ENCODE TO DECODE: SCALABLE KNOWLEDGE GRAPH FOUNDATION MODELS FROM EMBEDDING MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Knowledge Graph Foundation Models (KGFMs) have emerged to enable zero-shot generalization to unseen knowledge graphs (KGs). However, existing KGFMs typically incur a significant computational overhead by executing message passing over the entire KG for every single query. We introduce MUZE, a scalable KGFM whose encoder constructs query-independent entity and relation embeddings from random initialization through recursive message passing. At each layer, MUZE computes messages using the gradients of a Knowledge Graph Embedding (KGE) scoring function and aggregates them to update entity and relation representations. The link prediction decoder directly applies the same scoring function to the resulting embeddings to rank candidates for each query. This principled coupling unifies KGE-based decoding and message-passing-based encoding by incorporating the KGE scoring function into the encoder's recursive computation. Crucially, since KGE scoring functions operate directly on query-independent embeddings, MUZE encodes each KG once and reuses the resulting embeddings across queries, thereby substantially reducing inference time. We theoretically prove that, on a finite set of KGs, MUZE can recover arbitrary target embeddings and serve as a fully expressive encoder for KG tasks solvable via entity and relation embeddings. Experiments show that MUZE is 11x faster than the fastest baseline and over 1389x faster than the most accurate baseline, all while delivering comparable or superior accuracy.

1 INTRODUCTION

Knowledge graphs (KGs) represent human knowledge in the form of triplets (s, p, o), where s is a head entity, p is a relation, and o is a tail entity. To represent entities and relations in continuous vector spaces, KG Embedding (KGE) methods (Wang et al., 2017) optimize embeddings using their scoring functions. While effective for knowledge representations, KGE methods are limited to transductive inference, i.e., they cannot generalize to entities and relations unseen during training without retraining from scratch. Recently, KG Foundation Models (KGFMs) (Galkin et al., 2024; Kim et al., 2026; Cui et al., 2024) have been proposed to enable zero-shot inference across diverse KGs. These models focus on KG completion, the task of inferring a missing element in an incomplete triplet called a query, and adopt a query-conditional modeling approach. For example, methods such as ULTRA (Galkin et al., 2024) and TRIX (Zhang et al., 2025) cast KG completion as a structural pattern recognition problem conditioned on a query and solve it via query-specific message passing over the relation and entity graphs. However, such query-dependent approaches require processing the entire KG for each prediction, leading to significant computational bottlenecks and inference latency.

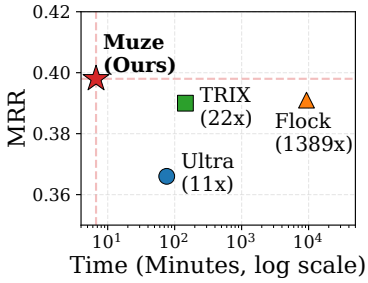

Figure 1: MRR and inference time of MUZE vs. baselines for zero-shot entity prediction on 54 KGs.

In this paper, we propose a fundamentally different approach that replaces the prevailing query-specific approach taken by all existing KGFMs (Galkin et al., 2024; Kim et al., 2026; Cui et al., 2024; Huang et al., 2025; Zhang et al., 2025) with a highly efficient, query-agnostic embedding computation. We introduce MUZE, a recursive **M**essage passing framework for f**U**lly expressive KG encoding and scalable **Z**ero-shot KG transfer via **E**ncode-to-decode. Starting from randomly initialized entity and

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

relation representations, MUZE recursively aggregates messages derived from the gradients of a KGE scoring function to update the representations. The link prediction decoder applies the same KGE scoring function to the resulting embeddings to rank candidate entities or relations. This shared use of the KGE scoring function provides a principled bridge between KGE-based decoding and message-passing-based encoding, where the gradient-based messages guide the computed embeddings to align with the scoring function. Crucially, since the scoring function operates directly on the computed embeddings, MUZE encodes each KG once and reuses the embeddings across queries.

This architectural shift drastically reduces inference overhead. Unlike existing KGFMs that typically incur an $\mathcal{O}(|\mathcal{Q}||\mathcal{T}|)$ computational complexity for inference on a KG, where $\mathcal{Q}$ and $\mathcal{T}$ denote the sets of queries and triplets, respectively, MUZE requires only a one-time offline computation of $\mathcal{O}(|\mathcal{T}|)$ to pre-compute the embeddings. Subsequent queries are evaluated in $\mathcal{O}(|\mathcal{Q}|(|\mathcal{V}| + |\mathcal{R}|))$ time via a scoring function, where $|\mathcal{V}|$ and $|\mathcal{R}|$ denote the number of entities and relations, respectively. As a result, MUZE accelerates inference by over $11\times$ without sacrificing accuracy (Figure 1). Furthermore, our theoretical analysis shows that, over any finite collection of KGs, MUZE is a fully expressive encoder for tasks that can be solved using entity and relation embeddings, enabled by its random initialization and representation updates. Our contributions are summarized as follows:

- We propose MUZE, the first query-agnostic KGFM that couples KGE-based decoding with message-passing-based encoding to recursively compute entity and relation representations. Our codes are available at `https://anonymous.4open.science/r/muze-i6`.
- We prove that MUZE can approximate any target KG embeddings on a finite set of KGs, establishing it as a fully expressive encoder for KG tasks solvable via embeddings.
- MUZE accelerates inference by over 11x compared to the fastest baseline and 1389x compared to the most accurate baseline, resolving the latency bottlenecks of existing KGFMs.
- Unlike existing KGFMs that require separate pre-training for entity and relation prediction, MUZE handles both tasks with a single pre-trained model, while achieving comparable or state-of-the-art performance in zero-shot and finetuned inference across these tasks.

2 RELATED WORK

Knowledge Graph Embedding KGE methods map KGs into continuous vector spaces using specific scoring functions, optimizing representations to maximize the scores of positive triplets over negative ones (Li et al., 2022; 2024). Translational methods (Lin et al., 2015; Wang et al., 2014; Bordes et al., 2013; Sun et al., 2019) and semantic matching methods (Yang et al., 2015; Trouillon et al., 2016; Liu et al., 2017; Kazemi & Poole, 2018) are two representative KGE approaches. A major strength of KGE models is their highly efficient $\mathcal{O}(|\mathcal{V}| + |\mathcal{R}|)$ inference complexity per query, requiring only a simple dot product or distance calculation. However, these models learn fixed embeddings tied to the training KG's entity and relation sets, and thus cannot perform inference on new KGs. MUZE leverages this efficient decoding structure while replacing fixed KG-specific embeddings with representations computed by a shared message-passing encoder, enabling zero-shot transfer to unseen KGs while retaining efficient KGE-based decoding.

Query-Specific Models in KGs Query-specific models require a complete encoding-decoding process for each individual query. These models were originally introduced to tackle inductive inference scenarios involving unseen entities (Teru et al., 2020). For example, query-specific message passing methods (Zhu et al., 2021; Zhang & Yao, 2022) perform per-query message passing on the KG to predict the score of each entity for a given query. KGFMs such as ULTRA (Galkin et al., 2024), TRIX (Zhang et al., 2025), and MOTIF (Huang et al., 2025) extend query-specific message passing by constructing relation graphs to capture structural patterns of relations in KGs. KG-ICL (Cui et al., 2024) uses candidate-specific subgraphs as prompts, while FLOCK (Kim et al., 2026) encodes query-dependent random walk sequences. Despite their zero-shot capabilities, they share a critical computational bottleneck due to the encoding-decoding process over the entire KG for *each* query.

Query-Agnostic Message Passing in KGs Methods such as R-GCN (Schlichtkrull et al., 2018) and CompGCN (Vashishth et al., 2020) generate query-agnostic representations but are strictly transductive, requiring retraining for unseen entities or relations. REFACTOR GNNS (Chen et al., 2022) handle new entities by unrolling the update steps of DISTMULT (Yang et al., 2015), but it relies on the learned relation embeddings and is thus incapable of zero-shot transfer to KGs with

unseen relations. For scenarios where both entities and relations are unseen during training, models such as InGram (Lee et al., 2023) and MAYPL (Lee & Whang, 2025) perform query-agnostic message passing over relation graphs or hyper-relational KGs, respectively. However, these methods implicitly assume a shared structural distribution between the training and inference KGs, limiting their zero-shot generalization to arbitrary, out-of-distribution KGs. Unlike these methods, MUZE derives messages from the gradients of the KGE scoring function used for prediction, utilizing the decoding function as the basis for recursively constructing entity and relation representations.

3 PRELIMINARIES

Given a KG denoted by $G = (\mathcal{V}, \mathcal{R}, \mathcal{T})$, where $\mathcal{V}$ is the set of entities, $\mathcal{R}$ is the set of relations, and $\mathcal{T} \subseteq \mathcal{V} \times \mathcal{R} \times \mathcal{V}$ is the set of triplets, KG completion is defined as follows, where $(s, p, o) \in \mathcal{T}$.

Definition 1 (KG Completion)**.** KG completion is the task of predicting missing triplets in a given incomplete KG $G = (\mathcal{V}, \mathcal{R}, \mathcal{T})$ via entity or relation prediction. Entity prediction is to predict a missing head or tail entity, represented as $(?, p, o)$ or $(s, p, ?)$, while relation prediction aims to identify a missing relation, denoted as $(s, ?, o)$.

