Knowledge Integration in Deep Clustering
Supplementary material

April 6, 2022

1 Proofs

1.1 Propositions for B formulation

Definition In our main paper, we give the definitions of β and $\mathbf{B}$ as follows:

For each point i, we will define the formulas $\beta_{i1}, ..., \beta_{ik}$, such that β_{ij} interpreted to $\top$ means that the point i is assigned to cluster j. Let $\mathbf{B}$ be a set of logical variables $\{B_{ij} : i \in [1, n], j \in [1, k]\}$. The formula β_{ij} is defined as follows:

$$\begin{aligned} &\beta_{ij} \stackrel{\text{def}}{=} B_{ij} \wedge \textstyle\bigwedge_{t\in[1,j-1]} \neg B_{it} \quad \text{for all } j \in [1, k-1],\\ &\beta_{ik} \stackrel{\text{def}}{=} \textstyle\bigwedge_{t\in[1,k-1]} \neg B_{it} \end{aligned} \tag{1}$$

We define, as follows, the weight w_B for the variables:

$$w_B(B_{ij}) = \begin{cases} \frac{S_{ij}}{1-\sum_{t\in[1,j-1]} S_{it}} & \text{if} \quad \sum_{t\in[1,j-1]} S_{it} < 1 \\ 1 & \text{otherwise} \end{cases} \tag{2}$$

Theorem 1. For any expert constraint c, we have $WMC(c, w_B) = Score(c, S)$.

Proof. From definition of weighted mode count, we have:

$WMC(c, w_B) = \sum_{\mathbf{p}\in\mathbb{P}_c} WMC(\mathbf{p}, w_B) = \sum_{p\in\mathbb{P}_c} WMC(\wedge_{i\in[1,n]}\beta_{ip_i})$

With the definition given in (1), we can observe that for $i \neq i'$, the formulas β_{ip_i} and $\beta_{i'p_{i'}}$ do not share any common variables. Using axiom 6 in [1], we have $WMC(\wedge_{i\in[1,n]}\beta_{ip_i}) = \prod_{i\in[1,n]} WMC(\beta_{ip_i})$.

According to Lemma 2, we have $WMC(\beta_{ipi}, w_B) = S_{ip_i}$. Therefore $WMC(c, w_B) = \sum_{p\in\mathbb{P}_c} \prod_{i\in[1,n]} S_{ip_i} = Score(c, S)$.

Theorem 2. The clustering condition is always satisfied with any instantiation of $\mathbf{B}$.

Proof. The clustering condition states that all points must be assigned to at most one cluster and all points must be assigned to at least one cluster. The proofs are the same for all points i. Therefore, for sake of simplicity we remove the index i. This leads to the definitions as follow.

Let $\mathbf{B}$ be a set of logical variables $\{B_j : j \in [1, k]\}$. The formula β_j is defined as follows:

$$\begin{aligned} &\beta_{j} \stackrel{\text{def}}{=} B_{j} \wedge \textstyle\bigwedge_{t\in[1,j-1]} \neg B_{t} \quad \text{for all } j \in [1, k-1],\\ &\beta_{k} \stackrel{\text{def}}{=} \textstyle\bigwedge_{t\in[1,k-1]} \neg B_{t} \end{aligned} \tag{3}$$

If we call s the row S_i (corresponding to i) in the matrix S, the weight w_B is corresponding to:

$$w_B(B_{j}) = \begin{cases} \frac{s_{j}}{1-\sum_{t\in[1,j-1]} s_{t}} & \text{if} \quad \sum_{t\in[1,j-1]} s_{t} < 1 \\ 1 & \text{otherwise} \end{cases} \tag{4}$$

Under the simplified definition, any point must be assigned to at most one cluster is proven in Lemma 3, and any point must be assigned to at least one cluster is proven in Lemma 4.

1.2 Lemmas and proofs

Lemma 1. For all $j \in [1..k]$, there exists an assignment $\mathbf{b}$ for each B_l such that $\mathbf{b} \models \beta_j$.

Proof. For $j = 1$, $\beta_1 \stackrel{\text{def}}{=} B_1$. Let $\mathbf{b}$ be such that $B_1 = true$. We have $\mathbf{b} \models \beta_1$.

