

Supplementary Materials: IBMEA

Anonymous Authors

1 SUPPLEMENTARY ANALYSIS

1.1 Hyper-parameters Analysis

To analyze how the information bottleneck principle works, we explore how the Lagrangian multipliers β_m in Eq. (7) of the paper with $m \in \{g, v, a, r\}$ influence the final performances. We conduct detailed experiments on the FB15K-DB15K dataset (20% seed alignments). Take β_g as an example, we maintain constant values for $\beta_{v,a,r}$ parameters, while varying β_g within the range of $[1e-4, 1e-3, 1e-2, 1e-1]$. The overall comparative results are shown in Figure 1. As the values of β_m increase, the evaluation curves tend to increase at first and decline later on. The observed effect is attributed to the fact that a very small β_m value weakens the minimality term’s impact, hindering the exclusion of alignment-irrelevant information. On the other hand, an excessively large β_m overly amplifies the minimality term’s influence. In this way, we verify the necessity of balancing information compression and prediction ability. Generally, the comprehensive results demonstrate the effectiveness of achieving tradeoffs two-fold purposes.

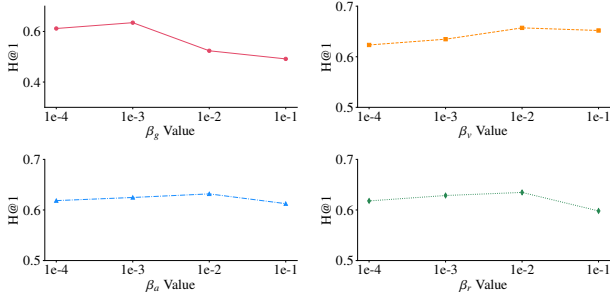


Figure 1: Impact of β in IB regularizers on FB15K-DB15K.

1.2 Efficiency Analysis

To gain further insights into our model, we examine the efficiency behaviors of the three MMEA models on FB15K-DB15K dataset (20% seed alignments). As shown in Table 1, IBMEA training time was approximately 2 to 3 times faster than the others models (MCLEA, Meaformer) with similar parameter amounts, and it produces a considerable advantage in H@1 and MRR metrics. The extended training time of MCLEA may be due to its complex contrastive learning for learning, and MEAFormer may be due to its intricate transformer-based calculation of multi-modal weights. We contribute the enhanced performance of our model to our efficient variational encoder and the information regularizer. Our model efficiently computes likelihood and KL divergence to suppress alignment-irrelevant information while retaining critical alignment-related information. This approach facilitates the generation of more expressive entity representations, leading to a faster convergence rate compared to the other models.

Table 1: Efficiency comparison on FB15K-DB15K dataset.

Methods	Params	Training Time	H@1	MRR
MCLEA	8.9 M	4707 s	.445	.534
MEAformer	10.5 M	5160 s	.578	.661
IBMEA	10.9 M	1797 s	.631	.697

2 SUPPLEMENTARY DEATAILS

2.1 Datasets details

In our experiments, we use two types of multi-modal EA datasets. (1) Cross-KG datasets: we select FB15K-DB15K and FB15K-YAGO15K public datasets, which are deemed as the most typical datasets in multi-modal entity alignment tasks built-in [4]. FB15K is a representative subset extracted from the Freebase knowledge base. Aiming to maintain an approximate entity number of FB15K, DB15K from DBpedia, and YAGO15K from YAGO are mainly selected based on the entities aligned with FB15K. (1) Bilingual datasets: DBP15k is a widely used cross-lingual EA benchmark. It consists of four language-specific knowledge graphs from DBpedia and includes three bilingual entity alignment settings: French-English (FR-EN), Japanese-English (JA-EN), and Chinese-English (ZH-EN). Additionally, DBpedia has released images for the English, French, and Japanese versions. Since Chinese images are not released in DBpedia, EVA [3] extracted them from the raw Chinese Wikipedia dump with the same process as described by Lehmann et al. The details of all multi-modal EA datasets are listed in 2.

Table 2: Statistics of the Datasets (Rel.: Relation, Rel tr.: Relation triple, Attr.: Attribute, Rel tr.: Attribute triple.).

