Appendix to Conformal Credal Self-Supervised Learning

Julian Lienen* JULIAN.LIENEN@UPB.DE
Caglar Demir CAGLAR.DEMIR@UPB.DE
Paderborn University, Germany

Eyke Hüllermeier EYKE@LMU.DE
University of Munich (LMU), Germany

Appendix A. Pseudo-Code of CCSSL

Algorithm 2 CCSSL with consistency regularization

Require: Batch of labeled instances with degenerate ground truth distributions $\mathcal{B}_l = \{(\boldsymbol{x}_i, p_i)\}_{i=1}^{B} \in (\mathcal{X} \times \mathcal{Y})^B$, unlabeled batch ratio μ, batch $\mathcal{B}_u = \{\boldsymbol{x}_i\}_{i=1}^{\mu B}$ of unlabeled instances, unlabeled loss weight λ_u, model $\hat{p} : \mathcal{X} \longrightarrow \mathbb{P}(\mathcal{Y})$, strong and weak augmentation functions $\mathcal{A}_s, \mathcal{A}_w : \mathcal{X} \longrightarrow \mathcal{X}$, calibration data $\mathcal{D}_{\text{calib}} \subset (\mathcal{X} \times \mathcal{Y})^L$, inductive conformal prediction procedure $ICP : (\mathcal{X} \times \mathcal{Y})^L \times \mathbb{P}(\mathcal{Y}) \longrightarrow (\mathcal{Y} \longrightarrow [0, 1])$

$\mathcal{L}_l = \frac{1}{B} \sum_{(\boldsymbol{x},p) \in \mathcal{B}_l} D_{KL}(p \,||\, \hat{p}(\mathcal{A}_w(\boldsymbol{x})))$
Initialize pseudo-labeled batch $\mathcal{U} = \emptyset$
for all $\boldsymbol{x} \in \mathcal{B}_u$ **do**
  Derive possibility distribution $\pi = ICP(\mathcal{D}_{\text{calib}}, \hat{p}(\mathcal{A}_w(\boldsymbol{x})))$
  Apply normalization to π such that $\max_{y \in \mathcal{Y}} \pi(y) = 1$ (e.g., as in Eq. (9))
  Construct credal set $\mathcal{Q}_\pi$ as in Eq. (1)
  $\mathcal{U} = \mathcal{U} \cup \{(\boldsymbol{x}, Q_\pi)\}$
end for
$\mathcal{L}_u = \frac{1}{\mu B} \sum_{(\boldsymbol{x},Q_\pi) \in \mathcal{U}} \mathcal{L}^*(Q_\pi, \hat{p}(\mathcal{A}_s(\boldsymbol{x})))$ {Can be solved via generalized credal learning procedure (Alg. 1 in main paper)}
return $\mathcal{L}_l + \lambda_u \mathcal{L}_u$

Appendix B. Experimental Details

B.1. Settings

To conduct the experiments as presented in the paper, we followed the basic semi-supervised learning evaluation scheme as in Lienen and Hüllermeier (2021); Sohn et al. (2020). However, as opposed to previous evaluations, we reduce the number of iterations to 2^{18} with a batch size of 32, which allows for a proper hyperparamter optimization of all methods. To this end, we employ a Bayesian optimization[1] with 20 runs for each combination of

* Corresponding author

1. We used the Bayesian optimization implementation as offered by *Weights & Biases* Biewald (2020) with default parameters.

Table 3: Hyperparameter search spaces considered in the optimization.

Method	Parameter	Values
All	Initial learning rate	$\{0.005, 0.01, 0.03, 0.05, 0.1\}$
	Unlabeled batch multiplicity μ	$\{3, 7\}$
	Weight decay	$\{0.0005, 0.0001\}$
	Unlabeled loss weight λ_u	$\{1\}$
FixMatch (DA), UDA	Confidence threshold τ	$\{0.7, 0.8, 0.9, 0.95\}$
	Temperature	$\{0.5, 1\}$
FlexMatch	Cutoff threshold	$\{0.8, 0.9, 0.95\}$
	Threshold warmup	$\{\text{True}, \text{False}\}$
CCSSL-diff	Calibration split	$\{0.1, 0.25, 0.5\}$
CCSSL-prop	Calibration split	$\{0.1, 0.25, 0.5\}$
	Non-conf. sensitivity γ	$\{0.01, 0.1, 1\}$

dataset and number of labels on a separate validation split. Moreover, we use Hyperband Li et al. (2017) with $\eta = 3$ and 20 minimum epochs (that is, iterating over all unlabeled instances once) for early stopping. Due to the computational complexity of this procedure, we determined the best hyperparameter on a fixed seed and applied those parameters to all repetitions with different seeds for the same dataset and number of labels combination. Albeit not being ideal, such routine still improves fairness compared to previous evaluations which do not apply the same hyperparameter tuning procedure to all regarded baselines.

Tab. 3 shows the considered parameter spaces. To train the models, we use SGD with a Nesterov momentum of 0.9. We further employ cosine annealing as learning rate schedule Loshchilov and Hutter (2017). Moreover, we apply exponential moving averaging with a fixed decay of 0.999 to the weights.

B.2. Code and Environment

Our official implementation is publicly available[2]. Therein, we implemented all methods using PyTorch[3], where we reused the official implementations if available. We proceeded from a popular FixMatch re-implementation in PyTorch[4] for the image classification experiments, which we carefully checked for any differences to the original repository, and embedded all other baselines into it. To conduct the experiments, we used several Nvidia A100 GPUs in a modern high performance cluster environment.

2. https://github.com/julilien/C2S2L
3. https://pytorch.org/, BSD-style license
4. https://github.com/kekmodel/FixMatch-pytorch, MIT license

Table 4: Averaged ECE scores with 15 bins over 3 seeds for different numbers of labels. **Bold** entries indicate the best performing method per column. The standard deviation is a factor of $1e^{-2}$.

	CIFAR-10			CIFAR-100			SVHN			STL-10
	40 lab.	250 lab.	4000 lab.	400 lab.	2500 lab.	10000 lab.	40 lab.	250 lab.	1000 lab.	1000 lab.
UDA	0.159 ±7.9	**0.051** ±0.2	0.046 ±0.1	0.420 ±2.6	0.232 ±0.5	0.173 ±0.5	0.124 ±1.1	0.037 ±1.3	0.030 ±0.1	0.140 ±0.9
FixMatch	0.136 ±4.3	0.059 ±1.0	0.048 ±0.1	0.417 ±2.3	0.254 ±0.5	0.168 ±0.1	0.275 ±34.9	0.038 ±1.3	**0.027** ±0.1	0.128 ±0.8
FixMatch DA	0.131 ±11.1	0.057 ±0.6	0.047 ±0.1	**0.347** ±2.4	0.234 ±0.4	0.169 ±0.1	**0.101** ±3.1	0.041 ±1.5	0.029 ±0.1	0.127 ±0.8
FlexMatch	0.114 ±3.7	0.057 ±0.5	0.052 ±0.2	0.404 ±0.3	0.232 ±0.5	0.169 ±0.2	0.126 ±1.3	0.039 ±0.3	0.033 ±0.4	0.130 ±0.9
CSSL	**0.079** ±4.4	0.053 ±0.1	0.045 ±0.0	0.368 ±0.8	0.233 ±0.6	0.167 ±0.2	0.121 ±5.1	0.041 ±1.1	0.032 ±0.1	0.126 ±0.9
CCSSL-diff	0.139 ±3.2	0.058 ±0.1	0.045 ±0.1	0.352 ±1.6	**0.222** ±0.3	0.165 ±0.1	0.117 ±7.2	0.044 ±1.2	0.030 ±0.3	0.129 ±0.6
CCSSL-prop	0.101 ±2.2	0.054 ±0.2	**0.044** ±0.1	**0.347** ±2.3	0.227 ±0.2	**0.162** ±0.2	0.128 ±8.6	**0.033** ±0.1	0.028 ±0.2	**0.124** ±0.8

Appendix C. Additional Results

C.1. Predictor Calibration

For completeness, we present the quality of the prediction probability distributions in the large-scale image classification experiments with respect to their expected calibration errors in Tab. 4. Both CCSSL variants demonstrate favorable calibration properties, whereas CCSSL-prop often outperforms all other methods.