KGE Methods A KGE method is designed to represent entities and relations as d-dimensional vectors, where $\boldsymbol{E} \in \mathbb{R}^{d \times (|\mathcal{V}|+|\mathcal{R}|)}$ denotes the embedding vector matrix. To measure the plausibility of triplets, these methods define a scoring function f and optimize the entity and relation embeddings by maximizing the scores of valid triplets while minimizing those of negative samples. For example, TRANSE (Bordes et al., 2013) defines its scoring function as $f(\mathbf{s}, \mathbf{p}, \mathbf{o}) = -\|\mathbf{s} + \mathbf{p} - \mathbf{o}\|_2$, and DISTMULT (Yang et al., 2015) defines $f(\mathbf{s}, \mathbf{p}, \mathbf{o}) = \langle \mathbf{s} \odot \mathbf{p}, \mathbf{o} \rangle$, where $\mathbf{s}, \mathbf{p}, \mathbf{o}$ denote the embedding vectors of s, p, and o, respectively. To train KGE methods, various loss functions, such as cross-entropy loss (Lacroix et al., 2018) and negative log-likelihood loss (Sun et al., 2019), are utilized. To derive the gradient-based messages used by MUZE, we adopt the negative log-likelihood loss with negative sampling, a widely adopted loss in KG completion (Zhu et al., 2021; Galkin et al., 2024):

$$L_{\boldsymbol{E}}(s, p, o) = -\log\big(\sigma(f(\mathbf{s}, \mathbf{p}, \mathbf{o}))\big) - \frac{1}{n}\sum_{i=1}^{n} \log\big(1 - \sigma(f(\mathbf{s}'_i, \mathbf{p}'_i, \mathbf{o}'_i))\big), \tag{1}$$

where n is the number of negative samples per positive triplet, (s'_i, p'_i, o'_i) is the i-th negative sample for the positive triplet (s, p, o), and $\sigma(\cdot)$ is the sigmoid function.

4 MUZE: A SCALABLE KG FOUNDATION MODEL VIA ENCODE TO DECODE

We propose MUZE, a recursive message passing framework that couples message-passing-based encoding with KGE-based decoding by using gradients of a KGE scoring function as messages and the scoring function itself as the link prediction decoder. Figure 2 provides an overview of MUZE. Starting from randomly initialized representations of entities and relations, each layer of MUZE computes and aggregates messages derived from the gradients of f for positive and negative triplets and updates the entity and relation representations. The messages align the representations with f by guiding them toward higher scores for positive triplets and lower scores for negative triplets. MUZE recursively applies the same message passing functions with shared parameters for K layers to produce the final query-independent embeddings. The decoder then applies f to these embeddings to score candidate entities or relations. Since f only requires the query-independent embeddings to score candidates, MUZE encodes each KG once and reuses the resulting embeddings across queries. This provides a significant scalability advantage over existing KGFMs, empirically verified in Section 6.2.

4.1 RECURSIVE KG ENCODING

MUZE encodes a KG by repeatedly applying the same message passing functions with shared parameters across layers to the entity and relation representations. At each layer, for each positive triplet and its sampled negative triplets, MUZE computes a loss from the scores produced by the KGE scoring function f and uses its gradients with respect to the representations of the entities and the relation involved in the triplet as messages. These messages are aggregated to update the entity and relation representations. The negative samples provide contrasting signals for representation updates. Repeating this process for K layers yields the final entity and relation embeddings.

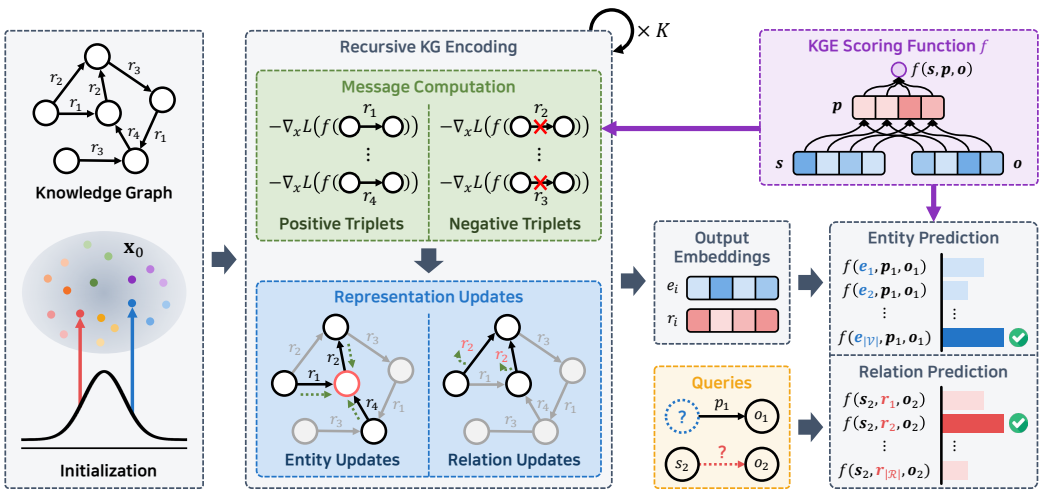

Figure 2: Overview of MUZE. Starting from randomly initialized entity and relation representations, MUZE recursively constructs their embeddings through gradient-based message passing over the KG. The KGE scoring function f induces messages during encoding and serves as the link prediction decoder. The embeddings are computed once for a KG and reused across queries.

Computing Messages From a KGE Scoring Function Let $\boldsymbol{E}_{t-1}$ denote the entity and relation representations output by layer $t-1$, where $\boldsymbol{E}_0$ is sampled from a standard normal distribution. For an entity or a relation x, let $\mathcal{N}(x)$ denote the set of positive triplets containing x, and let $\mathcal{N}'_t(x)$ denote the negative samples involving x. Using the loss defined in Equation 1, its gradient at layer t with respect to the representation of x can be decomposed as:

$$\mathbf{g}^+_{x,t} = -\sum_{(s,p,o)\in\mathcal{N}(x)} \sigma(-f(\mathbf{s}_{t-1},\mathbf{p}_{t-1},\mathbf{o}_{t-1}))\nabla_x f(\mathbf{s}_{t-1},\mathbf{p}_{t-1},\mathbf{o}_{t-1}) \tag{2}$$

$$\mathbf{g}^-_{x,t} = \frac{1}{n}\sum_{(s',p',o')\in\mathcal{N}'_t(x)} \sigma(f(\mathbf{s}'_{t-1},\mathbf{p}'_{t-1},\mathbf{o}'_{t-1}))\nabla_x f(\mathbf{s}'_{t-1},\mathbf{p}'_{t-1},\mathbf{o}'_{t-1}) \tag{3}$$

where $\mathbf{g}^+_{x,t}$ is the sum of gradients from positive triplets, and $\mathbf{g}^-_{x,t}$ is the sum of gradients from negative samples. We then define the aggregated message as the negative of the gradient $\mathbf{a}_{x,t} = -(\mathbf{g}^+_{x,t}+\mathbf{g}^-_{x,t})$. By taking the negative loss gradient to form the message, positive triplets guide the representations toward higher scores, while negative triplets guide them toward lower scores. Thus, using f during encoding guides the representations to align with it, naturally allowing f to serve as the decoder.

Representation Update MUZE utilizes the aggregated message $\mathbf{a}_{x,t}$ to update the representations of entities and relations. Instead of directly utilizing $\mathbf{a}_{x,t}$, we maintain running statistics of the aggregated messages across layers, inspired by the moment statistics used in adaptive optimizers such as AdamW (Loshchilov & Hutter, 2019). Starting from $\mathbf{m}_{x,0} = \mathbf{0}$ and $v_{x,0} = 0$, we compute

$$\mathbf{m}_{x,t} = \widetilde{\beta}_1\mathbf{m}_{x,t-1} + (1-\widetilde{\beta}_1)\mathbf{a}_{x,t}, \qquad v_{x,t} = \widetilde{\beta}_2 v_{x,t-1} + (1-\widetilde{\beta}_2)\frac{\|\mathbf{a}_{x,t}\|_2^2}{d}, \tag{4}$$

where $\widetilde{\beta}_1, \widetilde{\beta}_2 \in (0,1)$ are learnable scalars. $\mathbf{m}_{x,t} \in \mathbb{R}^d$ accumulates the messages to x across layers, whereas $v_{x,t} \in \mathbb{R}$ tracks their overall magnitude. We apply bias correction to obtain $\widehat{\mathbf{m}}_{x,t} = \frac{\mathbf{m}_{x,t}}{1-\widetilde{\beta}_1^t}$ and $\widehat{v}_{x,t} = \frac{v_{x,t}}{1-\widetilde{\beta}_2^t}$, and normalize the accumulated message as $\widetilde{\mathbf{m}}_{x,t} = \frac{\widehat{\mathbf{m}}_{x,t}}{\sqrt{\widehat{v}_{x,t}+\epsilon}}$. Since $\widehat{v}_{x,t}$ is a scalar, the normalization preserves the direction of the accumulated message.

