum) (baseline)	Target Domain	0.044 ± 0.041	0.506 ± 0.063	$0.003 \pm 0.003 (0.014 \pm 0.011)$	$\mathbf{0.978 \pm 0.004}$	0.0 ± 0.0 (0 ± 0)
Resnet50_count_(sum) (baseline)	OOD	0.01 ± 0.009	0.015 ± 0.004	$0.003 \pm 0.002 (0.011 \pm 0.007)$	0.0 ± 0.0	n/a
Resnet50_count_(sum) (baseline)	Source Domain	0.127 ± 0.132	0.649 ± 0.108	$0.005 \pm 0.004 (0.028 \pm 0.018)$	0.828 ± 0.021	0.0 ± 0.0 (0 ± 0)
RCNN (Pretrained)	Target Domain	0.292 ± 0.005	0.298 ± 0.005	0.632 ± 0.014 (0.685 ± 0.002)	0.592 ± 0.007	0.568 ± 0.005 (0.785 ± 0.004)
RCNN (Pretrained)	OOD	0.205 ± 0.004	0.212 ± 0.004	0.631 ± 0.013 (0.683 ± 0.002)	0.348 ± 0.036	n/a
RCNN (Pretrained)	Source domain	0.961 ± 0.008	0.961 ± 0.008	$\mathbf{0.917 \pm 0.021}$ $(\mathbf{0.988 \pm 0.002})$	$\mathbf{0.948 \pm 0.004}$	$\mathbf{0.737 \pm 0.005}$ $(\mathbf{0.979 \pm 0.0})$
RCNN (Finetuned)	Target Domain	0.629 ± 0.02	0.631 ± 0.02	0.802 ± 0.024 (0.873 ± 0.009)	0.806 ± 0.014	$\mathbf{0.686 \pm 0.009}$ (0.972 ± 0.005)
RCNN (Finetuned)	OOD	0.532 ± 0.021	0.536 ± 0.02	0.795 ± 0.023 (0.867 ± 0.007)	0.538 ± 0.14	n/a
RCNN (Finetuned)	Source domain	0.907 ± 0.021	0.908 ± 0.021	0.894 ± 0.025 (0.967 ± 0.007)	0.838 ± 0.019	0.724 ± 0.007 (0.959 ± 0.005)
RCNN (Retrained)	Target Domain	$\mathbf{0.888 \pm 0.008}$	$\mathbf{0.889 \pm 0.008}$	$\mathbf{0.871 \pm 0.014}$ $(\mathbf{0.971 \pm 0.002})$	0.874 ± 0.018	0.597 ± 0.02 $(\mathbf{0.978 \pm 0.003})$
RCNN (Retrained)	OOD	$\mathbf{0.844 \pm 0.01}$	$\mathbf{0.845 \pm 0.011}$	$\mathbf{0.866 \pm 0.015}$ $(\mathbf{0.968 \pm 0.002})$	$\mathbf{0.626 \pm 0.034}$	n/a
RCNN (Retrained)	Source domain	$\mathbf{0.963 \pm 0.006}$	$\mathbf{0.963 \pm 0.005}$	0.907 ± 0.009 $(\mathbf{0.988 \pm 0.002})$	0.947 ± 0.017	0.692 ± 0.025 (0.976 ± 0.008)

Table 1. Results of the experiments on the MNIST object detection dataset and Molecules dataset. Reported test accuracies over the 5 folds.

predictions using a class activation maps as in Bae et al. (2020) to compare its performance on objects localization.

For the MNIST object detection dataset, where we only need the sum of digits to be predicted, we create a small variations of the baseline model. In the variation we add an extra layer on top that sums the individual counts to give the resulting sum. The advantage for this model is that we still have predictions that can be interpreted as the individual counts and secondly for source domain (where we have still the true counts available) we can add these counts to the loss so it can help the training of the model. The baseline model variation can still be considered to be trained in fully supervised way.