For $2 \leq j \leq k-1$, $\beta_j \stackrel{\text{def}}{=} B_j \wedge \bigwedge_{l<j} \neg B_l$. Let $\mathbf{b}$ be such that $B_j = true$ and $B_l = false$ for all $l < j$. We have $\mathbf{b} \models \beta_j$.

For $j = k$, $\beta_j \stackrel{\text{def}}{=} \bigwedge_{l<k} \neg B_l$. Let $\mathbf{b} = \{\neg B_l : 1 \leq l \leq k-1\}$. We have $\mathbf{b} \models \beta_k$

Lemma 2. The weighted model counting of β_j is equal to s_j.

$$\sum_{\mathbf{b} \models \beta_j} \prod_{B \in \mathbf{b}} w_B(B) \prod_{\neg B \in \mathbf{b}} (1 - w_B(B)) = s_j \tag{5}$$

Proof. We consider two cases: $j \in [1, k-1]$ and $j = k$. For $j \in [1, k-1]$, we denote $\mathbf{b_{prefix}} = \{b_p : p \in [1, j]\}$ and $\mathbf{b_{postfix}} = \{b_p : p \in [j+1, k-1]\}$.

$$\begin{aligned}
&\sum_{\mathbf{b} \models \beta_j} \prod_{B \in \mathbf{b}} w_B(B) \prod_{\neg B \in \mathbf{b}} (1 - w_B(B)) \\
=&\sum_{\mathbf{b_{prefix}} \models \beta_j} \left[\prod_{B \in \mathbf{b_{prefix}}} w_B(B) \prod_{\neg B \in \mathbf{b_{prefix}}} (1 - w_B(B)) \sum_{\mathbf{b_{postfix}} \models \beta_j} \prod_{B \in \mathbf{b_{postfix}}} w_B(B) \prod_{\neg B \in \mathbf{b_{postfix}}} (1 - w_B(B)) \right] \\
=&\sum_{\mathbf{b_{prefix}} \models \beta_j} \left[\prod_{B \in \mathbf{b_{prefix}}} w_B(B) \prod_{\neg B \in \mathbf{b_{prefix}}} (1 - w_B(B)) \sum_{\mathbf{b_{postfix}} \models \top} \prod_{B \in \mathbf{b_{postfix}}} w_B(B) \prod_{\neg B \in \mathbf{b_{postfix}}} (1 - w_B(B)) \right] \\
&\text{(because no variables of } \mathbf{b_{postfix}} \text{ appear in } \beta_j) \\
=&\sum_{\mathbf{b_{prefix}} \models \beta_j} \left[\prod_{B \in \mathbf{b_{prefix}}} w_B(B) \prod_{\neg B \in \mathbf{b_{prefix}}} (1 - w_B(B)) \times 1 \right] \\
&\text{(because } WMC(\top) = 1) \\
=&\sum_{\mathbf{b_{prefix}} = \{(\{\neg B_l : l \in [1, j-1]\}, B_j\})} \left[\prod_{B \in \mathbf{b_{prefix}}} w_B(B) \prod_{\neg B \in \mathbf{b_{prefix}}} (1 - w_B(B)) \right] \\
=&w_B(B_j) \prod_{l \in [1, j-1]} (1 - w_B(B_l)) \\
=&\frac{s_j}{1 - \sum_{t \in [1, j-1]} s_t} \prod_{l \in [1, j-1]} \left(1 - \frac{s_l}{1 - \sum_{t \in [1, l-1]} s_t} \right) \\
=&\frac{s_j}{1 - \sum t \in [1, j-1] s_t} \times (1 - s_1) \times \frac{1 - s_1 - s_2}{1 - s_1} \times \ldots \times \frac{1 - \sum_{t \in [1, j-1]} s_t}{1 - \sum_{t \in [1, j-2]} s_t} \\
=&s_j
\end{aligned} \tag{6}$$