C.2. Learning Curves

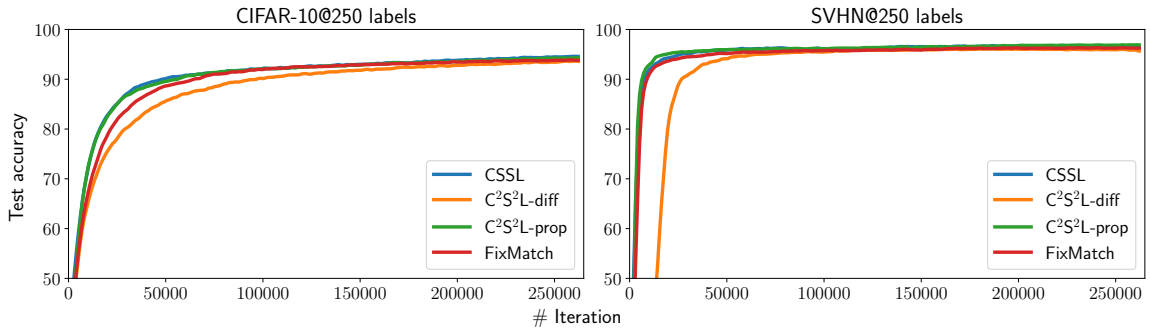

Figure 3: Test accuracies over the course of the training. The results are averaged over 3 different random seeds.

In addition, we provide the learning curves in terms of test accuracy per training iteration averaged over 3 random seeds on CIFAR-10 and SVHN with 250 labels each in Fig. 3. As can be seen, CCSSL-prop shows a similar training efficiency as CSSL, whereas CCSSL-diff realizes a more cautious learning than all other methods. This becomes particularly visible for SVHN. Here, CCSSL-prop shows an even improved training efficiency compared to CSSL. Remarkably, the CCSSL variants have less labeled supervision available as part of it is separated in form of the calibration split, which can be one reason for the cautiousness of CCSSL-diff in the first iterations. Together with the validity gains in the pseudo-label quality, especially CCSSL-prop demonstrates its effectiveness to the task of semi-supervised learning.

C.3. Mitigation of Confirmation Biases: Imbalanced Data

As discussed in the main paper, consistency regularization (CR) and the validity of conformal (credal) pseudo-labels serve as means to tackle confirmation biases. In addition to the previously shown experiments, we consider here another experimental setting aiming to isolate their individual contributions to the mitigation of confirmation biases. Namely, we look at EMNIST-ByClass Cohen et al. (2017) consisting of 814,255 handwritten letters (both lower and upper case) and digits, constituting 62 imbalanced classes in total[5]. The class imbalance leads to an attenuation of CR as frequently occurring classes dominate underrepresented ones. More technically, the neighborhood of instances belonging to an underrepresented region may be mostly populated by instances from different classes, thereby violating the expansion assumption Wei et al. (2021). Consequently, from an empirical risk minimization point of view, it might be more reasonable to attribute larger regions populated by unlabeled instances from the minority class to a majority classes. This leaves the former overlooked, so that a "de-noising" of the pseudo-labels is not possible anymore. Again, the validity guarantees provided by the conformal prediction framework serve as a fallback here.

For this dataset, we consider either 250 or 500 labeled instances and train for 2^{15} iterations, keeping all other experimental parameters the same as before. Adopting the reported optimal hyperparameters for CSSL in Lienen and Hüllermeier (2021), we used a fixed learning rate of 0.03, SGD with Nesterov momentum of 0.9 and trained a Wide ResNet-28-2 with a batch size of 32 for three different seeds. Table 5 shows the resulting generalization performances for the individual methods. Moreover, Fig. 4 presents the learning curves, which show similar but even more extreme trends as also observed in the CIFAR-100 experiments presented in Sec. 5.2 of the main paper.

Table 5: Test accuracies on EMNIST-ByClass. The presented results and their standard deviations are computed over 3 seeds.

	250 lab.	500 lab.
CSSL	49.96 ±4.08	62.74 ±2.15
CCSSL-diff	**59.22** ±3.93	67.39 ±3.74
CCSSL-prop	57.90 ±5.85	**67.62** ±4.61

C.4. Ablation Studies

In the following, we present further ablation studies to investigate properties of CCSSL more thoroughly. If not stated otherwise, we set the initial learning rate to 0.03, $\lambda_u = 1$, $\mu = 7$ and the weight decay to 0.0005. Also, we consider calibration size fractions of 0.25 by default.

Possibility Distribution Normalization Conformal Credal Pseudo-Labeling involves normalizing possibility distributions $\pi : \mathcal{Y} \longrightarrow [0, 1]$ to satisfy $\max_{y \in \mathcal{Y}} \pi(y) = 1$. In Eq. (9)

5. We refer to Fig. 2 in Cohen et al. (2017) for a detailed overview over the class distribution.

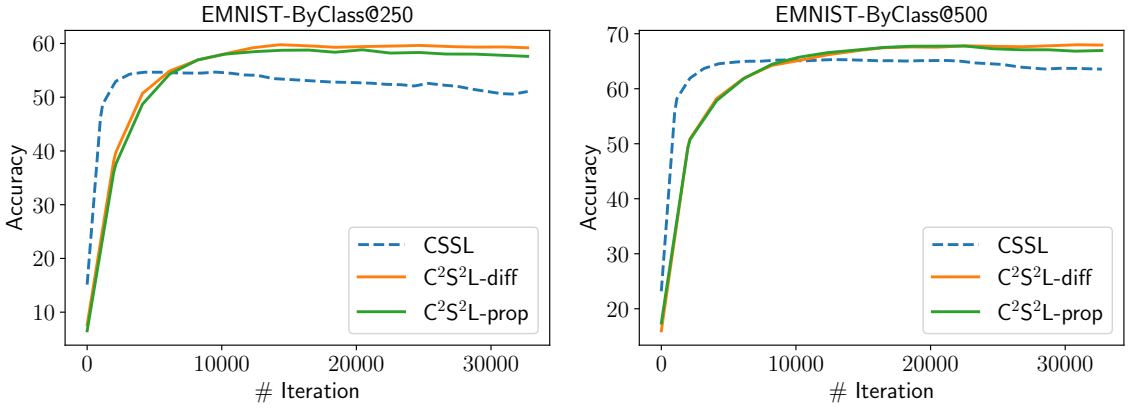

Figure 4: Averaged learning curves in terms of test accuracies over three seeds on EMNIST-ByClass with 250 and 500 labels.

of the paper, we introduced a proportion-based normalization technique, to which we refer as *normalization 1*. As an alternative, one can consider the following normalization Cella and Martin (2021), to which we refer as *normalization 2*:

$$\pi(\hat{y}) = \begin{cases} 1 & \text{if } \hat{y} = \operatorname{argmax}_{y' \in \mathcal{Y}} \pi(y'), \\ \pi(\hat{y}) & \text{otherwise.} \end{cases} \tag{13}$$

It is easy to see that (13) leads to smaller credal sets due to $0 \leq \pi(\cdot) \leq 1$.

