A Theoretical Proof

In this section, we provide theoretical proofs of the derivation of information constraints in the main text.

A.1 Derivation Proof of $I(\boldsymbol{Z};\boldsymbol{Y})$

We devise predictable terms $I(\boldsymbol{Z};\boldsymbol{Y})$ to identify the truth of a given triple, and thus endow the model with the predictability of new triples. For simplicity, we omit the KG subscript and abbreviate $I(\boldsymbol{Z}_i;\boldsymbol{Y}_i)$, $I(\boldsymbol{Z}_j;\boldsymbol{Y}_i)$, and $I(\bar{\boldsymbol{Z}};\boldsymbol{Y}_i)$ as $I(\boldsymbol{Z};\boldsymbol{Y})$. The difference of them is to use the variables $\boldsymbol{Z}_i$, $\boldsymbol{Z}_j$, and $\bar{\boldsymbol{Z}}$ learned from different KGs.

Following the definition of mutual information, the predictability term $I(\boldsymbol{Z};\boldsymbol{Y})$ can be defined as:

$$\begin{aligned}
&I(\boldsymbol{Z};\boldsymbol{Y}) = \int p(\boldsymbol{z},\boldsymbol{y})\log\frac{p(\boldsymbol{z},\boldsymbol{y})}{p(\boldsymbol{z})p(\boldsymbol{y})}d\boldsymbol{y}d\boldsymbol{z}\\
&= \int p(\boldsymbol{z},\boldsymbol{y})\log\frac{p(\boldsymbol{z},\boldsymbol{y})}{p(\boldsymbol{z})}d\boldsymbol{y}d\boldsymbol{z} - \int p(\boldsymbol{z},\boldsymbol{y})\log p(\boldsymbol{y})d\boldsymbol{y}d\boldsymbol{z}\\
&= \int p(\boldsymbol{z},\boldsymbol{y})\log p(\boldsymbol{y}|\boldsymbol{z})d\boldsymbol{y}d\boldsymbol{z} - \int p(\boldsymbol{y})\log p(\boldsymbol{y})d\boldsymbol{y}.
\end{aligned} \tag{1}$$

Let $q_\phi(\boldsymbol{y}|\boldsymbol{z})$ be the variational approximation of $p(\boldsymbol{y}|\boldsymbol{z})$ with the decoder ϕ. Considering the Kullback-Leibler (KL) divergence, we have

$$\begin{aligned}
&KL(p(\boldsymbol{y}|\boldsymbol{z})||q(\boldsymbol{y}|\boldsymbol{z})) \geq 0 \Rightarrow \int p(\boldsymbol{y}|\boldsymbol{z})\log\frac{p(\boldsymbol{y}|\boldsymbol{z})}{q(\boldsymbol{y}|\boldsymbol{z})}d\boldsymbol{y} \geq 0\\
&\Rightarrow \int p(\boldsymbol{y}|\boldsymbol{z})\log p(\boldsymbol{y}|\boldsymbol{z})d\boldsymbol{y} \geq \int p(\boldsymbol{y}|\boldsymbol{z})\log q(\boldsymbol{y}|\boldsymbol{z})d\boldsymbol{y}.
\end{aligned} \tag{2}$$

Hence, the lower bound of $I(\boldsymbol{Z};\boldsymbol{Y})$ can be written as:

$$\begin{aligned}
&I(\boldsymbol{Z};\boldsymbol{Y})\\
&\geq \int p(\boldsymbol{z},\boldsymbol{y})\log q(\boldsymbol{y}|\boldsymbol{z})d\boldsymbol{y}d\boldsymbol{z} - \int p(\boldsymbol{y})\log p(\boldsymbol{y})d\boldsymbol{y}\\
&\propto \int p(\boldsymbol{z},\boldsymbol{y})\log q(\boldsymbol{y}|\boldsymbol{z})d\boldsymbol{y}d\boldsymbol{z}\\
&= \int p(\boldsymbol{z},\boldsymbol{y},\boldsymbol{x})\log q(\boldsymbol{y}|\boldsymbol{z})d\boldsymbol{x}d\boldsymbol{y}d\boldsymbol{z}\\
&= \int p(\boldsymbol{z}|\boldsymbol{x})p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x})\log q(\boldsymbol{y}|\boldsymbol{z})d\boldsymbol{x}d\boldsymbol{y}d\boldsymbol{z}\\
&= \int p(\boldsymbol{z}|\boldsymbol{x})p(\boldsymbol{x},\boldsymbol{y})\log q(\boldsymbol{y}|\boldsymbol{z})d\boldsymbol{x}d\boldsymbol{y}d\boldsymbol{z}\\
&= \mathbb{E}_{\boldsymbol{x},\boldsymbol{y}\sim p(\boldsymbol{x},\boldsymbol{y}),\boldsymbol{z}\sim p_\theta(\boldsymbol{z}|\boldsymbol{x})}[\log q_\phi(\boldsymbol{y}|\boldsymbol{z})],
\end{aligned} \tag{3}$$