For $j = k$,

$$
\begin{aligned}
&\sum_{\mathbf{b}\models\beta_k}\prod_{B\in\mathbf{b}} w_B(B) \prod_{\neg B\in\mathbf{b}}(1-w_B(B)) \\
=&\prod_{j\in[1,k-1]}(1-w_B(B_j)) \\
=&\prod_{j\in[1,k-1]}\left(1-\frac{s_j}{1-\sum_{t\in[1,j-1]}s_t}\right) \\
=&(1-s_1)\times\frac{1-s_1-s_2}{1-s_1}\times\ldots\times\frac{1-\sum_{t\in[1,k-1]}s_t}{1-\sum_{t\in[1,k-2]}s_t} \\
=&1-\sum_{t\in[1,k-1]}s_t \\
=&s_k
\end{aligned}
\tag{7}
$$

Lemma 3. A point is assigned to at most one cluster. That means that for all $i, j \in [1, k], i \neq j$ we have:

$$\neg\beta_i \vee \neg\beta_j \equiv \top \tag{8}$$

Proof. Without loss of generality, we assume $i < j$. We consider two cases: when $j < k$ and when $j = k$.

Case 1: $i < j < k$, we have $\neg\beta_i \vee \neg\beta_j$

$$
\begin{aligned}
\neg\beta_i \vee \neg\beta_j \equiv\ & \neg(\wedge_{t\in[1,i-1]}\neg B_t \wedge B_i) \vee \neg(\wedge_{t\in[1,j-1]}\neg B_t \wedge B_j) \\
\equiv\ & \vee_{t\in[1,i-1]}B_t \vee \neg B_i \vee_{t\in[1,j-1]} B_t \vee \neg B_j \\
\equiv\ & \vee_{t\in[1,i-1]}B_t \vee \neg B_i \vee B_i \vee_{t\in[i+1,j-1]} B_t \vee \neg B_j \\
\equiv\ & \top
\end{aligned}
$$

Case 2: $i < j and j = k$, we have $\neg\beta_i \vee \neg\beta_k$

$$
\begin{aligned}
\neg\beta_i \vee \neg\beta_k \equiv\ & \neg(\wedge_{t\in[1,i-1]}\neg B_t \wedge B_i) \vee \neg(\wedge_{t\in[1,k-1]}\neg B_t) \\
\equiv\ & \vee_{t\in[1,i-1]}B_t \vee \neg B_i \vee_{t\in[1,k-1]} B_t \\
\equiv\ & \vee_{t\in[1,i-1]}B_t \vee \neg B_i \vee B_i \vee_{t\in[i+1,k-1]} B_t \\
\equiv\ & \top
\end{aligned}
$$

Lemma 4. A point must be assigned to at least one cluster, that means:

$$\vee_{i\in[1,k]}\beta_i \equiv \top \tag{9}$$

Proof. We have:

$$
\begin{aligned}
\vee_{i\in[1,k]}\beta_i \equiv\ & \vee_{i\in[1,k-1]}(\wedge_{t\in[1,i-1]}\neg B_t \wedge B_i) \vee (\wedge_{t\in[1,k-1]}\neg B_t) \\
\equiv\ & (B_1 \vee (\neg B_1 \wedge B_2)) \vee_{i\in[3,k-1]} (\wedge_{t\in[1,i-1]}\neg B_t \wedge B_i) \vee (\wedge_{t\in[1,k-1]}\neg B_t) \\
\equiv\ & (B_1 \vee B_2) \vee_{i\in[3,k-1]} (\wedge_{t\in[1,i-1]}\neg B_t \wedge B_i) \vee (\wedge_{t\in[1,k-1]}\neg B_t) \\
\equiv\ & \ldots \\
\equiv\ & (B_1 \vee B_2 \vee \ldots \vee B_{k-1}) \vee (\wedge_{t\in[1,k-1]}\neg B_t) \\
\equiv\ & (\vee_{t\in[1,k-1]}B_t) \vee \neg(\vee_{t\in[1,k-1]}B_t) \\
\equiv\ & \top
\end{aligned}
$$

2 Experiment results

2.1 Performance of SDAE+Kmeans

In Table 1, 2, and 3, we use the same set of hyperparamters.