Table 6: Test accuracies per normalization and CCSSL variant for 250 labels each. The presented results and their standard deviations are computed over 3 seeds.

	CIFAR-10		SVHN	
	Norm. 1	Norm. 2	Norm. 1	Norm. 2
CCSSL-diff	92.38 ±1.14	92.71 ±0.81	95.93 ±1.78	95.70 ±1.81
CCSSL-prop	92.26 ±2.30	91.84 ±1.13	96.61 ±1.08	95.72 ±1.50

Tab. 6 shows the results. Although the second normalization strategy leads to smaller credal sets, it leads to inferior generalization performance for the proportion-based non-conformity measure, suggesting that an overly extreme credal set construction may be suboptimal. This supports the adequacy of the first normalization strategy as employed in CCSSL by default.

Calibration Split Size The calibration size used to determine the number of calibration instances affects the quality of the credal sets. Here, we consider calibration split proportions in $\{0.1, 0.25, 0.5, 0.9\}$. As can be seen in Fig. 5, the differences in the performances do not vary too much. On CIFAR-10 with 250 labels, no clear trend can be observed. However, the deviations of runs with a fraction of 0.25 appear significantly higher than for the other sizes. This could be due to the sacrifice of too many labeled instances without achieving

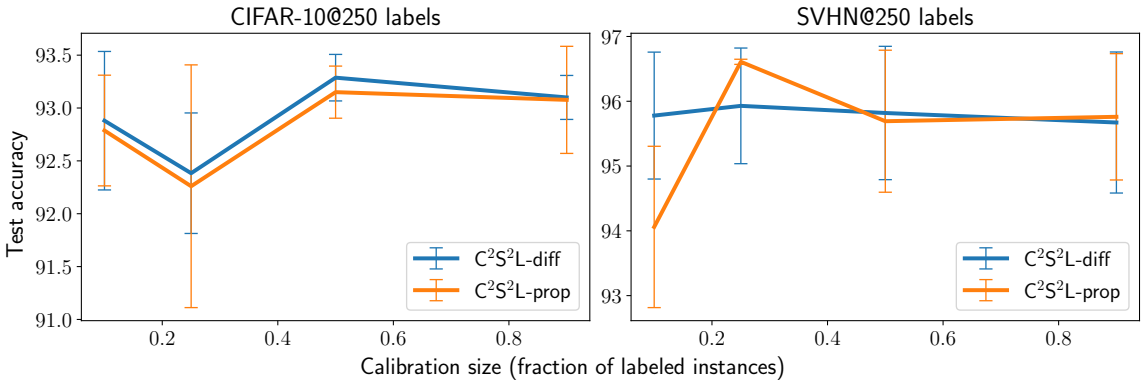

Figure 5: Test accuracy and the standard deviation per calibration size for the two variants of CCSSL. The results are averaged over 3 different random seeds.

too precise pseudo-supervision. Lower and higher calibration sizes may overcome this by benefiting from either of these two extremes (higher pseudo-label quality through many labeled instances or larger calibration sets). In case of SVHN, CCSSL-prop seems to be more sensitive to the calibration size and achieves the best results with a calibration size fraction of 0.25. Here, too small calibration sizes clearly lead to unsatisfying results, demonstrating again the influence of the pseudo-label quality on the overall generalization performance.

Proportion-based Non-Conformity Sensitivity The proportion-based non-conformity measure $\alpha(\mathcal{D}, (\boldsymbol{x}, y))$ in Eq. (3) involves the parameter $\gamma \geq 0$, which represents the sensitivity towards the influence of the prediction $\hat{p}_{\mathcal{D}}(\boldsymbol{x})(y)$ on the scoring for a given tuple $(\boldsymbol{x}, y)$.

Table 7: Averaged test accuracies and their standard deviations over 3 random seeds for various γ values used in CCSSL-prop. For each dataset, we considered 250 labeled examples to be given.

γ	CIFAR-10	SVHN
0.01	**93.64** ±0.33	95.60 ±2.17
0.1	93.24 ±1.16	95.91 ±1.76
0.5	93.19 ±1.02	**96.82** ±0.10
1	93.01 ±1.23	96.61 ±0.08
10	92.29 ±1.94	95.41 ±2.07

In Tab. 7, we report results of CCSSL-prop for $\gamma \in \{0.01, 0.1, 0.5, 1, 10\}$. While smaller γ values lead to better results for CIFAR-10, SVHN benefits from slightly higher γ values. These results show that this parameter indeed has an effect on the overall results, i.e., it is reasonable to consider it as a hyperparameter.

Table 8: Overview of the datasets considered in the KGE experiments.

	# Entities	# Relations	$\lvert\mathcal{D}_{\text{train}}\rvert$	$\lvert\mathcal{D}_{\text{val}}\rvert$	$\lvert\mathcal{D}_{\text{test}}\rvert$
UMLS	136	93	10,432	1,304	1,965
KINSHIP	105	51	17,088	2,136	3,210

C.5. Knowledge Graph Embedding Experiments

We were interested in evaluating conformal credal self-supervised learning in the link prediction problem on knowledge graphs. Knowledge graphs represent structured collections of facts Hogan et al. (2020), which are stored in graphs connecting entities via relations. These collections of facts have been used in a wide range of applications, including web search, question answering, and recommender systems Nickel et al. (2015). The task of identifying missing links in knowledge graphs is referred to as *link prediction*. Knowledge graph embedding (KGE) models have been particularly successful at tackling the link prediction task, among many others Nickel et al. (2015). In semi-supervised link prediction, only a fraction of facts is given. The task is then to "enrich" the input graph to detect relations that connect entities, which are subsequently also used in the learning of graph embeddings.

Experimental Setting In our experiments, we follow a standard training and evaluation setup commonly used in the KGE domain Ruffinelli et al. (2020); Cao et al. (2021). We consider three multiplicative interaction-based KGE embedding models: DistMult Yang et al. (2014), ComplEx Trouillon et al. (2016), and QMult Demir et al. (2021). To train the models, we model the problem as a 1vsAll classification (see Demir et al. (2022) and Ruffinelli et al. (2020) for more details about the 1vsAll training regime). Here, we compare conventional pseudo-labeling as described in Lee (2013) to CCSSL-diff. In the following, we refer to the former as PL. We do not employ consistency regularization or any other confirmation bias mitigation technique, demonstrating the flexibility of our framework. In our experiments, we used the two benchmark datasets UMLS and KINSHIP Dettmers et al. (2018), whose characteristics are provided in Table 8.