where $p_\theta(\boldsymbol{z}|\boldsymbol{x})$ is achieved by the variational encoder θ. We omit $\int p(\boldsymbol{y})\log p(\boldsymbol{y})d\boldsymbol{y}$ since the entropy of labels is independent of our optimization. Therefore, the lower bound of $I(\boldsymbol{Z};\boldsymbol{Y})$ is equal to the log-likelihood of task prediction.

To achieve this log-likelihood, we would like to use margin loss for approximation. To prove this approximation, we first rewrite $\boldsymbol{z}$ in relational triples as $\Gamma = (\boldsymbol{z}_h, \boldsymbol{r}, \boldsymbol{z}_t)$. Here, we use Γ^+ and Γ^- to denote positive and negative triples, respectively, where the negative triples can be obtained by replacing the tail entity with other false entities. Let $\boldsymbol{y} \in \{0,1\}$ indicate the truth of the triple. The expectation to maximize can be derived as

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{x},\boldsymbol{y}\sim p(\boldsymbol{x},\boldsymbol{y}),(\boldsymbol{z}_h,\boldsymbol{z}_t)\sim p_\theta(\boldsymbol{z}|\boldsymbol{x})}[\log q_\phi(\boldsymbol{y}|\Gamma)]\\
&= \sum_{\Gamma^+\in\mathcal{T},\Gamma^-\notin\mathcal{T}} \log p(y=1|\Gamma^+) + \log p(y=0|\Gamma^-)\\
&= \sum_{\Gamma^+\in\mathcal{T},\Gamma^-\notin\mathcal{T}} \log\sigma(-s_\phi(\Gamma^+)) + \log[1-\sigma(-s_\phi\Gamma^-))]\\
&= \sum_{\Gamma^+\in\mathcal{T},\Gamma^-\notin\mathcal{T}} \log\sigma(-s_\phi(\Gamma^+)) + \log\sigma(s_\phi(\Gamma^-))],
\end{aligned} \tag{4}$$

where σ is sigmoid function for binary classification, s_ϕ is the function to measure the plausibility of the given triple Γ, and we use the 2-norm distance in TransE (Bordes et al. 2013) to achieve it, where the lower s_ϕ is, the higher plausibility will be. In fact, maximizing this objective implies an unlimited decrease in $s_\phi(\Gamma^+)$ and an unlimited increase in $s_\phi(\Gamma^-)$. However, we hope that the margin between $s_\phi(\Gamma^+)$ and $s_\phi(\Gamma^-)$ is not as large as possible, according to the open-world assumption (Bordes et al. 2013) of incomplete KGs. Therefore, we set a maximal margin λ, and hope that

$$P(s_\phi(\Gamma^+) - s_\phi(\Gamma^-) < \lambda) \to 1. \tag{5}$$

To achieve this, we set a negative log-likelihood as the optimization objective, denoted as $l(a) = -\log\sigma(a)$, where $a = s_\phi(\Gamma^+) - s_\phi(\Gamma^-) - \lambda$. Then, there exists:

$$\begin{aligned}
l(a) &= \log(1+e^{-a})\\
&\approx \max\{-a, 0\}, \text{when } |a| \text{ is large.}
\end{aligned} \tag{6}$$

Here, the approximation holds since we can see that

- when $a \to +\infty$, $e^{-a} \to 0$, $l(a) \to 0$,
- when $a \to -\infty$, $l(a) \approx -a$.

In addition, Eq. (6) can also be understood directly as a LogSumExp (LSE) function, which is a smooth approximation of the maximum function.

In summary, we can maximize the following formula as the objective of $I(\boldsymbol{Z};\boldsymbol{Y})$:

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{x},\boldsymbol{y}\sim p(\boldsymbol{x},\boldsymbol{y}),(\boldsymbol{z}_h,\boldsymbol{z}_t)\sim p_\theta(\boldsymbol{z}|\boldsymbol{x})}[\log q_\phi(\boldsymbol{y}|\Gamma)]\\
&= \sum_{\Gamma^+\in\mathcal{T},\Gamma^-\notin\mathcal{T}} l(s_\phi(\Gamma^+) - s_\phi(\Gamma^-) - \lambda)\\
&= \sum_{\Gamma^+\in\mathcal{T},\Gamma^-\notin\mathcal{T}} -\log\sigma(s_\phi(\Gamma^+) - s_\phi(\Gamma^-) - \lambda)\\
&\approx \sum_{\Gamma^+\in\mathcal{T},\Gamma^-\notin\mathcal{T}} \max\{s_\phi(\Gamma^-) - s_\phi(\Gamma^+) + \lambda, 0\}.
\end{aligned} \tag{7}$$

Here, this formula is equivalent to the formula of Eq. (9) in the main text.