To update the current representation $\mathbf{x}_{t-1}$ of x, MUZE first adaptively scales the normalized message using $\text{MLP}_{\text{scale}}$ and then applies a gated residual refinement through MLP_{upd}, gated by MLP_{gate}:

$$\mathbf{u}_{x,t} = (1-\widetilde{\lambda})\mathbf{x}_{t-1} + \eta_t\widetilde{\alpha}\sigma(\text{MLP}_{\text{scale}}[\mathbf{x}_{t-1},\widetilde{\mathbf{m}}_{x,t}]) \odot \widetilde{\mathbf{m}}_{x,t}, \tag{5}$$

$$\mathbf{z}_{x,t} = \text{MLP}_{\text{upd}}(\mathbf{u}_{x,t}), \qquad \mathbf{x}_t = \mathbf{u}_{x,t} + \eta_t\widetilde{\gamma}\sigma(\text{MLP}_{\text{gate}}[\mathbf{u}_{x,t},\mathbf{z}_{x,t}]) \odot \mathbf{z}_{x,t}. \tag{6}$$

Here, $\eta_t \in \mathbb{R}^+ \cup \{0\}$ is a layer-wise decay factor, $\widetilde{\alpha}, \widetilde{\gamma} \in \mathbb{R}^+$ are learnable scalars that control the scales of the message and residual updates, respectively, and $\widetilde{\lambda} \in [0, 1]$ is a learnable representation-decay factor. The dimension-wise scaling and gating functions adaptively control the contributions of the normalized message and nonlinear residual transformation. Repeating this procedure for K layers produces the final entity and relation embeddings $\boldsymbol{E}_K$.

4.2 KGE-Based Decoding

Given the final embeddings $\boldsymbol{E}_K$, MUZE directly applies f to perform link prediction. For a query $(?, p, o)$ with a candidate entity c, MUZE computes $f(\mathbf{c}, \mathbf{p}, \mathbf{o})$. Similarly, it computes $f(\mathbf{s}, \mathbf{p}, \mathbf{c})$ for an entity candidate c given $(s, p, ?)$, and $f(\mathbf{s}, \mathbf{c}, \mathbf{o})$ for a relation candidate c given $(s, ?, o)$, where all representations are taken from $\boldsymbol{E}_K$. Candidate entities or relations are ranked according to these scores. This shared use of f directly couples message-passing-based encoding with KGE-based decoding. Since f only requires the computed entity and relation embeddings $\boldsymbol{E}_K$, MUZE encodes each KG once to obtain $\boldsymbol{E}_K$ and shares $\boldsymbol{E}_K$ across all link prediction queries.

4.3 Implementation Details of Muze

In our implementation, we apply an adaptive version of RMSNorm (Zhang & Sennrich, 2019) to the intermediate representations before computing the triplet scores. Gradients are computed with respect to these normalized representations, and we apply the chain rule to derive the gradient updates for the unscaled embeddings. To ensure the magnitude of updates remains stable and consistent across KGs of varying densities and sizes, we normalize the message of each triplet by the square root of the sum of squared messages along each dimension. We employ a parameterized variation of DistMult as our scoring function, incorporating additional terms to effectively model anti-symmetric relationships: $f(\mathbf{s}, \mathbf{p}, \mathbf{o}) = (\langle \mathbf{w}_1, \mathbf{s} \odot \mathbf{p} \odot \mathbf{o} \rangle + \langle \mathbf{w}_2, \mathbf{s} \odot \mathbf{p} \rangle + \langle \mathbf{w}_3, \mathbf{s} \odot \mathbf{o} \rangle + \langle \mathbf{w}_4, \mathbf{p} \odot \mathbf{o} \rangle)/\sqrt{d}$, where $\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3$, and $\mathbf{w}_4$ are learnable d-dimensional vectors. During training, we optimize MUZE using the cross-entropy loss. To reduce the impact of random initializations, we perform Monte Carlo sampling with 32 samples during inference. Further implementation details are provided in Appendix B.2.

5 Theoretical Analysis of Muze

We theoretically prove that MUZE is a universal encoder for downstream tasks on KGs that can be solved via entity and relation embeddings, e.g., KG completion. We also analyze the computational complexity of MUZE compared to other message passing-based KGFMs.

5.1 Universal Encoder for KG tasks

Existing KGFMs are link predictors designed only for the KG completion task and do not produce embeddings, and thus, they cannot be directly applied to other tasks on KGs such as entity alignment (Cui et al., 2026). However, since MUZE outputs embeddings, it can be applied to tasks that can be solved by utilizing entity or relation embeddings. In Theorem 1, we show that for a finite set of KGs $\mathcal{G}$ whose entity and relation embeddings are initialized via standard normal sampling, a single parameter configuration of MUZE can approximate any target embeddings for all KGs in $\mathcal{G}$.

Theorem 1 (Universal KG Encoder)**.** *Let $\mathcal{G}$ be a finite set of KGs. For each $G \in \mathcal{G}$, let $\boldsymbol{E}_G^*$ denote arbitrary target embeddings of its entities and relations. Given a set of distinguishable random initializations for the entities and relations in each G, and an error tolerance $\epsilon_{err} > 0$, there exists a single parameter configuration of* MUZE*, such that for every $G \in \mathcal{G}$, the final output embeddings $\boldsymbol{E}_G$ satisfy $\|\boldsymbol{E}_G - \boldsymbol{E}_G^*\|_F < \epsilon_{err}$.*

Note that the distinguishability assumption is satisfied almost surely by MUZE's initialization scheme, which samples entity and relation embeddings from a standard normal distribution. Because the given initial random embeddings are uniquely distinguishable, there exists a well-defined continuous mapping from these initial states to the target embeddings $\boldsymbol{E}_G^*$. Then, MLP_{upd} in MUZE can approximate this continuous mapping to an arbitrary precision. Subsequently, Corollary 1 demonstrates that combining this capability with a fully expressive decoder guarantees universal approximation for task-specific functions. The proofs of Theorem 1 and Corollary 1 are in Appendix A.

Corollary 1 (Universal Approximation for Task-specific Functions)**.** *Let a decoder be fully expressive if, for any ground-truth function of a given KG task, there exists a set of embeddings that exactly reconstructs the function, e.g., SimplE (Kazemi & Poole, 2018) or ComplEx (Trouillon et al., 2016) for KG completion. Then, for any distinguishable random initializations of embeddings, there exists a single parameter configuration of* MUZE *combined with a continuous and fully expressive decoder that can approximate any task-specific function to an arbitrary degree of accuracy $\epsilon_{err} > 0$, across all KGs in $\mathcal{G}$.*

Since Theorem 1 guarantees that MUZE can approximate any target embeddings $\boldsymbol{E}_G^*$, the composition of MUZE and a fully expressive decoder can approximate any task-specific function. In contrast to most prior KGFMs (Galkin et al., 2024; Zhang et al., 2025; Huang et al., 2025) with bounded expressivity, we theoretically establish the full expressivity of MUZE. Note that our theoretical results are not intended to guarantee zero-shot transfer of MUZE to unseen KGs; we evaluate this empirically.

5.2 COMPLEXITY ANALYSIS

The inference complexity of MUZE is decoupled into a one-time KG encoding and per-query processing. Assuming $|\mathcal{V}| \gg |\mathcal{R}|$, the encoding takes $\mathcal{O}(Kd((n+1)|\mathcal{T}| + |\mathcal{V}|d))$. Once embeddings are computed, processing a query set $\mathcal{Q}$ requires only $\mathcal{O}(|\mathcal{Q}||\mathcal{V}|d)$. In contrast, existing message passing KGFMs (Galkin et al., 2024; Zhang et al., 2025; Huang et al., 2025) must re-encode the entire KG for every query, taking $\mathcal{O}(K|\mathcal{Q}|(|\mathcal{T}|d + |\mathcal{V}|d^2))$. By reducing the per-query cost to merely $\mathcal{O}(|\mathcal{V}|d)$, MUZE achieves a remarkable computational efficiency, which translates to an $11\times$ empirical speedup over the fastest baseline, as shown in Section 6.2.

6 EXPERIMENTAL RESULTS

Following the standard evaluation protocol for KGFMs (Galkin et al., 2024), we pre-train MUZE on three KGs, FB15k-237 (Toutanova & Chen, 2015), WN18RR (Miller, 1995), and CoDEx-M (Safavi & Koutra, 2020), and evaluate its zero-shot performance across 54 KGs. We also finetune the pre-trained MUZE on 57 KGs, including the three pre-training KGs, and evaluate its performance. These evaluation datasets are categorized into three distinct groups based on their original inference scenarios: (1) **Inductive** $\mathcal{V}, \mathcal{R}$, which tests fully inductive reasoning where both entities and relations are new at inference time; (2) **Inductive** $\mathcal{V}$, where relations are known but entities are new; and (3) **Transductive**, where the entity and relation sets remain identical between training and inference. We evaluate performance on both entity and relation prediction tasks, comparing MUZE against four KGFMs: ULTRA (Galkin et al., 2024), TRIX (Zhang et al., 2025), MOTIF (Huang et al., 2025), and FLOCK (Kim et al., 2026). MOTIF returns out-of-memory (OOM) errors when processing certain KGs in the Transductive category. Additionally, it does not report its performance on relation prediction. Notably, while the baselines are trained separately for entity and relation prediction, MUZE performs both tasks simultaneously using a single pre-trained model. Dataset details are in Appendix B.1.