Dataset	KG	#Ent.	#Rel.	#Rel tr.	#Attr.	#Attr tr.	#Image	#EA pairs
FB15K-DB15K	FB15K	14,951	1,345	592,213	116	29,395	13,444	12,846
	DB15K	12,842	279	89,197	225	48,080	12,837	
FB15K-YAGO15K	FB15K	14,951	1,345	592,213	116	29,395	13,444	11,199
	YAGO15K	15,404	32	122,886	7	23,532	11,194	
DBP15K _{ZH-EN}	ZH (Chinese)	19,388	1,701	70,414	8,111	248,035	15,912	15,000
	EN (English)	19,572	1,323	95,142	7,173	343,218	14,125	
DBP15K _{JA-EN}	JA (Japanese)	19,814	1,299	77,214	5,882	248,991	12,739	15,000
	EN (English)	19,780	1,153	93,484	6,066	320,616	13,741	
DBP15K _{FR-EN}	FR (French)	19,661	903	105,998	4,547	273,825	14,174	15,000
	EN (English)	19,993	1,208	115,722	6,422	351,094	13,858	

2.2 Metric Details

To evaluate our IBMEA approach, we adopt the classical rank-based evaluation protocol of knowledge graph entity alignment. The following metrics are used:

- **Hits@N**: Hits@N is the proportion of true aligned entities that appear in the first N entities of the sorted rank list. Hits@N can be defined as

$$\text{Hits@N} = \frac{1}{|S|} \sum_{q \in S} \mathbb{I}[\text{rank}(i) \leq N], \quad (1)$$

where \mathcal{S} is the number of all testing alignment sets, rank_i refers to the rank position of the first correct mapping for the i -th query entities, and $\mathbb{I}[\text{rank}(i) \leq N]$ yields 1 if i is ranked between 1 and \mathcal{S} , 0 otherwise. This metric is bounded in the $[0, 1]$ range and its values increase with \mathcal{S} , where the higher the better. Note that, Hits@1 should be preferable, and it is equivalent to precision widely-used in conventional entity alignment.

- **MRR**: Mean reciprocal rank (MRR) measures the number of aligned entity pairs predicted correctly. MRR is the average of the reciprocal ranks of results for a sample of candidate alignment entities:

$$\text{Hits@N} = \frac{1}{|\mathcal{S}|} \sum_{q \in \mathcal{S}} \frac{1}{\text{rank}(i)}, \quad (2)$$

MRR is a useful metric because it not only considers if the EA algorithm correctly aligns entities, but also the rank of the first correctly aligned entity. This means that MRR penalizes lower ranks more severely than higher ones, which is often more reflective of real-world performance. Higher MRR values indicate better performance, with 1 being the maximum achievable value.

2.3 Implementation details

We report our best hyper-parameter settings across two MMKGs datasets and hyper-parameter search space in Table 3. It's noteworthy that all hyperparameter configurations were carefully tuned using a 10-trial grid search technique. Instead of always choosing the best-performing model, we balance the memory limit and model performance. We train and evaluate all our models on a machine with the specifications listed in Table 4.

Table 3: Best hyper-parameter settings of model and the search space for hyper-parameters used.

Hyper-parameters	Best setting	Search space
Batch size	7500	1000, 1500, 3500, 7500, 10000
Train epoch	1000	500, 1000, 1500, 2000
Learning rate	5e-3	3e-4, 6e-4, 3e-3, 6e-3, 3e-2
Weight Decay	1e-2	1e-3, 5e-3, 1e-2, 5e-2
Random Dropouts Rate	0.45	0.25, 0.35, 0.45, 0.55
GAT input hidden dimension	300	200, 300, 400, 500
Graph feature size	300	100, 200, 300, 400, 500
Visual feature size	100	100, 200, 300, 400, 500
Attribute feature size	100	100, 200, 300, 400, 500
Relation feature size	100	100, 200, 300, 400, 500
β_g	1e-3	1e-4, 1e-3, 1e-2, 1e-1
β_o	1e-2	1e-4, 1e-3, 1e-2, 1e-1
β_a	1e-2	1e-4, 1e-3, 1e-2, 1e-1
β_r	1e-2	1e-4, 1e-3, 1e-2, 1e-1

Table 4: Hardware specifications of the used machine.

hardware	specification
RAM	251 GB
CPU	Intel(R) Xeon(R) Silver 4110 CPU @ 2.10GHz
GPU	NVIDIA(R) A100(80GB) x 4

3 SUPPLEMENTARY THEORY

3.1 Mutual Information

Mutual information (MI) measures the amount of information obtained about one random variable after observing another random variable. Formally given two random variables x and y with joint distribution $p(x, y)$ and marginal densities $p(x)$ and $p(y)$ their MI is defined as the KL-divergence between the joint density and the product of their marginal densities

$$\begin{aligned} I(x; y) &= I(y; x) \\ &= KL(p(x, y) || p(x)p(y)) \\ &= \mathbb{E}_{(x, y) \sim p(x, y)} \left[\log \frac{p(x, y)}{p(x)p(y)} \right] \\ &= \int dx dy p(x, y) \log \frac{p(x, y)}{p(x)p(y)} \end{aligned} \quad (3)$$