A.2 Derivation Proof of $I(\boldsymbol{Z};\boldsymbol{X})$

We devise minimality terms to compress redundant information in entity representations $\boldsymbol{Z}$. Following the definition of mutual information, the minimality term $I(\boldsymbol{Z};\boldsymbol{X})$ can be defined as:

$$
\begin{aligned}
& I(\boldsymbol{Z};\boldsymbol{X}) = \int p(\boldsymbol{x},\boldsymbol{z}) \log \frac{p(\boldsymbol{x},\boldsymbol{z})}{p(\boldsymbol{x})p(\boldsymbol{z})} d\boldsymbol{x}d\boldsymbol{z} \\
&= \int p(\boldsymbol{x},\boldsymbol{z}) \log \frac{p(\boldsymbol{z}|\boldsymbol{x})}{p(\boldsymbol{z})} d\boldsymbol{x}d\boldsymbol{z} \\
&= \int p(\boldsymbol{x},\boldsymbol{z}) \log p(\boldsymbol{z}|\boldsymbol{x}) d\boldsymbol{x}d\boldsymbol{z} - \int p(\boldsymbol{z}) \log p(\boldsymbol{z}) d\boldsymbol{z}.
\end{aligned}
\tag{8}
$$

Afterward, we also use the KL divergence as:

$$
\begin{aligned}
& KL(p(\boldsymbol{z})||q(\boldsymbol{z})) \geq 0 \\
& \Rightarrow \int p(\boldsymbol{z}) \log p(\boldsymbol{z}) dz \geq \int p(\boldsymbol{z}) \log q(\boldsymbol{z}) dz.
\end{aligned}
\tag{9}
$$

Hence, the upper bound of $I(\boldsymbol{Z};\boldsymbol{X})$ can be written as:

$$
\begin{aligned}
& I(\boldsymbol{Z};\boldsymbol{X}) \\
&\leq \int p(\boldsymbol{x},\boldsymbol{z}) \log p(\boldsymbol{z}|\boldsymbol{x}) d\boldsymbol{x}d\boldsymbol{z} - \int p(\boldsymbol{z}) \log q(\boldsymbol{z}) d\boldsymbol{z} \\
&= \int p(\boldsymbol{x},\boldsymbol{z}) \log p(\boldsymbol{z}|\boldsymbol{x}) d\boldsymbol{x}d\boldsymbol{z} - \int p(\boldsymbol{x},\boldsymbol{z}) \log q(\boldsymbol{z}) d\boldsymbol{x}d\boldsymbol{z} \\
&= \int p(\boldsymbol{x},\boldsymbol{z}) \log \frac{p(\boldsymbol{z}|\boldsymbol{x})}{q(\boldsymbol{z})} d\boldsymbol{x}d\boldsymbol{z} \\
&= \int p(\boldsymbol{x})p(\boldsymbol{z}|\boldsymbol{x}) \log \frac{p(\boldsymbol{z}|\boldsymbol{x})}{q(\boldsymbol{z})} d\boldsymbol{x}d\boldsymbol{z} \\
&= \int p(\boldsymbol{x})p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{z}|\boldsymbol{x}) \log \frac{p(\boldsymbol{z}|\boldsymbol{x})}{q(\boldsymbol{z})} d\boldsymbol{x}d\boldsymbol{y}d\boldsymbol{z} \\
&= \int p(\boldsymbol{x},\boldsymbol{y}) \mathbb{D}_{KL}(p_\theta(\boldsymbol{z}|\boldsymbol{x})||q(\boldsymbol{z})) d\boldsymbol{x}d\boldsymbol{y} \\
&= \mathbb{E}_{\boldsymbol{x},\boldsymbol{y}\sim p(\boldsymbol{x},\boldsymbol{y})} \mathbb{D}_{KL}(p_\theta(\boldsymbol{z}|\boldsymbol{x})||q(\boldsymbol{z})),
\end{aligned}
\tag{10}
$$

where $p_\theta(\boldsymbol{z}|\boldsymbol{x})$ is the variational encoder to approximate the true posterior $p(\boldsymbol{z}|\boldsymbol{x})$. The prior distribution $q(\boldsymbol{z})$ is assumed as a standard Gaussian $\mathcal{N}(0, \mathrm{diag}(\boldsymbol{I}))$ following studies (Kingma and Welling 2014; Alemi et al. 2017). Note that this actually regularizes the representations with limited total amount of information. Since task-relevant information is required to be preserved in predicable terms, redundant information is naturally surpassed in the limited total information. Here, this formula is exactly the same as the formula of Eq. (10) in the main text.