In SDAE, IDEC, DCC and IDEC-LK, we use the same neural architecture for the autoencoder. The encoder network is a fully connected multilayer perceptron with dimensions d-500-500-2000-10 for all datasets, where d is the dimension of input data. The decoder network is a mirror of the encoder. All the internal layers are activated by the ReLU [2] function.

The number of epochs for training each layer is 300. The number of epochs for training the whole autoencoder is 500. The optimizer is Stochastic Gradient Descent (SGD) with a momentum of 0.9. The initial learning rate is 0.1 and decreases by one-tenth every 100th epoch. In all the training, the ratio of corruption is 0.2, meaning that 20% of inputs are set to 0.

Table 1: Raw training results on MNIST with SDAE + Kmeans

Data	Run	NMI	ACC
MNIST	0	0.7653	0.8270
MNIST	1	0.7652	0.8290
MNIST	2	0.7554	0.8141
MNIST	3	0.7597	0.8173
MNIST	4	0.7615	0.8198
MNIST	Average	0.7614 ± 0.0037	0.8214 ± 0.0057

Table 2: Raw training results on Fashion with SDAE + Kmeans

Data	Run	NMI	ACC
Fashion	0	0.5842	0.5170
Fashion	1	0.5723	0.5089
Fashion	2	0.5688	0.4979
Fashion	3	0.5885	0.5312
Fashion	4	0.5899	0.5224
Fashion	Average	0.5807 ± 0.0086	0.5155 ± 0.0114

Table 3: Raw training results on Reuters with SDAE + Kmeans

Data	Run	NMI	ACC
Reuters	0	0.5484	0.7371
Reuters	1	0.5222	0.7162
Reuters	2	0.5229	0.7138
Reuters	3	0.5171	0.7626
Reuters	4	0.4473	0.6612
Reuters	Average	0.5116 ± 0.0339	0.7182 ± 0.0335

2.2 Performance of IDEC

The optimizer is Adam with the learning rate of 0.001 [3]. The maximum number of epochs is 200 but it can stop before if the change of cluster assignment compared to the last epoch is less than 0.1%.

In Table 4, we report IDEC results (with five trials as well) using the same pretrained model computed by SDAE. Compared to SDAE+Kmeans, IDEC improved significantly on the MNIST dataset and has modest or no improvement on other datasets.

2.3 Perfornance of SDAE+Kmeans, IDEC and SCAN on CIFAR10

We report the performances of SDAE+Kmeans, IDEC and SCAN with CIFAR10 datset in Table 5.

Table 4: SDAE+IDEC performance on MNIST, Fashion and Reuters

Data	Model	NMI	ACC
MNIST	SDAE+IDEC	0.8668 $\pm$ 0.0005	0.8814 $\pm$ 0.0011
Fashion	SDAE+IDEC	0.5966 $\pm$ 0.0027	0.5183 $\pm$ 0.0033
Reuters	SDAE+IDEC	0.5309 $\pm$ 0.0015	0.7121 $\pm$ 0.0010
CIFAR10	SDAE+IDEC	0.1174 $\pm$ 0.0005	0.2403 $\pm$ 0.0013

Table 5: SDAE+IDEC performance on CIFAR10

Model	NMI	ACC
SDAE+Kmeans	0.1220 $\pm$ 0.0021	0.2459 $\pm$ 0.0041
SDAE+IDEC	0.1174 $\pm$ 0.0005	0.2403 $\pm$ 0.0013
SCAN	68.30	79.39

2.4 Constrained Clustering Results

Pairwise constraints In Table 6, we report Normalized Mutual Information (NMI), Accuracy (ACC), comparison to IDEC (vsIDEC), number of unsatisfied constraints and time running (in seconds). For each dataset and one specific number of constraints, we generate five random sets of constraints (we called them test cases). Then, we run each method once for each test case to measure the average and standard deviation of the metrics mentioned above. In the vsIDEC column, the first number is the p-value of the KS test [4] testing if the NMI of IDEC is similar to the compared method, the second number is the comparison with accuracy. We highlight in bold color when the average value of the constrained clustering method is better than IDEC value.

Table 7 shows how differs each run of IDEC-LK with the same test case (i.e. the same set of constraints). The difference between runs of the same test case is less than the difference between different test cases (different sets of constraints). Overall, IDEC-LK shows a relatively small change between each run.