For each model (DistMult, ComplEx, and QMult), we applied a grid-search over the learning rates $\{0.01, 0.1, 0.001\}$, batch sizes $\{512, 1024\}$ and the number of epochs $\{5, 50\}$ to tune the parameters on a separate validation set. For CCSSL-diff, we initially divided $\mathcal{D}_{\text{train}}$ into training, calibration and unlabeled splits with 40:40:20 ratios, respectively. For the conventional pseudo-labeling procedure, we divided $\mathcal{D}_{\text{train}}$ into training, and unlabeled splits with 40:60 ratios, respectively. To ensure that each model is trained with the exact training split, we select the first 40% of all triples as training split.

As opposed to domains like image classification, where a small number of classes, for which labeled instances are provided, allows for a certain degree of interpolation, knowledge graphs involve a typically larger vocabulary of entities. By subselecting data from the knowledge graph, there is a much higher chance to miss some entities. Not observing parts of the vocabulary renders the task of learning their embeddings effectively as an unsupervised learning problem. This is why our considered training splits are relatively large. Arguably, CCSSL gets an unfair advantage here by providing (labeled) calibration

Table 9: Link prediction results on UMLS and KINSHIP. The 40:40:20 split ratio for CCSSL-diff and a 40:60 for conventional pseudo-labeling (PL). Bold entries denote best results per method, dataset and metric.

	UMLS			**KINSHIP**		
	MRR	H@1	H@3	MRR	H@1	H@3
DistMult-PL	0.229	0.128	0.243	0.262	0.160	0.284
DistMult-CCSSL	**0.246**	**0.161**	**0.265**	**0.274**	**0.170**	**0.299**
ComplEx-PL	**0.282**	0.160	**0.358**	0.333	0.226	0.382
ComplEx-CCSSL	0.253	**0.191**	0.261	**0.344**	**0.255**	**0.381**
QMult-PL	0.269	0.157	**0.309**	0.323	0.228	0.351
QMult-CCSSL	**0.294**	**0.217**	**0.309**	**0.328**	**0.238**	**0.363**

data beyond the labeled training data. However, this data is excluded from being used as pseudo-labeled training data either, and can not contribute to the learned embeddings directly. It leaves large parts of the knowledge graph unconnected as it prevents to enrich that part of the graph by pseudo-labels, having an influence on the generalizability of the learned KGE model.

Link Prediction Results Table 9 reports the link prediction performance on UMLS and KINSHIP. Overall, the results suggest that incorporating CCSSL in knowledge graph embedding model leads to better generalization performance in 16 out of 18 metrics on two benchmark datasets. Also, the credal pseudo-label construction is effective as it improves the results over the PL baseline. To address our discussions about fairness in the split modeling, we conduct two more experiments with 40:10:50 and 30:20:50 split ratios to quantify the impact of the calibration signal in the link prediction task, where the PL baselines observes splits with ratios 40:60 and 50:50, respectively.

Table 10 reports the link prediction performances for the 40:10:50 (CCSSL) and 40:60 (PL) split ratio. Interestingly, the results do not vary much (only in 3 out of 18 cases). This confirms that the labeled data split has critical influence on the overall results in KGE link prediction, whereas the contribution of the self-supervised part is limited. As said before, semi-supervised learning in knowledge graph embedding is much more depending on the training data compared to image classification.

Motivated by the findings in the results shown in Tables 9 and 10, we reduced the training set for CCSSL by 25% and increased the training set for pseudo-labeling by 25%. This setting leads to better generalization performance for PL in all metrics, again giving further evidence for our reasoning about the splits. This case clearly demonstrates a limitation of our method: CCSSL is especially favorable in settings where a sufficient amount of labeled instances are provided, particularly when facing structural data such as knowledge graphs.

Table 10: Link prediction results on UMLS and KINSHIP. The 40:10:50 split ratio for CCSSL-diff and a 40:60 for pseudo-labelling (PL). Bold entries denote best results per method, dataset and metric.

	UMLS			**KINSHIP**		
	MRR	H@1	H@3	MRR	H@1	H@3
DistMult-PL	0.229	0.128	0.243	0.262	0.160	0.284
DistMult-CCSSL	**0.246**	**0.161**	**0.265**	**0.275**	**0.170**	**0.299**
ComplEx-PL	**0.282**	0.160	**0.358**	0.333	0.226	0.382
ComplEx-CCSSL	0.253	**0.191**	0.261	**0.335**	**0.232**	**0.378**
QMult-PL	0.269	0.157	**0.309**	0.323	0.228	.351
QMult-CCSSL	**0.294**	**0.217**	**0.309**	**0.328**	**0.238**	**0.363**

Table 11: Link prediction results on UMLS and KINSHIP. The 30:20:50 split ratio for CCSSL-diff and a 50:50 for pseudo-labelling (PL). Bold entries denote best results per method, dataset and metric.

	UMLS			**KINSHIP**		
	MRR	H@1	H@3	MRR	H@1	H@3
DistMult-PL	**0.263**	**0.164**	**0.268**	**0.302**	**0.198**	**0.328**
DistMult-CCSSL	0.240	0.161	0.253	0.245	0.140	0.267
ComplEx-PL	**0.308**	**0.205**	**0.351**	**0.392**	**0.302**	**0.431**
ComplEx-CCSSL	0.228	0.159	0.241	0.255	0.148	0.285
QMult-PL	**0.330**	**0.223**	**0.362**	**0.381**	**0.292**	**0.422**
QMult-CCSSL	0.224	0.150	0.225	0.221	0.118	0.242

Appendix D. Generalized Credal Learning: Theoretical Results

In this section, we provide theoretical results on generalized credal learning as introduced in Section 4.2 of the main paper.

D.1. Proof of Theorem 1

In the following, we consider possibility distributions $\pi : \mathcal{Y} \longrightarrow [0, 1]$, where $\pi_i := \pi(y_i)$ abbreviates the possibility of class y_i. Without loss of generality, we assume the possibilities be ordered and normalized, i.e., $0 \leq \pi_1 \leq \ldots \leq \pi_K = 1$ for K classes. Furthermore, we also denote the respective probability of a class y_i given a distribution $p \in \mathbb{P}(\mathcal{Y})$ by $p_i := p(y_i)$.

As described in Section 4.2, the set of inequalities that defines the boundary of a credal set $\mathcal{Q}_\pi$ induces a convex polytope. In Fig. 6, such a credal set is illustrated for three and four

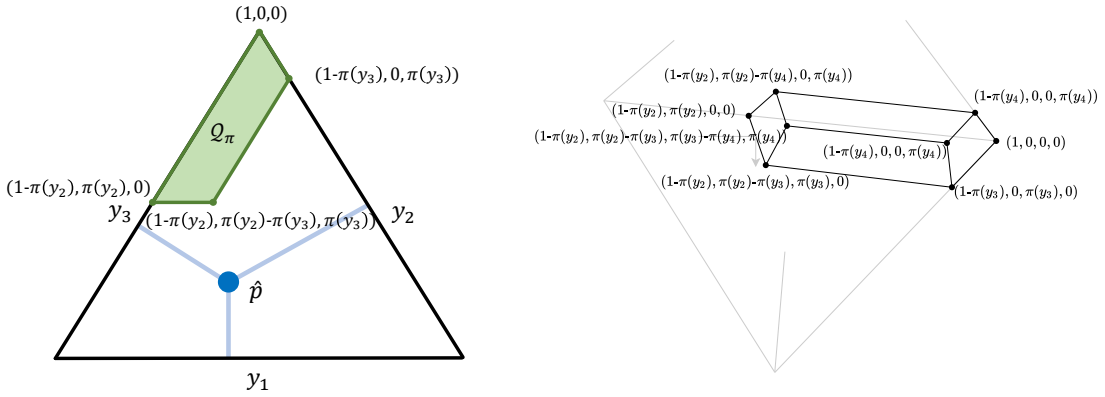

Figure 6: Schematic illustration of a credal set $\mathcal{Q}_\pi$ as a convex polytope in a barycentric coordinate space of all distributions $\mathbb{P}(\mathcal{Y})$ for three and four classes. $\mathcal{Q}_\pi$ is induced by a normalized possibility distribution π with $1 = \pi(y_1) \geq \ldots \geq \pi(y_K) \geq 0$.

classes in a barycentric visualization. The extreme points are marked with the respective probabilities.