B Experimental Details

In this section, we provide details of data statistics, competitors, evaluation protocol, and optimal hyper-parameters.

B.1 Data Statistics

For evaluation, we adopt three benchmarks with 14 KGs in our experiments: two multilingual datasets DBP-5L (Chen et al. 2017), E-PKG (Huang et al. 2022), and a constructed multi-domain dataset DWY (Sun et al. 2018).

- The **DBP-5L** dataset consists of 5 KGs extracted from DBpedia constructed in Greek (EL), English (EN), Spanish (ES), French (FR) and Japanese (JA).

Dataset	KG	# Ent.	# Rel.	# Tra.	# Val.	# Tes.
DBP-5L	EL	5,231	111	8,670	4,152	1,017
	EN	13,996	831	48,652	24,051	7,464
	ES	12,382	144	33,036	16,220	4,810
	FR	13,176	178	30,139	14,705	4,171
	JA	11,805	128	17,979	8,633	2,162
E-PKG	DE	17,223	21	45,515	22,753	7,602
	EN	16,544	21	60,310	39,150	10,071
	ES	9,595	21	18,090	9,039	3,034
	FR	17,068	21	47,999	23,994	8022
	IT	15,670	21	42,767	21,377	7,148
	JA	2,642	21	10,013	5,002	1,688
DYW	DB	23,315	180	85,506	10,688	10,780
	YG	13,864	27	91,179	11,398	11,410
	WK	17,743	90	82,047	10,255	10,296

Table 1: The statistics of DBP-5L (Chen et al. 2017), E-PKG (Huang et al. 2022) and the constructed DWY dataset.

- The **E-PKG** dataset is an e-commerce dataset about the mobile phone-related product information in 6 languages, including German (DE), English (EN), Spanish (ES), French (FR), Italian (IT) and Japanese (JA).
- The **DWY** dataset is a multi-domain dataset constructed in this paper based on Sun et al. (2018), which includes DBpedia (DB), YAGO (YG) and Wiki (WK). We select 80%, 10%, 10% triples of each KG as training, validation, testing data, respectively. To connect different KGs in the MKGC setting, we adopt the original aligned entities between paired KGs, and prepare equivalent entities between DB-YG, DB-WK and YG-WK. We will release this dataset for future public research.

For all datasets, the equivalent entities are given to connect each of two KGs. The relations are unified in a scheme across all KGs (Chen et al. 2017; Tang et al. 2023). Detailed statistics of all datasets are shown in Table 1.

B.2 Competitors

To evaluate the effectiveness of our proposed method, we select the following state-of-the-art methods as competitors. The competitors can be manifested into three groups:

(i) The **single-source methods**, which learn each KG separately and conduct KGC prediction on each of them:

- **TransE** (Bordes et al. 2013) models relations as translations in the Euclidean space.
- **DisMult** (Yang et al. 2015) adopts a simple bilinear formulation for semantic matching.
- **RotatE** (Sun et al. 2019) models relations as rotations in the complex space.
- **KG-BERT** (Yao, Mao, and Luo 2020) employs pretrained language models for KGC based on text information of relations and entities.

These methods assume to predict missing triples with a single purified, which remain further designations to transfer knowledge between multiple KGs.

(ii) The **multi-source methods**, which utilize all KGs to transfer knowledge to benefit KGC on each of the KGs:

- **KEnS** (Chen et al. 2020) learns all KGs in a unified space and exploits an ensemble method to transfer knowledge.
- **CG-MuA** (Zhu et al. 2020) achieves the GNN-based KG alignment with collective aggregation and makes MKGC with revised losses (Huang et al. 2022).
- **AlignKGC** (Singh et al. 2021) performs KGC together with entity alignment and relation alignment across KGs.
- **SS-AGA** (Huang et al. 2022) improves MKGC by dynamically generating more potential alignment pairs.
- **LSMGA** (Tang et al. 2023) designs an attentive relational GNN-based KG encoder and fuses all equivalent entities with an attention aggregation for MKGC.
- **GLKGC** (He and Yang 2024) leverages transformer-based GNN to encode each KG and trains the model with losses of EA and KGC. We re-implement it with LSMGA source code, as its code is not available.