3.3. Results

We employ several metrics to evaluate the performance of the different models on the different datasets. The primary metric (used for validation) for the Molecules dataset is the count accuracy. The count accuracy measures the ratio of correct images where all individual counts of (all detected) objects are correct. To evaluate also how well the model is performing in localizing the different objects in the image, we also measure the mean average precision (mAP) performance, which is a standard performance metric in objects detection. The primary metric for the MNIST object detection dataset is the sum accuracy. The sum accuracy measures the ratio of correct images where the predicted sum (of all detected digits) is correct. The results for the experiments are summarized in Table 1.

3.3.1. MNIST OBJECT DETECTION RESULTS

For the MNIST dataset we observe that all metrics improve on target and OOD domain for RCNN after fine-tuning the pre-trained model. On source domain however there is slight decrease of performance after fine-tuning. Retraining the RCNN using pseudo-labels on target domain can improve the performance significantly even further on all domains.

3.3.2. MOLECULES RESULTS

On Molecules dataset we see that after fine-tuning the pre-trained RCNN model on target domain, performance increases on all metrics on target domain and OOD while it decreases on source domain. Retraining the RCNN on target domain can help to increase the count accuracy on all domains while the mAP metric decreases. This could be related with the fact that the bounding boxes in the Molecules dataset are very small. Therefore we also report the mAP_50 (IoU threshold of 0.5) where we observe no significant reduction in performance after fine-tuning or retraining. We can also observe that the Resnet50 baseline performs very good on the target domain but completely fails on OOD. As a final remark for Table 1 we clarify that for the external OOD dataset (UoB selection) for Molecules no bounding boxes are available so mAP can not be evaluated here.

4. Conclusion and Discussion

Objects detection architectures are usually large and require significant computational resources for training. When the distribution of images fed to these models change, it is thus preferable to update them rather than training again from the ground up. However, these updates can be technically challenging if only weak supervision is available on the target domain.

In this work, we showed that probabilistic programming is an elegant and effective solution for adapting pre-trained object detection models to domains with arbitrary types of weak supervision. We also demonstrated that the fine-tuning step could produce accurate pseudo-labels that can be used to improve the performance of the model further.

Probabilistic programming is a versatile and powerful way to encode uncertainty in knowledge graphs. The ideas laid out here could then be used for updating models from other areas such as image to graph translations. We leave this investigation as future work.

References

W. Bae, J. Noh, and G. Kim. Rethinking class activation mapping for weakly supervised object localization. In *European Conference on Computer Vision*, pages 618–634. Springer, 2020.

A. Behl, O. Hosseini Jafari, S. Karthik Mustikovela, H. Abu Alhaija, C. Rother, and A. Geiger. Bounding boxes, segmentations and object coordinates: How important is recognition for 3d scene flow estimation in autonomous driving scenarios? In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2574–2583, 2017.

N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

E. De Brouwer, J. Gonzalez, and S. Hyland. Predicting the impact of treatments over time with uncertainty aware neural differential equations. In *International Conference on Artificial Intelligence and Statistics*, pages 4705–4722. PMLR, 2022.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

L. Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE signal processing magazine*, 29(6):141–142, 2012.

T. Deselaers, B. Alexe, and V. Ferrari. Weakly supervised localization and learning with generic knowledge. *International journal of computer vision*, 100(3):275–293, 2012.

M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88 (2):303–338, June 2010.

I. M. Hameed, S. H. Abdulhussain, and B. M. Mahmmod. Content-based image retrieval: A review of recent trends. *Cogent Engineering*, 8(1):1927469, 2021.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

M. Horn, E. De Brouwer, M. Moor, Y. Moreau, B. Rieck, and K. Borgwardt. Topological graph neural networks. *arXiv preprint arXiv:2102.07835*, 2021.

T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

R. Manhaeve, S. Dumancic, A. Kimmig, T. Demeester, and L. De Raedt. Deepproblog: Neural probabilistic logic programming. *Advances in Neural Information Processing Systems*, 31, 2018.

M. Oldenhof, A. Arany, Y. Moreau, and J. Simm. Chemgrapher: optical graph recognition of chemical compounds by deep learning. *Journal of chemical information and modeling*, 60(10):4506–4517, 2020.

M. Oldenhof, A. Arany, Y. Moreau, and J. Simm. Self-labeling of fully mediating representations by graph alignment. In *Benelux Conference on Artificial Intelligence*, pages 46–65. Springer, 2021.