Table 6: Comparison on clustering quality between the baselines and our IDEC-LK with pairwise constraints. Green and blue number are for the best and second-best values, respectively.

Data	Models	NMI	ACC	Time (s)
MNIST	DCC	0.8691 ± 0.0008	0.8819 ± 0.0011	277 ± 23
MNIST	MPCK-means	0.7296 ± 0.0330	0.7464 ± 0.0399	56.83 ± 4.66
MNIST	PCK-means	0.7241 ± 0.0389	0.7315 ± 0.0825	36.99 ± 10.05
MNIST	IDEC-LK	0.8672 ± 0.0011	0.8805 ± 0.0005	239 ± 3
MNIST	DCC	0.8682 ± 0.0011	0.8817 ± 0.0017	288 ± 8
MNIST	MPCK-means	0.7154 ± 0.0198	0.7356 ± 0.0313	58.53 ± 1.37
MNIST	PCK-means	0.7477 ± 0.0199	0.7743 ± 0.0509	44.31 ± 15.60
MNIST	IDEC-LK	0.8672 ± 0.0012	0.8814 ± 0.0011	263 ± 9
MNIST	DCC	0.8692 ± 0.0012	0.8818 ± 0.0013	273 ± 21
MNIST	MPCK-means	0.7442 ± 0.0310	0.8013 ± 0.0528	60.26 ± 0.37
MNIST	PCK-means	0.7333 ± 0.0261	0.7248 ± 0.0482	28.74 ± 7.66
MNIST	IDEC-LK	0.8683 ± 0.0015	0.8823 ± 0.0009	309 ± 19
MNIST	DCC	0.8689 ± 0.0008	0.8815 ± 0.0007	277 ± 9
MNIST	MPCK-means	0.7589 ± 0.0171	0.7788 ± 0.0413	211 ± 3
MNIST	PCK-means	0.7463 ± 0.0228	0.7698 ± 0.0543	32.97 ± 15.90
MNIST	IDEC-LK	0.8680 ± 0.0017	0.8826 ± 0.0012	388 ± 27
Fashion	DCC	0.5955 ± 0.0018	0.5222 ± 0.0082	191 ± 58
Fashion	MPCK-means	0.5736 ± 0.0201	0.5210 ± 0.0335	59.73 ± 0.42
Fashion	PCK-means	0.5700 ± 0.0212	0.5177 ± 0.0287	36.78 ± 6.84
Fashion	IDEC-LK	0.5956 ± 0.0015	0.5174 ± 0.0030	239 ± 6
Fashion	DCC	0.5945 ± 0.0032	0.5183 ± 0.0037	176 ± 45
Fashion	MPCK-means	0.5747 ± 0.0124	0.5122 ± 0.0403	60.12 ± 1.34
Fashion	PCK-means	0.5756 ± 0.0110	0.5228 ± 0.0067	38.59 ± 11.26