In Algorithm 1, we provide an algorithm to solve the problem of finding the closest point in the convex polytope $\mathcal{Q}_\pi$ to a query distribution $\hat{p}$. In order to proof Theorem 1 that states the optimality of this approach, we will introduce the following three lemmas:

1. Termination
2. Optimal projection
3. Optimal face

Lemma 1 (Termination) *Given a normalized possibility distribution $\pi : \mathcal{Y} \longrightarrow [0, 1]$, Algorithm 1 terminates for an arbitrary probability distribution $\hat{p} \in \mathbb{P}(\mathcal{Y})$.*

Proof In case of $\hat{p} \in \mathcal{Q}_\pi$, we immediately return with a result. For $\hat{p} \notin \mathcal{Q}_\pi$, we fix the probabilities of all $y \in Y'$ in each iteration of Algorithm 1, which are then removed from Y. Thereby, at least for the class $\bar{y} \in Y$ with smallest possibility $\pi(\bar{y})$

$$\bar{p}(\bar{y}) = \left(\pi(\bar{y}) - \sum_{y' \notin Y} p^r(y') \right) \cdot \frac{\hat{p}(\bar{y})}{\hat{p}(\bar{y})} \leq \pi(\bar{y})$$

holds, which does not violate the possibility constraints as in Eq. (12). Here, the set of classes whose probabilities p^r are set is given by $Y' = \{\bar{y}\}$. Consequently, at least one element in Y is removed per step, which eventually results in an empty set Y and the termination of Algorithm 1. ■

In the next lemma, we characterize the optimality of a projection according to the distribution $p^r \in \mathbb{P}(\mathcal{Y})$ as determined in Algorithm 1. In this course, $\bar{Y} \subseteq \mathcal{Y}$ denotes the

set of arbitrary, but already fixed probabilities $p^r(y)$ for $y \in \bar{Y}$. This set shall represent classes with optimal probability scores determined in previous iterations. Without loss of generality, we assume that $\forall y \in \bar{Y} : \pi(y) \leq \min_{y' \in \mathcal{Y} \setminus \bar{Y}} \pi(y')$. Moreover, for a certain possibility degree π_i, let us define the set of classes with at most π_i possibility as

$$Y_{\pi_i} := \{y \in \mathcal{Y} \,|\, \pi(y) \leq \pi_i\} \ . \tag{14}$$

For the remaining classes in $\mathcal{Y} \setminus \bar{Y}$, let p^r be constructed as follows:

$$p^r(y) = \begin{cases} \left(\pi_i - \sum_{y' \in \bar{Y}} p^r(y')\right) \cdot \frac{\hat{p}(y)}{\sum_{y' \in Y_{\pi_i} \setminus \bar{Y}} \hat{p}(y')} & \text{if } y \in Y_{\pi_i} \setminus \bar{Y} \\ (1 - \pi_i) \cdot \frac{\hat{p}(y)}{\sum_{y' \in \mathcal{Y} \setminus Y_{\pi_i}} \hat{p}(y')} & \text{if } y \in \mathcal{Y} \setminus Y_{\pi_i} \end{cases} . \tag{15}$$

Moreover, we use the notion of a *half space* associated with a possibility constraint π_i, which is defined as follows.

Definition 2 (Half-space) *For a possibility constraint π_i of a (normalized) possibility distribution $\pi : \mathcal{Y} \longrightarrow [0, 1]$ with $\max_{y \in \mathcal{Y}} \pi(y) = 1$, we define*

$$\textit{half-space}_{\pi_i} := \left\{ p \in \mathbb{P}(\mathcal{Y}) \,|\, \sum_{y \in \mathcal{Y} : \pi(y) \leq \pi_i} p(y) = \pi_i \right\}$$

as half-space associated with π_i.

Given distributions p^r of the form (15) for a possibility constraint π_i, which is by construction element of half-space$_\pi$ (due to $\forall y \in \bar{Y} : \pi(y) \leq \pi_i$), we can make the following statement about its optimality.

Lemma 3 (Optimal projection) *Given a set $\bar{Y} \subseteq \mathcal{Y}$ of classes with arbitrarily fixed probabilities, a (normalized) possibility distribution $\pi : \mathcal{Y} \longrightarrow [0, 1]$ with $\max_{y \in \mathcal{Y}} \pi(y) = 1$ and the set half-space$_{\pi_i}$, the projection $p^r(y) \in$ half-space$_{\pi_i}$ as defined before is optimal in the sense that $\nexists p \in$ half-space$_{\pi_i}$ with $p(y) = p^r(y)\, \forall y \in \bar{Y}$ for which $\exists y \in Y_{\pi_i} \setminus \bar{Y} : p(y) \neq p^r(y)$ such that $D_{KL}(p \,||\, \hat{p}) < D_{KL}(p^r \,||\, \hat{p})$.*

Proof Let us define $A := \sum_{y \in \bar{Y}} p^r(y)$ as the sum of the (previously) fixed probabilities. From the definition of p^r, we know:

$$
\begin{aligned}
D_{KL}(p^r \,||\, \hat{p}) = & \sum_{y \in \mathcal{Y} \setminus Y_{\pi_i}} \frac{(1-\pi_i)\hat{p}(y)}{\sum_{y' \in \mathcal{Y} \setminus Y_{\pi_i}} \hat{p}(y')} \log \frac{\frac{(1-\pi_i)\hat{p}(y)}{\sum_{y' \in \mathcal{Y} \setminus Y_{\pi_i}} \hat{p}(y')}}{\hat{p}(y)} \\
& + \sum_{y \in Y_{\pi_i} \setminus \bar{Y}} \frac{(\pi_i - A)\hat{p}(y)}{\sum_{y' \in Y_{\pi_i} \setminus \bar{Y}} \hat{p}(y')} \log \frac{\frac{(\pi_i - A)\hat{p}(y)}{\sum_{y' \in Y_{\pi_i} \setminus \bar{Y}} \hat{p}(y')}}{\hat{p}(y)} \\
& + \sum_{y \in \bar{Y}} p^r(y) \log \frac{p^r(y)}{\hat{p}(y)} \\
= & \frac{(1-\pi_i)}{\sum_{y' \in \mathcal{Y} \setminus Y_{\pi_i}} \hat{p}(y')} \left(\sum_{y \in \mathcal{Y} \setminus Y_{\pi_i}} \hat{p}(y) \right) \log \frac{(1-\pi_i)}{\sum_{y' \in \mathcal{Y} \setminus Y_{\pi_i}} \hat{p}(y')} \\
& + \frac{(\pi_i - A)}{\sum_{y' \in Y_{\pi_i} \setminus \bar{Y}} \hat{p}(y')} \left(\sum_{y \in Y_{\pi_i} \setminus \bar{Y}} \hat{p}(y) \right) \log \frac{(\pi_i - A)}{\sum_{y' \in Y_{\pi_i} \setminus \bar{Y}} \hat{p}(y')} \\
& + \sum_{y \in \bar{Y}} p^r(y) \log \frac{p^r(y)}{\hat{p}(y)} \\
= & (1-\pi_i) \log \frac{(1-\pi_i)}{\sum_{y' \in \mathcal{Y} \setminus Y_{\pi_i}} \hat{p}(y')} + (\pi_i - A) \log \frac{(\pi_i - A)}{\sum_{y' \in Y_{\pi_i} \setminus \bar{Y}} \hat{p}(y')} \\
& + \sum_{y \in \bar{Y}} p^r(y) \log \frac{p^r(y)}{\hat{p}(y)}
\end{aligned} \tag{16}
$$