S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

N. M. Sadawi, A. P. Sexton, and V. Sorge. Chemical structure recognition: a rule-based approach. In *Document Recognition and Retrieval XIX*, volume 8297, page 82970E. International Society for Optics and Photonics, 2012.

J. Simm, A. Arany, E. D. Brouwer, and Y. Moreau. Expressive graph informer networks. In *International Conference on Machine Learning, Optimization, and Data Science*, pages 198–212. Springer, 2021.

M. Tomei, M. Cornia, L. Baraldi, and R. Cucchiara. Art2real: Unfolding the reality of artworks via semantically-aware image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5849–5859, 2019.

J. Uijlings, S. Popov, and V. Ferrari. Revisiting knowledge transfer for training object class detectors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1101–1110, 2018.

Y. Xue, N. Ray, J. Hugh, and G. Bigras. Cell counting by regression using convolutional neural network. In *European Conference on Computer Vision*, pages 274–290. Springer, 2016.

Y. Zhong, J. Wang, J. Peng, and L. Zhang. Boosting weakly supervised object detection with progressive knowledge transfer. In *European conference on computer vision*, pages 615–631. Springer, 2020.

A. ProbKT*: the most probable world and connection to Hungarian matching

The probabilistic inference step requires a smart aggregation of all worlds compatible with the query q. Yet, in certain cases, one can reduce the computational cost by only considering the most probable world. Indeed, consider the case when the query consists of the list of different class labels in the images. For a number of boxes $\hat{n}$ proposed by the objects detection model, the probability of query can be written as the set of labels $q = \{y^i : i = 1, ..., \hat{n}\}$. If we further write $\hat{p}^k_{y,n}$ as the probability of the label of box n belonging to class k given by the model (as introduced in Section 2.2.1), we have:

$$\begin{aligned} P_{\mathcal{P}}(q) &= \sum_{k=1}^{\hat{n}!} \hat{p}_{y,0}^{\sigma_k(0)} \cdot \hat{p}_{y,1}^{\sigma_k(1)} \cdot ... \cdot \hat{p}_{y,\hat{n}}^{\sigma_k(\hat{n})} \\ &= \sum_{k=1}^{\hat{n}!} \prod_n \hat{p}_{y,n}^{\sigma_k(n)} \end{aligned}$$

where σ_k corresponds to the k^{th} permutation of the query vector q. To avoid the computation of each possible world contribution, one can only use the configuration with the largest contribution to $P_{\mathcal{P}}(q)$ and discard the other ones. Indeed, this also corresponds to the largest contribution to the gradient update.

This possible world corresponds to the permutation σ^* that satisfies:

$$\begin{aligned} \sigma^* &= \arg\max_{\sigma} \log(\prod_n \hat{p}_{y,n}^{\sigma_k(n)}) \\ &= \arg\max_{\sigma} \sum_n \hat{p}_{y,n}^{\sigma_k(n)} \\ &= \arg\min_{\sigma} \sum_n (1 - \hat{p}_{y,n}^{\sigma_k(n)}). \end{aligned}$$

Remarkably, this corresponds to the solution of the best alignment using the Hungarian matching algorithm with cost $c(n) = (1 - \hat{p}_{y,n}^{\sigma_k(n)})$, as used, among others, in DETR (Carion et al., 2020). Thus, when the query is is the set of class labels, the most plausible world can thus be inferred with the Hungarian matching algorithm.

You can have as much text here as you want. The main body must be at most 8 pages long. For the final version, one more page can be added. If you want, you can use an appendix like this one, even using the one-column format.

B. Datasets

B.1. Molecules

Representation learning on molecules is important for applications such as drug discovery (Simm et al., 2021; De Brouwer et al., 2022), usually operating on graphs with graph neural network architectures (Horn et al., 2021). However, at times, only images of molecules are available (Oldenhof et al., 2020). On Figure 2, we show examples of images of molecules. Our goal is to predict the chemical formula from the images only.