Fashion	IDEC-LK	0.5976 ± 0.0013	0.5210 ± 0.0030	270 ± 23
Fashion	DCC	0.5993 ± 0.0045	0.5216 ± 0.0068	159 ± 17
Fashion	MPCK-means	0.5698 ± 0.0177	0.5451 ± 0.0394	59.12 ± 1.01
Fashion	PCK-means	0.5756 ± 0.0120	0.5231 ± 0.0052	56.82 ± 27.42
Fashion	IDEC-LK	0.5986 ± 0.0010	0.5211 ± 0.0036	301 ± 5
Fashion	DCC	0.6000 ± 0.0019	0.5241 ± 0.0039	140 ± 16
Fashion	MPCK-means	0.5749 ± 0.0138	0.5312 ± 0.0292	205 ± 4
Fashion	PCK-means	0.5714 ± 0.0212	0.5314 ± 0.0293	37.02 ± 13.19
Fashion	IDEC-LK	0.6009 ± 0.0019	0.5230 ± 0.0034	358 ± 17
Reuters	DCC	0.5371 ± 0.0069	0.7162 ± 0.0062	4.05 ± 0.94
Reuters	MPCK-means	0.4824 ± 0.0431	0.7065 ± 0.0329	53.03 ± 0.61
Reuters	PCK-means	0.5015 ± 0.0288	0.7055 ± 0.0284	10.64 ± 2.80
Reuters	IDEC-LK	0.5330 ± 0.0035	0.7128 ± 0.0024	5.45 ± 1.08
Reuters	DCC	0.5450 ± 0.0050	0.7248 ± 0.0039	2.90 ± 0.55
Reuters	MPCK-means	0.5086 ± 0.0357	0.6943 ± 0.0744	53.27 ± 0.62
Reuters	PCK-means	0.5224 ± 0.0218	0.7557 ± 0.0425	14.82 ± 5.72
Reuters	IDEC-LK	0.5337 ± 0.0049	0.7133 ± 0.0028	7.33 ± 2.23
Reuters	DCC	0.5552 ± 0.0037	0.7319 ± 0.0026	3.33 ± 0.44
Reuters	MPCK-means	0.5154 ± 0.0350	0.7483 ± 0.0374	52.45 ± 0.25
Reuters	PCK-means	0.5056 ± 0.0347	0.7152 ± 0.0240	14.68 ± 5.60
Reuters	IDEC-LK	0.5628 ± 0.0055	0.7321 ± 0.0047	10.27 ± 2.91
Reuters	DCC	0.5655 ± 0.0086	0.7477 ± 0.0030	3.46 ± 0.41
Reuters	MPCK-means	0.5262 ± 0.0330	0.7251 ± 0.0412	167 ± 2
Reuters	PCK-means	0.5174 ± 0.0288	0.7343 ± 0.0377	14.80 ± 2.34
Reuters	IDEC-LK	0.5927 ± 0.0105	0.7563 ± 0.0079	27.52 ± 9.88