Now, suppose $\exists p \in \text{half-space}_{\pi_i}$ with $p(y) = p^r(y)\, \forall y \in \bar{Y}$ for which $\exists y \in Y_{\pi_i} \setminus \bar{Y}$: $p(y) \neq p^r_{\pi_i}(y)$ such that $D_{KL}(p \,||\, \hat{p}) < D_{KL}(p^r \,||\, \hat{p})$, which would lead to a contradiction of the lemma.

$$
\begin{aligned}
& D_{KL}(p \,||\, \hat{p}) - D_{KL}(p^r \,||\, \hat{p}) \\
&= \sum_{y \in \mathcal{Y} \setminus Y_{\pi_i}} p(y) \log \frac{p(y)}{\hat{p}(y)} + \sum_{y \in Y_{\pi_i} \setminus \bar{Y}} p(y) \log \frac{p(y)}{\hat{p}(y)} + \sum_{y \in \bar{Y}} p(y) \log \frac{p(y)}{\hat{p}(y)} \\
&\quad - \sum_{y \in \mathcal{Y} \setminus Y_{\pi_i}} p^r(y) \log \frac{p^r(y)}{\hat{p}(y)} - \sum_{y \in Y_{\pi_i} \setminus \bar{Y}} p^r(y) \log \frac{p^r(y)}{\hat{p}(y)} - \sum_{y \in \bar{Y}} p^r(y) \log \frac{p^r(y)}{\hat{p}(y)} \\
&= \sum_{y \in \mathcal{Y} \setminus Y_{\pi_i}} p(y) \log \frac{p(y)}{\hat{p}(y)} + \sum_{y \in Y_{\pi_i} \setminus \bar{Y}} p(y) \log \frac{p(y)}{\hat{p}(y)} \\
&\quad - \sum_{y \in \mathcal{Y} \setminus Y_{\pi_i}} p^r(y) \log \frac{p^r(y)}{\hat{p}(y)} - \sum_{y \in Y_{\pi_i} \setminus \bar{Y}} p^r(y) \log \frac{p^r(y)}{\hat{p}(y)} \\
&\geq \left(\sum_{y \in \mathcal{Y} \setminus Y_{\pi_i}} p(y) \right) \log \frac{\left(\sum_{y \in \mathcal{Y} \setminus Y_{\pi_i}} p(y) \right)}{\left(\sum_{y \in \mathcal{Y} \setminus Y_{\pi_i}} \hat{p}(y) \right)} + \left(\sum_{y \in Y_{\pi_i} \setminus \bar{Y}} p(y) \right) \log \frac{\left(\sum_{y \in Y_{\pi_i} \setminus \bar{Y}} p(y) \right)}{\left(\sum_{y \in Y_{\pi_i} \setminus \bar{Y}} \hat{p}(y) \right)} \\
&\quad - \sum_{y \in \mathcal{Y} \setminus Y_{\pi_i}} p^r(y) \log \frac{p^r(y)}{\hat{p}(y)} - \sum_{y \in Y_{\pi_i} \setminus \bar{Y}} p^r(y) \log \frac{p^r(y)}{\hat{p}(y)}
\end{aligned}
$$

Here, the last equation can be derived from the log sum inequality[6]. As we are projecting onto the half space associated with π_i, we can further see that $\sum_{y \in \mathcal{Y} \setminus Y_{\pi_i}} p(y) = 1 - \pi_i$ and $\sum_{y \in Y_{\pi_i} \setminus \bar{Y}} p(y) = \pi_i - A$, which also applies to p^r for the same class subsets.

As a result, together with (16), we get

$$
\begin{aligned}
&= (1 - \pi_i) \log \frac{(1 - \pi_i)}{\left(\sum_{y \in \mathcal{Y} \setminus Y_{\pi_i}} \hat{p}(y) \right)} + (\pi_i - A) \log \frac{(\pi_i - A)}{\left(\sum_{y \in Y_{\pi_i} \setminus \bar{Y}} \hat{p}(y) \right)} \\
&\quad - (1 - \pi_i) \log \frac{(1 - \pi_i)}{\left(\sum_{y \in \mathcal{Y} \setminus Y_{\pi_i}} \hat{p}(y) \right)} - (\pi_i - A) \log \frac{(\pi_i - A)}{\left(\sum_{y \in Y_{\pi_i} \setminus \bar{Y}} \hat{p}(y) \right)} = 0 \not< 0 \ ,
\end{aligned}
$$

which leads to a contradiction. ■

For the next lemma, we introduce the notion of a *face* as follows:

Definition 4 (Face) *For a possibility constraint π_i of a (normalized) possibility distribution $\pi : \mathcal{Y} \longrightarrow [0, 1]$ and distributions $p \in \mathbb{P}(\mathcal{Y})$ with $\sum_{y \in \mathcal{Y} : \pi(y) \leq \pi_i} p(y) = \pi_i$ (i.e., $p \in$ half-space$_{\pi_i}$), we define*

$$
face_{\pi_i} := \left\{ p \,|\, \sum_{k=1}^{j} p_k \leq \pi_j, \quad j \in \{1, ..., i\} \right\}
$$

6. For $a_1, \ldots, a_n, b_1, \ldots, b_n \in \mathbb{R}_+$, the log sum inequality states that $\sum_{i \in [n]} a_i \log \frac{a_i}{b_i} \geq a \log \frac{a}{b}$, whereby $a := \sum_{i \in [n]} a_i$ and $b := \sum_{i \in [n]} b_i$.

as face associated with π_i*.*

Effectively, face_{π_i} considers the subspace of $\text{half-space}_{\pi_i}$ that does not violate any of the possibility constraints $\pi_j \leq \pi_i$. One can readily see that each iteration of Algorithm 1 determines the highest possibility constraint $\pi(y^*)$ whose projection p^r as defined before is element of $\text{face}_{\pi(y^*)}$. Given this fact, we show the optimality of this selected face in the next lemma.