3.2 Information Bottleneck

Information Bottleneck (IB) regards supervised learning as a representation learning problem, seeking a stochastic map from input data x to some latent representation z that can still be used to predict the labels y , under a constraint on its total complexity. The joint distribution $p(x, y, z)$ can be factorised as follows:

$$p(x, y, z) = p(z | x, y) p(y | x) p(x) = p(z | x) p(y | x) p(x), \quad (4)$$

which corresponds to the following Markov Chain

$$y \rightarrow x \rightarrow z. \quad (5)$$

The goal is to learn an encoding that is maximally informative about the target y measured by $I(y; z)$. While a straightforward approach would be to opt for the identity encoding $x = z$, such a solution lacks practical utility. To address this, we introduce a constraint aiming to balance informativeness and usefulness. The optimization problem becomes:

$$\begin{aligned} \max \quad & I(y; z) \\ \text{subject to} \quad & I(x; z) \leq I_c, \end{aligned} \quad (6)$$

where I_c is the information constraint. The Lagrangian of the above-constrained optimization problem, which we would like to **maximize** is

$$\begin{aligned} L_{IB} &= \beta(I(x; z) - I_c) - I(y; z) \\ &= \beta I(x; z) - I(y; z), \end{aligned} \quad (7)$$

where $\beta \geq 0$ is a Lagrange multiplier. Intuitively, the first term encourages z to be predictive of y , whilst the second term encourages z to "forget" x . In essence, IB principle explicitly enforces the learned representation z retains only the relevant information from x necessary for predicting y , effectively capturing the minimal sufficient statistics of x for y . **Eq.(7) here corresponds to Eq. (1) in the body of the paper.**

3.3 Variational Information Bottleneck

To optimize the objective function in Eq. (7), leverage the approach introduced in the VIB [1] framework. Focusing on the first term in

Eq. (7), we can write out the term as:

$$\begin{aligned} I(x; z) &= \int dx dz p(x, z) \log \frac{p(x, z)}{p(x)p(z)} \\ &= \int dx dz p(x, z) \log \frac{p(z | x)}{p(z)}. \end{aligned} \quad (8)$$

Introducing a variational approximation $q(z)$ to the marginal distribution $p(z)$, we use the Kullback-Leibler (KL) divergence to derive an upper bound on $I(x; z)$:

$$\begin{aligned} KL(p(z) || q(z)) &\geq 0 \implies \int dz p(z) \log p(z) \\ &\geq \int dz p(z) \log q(z), \end{aligned} \quad (9)$$

which yields:

$$\begin{aligned} I(x; z) &= \int dx dz p(x, z) \log p(z | x) - \int dz p(z) \log p(z) \\ &\leq \int dx dz p(x, z) \log p(z | x) - \int dz p(z) \log q(z) \\ &= \int dx dz p(x) p(z | x) \log \frac{p(z | x)}{q(z)}, \end{aligned} \quad (10)$$

For the second term, $I(y; z)$, it is expressed as:

$$\begin{aligned} I(y; z) &= \int dy dz p(y, z) \log \frac{p(y, z)}{p(y)p(z)} \\ &= \int dy dz p(y, z) \log \frac{p(y | z)}{p(y)}, \end{aligned} \quad (11)$$

where $p(y | z)$ is defined as:

$$\begin{aligned} p(y | z) &= \int dx \frac{p(x, y, z)}{p(z)} \\ &= \int dx \frac{p(z | x)p(y | x)p(x)}{p(z)}, \end{aligned} \quad (12)$$

which is intractable. Introducing a variational approximation $q(y | z)$ to $p(y | z)$, we utilize the KL divergence to obtain a lower bound on $I(y; z)$:

$$\begin{aligned} KL(p(y | z) || q(y | z)) &\geq 0 \implies \int dy p(y | z) \log p(y | z) \\ &\geq \int dy p(y | z) \log q(y | z). \end{aligned} \quad (13)$$

Thus, we have:

$$\begin{aligned} I(y; z) &= \int dy dz p(y, z) \log p(y | z) - \int dy p(y) \log p(y) \\ &\geq \int dy dz p(y, z) \log q(y | z) - \int dy p(y) \log p(y) \\ &= \int dx dy dz p(z | x)p(y | x)p(x) \log q(y | z), \end{aligned} \quad (14)$$

where the entropy of the labels $H(y) = - \int dy p(y) \log p(y)$ is independent of our optimization and thus can be disregarded.