Table 7: Stability of pairwise IDEC-LK for five different runs each test case

Data	N	Models	NMI	ACC	vsIDEC	#Unsat	Time (s)
MNIST	1000	IDEC-LK	0.8671 ± 0.0009	0.8824 ± 0.0008	**0.87 0.36**	0 ± 0	396 ± 20
Fashion	1000	IDEC-LK	0.6013 ± 0.0008	0.5273 ± 0.0011	**0.08 0.01**	0.6000 ± 0.4899	362 ± 5
Reuters	1000	IDEC-LK	0.5870 ± 0.0024	0.7519 ± 0.0020	**0.01 0.01**	0 ± 0	27.25 ± 3.76

Triplet constraints The performances with triplet constraints are reported similar to those of pairwise constraints in Table 8.

Table 8: Comparison on triplet constraints with DCC and IDEC-LK

Data	N	Models	NMI	ACC	vsIDEC	#Unsat	Time (s)
MNIST	10	DCC	0.8662 ± 0.0003	0.8805 ± 0.0004	0.08 0.87	0 ± 0	133 ± 6
MNIST	10	IDEC-LK	0.8665 ± 0.0012	0.8800 ± 0.0006	0.36 0.36	0 ± 0	253 ± 3
MNIST	100	DCC	0.8659 ± 0.0002	0.8805 ± 0.0011	0.01 0.87	0 ± 0	127 ± 5
MNIST	100	IDEC-LK	0.8666 ± 0.0016	0.8812 ± 0.0011	0.87 1.00	0 ± 0	436 ± 2
MNIST	500	DCC	0.8669 ± 0.0005	0.8817 ± 0.0016	**0.87 1.00**	1.80 ± 0.75	151 ± 11
MNIST	500	IDEC-LK	0.8685 ± 0.0010	0.8815 ± 0.0008	**0.01 0.87**	0 ± 0	1263 ± 2
MNIST	1000	DCC	0.8692 ± 0.0006	0.8855 ± 0.0015	**0.01 0.08**	2.40 ± 1.36	191 ± 21
MNIST	1000	IDEC-LK	0.8682 ± 0.0013	0.8812 ± 0.0011	**0.08** 1.00	0 ± 0	2398 ± 74
Fashion	10	DCC	0.5934 ± 0.0008	0.5119 ± 0.0032	0.08 0.08	0 ± 0	98.69 ± 8.97
Fashion	10	IDEC-LK	0.5965 ± 0.0020	0.5194 ± 0.0040	1.00 **0.87**	0 ± 0	266 ± 5
Fashion	100	DCC	0.5927 ± 0.0013	0.5111 ± 0.0026	0.08 0.08	1.80 ± 1.33	90.90 ± 7.01
Fashion	100	IDEC-LK	0.5969 ± 0.0014	0.5175 ± 0.0021	**1.00** 0.87	0.2000 ± 0.4000	456 ± 3
Fashion	500	DCC	0.5989 ± 0.0020	0.5219 ± 0.0067	**0.36 0.36**	8.80 ± 2.32	141 ± 26
Fashion	500	IDEC-LK	0.5992 ± 0.0015	0.5228 ± 0.0022	**0.08 0.36**	1.20 ± 1.17	1312 ± 1
Fashion	1000	DCC	0.6009 ± 0.0042	0.5370 ± 0.0048	**0.36 0.01**	15.20 ± 4.40	270 ± 38
Fashion	1000	IDEC-LK	0.6003 ± 0.0013	0.5283 ± 0.0065	**0.08 0.08**	4.60 ± 3.38	2413 ± 9
Reuters	10	DCC	0.4687 ± 0.0094	0.4590 ± 0.0165	0.01 0.01	0 ± 0	4.52 ± 0.58
Reuters	10	IDEC-LK	0.5337 ± 0.0034	0.7143 ± 0.0028	**0.08 0.36**	0 ± 0	6.31 ± 1.26
Reuters	100	DCC	0.4839 ± 0.0091	0.4698 ± 0.0066	0.01 0.01	0 ± 0	4.62 ± 0.55
Reuters	100	IDEC-LK	0.5306 ± 0.0043	0.7118 ± 0.0040	0.87 0.87	0 ± 0	15.12 ± 3.90
Reuters	500	DCC	0.4829 ± 0.0087	0.4791 ± 0.0062	0.01 0.01	0.8000 ± 0.7483	5.71 ± 0.74
Reuters	500	IDEC-LK	0.5278 ± 0.0069	0.7099 ± 0.0051	0.87 0.36	0 ± 0	43.06 ± 15.99
Reuters	1000	DCC	0.4936 ± 0.0145	0.4922 ± 0.0211	0.01 0.01	0.6000 ± 0.4899	10.80 ± 1.70
Reuters	1000	IDEC-LK	0.5359 ± 0.0075	0.7142 ± 0.0074	**0.36 0.36**	0.2000 ± 0.4000	102 ± 29

3 Sensitivity experiments for hyper-parameters

The impact of λ_e on the final partition has been measured using IDEC-LK with the Fashion and Reuters dataset. The test scenario has 5 cases, each with 1,000 randomly selected pairwise constraints. For each test scenario, λ_e is tested with $0.01, 0.1, 1.0$ values.

In all cases, when λ_e increases, the average value of WMC and the number of satisfied constraints increase.

For the clustering performance, in the Fashion dataset, the NMI and Accuracy are relatively unchanged. In contrast, the NMI and Accuracy of IDEC-LK achieve the best values when $\lambda_e = 0.1$ for the Reuters dataset.