These methods mostly learn and fuse multiple KGs with alignments and attention modules. Few methods explicitly address redundancy within multiple KGs, which impedes further improvements with numerous KGs.

(iii) The **LLM-based methods**, which leverage powerful LLMs with textual information of entities and relations to predict new triples on each of the KGs:

- **ChatGPT-3.5** (Zhu et al. 2024) direct queries LLMs with a textual head entity and relation and outputs a tail entity as the prediction.
- **KICGPT** (Wei et al. 2023) retrieves relevant triples from the given KG and adopts in-context learning to prompt GPT to predict new triples.
- **MKGL** (Guo et al. 2024) treats KG triples as a three-word language to complete, developed with a tailored dictionary and illustrative sentences.

These methods are also mainly designed with a single purified KG, which also remains underexplored to make predictions with multiple potentially redundant KGs.

B.3 Evaluation Protocol

To evaluate our IMKGC framework, we adopt the typical rank-based evaluation protocol of KGC. For generality, we evaluate the KGC model with the task of predicting tail entities. For a query $(h, r, ?)$, we place all candidate tail entities in the query to form triples and measure the plausibility scores of the triples. The tail entity in the triple with the highest score is treated as the final prediction of the tail entity. Note that the entity set of a target KG serves as the corresponding candidate tail entities. All testing data of KGs in the datasets are used to test the model, and we also use the averaged metrics of all KGs to measure the overall performance of the model (Tang et al. 2023). In detail, the following metrics are used:

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

$$\text{Hits@N} = \frac{1}{|\mathcal{Q}|} \sum_{q_i \in \mathcal{Q}} \mathbb{I}[\text{rank}(i) \leq N], \tag{11}$$

  where $\mathcal{Q}$ denotes all query triples $(h, r, ?)$ in the testing data, rank_i denotes the rank position of the correct entity in the candidates for the i-th query, and $\mathbb{I}[\text{rank}(i) \leq N]$ yields 1 if i is ranked within top-N, and 0 otherwise. This metric is bounded in the range [0, 1], where the higher, the better. Note that Hits@1 is equivalent to the precision widely used in conventional classification tasks.

- **MRR**: Mean reciprocal rank (MRR) measures the overall performance of the ranking, which is the average of the reciprocal ranks of results for all queries as:

$$\text{MRR} = \frac{1}{|\mathcal{Q}|} \sum_{q_i \in \mathcal{Q}} \frac{1}{\text{rank}(i)}, \tag{12}$$

  where $\mathcal{Q}$ also refers to the query triples $(h, r, ?)$. MRR is a useful metric since it reflects the overall ranks of all query triples. Higher MRR values indicate better performance, with 1 being the maximum achievable value.

Hyper-parameters	**DBP-5L**	**E-PKG**	**DWY**
batch size	300	300	300
learning rate	1×10^{-3}	1×10^{-3}	1×10^{-3}
margin λ	0.5	0.5	0.5
epoch	30	50	30
temperature τ	0.5	0.5	0.5
embedding size d	256	256	256
hidden dimension	256	256	256
GNN layer number L	2	2	2
reasoning step K	4	2	3
codebook size ratio	0.8	0.6	0.8
factor α	1×10^{-1}	5×10^{-2}	1×10^{-1}
factor β	1×10^{-4}	1×10^{-4}	1×10^{-4}
factor ω	5×10^{-2}	1×10^{-3}	5×10^{-2}
factor γ	5×10^{-3}	5×10^{-3}	5×10^{-3}

Table 2: Detailed hyper-parameters of our IMKGC model.

B.4 Detailed Hyper-parameters

We implement our model with Pytorch[1] based on the PyG[2] architecture following Tang et al. (2023). Most experiments are conducted on a server with Tesla T4 GPUs, and partial experiments with LLMs are with Tesla A100 GPUs.

The optimal model is selected according to the average MRR of all KGs on their validation sets by grid-search with three trials, following Tang et al. (2023). We use Adam (Kingma and Ba 2014) to learn the model. The tuning ranges of hyper-parameters are reported in the main text. Here, we report the used hyper-parameters in Table 2.