6.1 KNOWLEDGE GRAPH COMPLETION

Table 1 shows the entity and relation prediction performance of MUZE against existing KGFMs. For entity prediction, MUZE achieves highly competitive results across all settings without relying on expensive query-specific message passing. In the zero-shot setting, it yields the highest average MRR, outperforming the strongest baseline, FLOCK (Kim et al., 2026), particularly by large margins in the Inductive $\mathcal{V}, \mathcal{R}$ category. In the finetuned setting, MUZE achieves performance comparable to the baselines. For relation prediction, MUZE consistently outperforms all baselines across every evaluation category, in both zero-shot and finetuned settings. These results demonstrate that MUZE can construct transferable query-independent representations that support both entity and relation prediction on unseen KGs. More detailed results are in Appendix C.

6.2 SCALABILITY ANALYSIS

We evaluate the scalability of MUZE against baselines across 54 KGs for the zero-shot entity prediction task on a single NVIDIA RTX A6000 GPU, scaling baseline batch sizes to maximize their inference speed. In Table 2, MUZE demonstrates a significant advantage in both time and memory efficiency,

Table 1: Entity and relation prediction in zero-shot (54 KGs) and finetuned (57 KGs) settings. Average is the average MRR over 54 KGs. Pre-trained is the MRR averaged over the 3 pre-training KGs.

Entity Prediction		Inductive $\mathcal{V}, \mathcal{R}$		Inductive $\mathcal{V}$		Transductive		Average		Pre-trained	
		MRR	Hit10	MRR	Hit10	MRR	Hit10	MRR	Hit10	MRR	Hit10
Zero-shot	ULTRA	0.345	0.513	0.431	0.566	0.312	0.458	0.366	0.518	–	–
	TRIX	0.368	0.540	0.455	0.592	0.339	0.500	0.390	0.548	–	–
	MOTIF	0.349	0.525	0.436	0.577	OOM	OOM	OOM	OOM	–	–
	FLOCK	0.369	0.554	**0.456**	**0.604**	**0.340**	**0.509**	0.391	0.560	–	–
	MUZE	**0.400**	**0.569**	0.452	0.598	0.331	0.494	**0.401**	**0.561**	–	–
Finetuned	ULTRA	0.397	0.556	0.442	0.582	0.379	0.543	0.408	0.562	0.407	**0.568**
	TRIX	0.401	0.556	0.459	0.594	**0.390**	**0.558**	0.418	0.569	**0.415**	0.563
	MOTIF	0.401	0.558	0.455	0.594	OOM	OOM	OOM	OOM	**0.415**	0.565
	FLOCK	0.417	0.576	**0.473**	**0.619**	0.383	0.544	**0.427**	**0.582**	**0.415**	0.561
	MUZE	**0.422**	**0.587**	0.462	0.604	0.360	0.522	0.421	0.577	0.395	0.543
Relation Prediction		Inductive $\mathcal{V}, \mathcal{R}$		Inductive $\mathcal{V}$		Transductive		Average		Pre-trained	
		MRR	Hit1	MRR	Hit1	MRR	Hit1	MRR	Hit1	MRR	Hit1
Zero-shot	ULTRA	0.785	0.691	0.714	0.590	0.629	0.507	0.724	0.613	–	–
	TRIX	0.842	0.770	0.756	0.611	0.752	0.647	0.792	0.687	–	–
	FLOCK	0.898	0.846	0.864	0.782	0.873	0.813	0.881	0.817	–	–
	MUZE	**0.938**	**0.904**	**0.916**	**0.860**	**0.886**	**0.835**	**0.919**	**0.873**	–	–
Finetuned	ULTRA	0.823	0.741	0.716	0.591	0.707	0.608	0.759	0.659	0.876	0.817
	TRIX	0.850	0.785	0.759	0.615	0.785	0.693	0.804	0.706	0.879	0.797
	FLOCK	0.929	0.889	0.887	0.808	0.897	0.844	0.907	0.851	**0.977**	**0.959**
	MUZE	**0.949**	**0.920**	**0.921**	**0.867**	**0.913**	**0.869**	**0.931**	**0.890**	**0.977**	**0.959**

Table 2: Comparison of inference cost for zero-shot entity prediction on 54 KGs using an A6000 GPU.

	Time	MRR	GPU Memory
ULTRA	76 minutes	0.366	33GB
TRIX	144 minutes	0.390	35GB
FLOCK	155 hours	0.391	48GB
MUZE	**< 7 minutes**	**0.401**	**15GB**

Table 3: Zero-shot evaluation on Wikidata-5M using a single NVIDIA H200 GPU.

	MRR	Runtime	Memory
TRIX	-	$>$ 24 hours	112 GB
FLOCK	-	$>$ 24 hours	124 GB
ULTRA	0.174	120 minutes	101 GB
MUZE	**0.291**	**9 minutes**	**92 GB**

requiring 6 minutes and 42 seconds with 15GB of peak GPU memory to perform entity prediction on all 54 KGs using 32 Monte Carlo Samples. In contrast, the baseline with the highest MRR, FLOCK, requires 155 hours with 48GB of GPU memory, while even the fastest baseline, ULTRA, takes 76 minutes with 33GB. This computational advantage arises because MUZE only requires computation on a KG once, whereas FLOCK uses a GRU to sequentially process multiple random walks per query, and ULTRA and TRIX require message passing over the entire KG for each query. Note that MUZE did not utilize the `rspmm` (Zhu et al., 2021) kernel, which is used by ULTRA and TRIX to accelerate message passing for relational graphs, leaving further room for improvement in MUZE's scalability.

Large-Scale Evaluation To examine the scalability of MUZE on a large-scale KG, we evaluate MUZE on Wikidata-5M, which contains approximately 5 million entities and 20 million triplets. We conduct zero-shot evaluation using a single NVIDIA H200 GPU. As shown in Table 3, MUZE completes inference in 9 minutes, whereas ULTRA requires 120 minutes and TRIX and FLOCK do not complete within 24 hours. MUZE also outperforms ULTRA while requiring less GPU memory. This demonstrates that MUZE remains effective and computationally efficient on a million-scale KG.

Scaling Behavior We investigate how MUZE scales with model size and the number of pre-training KGs. To evaluate scaling with model size, we pre-train MUZE with embedding dimensions of 32, 64, and 128 and report the average zero-shot performance across 54 KGs. As shown in Table 4, increasing the embedding dimension improves both entity and relation prediction. We also vary the

Table 4: Scaling of MUZE with model dimension (zero-shot MRRs over 54 KGs).

	32	64	128
Ent.	0.364	0.383	**0.401**
Rel.	0.908	0.915	**0.919**

Table 5: Scaling of MUZE with the number of pre-training KGs (zero-shot MRRs over 41 inductive KGs).

	1	2	3	4
Ent.	0.396	0.407	0.423	**0.430**
Rel.	0.921	0.919	0.927	**0.935**

Table 6: Effect of Monte Carlo sample size on zero-shot MRR of MUZE over 54 KGs.

	1	2	4	8	16	32	64	128
Ent.	0.371	0.382	0.393	0.397	0.400	0.401	0.401	0.402
Rel.	0.899	0.904	0.911	0.917	0.918	0.919	0.919	0.919

number of pre-training KGs from one to four and evaluate the average zero-shot performance on 41 inductive KGs. We restrict this evaluation to inductive KGs to avoid fairness issues caused by using transductive KGs for pre-training. Table 5 shows that performance tends to improve as the number of pre-training KGs increases. These results indicate that MUZE tends to benefit from both increased model capacity and additional pre-training data within the evaluated ranges.

6.3 MONTE CARLO SAMPLING AND TASK TRANSFER

Effect of Monte Carlo Sample Size Since MUZE starts from randomly initialized entity and relation representations, we examine its sensitivity to the number of Monte Carlo samples used during inference. As shown in Table 6, increasing the number of samples steadily improves zero-shot performance before plateauing at a stable convergence point. This indicates that averaging over multiple random initializations mitigates their effect on prediction performance. Appendix D shows additional analysis of multi-seed variance of MUZE and runtime across varying numbers of queries.

Task Transfer To examine whether the representations computed by MUZE transfer beyond KG completion, we evaluate entity alignment on OpenEA (Sun et al., 2020). We use the same checkpoint pre-trained on three KGs for link prediction, without additional training or fine-tuning. We merge the source and target KGs by combining each training alignment pair into a single entity node, encode the unified KG with MUZE, and rank candidate alignments for the remaining entities using the dot-product similarity of their embeddings. As shown in Table 7, MUZE outperforms ULTRA-LP and ULTRA-EA. ULTRA-LP is pre-trained for link prediction, whereas ULTRA-EA is pre-trained for entity alignment. While MUZE is outperformed by EAFM, a model designed and pre-trained specifically for entity alignment, these results demonstrate that its link-prediction-pretrained representations transfer to entity alignment without task-specific parameter learning.

Table 7: Entity alignment results on OpenEA. Baseline results are adopted from Cui et al. (2026).

	MRR	Hit@1	Hit@10
ULTRA-LP	0.013	0.001	0.037
ULTRA-EA	0.477	0.385	0.635
EAFM	**0.602**	**0.517**	**0.764**
MUZE	<u>0.558</u>	<u>0.477</u>	<u>0.706</u>

6.4 QUALITATIVE ANALYSIS

Figure 3 shows t-SNE (van der Maaten & Hinton, 2008) plots of the entity representations in the NELL995 dataset computed by MUZE under zero-shot inference, where each point represents an entity embedding and the colors represent entity categories. Among the categories provided in NELL (Carlson et al., 2010), we select five to visualize: *country, sport, chemical, furniture*, and *mammal*. Starting from a random Gaussian initialization at $t = 0$, MUZE iteratively updates the embeddings $\mathbf{x}_t$. As the layers progress, representations of entities belonging to the same category gradually group together, separating from those in other categories. At the final layer ($t = 12$), the entities form five clusters, each corresponding to its respective category. This validates that MUZE successfully generalizes to unseen KGs, effectively computing meaningful representations.