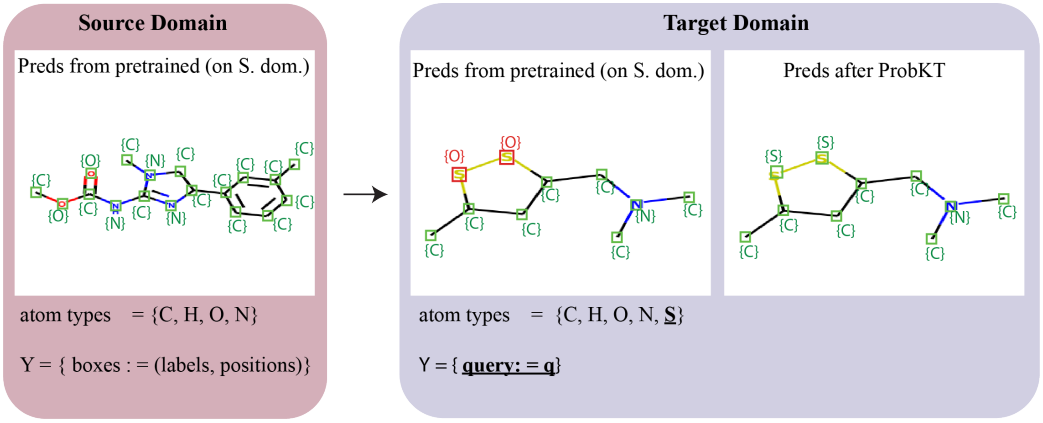

Figure 2. On the left we have source domain where a model can be trained using bounding box information(labels,positions) but only on a limited set of atom types (C,H, O, N). In the middle we can see that the pretrained model is not able to recognize the sulfur (S) from target domain correctly. On the right we see that the model is able to adapt to target domain after probabilistic reasoning using weak labels from target domain and is able to recognize the sulfur (S).

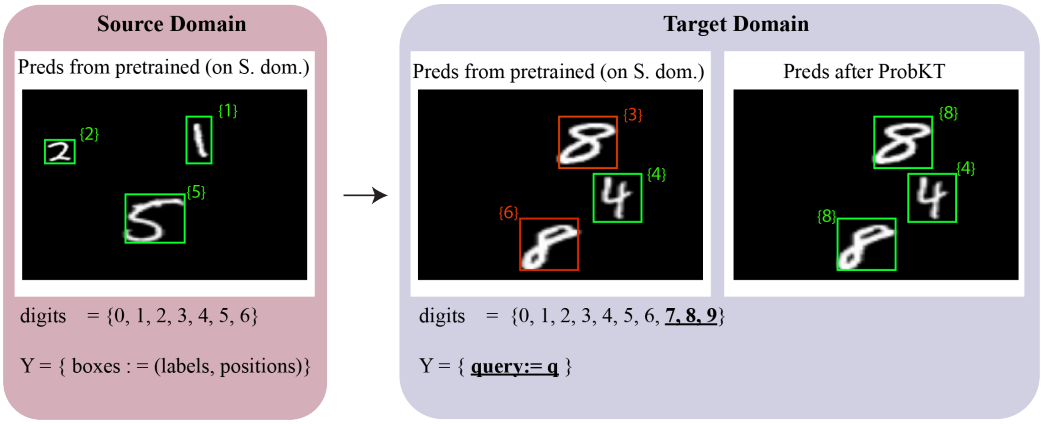

Figure 3. On the left we have source domain where a model can be trained using bounding box information(labels,positions) but only on a limited set of digits (0, 1, 2, 3, 4, 5, 6). In the middle we can see that the pretrained model is not able to recognize the digit eight (8) from target domain correctly. On the right we see that the model is able to adapt to target domain after probalistic reasoning using (arbitrary) weak labels and is able to recognize the digit eight (8).

Dataset	Type	Split	Size (number of samples)
MNIST object detection	Source	train	700
MNIST object detection	Source	validation	300
MNIST object detection	Source	test	1000
MNIST object detection	Target	train	700
MNIST object detection	Target	validation	300
MNIST object detection	Target	test	1000
MNIST object detection	OOD	test	1000
Molecules	Source	train	1400
Molecules	Source	validation	600
Molecules	Source	test	1000
Molecules	Target	train	1400
Molecules	Target	validation	600
Molecules	Target	test	1000
Molecules	OOD	test	1000

Table 2. Dataset sizes for the different splits. For train and validations splits 5 folds are used.