[1] https://docs.pytorch.org/docs/stable/index.html
[2] https://pytorch-geometric.readthedocs.io/en/latest/

Datasets	EL			EN			ES			FR			JA		
	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10	MRR
ALL	**38.2**	**90.9**	**59.9**	**18.2**	**62.0**	**33.8**	**30.6**	**77.5**	**47.5**	**35.4**	**82.5**	**52.5**	**37.3**	**81.5**	**53.9**
- EL	-	-	-	16.5	62.9	32.7	26.2	75.0	43.9	31.7	81.3	49.5	33.2	80.6	50.4
- EN	17.5	88.7	45.6	-	-	-	12.6	73.0	35.1	15.6	79.1	39.3	15.4	79.2	39.9
- ES	18.9	86.5	45.0	7.7	56.9	25.1	-	-	-	17.8	76.1	39.5	19.4	79.9	42.2
- FR	21.3	89.4	47.9	9.2	58.4	26.8	16.1	72.9	36.8	-	-	-	14.5	75.4	38.2
- JA	21.6	86.8	47.2	11.6	61.0	29.4	19.8	76.4	40.6	19.8	77.1	41.0	-	-	-

Table 3: Impact analysis of each KG in the DBP-5L. The best result is **bold-faced**.

Algorithm 1: Training procedure of IMKGC

Input: Training data of KGs, $\mathcal{G}_i \in \mathcal{D}_{train}, |\mathcal{D}_{train}| = N$.
Output: Model parameters Θ.

Initialize all model parameters.
while not convergence **do**
  Generate relation representations in RRD.
  for target KG $\mathcal{G}_i \in \mathcal{D}_{train}$ **do**
    Sample training triples from $\mathcal{G}_i$ with their entities.
    for source KG $\mathcal{G}_j \in \mathcal{D}_{train}$ **do**
      Generate entity representation $\boldsymbol{Z}_j$ by $\mathrm{ARGNN}(\mathcal{G}_j)$.
      Calculate Minimal Term $I(\boldsymbol{Z}_j; \boldsymbol{X}_j)$.
      if $j == i$:
        Calculate Endogenous Term $I(\boldsymbol{Z}_i; \boldsymbol{Y}_i)$.
      else:
        Calculate Exogenous Term $I(\boldsymbol{Z}_j; \boldsymbol{Y}_i)$.
    end for
    Sample another KG $\mathcal{G}_j, j \neq i$.
    Calculate Consistent Term $I(\boldsymbol{Z}_j; \boldsymbol{Z}_i)$.
    Fuse entity representations as $\bar{\boldsymbol{Z}}$.
    Calculate Prediction Term $I(\bar{\boldsymbol{Z}}; \boldsymbol{Y}_i)$.
  end for
  Collect overall loss $\mathcal{J}_{\text{final}}$, and optimize Θ.
end while

B.5 Pseudo-Code of Training Procedure

For better illustration of our model, we provide the pseudo-code of the training procedure, detailed in Algorithm 1. In testing, we derive representations from KGs, and make predictions with the fused representations. The source code and example data are also provided in **Supplementary Material**. We will release the source code and data after the reviewing phase, and hope that this work can be beneficial to future work in the research community.

C Further Experiments

In this section, we provide supplementary experiments to further evaluate the results of our model.

C.1 Impact of Each KG for MKGC

To investigate the impact of each KG on the enhancement of KGC in other KGs, we perform experiments by removing each KG. The results on DBP-5L are shown in Table 3. From the results, we can observe: *First, all KGs can be useful as a benefit to KGC in the target KG.* By removing each KG, the KGC performance of other KGs decreases. This reflects that the KGC task can indeed benefit from multi-source KGs, which provides complementary knowledge that can help to infer missing triples in the target KG. Our model also shows the effectiveness of transferring crucial knowledge to improve the KGC by other sourced KGs. *Second, the EL KG tends to benefit more from other KGs.* By removing a KG, the results of EL decrease significantly. We find that EL has the least training triples compared to other KGs, as shown in Table 1. We believe that KGs with limited endogenous triples can benefit more from other rich KGs. More progress can be explored in future work.

Variant	A-H@1	A-H@10	A-MRR	ΔA-MRR
Base (w/ avg)	31.9	78.9	49.5	-
Base (w/ att)	32.7	79.0	49.9	0.4↑
Base (w/ maxp)	31.7	78.9	49.2	0.3↓
w/o Prior	28.9	78.0	47.0	2.3↓
w/o Attention	29.1	78.1	46.9	2.4↓
repl. GCN	27.8	77.9	46.1	3.4↓
w/o RRD	29.6	77.0	47.6	1.9↓
w/o Endogenous	29.8	77.7	47.4	2.1↓
w/o Consistent	28.8	77.4	46.0	3.3↓
w/o Exogenous	24.9	76.9	45.8	3.7↓
w/o I (All Terms)	24.4	76.7	45.3	4.2↓

Table 4: More analyses of variants on DBP-5L, where A-H@1, A-H@10 and A-MRR denote averaged metrics.