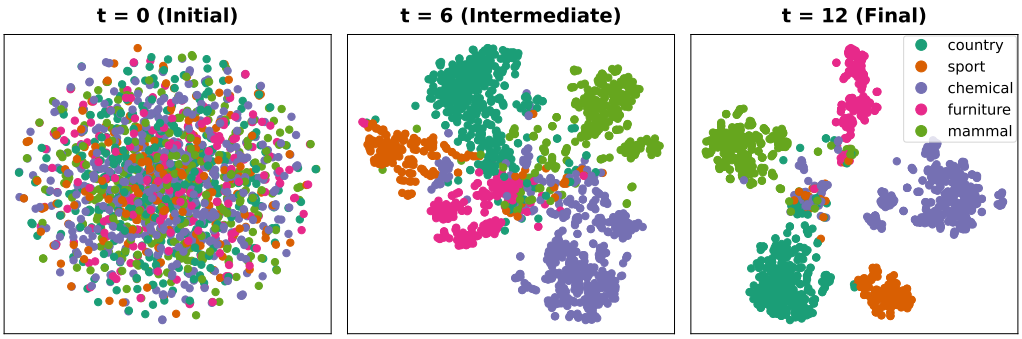

Figure 3: t-SNE visualization of the entity representations ($\mathbf{x}_t$) updated by MUZE at different layers ($t = 0, 6, 12$) on the NELL995 dataset under a zero-shot setting. As the layers proceed, the initially mixed random embeddings are refined and separated into five clusters.

6.5 Ablation Study

We perform ablation studies to validate representation update components of MUZE and the joint training objective. (i) removes $\text{MLP}_{\text{scale}}$ to examine the effect of the dimension-wise scaling of the normalized message $\widetilde{\mathbf{m}}_{x,t}$. (ii) removes the gating function MLP_{gate} that adaptively controls the nonlinear residual transformation, while (iii) bypasses the entire gated residual refinement and directly uses $\mathbf{u}_{x,t}$ as the representation of x at the t-th layer. Removing the adaptive scaling or residual refinement degrades performance on both tasks, with the largest entity-prediction drop occurring when the residual refinement is entirely removed. (iv) trains MUZE only for entity prediction, while (v) trains MUZE only for relation prediction. Single-task training retains most of the performance on its supervised task but substantially degrades performance on the other task. Interestingly, performance on the task without direct supervision remains non-trivial, suggesting that coupling message-passing-based encoding with KGE-based decoding already guides the representations to align with the shared decoder. These results show that the representation-update components are beneficial and that joint training enables a single MUZE model to effectively support both entity and relation prediction.

Table 8: Ablation Study of MUZE. Zero-shot MRRs for entity prediction (Ent.) and relation prediction (Rel.) on 54 KGs are reported.

	Ent.	Rel.
(i) w/o $\text{MLP}_{\text{scale}}$	0.377	0.910
(ii) w/o MLP_{gate}	0.369	0.904
(iii) $\mathbf{x}_t = \mathbf{u}_{x,t}$	0.342	0.908
(iv) w/o Rel. Pred.	0.396	0.799
(v) w/o Ent. Pred.	0.260	0.914
MUZE	**0.401**	**0.919**

7 Conclusion and Future Work

We present MUZE, the first query-agnostic KGFM that couples KGE-based decoding with message-passing-based encoding through a shared KGE scoring function. We demonstrate that MUZE achieves competitive performance in the zero-shot setting for both entity and relation prediction tasks, while accelerating inference by $11\times$ over the fastest baseline and by $1389\times$ over the baseline with the highest MRR. Furthermore, we theoretically prove that MUZE is a universal encoder for KGs. When paired with a fully expressive decoder, MUZE is a universal approximator for task-specific functions, highlighting its potential utility in tasks beyond KG completion. By bridging classical KGE models and KGFMs, our work offers a unified perspective that paves the way for future research on KGs.

Building upon these contributions, several avenues for future work remain. While we have shown the universal approximation property of MUZE, the theoretical conditions for its convergence to the optimal parameters remain to be characterized. Also, the relatively modest accuracy gains observed for entity prediction under the finetuning setting suggest that dedicated finetuning strategies could unlock even greater potential. We also plan to systematically investigate the interplay between our framework and various KGE scoring functions. Finally, extending MUZE to encompass broader downstream KG tasks, such as complex logical query answering, will further generalize our approach.

AI USE STATEMENT

In this work, we have not used generative AI tools for assisting with translation, helping develop theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, proposing or refining hypotheses, designing or providing feedback on research methodology or experiments, implementing methods, supporting qualitative and thematic data analysis, interpreting results, and generating synthetic data sets and cleaning and reformatting datasets are not applicable to this work. Additionally, we used generative AI tools for editing a research paper to improve readability. We have reviewed all AI-assisted work. We have verified all AI-written materials. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work introduces MUZE, a highly efficient and expressive framework for KGFMs. By achieving competitive zero-shot performance with significantly reduced inference time, our method substantially lowers the computational barriers and energy consumption required for deploying large-scale relational reasoning systems. The improved efficiency of KGFMs can accelerate progress in critical domains that may heavily rely on complex relational data, such as biomedical research, drug discovery, and intelligent recommendation systems, particularly in resource-constrained environments.

However, the enhanced predictive power and zero-shot generalization capabilities of MUZE also necessitate careful consideration of potential negative societal impacts. As with any model trained on large-scale, real-world datasets, our framework may inadvertently propagate existing historical and demographic biases that might be present in the underlying knowledge graphs. Therefore, we emphasize the importance of utilizing MUZE as an assistive reasoning tool rather than a fully autonomous decision-maker. We encourage future research and real-world deployments to be accompanied by rigorous bias evaluation, fairness mitigation strategies, and human-in-the-loop validation to ensure safe, transparent, and equitable applications.

REPRODUCIBILITY STATEMENT

We describe the architecture and implementation of MUZE in Section 4. Complete proofs of our theoretical results are provided in Appendix A. The experimental setup, dataset statistics, training and finetuning procedures, and hyperparameters are detailed in Section 6 and Appendix B. Our implementation is publicly available at `https://anonymous.4open.science/r/muze-i6`.

REFERENCES

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Proceedings of the 27th Conference on Neural Information Processing Systems*, pp. 2787–2795, 2013.

Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam Hruschka, and Tom Mitchell. Toward an architecture for never-ending language learning. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, pp. 1306–1313, 2010.

Yihong Chen, Pasquale Minervini, Sebastian Riedel, and Pontus Stenetorp. Relation prediction as an auxiliary training objective for improving multi-relational graph representations. In *Proceedings of the 3rd Conference on Automated Knowledge Base Construction*, 2021.

Yihong Chen, Pushkar Mishra, Luca Franceschi, Pasquale Minervini, Pontus Lars Erik Saito Stenetorp, and Sebastian Riedel. ReFactor GNNs: Revisiting factorisation-based models from a message-passing perspective. In *Proceedings of the 36th Conference on Neural Information Processing Systems*, pp. 16138–16150, 2022.

Yuanning Cui, Zequn Sun, and Wei Hu. A prompt-based knowledge graph foundation model for universal in-context reasoning. In *Proceedings of the 38th Conference on Neural Information Processing Systems*, pp. 7095–7124, 2024.

Yuanning Cui, Zequn Sun, Wei Hu, Kexuan Xin, and Zhangjie Fu. Breaking the reasoning horizon in entity alignment foundation models. *arXiv preprint arXiv:2601.21174*, 2026.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Convolutional 2d knowledge graph embeddings. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, pp. 1811–1818, 2018.

Boyang Ding, Quan Wang, Bin Wang, and Li Guo. Improving knowledge graph embedding using simple constraints. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 110–121, 2018.

Mikhail Galkin, Max Berrendorf, and Charles Tapley Hoyt. An open challenge for inductive link prediction on knowledge graphs. *arXiv preprint arXiv:2203.01520*, 2022.

Mikhail Galkin, Xinyu Yuan, Hesham Mostafa, Jian Tang, and Zhaocheng Zhu. Towards foundation models for knowledge graph reasoning. In *Proceedings of the 12th International Conference on Learning Representations*, 2024.

Takuo Hamaguchi, Hidekazu Oiwa, Masashi Shimbo, and Yuji Matsumoto. Knowledge transfer for out-of-knowledge-base entities : A graph neural network approach. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pp. 1802–1808, 2017.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (GELUs). *arXiv preprint arXiv:1606.08415*, 2016.

Daniel Scott Himmelstein, Antoine Lizee, Christine Hessler, Leo Brueggeman, Sabrina L Chen, Dexter Hadley, Ari Green, Pouya Khankhanian, and Sergio E Baranzini. Systematic integration of biomedical knowledge prioritizes drugs for repurposing. *eLife*, 6:e26726, 2017.

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359 – 366, 1989.

Xingyue Huang, Pablo Barcelo, Michael M. Bronstein, Ismail Ilkan Ceylan, Mikhail Galkin, Juan L Reutter, and Miguel Romero Orth. How expressive are knowledge graph foundation models? In *Proceedings of the 42nd International Conference on Machine Learning*, pp. 25021–25058, 2025.