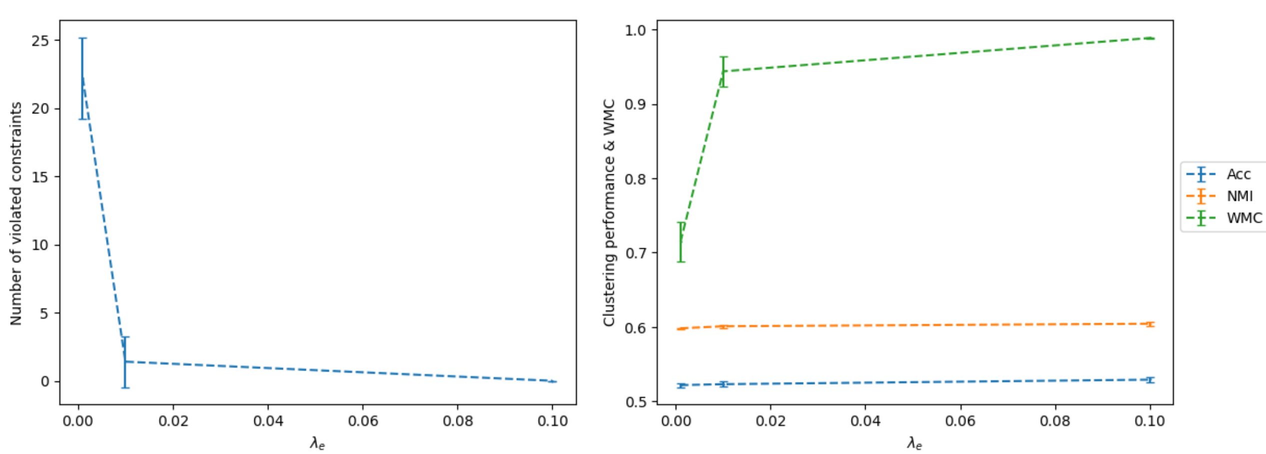

Figure 1: Effect of λ_e on clustering performance (NMI, Acc) and constraint satisfaction (WMC, #violated constraints) with IDEC-LK for Fashion dataset

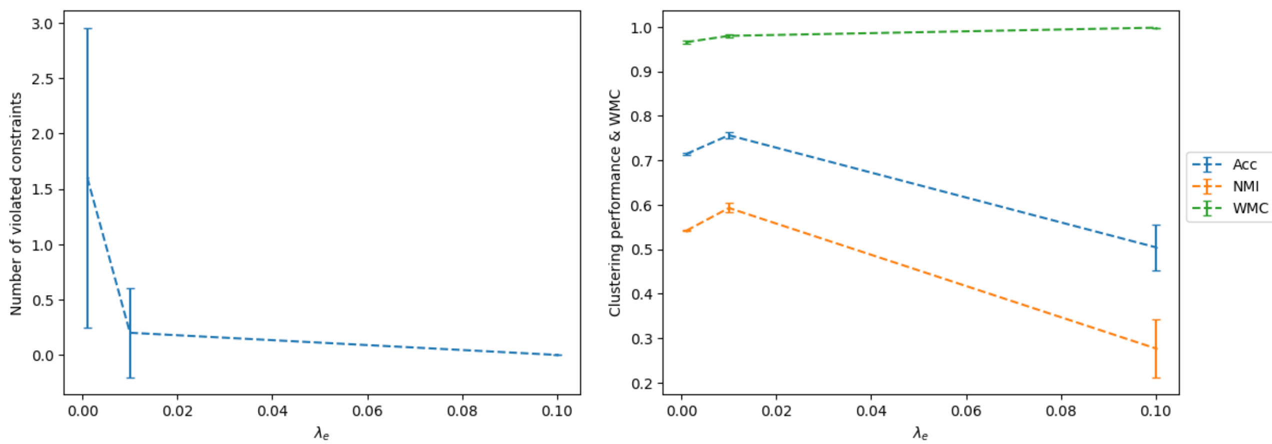

Figure 2: Sensitivity analysis of the λ_e hyperparameter with IDEC-LK for Reuters dataset

Implication constraints Before computing the loss, we need to compile and optimize the SDD structure of each constraint. It is the main bottleneck for learning with more complex knowledge. Table 9 shows the average SDD size and compliation time of our formuation with MNIST and Reuters dataset.

Table 9: Average SDD sizes and compilation times of a Horn clause

Data	Length	SDD size	Time (s)
MNIST	4	415.92 $\pm$ 169.54	0.502 $\pm$ 0.327
Reuters	10	272.40 $\pm$ 79.43	0.131 $\pm$ 0.056

References

[1] Jingyi Xu, Zilu Zhang, Tal Friedman, Yitao Liang, and Guy Broeck. A semantic loss function for deep learning with symbolic knowledge. In *International Conference on Machine Learning*, pages 5502–5511. PMLR, 2018.

[2] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Icml*, 2010.

[3] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[4] John L Hodges. The significance probability of the smirnov two-sample test. *Arkiv för Matematik*, 3(5):469–486, 1958.