C.2 Further Variant Analysis

To further investigate the impact of modules in our model, we conduct more variant experiments. The results are shown in Table 4. *First, we investigate the impact of different fusion modules of equivalent entities.* The base model of IMKGC uses *average* fusion on all equivalent entities, and we replace it by an *attention* fusion as in LSGMA (Tang et al. 2023) and a *max-pooling* operation. We find that attention fusion can slightly improve the overall results further. However, it also tends to slightly slow the convergence of training in our experiments, as the best result usually appears within 30 epochs for avg yet within 50 epochs for att. As our main motivation to verify the effectiveness of information constraints in limiting redundant information with representation, we opt for the simple average fusion to achieve our base model according to Occam's Razor principle. *Second, we investigate more variants of the GNN architecture.*

Methods	EL			EN			ES			FR			JA			AVG
	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10	MRR	MRR
TransE	13.1	43.7	24.3	7.3	29.3	16.9	13.5	45.0	24.4	17.5	48.8	27.6	21.1	48.5	25.3	23.7
DistMult	8.9	11.3	9.8	8.8	30.0	18.3	7.4	22.4	13.2	6.1	23.8	14.5	9.3	27.5	15.8	14.3
RotatE	14.5	36.2	26.2	12.3	30.4	20.7	21.2	53.9	33.8	23.2	55.5	35.1	26.4	60.2	39.8	31.1
KG-BERT	17.3	40.1	27.3	12.9	31.9	21.0	21.9	54.1	34.0	23.5	55.9	35.4	26.9	59.8	38.7	31.3
ChatGPT-3.5	10.5	-	-	**24.8**	-	-	18.7	-	-	26.3	-	-	12.2	-	-	-
KICGPT	10.4	27.8	20.8	9.5	21.2	13.9	7.2	20.3	12.0	14.9	30.5	20.6	24.2	50.6	32.9	20.0
MKGL	13.9	31.8	20.0	13.2	32.5	19.8	13.5	35.2	20.9	11.9	27.5	17.2	14.0	27.2	18.6	19.3
IMKGC	**38.2**	**90.9**	**59.9**	18.2	**62.0**	**33.8**	**30.6**	**77.5**	**47.5**	**35.4**	**82.5**	**52.5**	**37.3**	**81.5**	**53.9**	**49.5**

Table 5: Comparison results with LLM-based methods on the DBP-5L dataset. The best result is **bold-faced**.

We remove the prior of relations, and remove the attention in the neighbor score function, respectively. Both results decline, which reflects the importance of ways to capture the relational structures in KG encoder. *Third, we investigate relation reasoning decoder (RRD).* The results decline without RRD, which reflects the effectiveness of the residual vector quantization-based relation representation, which can help build relatedness between relations, and further improve reasoning. *Forth, we investigate the impact of different information constraints.* The results demonstrate the unity of each information constraints, which are also discussed in the main text. All results reflect the impacts of modules, and also indicate the effectiveness of modules in our model.

C.3 Full Results with LLM-based Methods

To further analyze the MKGC task, we conduct further investigation with LLM-based methods, including ChatGPT-3.5 (Zhu et al. 2024), KICGPT (Wei et al. 2023), and MKGL (Guo et al. 2024). Note that ChatGPT-3.5 and KICGPT use GPT-3.5 turbo as base model, and MKGL uses LLama2-7b (Touvron et al. 2023). We use these methods to predict missing triples with their training data in each single-source KG. The results are shown in Table 5, where we can find that: *First, the LLM-based methods generate diverse results on different sourced KGs.* The DBP-5L dataset has 5 KGs in different languages, but LLMs can hardly achieve promising results on every language KG. For example, ChatGPT-3.5 achieves the best results (H@1) in EN KG, since the LLM is constructed with most English data and can perform well in English tasks. However, it performs less well to deal with other languages, such as Greek (EL) KG. Moreover, LLMs also struggle to deal with entities whose entity names are not available or are literally difficult to understand. Both shortcomings impair the effectiveness of these LLM-based methods. *Second, the LLM-based methods can still hardly handle the MKGC task.* Existing LLM-based KGC methods are mostly designed with a single purified KG. Directly deploying them on multi-source KGs would produce a lot of repetitive redundant triples, which would interfere with the understanding of the KG structural knowledge and confuse the model prediction. This can be further investigated in the KG research community, and we leave this in future work.

References

Alemi, A. A.; Fischer, I.; Dillon, J. V.; and Murphy, K. 2017. Deep Variational Information Bottleneck. In *Proceedings of ICLR*. OpenReview.net.

Bordes, A.; Usunier, N.; Garcia-Durán, A.; Weston, J.; and Yakhnenko, O. 2013. Translating Embeddings for Modeling Multi-Relational Data. In *Proceedings of NeurIPS*, 2787–2795.