Seyed Mehran Kazemi and David Poole. SimplE embedding for link prediction in knowledge graphs. In *Proceedings of the 32nd Conference on Neural Information Processing Systems*, pp. 4284–4295, 2018.

Jinwoo Kim, Xingyue Huang, Krzysztof Olejniczak, Kyungbin Min, Michael M. Bronstein, Seunghoon Hong, and Ismail Ilkan Ceylan. Flock: A knowledge graph foundation model via learning on random walks. In *Proceedings of the 14th International Conference on Learning Representations*, 2026.

Timothee Lacroix, Nicolas Usunier, and Guillaume Obozinski. Canonical tensor decomposition for knowledge base completion. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 2863–2872, 2018.

Jaejun Lee and Joyce Jiyoung Whang. Structure Is All You Need: Structural representation learning on hyper-relational knowledge graphs. In *Proceedings of the 42nd International Conference on Machine Learning*, pp. 33765–33783, 2025.

Jaejun Lee, Chanyoung Chung, and Joyce Jiyoung Whang. InGram: Inductive knowledge graph embedding via relation graphs. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 18796–18809, 2023.

Rui Li, Jianan Zhao, Chaozhuo Li, Di He, Yiqi Wang, Yuming Liu, Hao Sun, Senzhang Wang, Weiwei Deng, Yanming Shen, Xing Xie, and Qi Zhang. HousE: Knowledge graph embedding with householder parameterization. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 13209–13224, 2022.

Rui Li, Chaozhuo Li, Yanming Shen, Zeyu Zhang, and Xu Chen. Generalizing knowledge graph embedding with universal orthogonal parameterization. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 28040–28059, 2024.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, pp. 2181–2187, 2015.

Hanxiao Liu, Yuexin Wu, and Yiming Yang. Analogical inference for multi-relational embeddings. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 2168–2178, 2017.

Shuwen Liu, Bernardo Grau, Ian Horrocks, and Egor Kostylev. INDIGO: Gnn-based inductive knowledge graph completion using pair-wise encoding. In *Proceedings of the 35th Conference on Neural Information Processing Systems*, pp. 2034–2045, 2021.

Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *Proceedings of the 5th International Conference on Learning Representations*, 2017.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Proceedings of the 7th International Conference on Learning Representations*, 2019.

Xin Lv, Xu Han, Lei Hou, Juanzi Li, Zhiyuan Liu, Wei Zhang, Yichi Zhang, Hao Kong, and Suhui Wu. Dynamic anticipation and completion for multi-hop reasoning over sparse knowledge graph. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pp. 5694–5703, 2020.

Chaitanya Malaviya, Chandra Bhagavatula, Antoine Bosselut, and Yejin Choi. Commonsense knowledge base completion with structural and semantic context. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, pp. 2925–2933, 2020.

George Armitage Miller. WordNet: a lexical database for English. *Communications of the ACM*, 38 (11):39–41, 1995. doi: 10.1145/219717.219748.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *Proceedings of the 33rd Conference on Neural Information Processing Systems*, pp. 8024–8035, 2019.

Tara Safavi and Danai Koutra. CoDEx: A comprehensive knowledge graph completion benchmark. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pp. 8328–8350, 2020.

Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *Proceedings of the 15th Extended Semantic Web Conference*, pp. 593–607, 2018.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: A core of semantic knowledge. In *Proceedings of the 16th International Conference on World Wide Web*, pp. 697–706, 2007.

Zequn Sun, Qingheng Zhang, Wei Hu, Chengming Wang, Muhao Chen, Farahnaz Akrami, and Chengkai Li. A benchmarking study of embedding-based entity alignment for knowledge graphs. *Proceedings of the VLDB Endowment*, 13(11):2326 – 2340, 2020.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. RotatE: Knowledge graph embedding by relational rotation in complex space. In *Proceedings of the 7th International Conference on Learning Representations*, 2019.

Komal Teru, Etienne Denis, and Will Hamilton. Inductive relation prediction by subgraph reasoning. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 9448–9457, 2020.

Kristina Toutanova and Danqi Chen. Observed versus latent features for knowledge base and text inference. In *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*, pp. 57–66, 2015.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *Proceedings of the 33rd International Conference on Machine Learning*, pp. 2071–2080, 2016.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008.

Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha Talukdar. Composition-based multi-relational graph convolutional networks. In *Proceedings of the 8th International Conference on Learning Representations*, 2020.

Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, 29(12): 2724 – 2743, 2017.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, pp. 1112–1119, 2014.

Wenhan Xiong, Thien Hoang, and William Yang Wang. DeepPath: A reinforcement learning method for knowledge graph reasoning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 564–573, 2017.

Bishan Yang, Wen tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. In *Proceedings of the 3rd International Conference on Learning Representations*, 2015.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. In *Proceedings of the 33rd Conference on Neural Information Processing Systems*, pp. 12381–12392, 2019.

Yongqi Zhang and Quanming Yao. Knowledge graph reasoning with relational digraph. In *Proceedings of The ACM Web Conference 2022*, pp. 912–924, 2022.

Yucheng Zhang, Beatrice Bevilacqua, Mikhail Galkin, and Bruno Ribeiro. TRIX: A more expressive model for zero-shot domain transfer in knowledge graphs. In *Proceedings of the Third Learning on Graphs Conference*, pp. 12:1–12:28, 2025.

Jincheng Zhou, Beatrice Bevilacqua, and Bruno Ribeiro. A multi-task perspective for link prediction with new relation types and nodes. *arXiv preprint arXiv:2307.06046*, 2023.

Zhaocheng Zhu, Zuobai Zhang, Louis-Pascal Xhonneux, and Jian Tang. Neural Bellman-Ford networks: A general graph neural network framework for link prediction. In *Proceedings of the 35th Conference on Neural Information Processing Systems*, pp. 29476–29490, 2021.

A PROOFS OF SECTION 5.1

We provide the detailed proofs for the theoretical results presented in Section 5.1.

A.1 PROOF OF THEOREM 1

Proof. Let $\mathcal{S} = \bigcup_{G=(\mathcal{V},\mathcal{R},\mathcal{T})\in\mathcal{G}} \bigcup_{x\in\mathcal{V}\cup\mathcal{R}} \{\mathbf{x}_0\}$ be the set of all initial random embeddings for all entities and relations across all KGs in the finite set $\mathcal{G}$. Since $\mathcal{G}$ is finite and each $G \in \mathcal{G}$ contains a finite number of entities and relations, the set $\mathcal{S}$ is finite. By the assumption of distinguishable random initializations, all elements in $\mathcal{S}$ are distinct. That is, for any two different elements x and y, $\mathbf{x}_0 \neq \mathbf{y}_0$.

Let $\mathbf{x}^*$ denote the target embedding of x in $\boldsymbol{E}^*_G$. Because all points in the finite set $\mathcal{S}$ are uniquely distinguishable, there exists a well-defined deterministic mapping $h : \mathcal{S} \to \mathbb{R}^d$ such that $h(\mathbf{x}_0) = \mathbf{x}^*$ for all x. Since $\mathcal{S}$ is a finite discrete subset of $\mathbb{R}^d$, we can extend h to a continuous function $F : \Omega \to \mathbb{R}^d$, where $\Omega \subset \mathbb{R}^d$ is a compact domain that entirely contains $\mathcal{S}$.

By the Universal Approximation Theorem (Hornik et al., 1989), an MLP with at least one hidden layer and a non-linear activation function can approximate any continuous function on a compact set to arbitrary precision. Let $N = \max_{G=(\mathcal{V},\mathcal{R},\mathcal{T})\in\mathcal{G}}(|\mathcal{V}| + |\mathcal{R}|)$ be the maximum number of total entities and relations in any KG within $\mathcal{G}$. We set a strictly positive approximation error bound $\epsilon' = \frac{\epsilon_{\text{err}}}{\sqrt{N}}$. According to the Universal Approximation Theorem, there exists a parameter configuration of MLP_{upd} such that for all $\mathbf{x} \in \Omega$, $\|\text{MLP}_{\text{upd}}(\mathbf{x}) - (F(\mathbf{x}) - \mathbf{x})\|_2 < \epsilon'$.

Now, we configure the remaining hyperparameters, the scoring function f used for message passing, and the learnable parameters of MUZE to eliminate updates other than those from MLP_{upd}. We set $\mathbf{a}_{x,t} = 0$ by utilizing a constant scoring function $f(\mathbf{s}, \mathbf{p}, \mathbf{o}) = 0$ used for aggregation, and set $\widetilde{\lambda} = 0$. This yields $\mathbf{u}_{x,1} = \mathbf{x}_0$. Also, by setting all parameters in MLP_{gate} to 0 and configuring $\eta_1 = 1$ and $\widetilde{\gamma} = 2$, the coefficient of the residual update becomes $2\sigma(0) = 1$. Under this configuration, the update at the first layer simplifies to $\mathbf{x}_1 = \mathbf{x}_0 + \text{MLP}_{\text{upd}}(\mathbf{x}_0)$. By setting $K = 1$, we ensure that $\mathbf{x}_1$ is the final output of MUZE.