Lemma 5 (Optimal face) *Given a set* $\bar{Y} \subseteq \mathcal{Y}$ *of classes with arbitrarily fixed probabilities, a (normalized) possibility distribution* $\pi : \mathcal{Y} \longrightarrow [0, 1]$ *with* $\max_{y \in \mathcal{Y}} \pi(y) = 1$ *and an arbitrary distribution* $\hat{p} \notin \mathcal{Q}_\pi$*, Algorithm 1 selects* $face_{\pi_i}$ *in each iteration that is optimal in the sense that* $\nexists j \neq i$ *with* $p^* \in \text{argmin}_{p \in face_{\pi_j}} D_{KL}(p \,||\, \hat{p})$*,* $p^*(y) = p^r(y)\, \forall y \in \bar{Y}$ *and* $\exists y \in Y_{\pi_i} \setminus \bar{Y} : p^*(y) \neq p^r(y)$*, such that* $D_{KL}(p^* \,||\, \hat{p}) < D_{KL}(p^r \,||\, \hat{p})$ *for* $p^r \in \text{argmin}_{p \in face_{\pi_i}} D_{KL}(p \,||\, \hat{p})$*.*

Proof Again, we define $A := \sum_{y \in \bar{Y}} p^r(y)$. Now, let us assume $\exists j \neq i$ with the properties described in this lemma, which leads us to a contradiction. To proof it, we can distinguish three cases.

Case 1: $\pi_j > \pi_i$. It is easy to see that a violation with respect to Eq. (12) for possibility $\leq \pi_j$ exists, i.e., the projection constructed as in (15) for π_j is not element of face_{π_j}. In this case, the distribution $p^* \in \text{argmin}_{\text{face}_{\pi_j}} D_{KL}(p \,||\, \hat{p})$ is located on an "edge" of the face, i.e., $\exists m : \sum_{k=1}^{m} p^*_k = \pi_m$.

If $m = i$, then Theorem 3 implies that $p^*(y) = p^r(y)\, \forall y \in Y_{\pi_i}$, which contradicts the assumptions.

In case of $m \neq i$, we can derive the following results using a similar scheme as in the proof of Theorem 3:

$$
\begin{aligned}
& D_{KL}(p^* \,||\, \hat{p}) - D_{KL}(p^r \,||\, \hat{p}) \\
&= \sum_{y \in \mathcal{Y} \setminus Y_{\pi_i}} p^*(y) \log \frac{p^*(y)}{\hat{p}(y)} + \sum_{y \in Y_{\pi_i} \setminus \bar{Y}} p^*(y) \log \frac{p(y)}{\hat{p}(y)} + \sum_{y \in \bar{Y}} p(y) \log \frac{p(y)}{\hat{p}(y)} \\
&\quad - \sum_{y \in \mathcal{Y} \setminus Y_{\pi_i}} p^r(y) \log \frac{p^r(y)}{\hat{p}(y)} - \sum_{y \in Y_{\pi_i} \setminus \bar{Y}} p^r(y) \log \frac{p^r(y)}{\hat{p}(y)} - \sum_{y \in \bar{Y}} p^r(y) \log \frac{p^r(y)}{\hat{p}(y)} \\
&\geq \left(\sum_{y \in \mathcal{Y} \setminus Y_{\pi_i}} p^*(y) \right) \log \frac{\left(\sum_{y \in \mathcal{Y} \setminus Y_{\pi_i}} p^*(y) \right)}{\left(\sum_{y \in \mathcal{Y} \setminus Y_{\pi_i}} \hat{p}(y) \right)} + \left(\sum_{y \in Y_{\pi_i} \setminus \bar{Y}} p^*(y) \right) \log \frac{\left(\sum_{y \in Y_{\pi_i} \setminus \bar{Y}} p^*(y) \right)}{\left(\sum_{y \in Y_{\pi_i} \setminus \bar{Y}} \hat{p}(y) \right)} \\
&\quad - \sum_{y \in \mathcal{Y} \setminus Y_{\pi_i}} p^r(y) \log \frac{p^r(y)}{\hat{p}(y)} - \sum_{y \in Y_{\pi_i} \setminus \bar{Y}} p^r(y) \log \frac{p^r(y)}{\hat{p}(y)}
\end{aligned}
$$

As we know that p^* does not violate any possibility constraints in $Y_{\pi_j} \supset Y_{\pi_i}$ due to $p^* \in \text{face}_{\pi_j}$, but is also not on the same edge as implied by π_i, it holds that

$$\sum_{y \in Y_{\pi_i} \setminus \bar{Y}} p^*(y) < \pi_i - A \ .$$

Moreover, as we know that there is no possibility violation by $p^r(y) \forall \in Y_{\pi_i}$ associated with π (cf. Theorem 3), it must hold $\sum_{Y_{\pi_i} \setminus \bar{Y}} \hat{p}(y) > \pi_i - A$. Otherwise, $\hat{p}$ would be element of $\mathcal{Q}_\pi$.

Altogether, one can readily follow

$$\begin{aligned} &= \underbrace{\left(\sum_{y \in \mathcal{Y} \setminus Y_{\pi_i}} p^*(y) \right)}_{>1-\pi_i} \log \frac{\overbrace{\left(\sum_{y \in \mathcal{Y} \setminus Y_{\pi_i}} p^*(y) \right)}^{>1-\pi_i}}{\left(\sum_{y \in \mathcal{Y} \setminus Y_{\pi_i}} \hat{p}(y) \right)} + \underbrace{\left(\sum_{y \in Y_{\pi_i} \setminus \bar{Y}} p^*(y) \right)}_{<\pi_i - A} \log \frac{\overbrace{\left(\sum_{y \in Y_{\pi_i} \setminus \bar{Y}} p^*(y) \right)}^{<\pi_i - A}}{\left(\sum_{y \in Y_{\pi_i} \setminus \bar{Y}} \hat{p}(y) \right)} \\ &\quad - (1-\pi_i) \log \frac{(1-\pi_i)}{\sum_{y' \in \mathcal{Y} \setminus Y_{\pi_i}} \hat{p}(y')} - (\pi_i - A) \log \frac{(\pi_i - A)}{\sum_{y' \in Y_{\pi_i} \setminus \bar{Y}} \hat{p}(y')} > 0 \not< 0 \ , \end{aligned} \tag{17}$$

leading to a contradiction.

Case 2: $\pi_j = \pi_i$. Theorem 3 implies the optimality of p^r in this case.

Case 3: $\pi_j < \pi_i$. Here, one can again distinguish whether there exists a violation of p^r constructed for π_j or not. In the first case, we can apply exactly the same idea as before by showing that the projection $p^* \in \text{argmin}_{p \in \text{face}_{\pi_j}} D_{KL}(p \,||\, \hat{p})$ is located on an edge that is associated with a π_m with $m < i$.

In case there is no violation, we know that $\sum_{k=1}^{j} p_k^* = \pi_j$. This leads to $\sum_{k=1}^{i} p_k^* \leq \pi_i$. Together with $\sum_{Y_{\pi_i} \setminus \bar{Y}} \hat{p}(y) > \pi_i - A$, one can derive a similar equation as in Eq. (17) to show that

$$D_{KL}(p^* \,||\, \hat{p}) - D_{KL}(p^r \,||\, \hat{p}) \geq 0 \not< 0 \ ,$$

leading again to a contradiction.

■

By combining the previous results, we are ready to proof the optimality of Algorithm 1.