Chen, M.; Tian, Y.; Yang, M.; and Zaniolo, C. 2017. Multilingual Knowledge Graph Embeddings for Cross-lingual Knowledge Alignment. In *Proceedings of IJCAI*.

Chen, X.; Chen, M.; Fan, C.; Uppunda, A.; Sun, Y.; and Zaniolo, C. 2020. Multilingual Knowledge Graph Completion via Ensemble Knowledge Transfer. In *Findings of EMNLP*, 3227–3238.

Guo, L.; Bo, Z.; Chen, Z.; Zhang, Y.; Chen, J.; Lan, Y.; Sun, M.; Zhang, Z.; Luo, Y.; Li, Q.; Zhang, Q.; Zhang, W.; and Chen, H. 2024. MKGL: Mastery of a Three-Word Language. In *Proceedings of NeurIPS*.

He, J.; and Yang, H. 2024. Multilingual Knowledge Graph Completion based on Global-Local Structure Encoding. In *Proceedings of ISCTIS*, 647–650.

Huang, Z.; Li, Z.; Jiang, H.; Cao, T.; Lu, H.; Yin, B.; Subbian, K.; Sun, Y.; and Wang, W. 2022. Multilingual Knowledge Graph Completion with Self-Supervised Adaptive Graph Alignment. In *Proceedings of ACL*, 474–485.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*.

Kingma, D. P.; and Welling, M. 2014. Auto-Encoding Variational Bayes. In *Proceedings of ICLR*.

Singh, H.; Jain, P.; Chakrabarti, S.; et al. 2021. Multilingual Knowledge Graph Completion with Joint Relation and Entity Alignment. *arXiv preprint arXiv:2104.08804*.

Sun, Z.; Deng, Z.-H.; Nie, J.-Y.; and Tang, J. 2019. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. In *Proceedings of ICLR*.

Sun, Z.; Hu, W.; Zhang, Q.; and Qu, Y. 2018. Bootstrapping Entity Alignment with Knowledge Graph Embedding. In *Proceedings of IJCAI*, 4396–4402.

Tang, R.; Zhao, Y.; Zong, C.; and Zhou, Y. 2023. Multilingual Knowledge Graph Completion with Language-Sensitive Multi-Graph Attention. In *Proceedings of ACL*, 10508–10519.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; Bikel, D.; Blecher, L.; Canton-Ferrer, C.; Chen, M.; Cucurull, G.; Esiobu, D.; Fernandes, J.; Fu, J.; Fu, W.; Fuller, B.; Gao, C.; Goswami, V.; Goyal, N.; Hartshorn, A.; Hosseini, S.; Hou, R.; Inan, H.; Kardas, M.; Kerkez, V.; Khabsa, M.; Kloumann, I.; Korenev, A.; Koura, P. S.; Lachaux, M.; Lavril, T.; Lee, J.; Liskovich, D.; Lu, Y.; Mao, Y.; Martinet, X.; Mihaylov, T.; Mishra, P.; Molybog, I.; Nie, Y.; Poulton, A.; Reizenstein, J.; Rungta, R.; Saladi, K.; Schelten, A.; Silva, R.; Smith, E. M.; Subramanian, R.; Tan, X. E.; Tang, B.; Taylor, R.; Williams, A.; Kuan, J. X.; Xu, P.; Yan, Z.; Zarov, I.; Zhang, Y.; Fan, A.; Kambadur, M.; Narang, S.; Rodriguez, A.; Stojnic, R.; Edunov, S.; and Scialom, T. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. *CoRR*, abs/2307.09288.

Wei, Y.; Huang, Q.; Zhang, Y.; and Kwok, J. T. 2023. KICGPT: Large Language Model with Knowledge in Context for Knowledge Graph Completion. In *Findings of EMNLP*, 8667–8683.

Yang, B.; Yih, W.-t.; He, X.; Gao, J.; and Deng, L. 2015. Embedding entities and relations for learning and inference in knowledge bases. In *Proceedings of ICLR*.

Yao, L.; Mao, C.; and Luo, Y. 2020. KG-BERT: BERT for Knowledge Graph Completion. *Proceedings of AAAI*.

Zhu, Q.; Wei, H.; Sisman, B.; Zheng, D.; Faloutsos, C.; Dong, X. L.; and Han, J. 2020. Collective Multi-Type Entity Alignment Between Knowledge Graphs. In *Proceedings of Web Conference*.

Zhu, Y.; Wang, X.; Chen, J.; Qiao, S.; Ou, Y.; Yao, Y.; Deng, S.; Chen, H.; and Zhang, N. 2024. LLMs for knowledge graph construction and reasoning: recent capabilities and future opportunities. *Proceedings of Web Conference*, 58.