For every element x, the squared Euclidean distance to the target embedding is bounded by:

$$\|\mathbf{x}_1 - \mathbf{x}^*\|_2^2 = \|\mathbf{x}_0 + \text{MLP}_{\text{upd}}(\mathbf{x}_0) - F(\mathbf{x}_0)\|_2^2 = \|\text{MLP}_{\text{upd}}(\mathbf{x}_0) - (F(\mathbf{x}_0) - \mathbf{x}_0)\|_2^2 < (\epsilon')^2 = \frac{\epsilon_{\text{err}}^2}{N} \quad (7)$$

Finally, we compute the Frobenius norm of the error matrix for the entire KG G. Summing the squared errors over all entities and relations in $G = (\mathcal{V}, \mathcal{R}, \mathcal{T})$, we obtain:

$$\|\boldsymbol{E}_G - \boldsymbol{E}^*_G\|_F = \sqrt{\sum_{x\in\mathcal{V}\cup\mathcal{R}} \|\mathbf{x}_1 - \mathbf{x}^*\|_2^2} < \sqrt{N \cdot \frac{\epsilon_{\text{err}}^2}{N}} = \epsilon_{\text{err}} \quad (8)$$

This inequality holds uniformly for every $G \in \mathcal{G}$ under the same parameter configuration of MUZE. □

A.2 PROOF OF COROLLARY 1

Proof. Let f^*_G be the ground-truth function for a given task on a specific KG $G \in \mathcal{G}$. By the assumption that the decoder, denoted as $\mathcal{D}$, is fully expressive, there exists a set of optimal target embeddings $\boldsymbol{E}^*_G$ such that $\mathcal{D}$ perfectly reconstructs the target function: $\mathcal{D}(\boldsymbol{E}^*_G) = f^*_G$. Since we assume a continuous decoder, for any arbitrarily small error tolerance $\epsilon_{\text{err}} > 0$, there exists an intermediate embedding error bound $\epsilon_{\text{emb}} > 0$ such that for any generated embedding matrix $\boldsymbol{E}_G$, if $\|\boldsymbol{E}_G - \boldsymbol{E}^*_G\|_F < \epsilon_{\text{emb}}$, the output functional error is strictly bounded by $\|\mathcal{D}(\boldsymbol{E}_G) - \mathcal{D}(\boldsymbol{E}^*_G)\| < \epsilon_{\text{err}}$.

According to Theorem 1, given distinguishable random initializations, there exists a single parameter configuration of MUZE that produces output embeddings $\boldsymbol{E}_G$ satisfying $\|\boldsymbol{E}_G - \boldsymbol{E}^*_G\|_F < \epsilon_{\text{emb}}$ uniformly for all $G \in \mathcal{G}$. By composing the output embeddings of MUZE with the continuous and fully expressive decoder $\mathcal{D}$, we obtain:

$$\|\mathcal{D}(\boldsymbol{E}_G) - f^*_G\| = \|\mathcal{D}(\boldsymbol{E}_G) - \mathcal{D}(\boldsymbol{E}^*_G)\| < \epsilon_{\text{err}} \quad (9)$$

This inequality holds for every $G \in \mathcal{G}$ with the same parameter configuration of MUZE. Consequently, when paired with a continuous and fully expressive decoder, MUZE can approximate any task-specific function to an arbitrary degree of accuracy ϵ_{err}. □

B EXPERIMENTAL DETAILS

We provide dataset statistics used in our experiments and additional implementation details for MUZE.

B.1 DATASET STATISTICS

Tables 9, 10, and 11 report the statistics of the KG datasets designed for the 'Inductive $|\mathcal{V}|$, $|\mathcal{R}|$', 'Inductive $|\mathcal{V}|$', and 'Transductive' settings, respectively. For the inductive datasets, we report the statistics of the training KG $G_{\text{tr}} = (\mathcal{V}_{\text{tr}}, \mathcal{R}_{\text{tr}}, \mathcal{T}_{\text{tr}})$ used for training, the validation KG $G_{\text{val}} = (\mathcal{V}_{\text{val}}, \mathcal{R}_{\text{val}}, \mathcal{T}_{\text{val}})$ used for validation, and the test KG $G_{\text{test}} = (\mathcal{V}_{\text{test}}, \mathcal{R}_{\text{test}}, \mathcal{T}_{\text{test}})$ used for test, respectively. $\mathcal{F}_{\text{val}}$ and $\mathcal{F}_{\text{test}}$ denote the sets of triplets used to evaluate validation and test performance, respectively. By default, entity prediction includes both head and tail entities, except for the three datasets in the Transductive category (as indicated in Table 11), where only tail entity prediction is evaluated following existing literature.

B.2 ADDITIONAL IMPLEMENTATION DETAILS OF MUZE

We use PyTorch 2.6 (Paszke et al., 2019) and CUDA 11.8. Following the standard practice (Vashishth et al., 2020) for KGs, we added inverse triplets to all datasets. MUZE was pre-trained on a single NVIDIA GeForce RTX 3090 GPU. For finetuning, we utilized an NVIDIA RTX A6000 for the Transductive datasets, and an RTX 3090 for both the Inductive $\mathcal{V}$ and Inductive $\mathcal{V}, \mathcal{R}$ settings. Throughout both the pre-training and finetuning phases, the model was optimized jointly for entity and relation prediction. Consequently, a single, unified model was used to evaluate both tasks on all datasets.

For model optimization, we employed the AdamW optimizer. Within each layer, we clip the aggregated gradient-based message $\mathbf{a}_{x,t}$ independently for each entity and relation at a threshold c, based on their respective root mean square (RMS) values, i.e., $\sqrt{\|\mathbf{a}_{x,t}\|_2^2/d}$. Furthermore, weight decay λ was exclusively applied to the MLP parameters.

We adopted distinct sampling strategies for pre-training and finetuning. During pre-training, at each step, we first randomly sampled one of the three pre-training KGs. We then dynamically split the sampled KG into two disjoint sets: one for message passing and one for computing the loss. To prevent the message passing subset from becoming overly small, the split ratio was drawn from a uniform distribution $\mathcal{U}(0, 1)$ and then square-rooted. From the loss-computation split, we randomly sampled a mini-batch of B triplets, discarding the remainder for that step. During finetuning, however, we employed a fixed split ratio δ and dynamically resplit the KG at the beginning of each epoch. All triplets assigned to the loss-computation split were utilized for training, processed in mini-batches of size B. MUZE was pre-trained for a maximum of 150,000 steps with an early stopping patience of 30,000 steps. The pre-training of MUZE requires approximately 13.5 hours, consuming a peak memory of 20GB.[1] For finetuning, the model was trained for up to 10 epochs, applying an early stopping patience of 2 epochs.

Table 12 details the hyperparameters of MUZE used for pre-training and finetuning. In the table, 'lr' denotes the constant learning rate used during finetuning, while N/A indicates hyperparameters that are not applicable to pre-training. Across all experiments, we set the dimension $d = 128$, the number of layers $K = 12$, the number of negative samples $n = 1$, and the number of layers for MLP_{upd} to 3, and the number of layers for the other MLPs to 2. We utilized GELU (Hendrycks & Gimpel, 2016) activations and applied a gradient clipping threshold of 1.0 for training MUZE. For pre-training, we employed a cosine learning rate scheduler with a linear warmup over the first 15,000 steps (Loshchilov & Hutter, 2017). The maximum and the minimum learning rates for the scheduler were set to 0.001 and 0.0005, respectively. The best validation performance was achieved at step 145,000. During finetuning, a constant learning rate was maintained across epochs.

[1] For comparison, it is reported that FLOCK (Kim et al., 2026) typically takes about three days and 28GB of GPU memory for pre-training.

Table 9: Statistics for KGs in the 'Inductive $\mathcal{V}$, $\mathcal{R}$' category.

	$\lvert\mathcal{V}_{\text{tr}}\rvert$	$\lvert\mathcal{R}_{\text{tr}}\rvert$	$\lvert\mathcal{T}_{\text{tr}}\rvert$	$\lvert\mathcal{V}_{\text{val}}\rvert$	$\lvert\mathcal{R}_{\text{val}}\rvert$	$\lvert\mathcal{T}_{\text{val}}\rvert$	$\lvert\mathcal{F}_{\text{val}}\rvert$	$\lvert\mathcal{V}_{\text{test}}\rvert$	$\lvert\mathcal{R}_{\text{test}}\rvert$	$\lvert\mathcal{T}_{\text{test}}\rvert$	$\lvert\mathcal{F}_{\text{test}}\rvert$
FB-25 (2023)	5190	163	91571	4097	216	17147	5716	4097	216	17147	5716
FB-50 (2023)	5190	153	85375	4445	205	11636	3879	4445	205	11636	3879
FB-75 (2023)	4659	134	62809	2792	186	9316	3106	2792	186	9316	3106
FB-100 (2023)	4659	134	62809	2624	77	6987	2329	2624	77	6987	2329
WK-25 (2023)	12659	47	41873	3228	74	3391	1130	3228	74	3391	1131
WK-50 (2023)	12022	72	82481	9328	93	9672	3224	9328	93	9672	3225
WK-75 (2023)	6853	52	28741	2722	65	3430	1143	2722	65	3430	1144
WK-100 (2023)	9784	67	49875	12136	37	13487	4496	12136	37	13487	4496
NL-0 (2023)	1814	134	7796	2026	112	2287	763	2026	112	2287	763
NL-25 (2023)	4396	106	17578	2146	120	2230	743	2146	120	2230	744
NL-50 (2023)	4396	106	17578	2335	119	2576	859	2335	119	2576	859
NL-75 (2023)	2607	96	11058	1578	116	1818	606	1578	116	1818	607
NL-100 (2023)	1258	55	7832	1709	53	2378	793	1709	53	2378	793
MT1-tax (2023)	10000	10	17178	10000	10	17178	1908	10000	9	16526	1834
MT1-health (2023)	10000	7	14371	10000	7	14371	1596	10000	7	14110	1566
MT2-org (2023)	10000	10	23233	10000	10	23233	2581	10000	11	21976	2441
MT2-sci (2023)	10000	16	16471	10000	16	16471	1830	10000	16	14852	1650
MT3-art (2023)	10000	45	27262	10000	45	27262	3026	10000	45	28023	3113
MT3-infra (2023)	10000	24	21990	10000	24	21990	2443	10000	27	21646	2405
MT4-sci (2023)	10000	42	12576	10000	42	12576	1397	10000	42	12516	1388
MT4-health (2023)	10000	21	15539	10000	21	15539	1725	10000	20	15337	1703
Metafam (2023)	1316	28	13821	1316	28	13821	590	656	28	7257	184
FBNELL (2023)	4636	100	10275	4636	100	10275	1055	4752	183	10685	597

Table 10: Statistics for KGs in the 'Inductive $\mathcal{V}$' category.