Theorem 6 (Optimality) *Given a credal set $\mathcal{Q}_\pi$ induced by a normalized possibility distribution $\pi : \mathcal{Y} \longrightarrow [0,1]$ with $\max_{y \in \mathcal{Y}} \pi(y) = 1$ according to (1), Algorithm 1 returns the solution of $\mathcal{L}^*(\mathcal{Q}_\pi, \hat{p})$ as defined in (11) for an arbitrary distribution $\hat{p} \in \mathbb{P}(\mathcal{Y})$.*

Proof Combining the three previous lemmas, as well as the fact that the solution of $\operatorname{argmin}_{p \in \mathcal{Q}_\pi} D_{KL}(p \,||\, \hat{p})$ is always characterized by an extreme point on one of the faces of the convex polytope $\mathcal{Q}_\pi$ for $\hat{p} \notin \mathcal{Q}_\pi$, leads to Theorem 6: In each iteration, we choose the optimal projection on the optimal face. Thus, we maintain the optimal probabilities $p^r(y)$ for all $y \notin Y$.

In case of $\hat{p} \in \mathcal{Q}_\pi$, Algorithm 1 returns $D_{KL}(\hat{p} \,||\, \hat{p}) = 0$, which is optimal by definition of D_{KL}. ■

D.2. Complexity

The complexity of Algorithm 1 requires a proper specification of how y^* is determined in the while loop. In our implementation, we sort the classes $y \in \mathcal{Y} = \{y_1, \ldots, y_K\}$ according to their possibilities $\pi(y)$ in a descending manner first, which can be done in $\mathcal{O}(K \log K)$. Then, the while loop iterates over the sorted classes and can continue with the next while-loop until a matching constraint $\pi(y^*)$ could be determined. This violation check involves iterating over all classes $y \in Y$ with $\pi(y) \leq \pi(y^*)$. By sorting the elements in advance, this becomes efficient.

Worst-Case Complexity In the worst-case, every iteration of Algorithm 1 has to iterate over all remaining elements in Y. As said before, checking violations of the possibility constraints requires iterating over all involved classes. Thus, the worst-case complexity can be (loosely) bounded by

$$\sum_{i=0}^{K-1} (i+1)(K-i) = \frac{K^3}{6} + \frac{K^2}{2} + \frac{K}{3} = \mathcal{O}(K^3) \ .$$

Average-Case Complexity Although we are not providing a rigorous analysis of the average-case complexity here, we characterize the efficiency of our algorithm in several cases.

We can observe that the worst case applies whenever the projection of a query distribution $\hat{p} \notin \mathcal{Q}_\pi$ on the convex polytope $\mathcal{Q}_\pi$ is the distribution p^* with $p^*(y_i) = \pi(y_i) - \pi(y_{i-1})$ for all $i \in \{1, \ldots, K-1\}$ and $p^*(y_K) = \pi(y_K)$ for sorted classes y_i according to their possibilities. This is the case when $\hat{p}$ is in the cone associated with this extreme point p^* Škulj (2022), which is given by

$$\left\{ p \notin \mathcal{Q}_\pi \,|\, p^* \in \operatorname*{argmin}_{p' \in \mathcal{Q}_\pi} D_{KL}(p' \,||\, p) \right\} \ .^{7}$$

This set, however, depends on the size of the faces resp. the credal set and is typically rather small. Moreover, it gets proportionally smaller with higher values of K.

In the (trivial) case of $\hat{p} \in \mathcal{Q}_\pi$, we achieve linear complexity $\mathcal{O}(K)$ as we have to check the possibility constraints for all K classes only once. In the other cases, the complexity depends on the face on which we need to project $\hat{p}$: When projecting on face_{π_i}, we do not have to consider any class y with $\pi(y) \leq \pi_i$. Roughly speaking, the larger the faces associated with high possibilities become, the higher the chance of (optimally) projecting directly on

7. More precisely, one would have to distinguish the cases where a p^r projection as in Eq. (15) is perfectly matching the extreme point p^*, but we omit it here for simplicity.

this face and not requiring any loop iterations over classes with smaller possibility values, leading to a sublinear amount of face projections and thus reducing the cubic complexity.

References

Lukas Biewald. Experiment tracking with weights and biases, 2020. URL https://www.wandb.com/. Software available from wandb.com.

Zongsheng Cao, Qianqian Xu, Zhiyong Yang, Xiaochun Cao, and Qingming Huang. Dual quaternion knowledge graph embeddings. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI, Virtual Event, February 2-9*, pages 6894–6902. AAAI Press, 2021.

Leonardo Cella and Ryan Martin. Valid inferential models for prediction in supervised learning problems. In *International Symposium on Imprecise Probability: Theories and Applications, ISIPTA, Granada, Spain, July 6-9*, volume 147 of *Proceedings of Machine Learning Research*, pages 72–82. PMLR, 2021.

Gregory Cohen, Saeed Afshar, Jonathan Tapson, and André van Schaik. EMNIST: an extension of MNIST to handwritten letters. *CoRR*, abs/1702.05373, 2017.

Caglar Demir, Diego Moussallem, Stefan Heindorf, and Axel-Cyrille Ngonga Ngomo. Convolutional hypercomplex embeddings for link prediction. In *Proc. of The 13th Asian Conference on Machine Learning, ACML, virtual, November 17-19*, volume 157 of *Proceedings of Machine Learning Research*, pages 656–671. PMLR, 2021.

Caglar Demir, Julian Lienen, and Axel-Cyrille Ngonga Ngomo. Kronecker decomposition for knowledge graph embeddings. In *Proc. of the 33rd ACM Conference on Hypertext and Social Media, HT, Barcelona, Spain, June 28 - July 1*, pages 1–10. ACM, 2022.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Convolutional 2d knowledge graph embeddings. In *Proc. of the 32nd AAAI Conference on Artificial Intelligence, AAAI, New Orleans, Louisiana, USA, February 2-7*, pages 1811–1818. AAAI Press, 2018.

Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard de Melo, Claudio Gutierrez, José Emilio Labra Gayo, Sabrina Kirrane, Sebastian Neumaier, Axel Polleres, et al. Knowledge graphs. *arXiv preprint arXiv:2003.02320*, 2020.

Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on Challenges in Representation Learning, International Conference on Machine Learning, ICML, Atlanta, GA, USA, June 16-21*, volume 3, 2013.

Lisha Li, Kevin G. Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. Hyperband: A novel bandit-based approach to hyperparameter optimization. *J. Mach. Learn. Res.*, 18:185:1–185:52, 2017.

Julian Lienen and Eyke Hüllermeier. Credal self-supervised learning. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems, NeurIPS, virtual, December 6-14*, pages 14370–14382, 2021.

Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *Proc. of the 5th International Conference on Learning Representations, ICLR, Toulon, France, April 24-26*. OpenReview.net, 2017.

Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104(1):11–33, 2015.

Daniel Ruffinelli, Samuel Broscheit, and Rainer Gemulla. You CAN teach an old dog new tricks! on training knowledge graph embeddings. In *Proc. of the 8th International Conference on Learning Representations, ICLR, Addis Ababa, Ethiopia, April 26-30*. OpenReview.net, 2020.

Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. FixMatch: Simplifying semi-supervised learning with consistency and confidence. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems, NeurIPS, virtual, December 6-12*, 2020.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *Proc. of the 33nd International Conference on Machine Learning, ICML, New York City, NY, USA, June 19-24*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 2071–2080. JMLR.org, 2016.

Colin Wei, Kendrick Shen, Yining Chen, and Tengyu Ma. Theoretical analysis of self-training with deep networks on unlabeled data. In *9th International Conference on Learning Representations, ICLR, Virtual Event, Austria, May 3-7*. OpenReview.net, 2021.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*, 2014.

Damjan Škulj. Normal cones corresponding to credal sets of lower probabilities, 2022.