Combining the above two bounds, the Lagrangian to **minimize** is expressed as:

$$\begin{aligned} L_{IB} &= \beta I(x; z) - I(y; z) \\ &\leq \beta \int dx dz p(x) p(z | x) \log \frac{p(z | x)}{q(z)} \\ &\quad - \int dx dy dz p(z | x)p(y | x)p(x) \log q(y | z) \\ &= \beta \int dx dy dz p(z | x)p(x, y) KL(p(z | x) || q(z)) \\ &\quad - \int dx dy dz p(z | x)p(y, x) \log q(y | z) \\ &= \mathbb{E}_{(x, y) \sim p(x, y), z \sim p(z | x)} \left[\beta KL(p(z | x) || q(z)) - \log q(y | z) \right] \\ &= J_{IB}. \end{aligned} \quad (15)$$

To compute the upper bound practically, we make certain assumptions. We approximate $p(x, y)$ using the empirical data distribution $p(x, y) = \frac{1}{n} \sum^n i = 1 \delta x_i(x) \delta y_i(y)$, resulting in:

$$\begin{aligned} J_{IB} &= \beta \int dx dz p(x) p(z | x) \log \frac{p(z | x)}{q(z)} \\ &\quad - \int dx dy dz p(z | x)p(y | x)p(x) \log q(y | z) \\ &\approx \frac{1}{n} \sum_{i=1}^n \left[\beta \int dz p(z | x_i) \log \frac{p(z | x_i)}{q(z)} \right. \\ &\quad \left. - \int dz p(z | x_i) \log q(y_i | z) \right] \\ &= \frac{1}{n} \sum_{i=1}^n \left[\beta KL(p(z | x_i) || q(z)) - \int dz p(z | x_i) \log q(y_i | z) \right] \end{aligned} \quad (16)$$

By utilizing an encoder parameterized as multivariate Gaussian:

$$p_\phi(z | x) = \mathcal{N}(z; \mu_\phi(x), \Sigma_\phi(x)), \quad (17)$$

then we can use the reparameterization trick such that $z = g_\phi(\epsilon, x)$, which is a deterministic function of x and the Gaussian random variable $\epsilon \sim p(\epsilon) = \mathcal{N}(0, I)$. Consequently, the ultimate objective to minimize becomes:

$$J_{IB} = \frac{1}{n} \sum_{i=1}^n \left[\beta KL(p(z | x_i) || q(z)) - \mathbb{E}_{\epsilon \sim p(\epsilon)} \left[\log q(y_i | g_\phi(\epsilon, x_i)) \right] \right], \quad (18)$$

where $p_\phi(z | x)$ represents the encoder, parameterized as a multivariate Gaussian. **Eq. (18) here corresponds to Eq. (2) in the body of the paper.** The decoder $q_\theta(y | z)$ is parameterized as independent Bernoulli distributions for each element y_j of y (in the case of binary data):

$$q_\theta(y_j | z) = \text{Ber}(\mu_\theta(z)), \quad (19)$$

and the approximated latent marginal $q(z)$ is typically fixed to a standard normal distribution:

$$q_\theta(z) = \mathcal{N}(z; 0, I_k), \quad (20)$$

By using the specified decoder parameterization $q_\theta(y | z)$ as independent Bernoulli distributions, the expression for the negative logarithm of $q_\theta(y | z)$ simplifies to:

$$-\log q_\theta(y | z) = -\left[y \log \hat{y} + (1 - y) \log(1 - \hat{y})\right], \quad (21)$$

which is commonly referred to as the Binary Cross Entropy loss.

We calculate $I(Z_m; X_m)$, $I(Z_m^{(1)}, Z_m^{(2)}; Y)$ in eq. (7) of the paper according to eq. (17) and eq. (21).

REFERENCES

- [1] Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. 2017. Deep Variational Information Bottleneck. In *Proceedings of ICLR*.
- [2] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Christian Bizer. 2015. DBpedia - A large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic Web* 6, 2 (2015), 167–195.
- [3] Fangyu Liu, Muhao Chen, Dan Roth, and Nigel Collier. 2021. Visual pivoting for (unsupervised) entity alignment. In *Proceedings of AAAI*, Vol. 35. 4257–4266.
- [4] Ye Liu, Hui Li, Alberto García-Durán, Mathias Niepert, Daniel Oñoro-Rubio, and David S. Rosenblum. 2019. MMKG: Multi-modal Knowledge Graphs. In *Proceedings of ESWC*, Vol. 11503. Springer, 459–474.