	$\lvert\mathcal{R}\rvert$	$\lvert\mathcal{V}_{\text{tr}}\rvert$	$\lvert\mathcal{T}_{\text{tr}}\rvert$	$\lvert\mathcal{V}_{\text{val}}\rvert$	$\lvert\mathcal{T}_{\text{val}}\rvert$	$\lvert\mathcal{F}_{\text{val}}\rvert$	$\lvert\mathcal{V}_{\text{test}}\rvert$	$\lvert\mathcal{T}_{\text{test}}\rvert$	$\lvert\mathcal{F}_{\text{test}}\rvert$
FB-v1 (2020)	180	1594	4245	1594	4245	489	1093	1993	411
FB-v2 (2020)	200	2608	9739	2608	9739	1166	1660	4145	947
FB-v3 (2020)	215	3668	17986	3668	17986	2194	2501	7406	1731
FB-v4 (2020)	219	4707	27203	4707	27203	3352	3051	11714	2840
WN-v1 (2020)	9	2746	5410	2746	5410	630	922	1618	373
WN-v2 (2020)	10	6954	15262	6954	15262	1838	2757	4011	852
WN-v3 (2020)	11	12078	25901	12078	25901	3097	5084	6327	1143
WN-v4 (2020)	9	3861	7940	3861	7940	934	7084	12334	2823
NL-v1 (2020)	14	3103	4687	3103	4687	414	225	833	201
NL-v2 (2020)	88	2564	8219	2564	8219	922	2086	4586	935
NL-v3 (2020)	142	4647	16393	4647	16393	1851	3566	8048	1620
NL-v4 (2020)	76	2092	7546	2092	7546	876	2795	7073	1447
HM 1k (2017)	11	36237	93364	36311	93364	1771	9899	18638	476
HM 3k (2017)	11	32118	71097	32250	71097	1201	19218	38285	1349
HM 5k (2017)	11	28601	57601	28744	57601	900	23792	48425	2124
Indigo-BM (2021)	229	12721	121601	12797	121601	14121	14775	250195	14904
ILPC Small (2022)	48	10230	78616	6653	20960	2908	6653	20960	2902
ILPC Large (2022)	65	46626	202446	29246	77044	10179	29246	77044	10184

For the scaling analysis in Sec. 6.3, we progressively increase the number of pre-training KGs using nested subsets: FB15K237 for one KG; FB15K237 and WN18RR for two; FB15K237, WN18RR, and CoDEx-M for three; and FB15K237, WN18RR, CoDEx-M, and NELL995 for four.

Table 11: Statistics for KGs in the 'Transductive' category. Entity Position denotes the position of the prediction for entity prediction task.

	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{R}\rvert$	$\lvert\mathcal{T}_{\text{tr}}\rvert$	$\lvert\mathcal{F}_{\text{val}}\rvert$	$\lvert\mathcal{F}_{\text{test}}\rvert$	Entity Position
FB15K237 (2015)	14541	237	272115	17535	20466	head/tail
WN18RR (2018)	40943	11	86835	3034	3134	head/tail
CoDEx-M (2020)	17050	51	185584	10310	10311	head/tail
NELL995 (2017)	74536	200	149678	543	2818	head/tail
NELL23k (2020)	22925	200	25445	4961	4952	head/tail
WDsinger (2020)	10282	135	16142	2163	2203	head/tail
ConceptNet100k (2020)	78334	34	100000	1200	1200	head/tail
CoDEx-S (2020)	2034	42	32888	1827	1828	head/tail
CoDEx-L (2020)	77951	69	551193	30622	30622	head/tail
YAGO3-10 (2007)	123182	37	1079040	5000	5000	head/tail
AristoV4 (2021)	44949	1605	242567	20000	20000	head/tail
DBpedia100k (2018)	99604	470	597572	50000	50000	head/tail
Hetionet (2017)	45158	24	2025177	112510	112510	head/tail
FB15K237(10) (2020)	11512	237	27211	15624	18150	tail
FB15K237(20) (2020)	13166	237	54423	16963	19776	tail
FB15K237(50) (2020)	14149	237	136057	17449	20324	tail

Table 12: Hyperparameters of MUZE for pre-training and finetuning on 57 KG datasets.

	lr	dropout	B	c	δ	epoch	label smoothing	weight decay
Pre-training	N/A	0.05	1024	1.0	N/A	N/A	0.1	0.01
FB-25	0.001	0.05	1024	1.0	0.4	3	0.01	0.0
FB-50	0.001	0.05	512	10.0	0.3	6	0.01	0.001
FB-75	0.0003	0.01	1024	1.0	0.3	1	0.0	0.0001
FB-100	0.0002	0.05	1024	1.0	0.2	1	0.0	0.0001
WK-25	0.002	0.1	1024	1.0	0.1	4	0.0	0.0001
WK-50	0.0005	0.1	2048	1.0	0.4	3	0.0	0.01
WK-75	0.0002	0.01	1024	1.0	0.5	2	0.0	0.01
WK-100	0.0005	0.1	2048	1.0	0.7	1	0.0	0.01
NL-0	0.002	0.05	1024	1.0	0.5	3	0.0	0.001
NL-25	0.001	0.05	1024	1.0	0.3	4	0.0	0.001
NL-50	0.001	0.05	1024	1.0	0.3	6	0.0	0.01
NL-75	0.001	0.05	1024	1.0	0.3	7	0.0	0.01
NL-100	0.001	0.01	1024	1.0	0.4	2	0.0	0.01
MT1-tax	0.001	0.01	1024	1.0	0.3	5	0.1	0.01
MT1-health	0.0005	0.01	1024	1.0	0.3	10	0.0	0.01
MT2-org	0.0005	0.01	1024	1.0	0.45	5	0.0	0.1
MT2-sci	0.001	0.01	1024	1.0	0.3	6	0.0	0.01
MT3-art	0.002	0.01	1024	1.0	0.4	6	0.01	0.01
MT3-infra	0.001	0.01	512	1.0	0.55	9	0.01	0.00001
MT4-sci	0.0004	0.01	128	1.0	0.45	5	0.0	0.0
MT4-health	0.0005	0.01	256	1.0	0.5	10	0.1	0.00001
Metafam	0.001	0.01	256	1.0	0.7	10	0.05	0.0001
FBNELL	0.0015	0.01	256	1.0	0.5	5	0.05	0.001
FB-v1	0.001	0.01	1024	1.0	0.5	7	0.1	0.01
FB-v2	0.002	0.01	1024	1.0	0.6	6	0.0	0.001
FB-v3	0.0005	0.01	512	1.0	0.5	5	0.0	0.01
FB-v4	0.0004	0.01	128	1.0	0.55	10	0.0	0.001
WN-v1	0.001	0.01	256	1.0	0.6	6	0.0	0.01
WN-v2	0.0005	0.01	1024	1.0	0.4	4	0.0	0.001
WN-v3	0.001	0.01	1024	1.0	0.5	3	0.0	0.001
WN-v4	0.001	0.01	512	1.0	0.4	6	0.0	0.00001
NL-v1	0.001	0.01	512	1.0	0.5	6	0.01	0.001
NL-v2	0.0015	0.01	512	1.0	0.6	5	0.01	0.0001
NL-v3	0.001	0.01	256	1.0	0.5	9	0.0	0.0001
NL-v4	0.001	0.05	1024	1.0	0.5	10	0.0	0.001
HM 1k	0.0005	0.01	512	1.0	0.3	9	0.0	0.1
HM 3k	0.0004	0.01	256	1.0	0.35	10	0.0	0.0001
HM 5k	0.001	0.01	2048	1.0	0.3	2	0.0	0.01
Indigo-BM	0.0005	0.01	512	1.0	0.6	4	0.0	0.01
ILPC Small	0.001	0.01	1024	1.0	0.3	4	0.0	0.01
ILPC Large	0.0004	0.01	256	1.0	0.35	8	0.0	0.001
NELL995	0.0002	0.1	1024	1.0	0.5	5	0.0	0.1
NELL23k	0.002	0.01	1024	1.0	0.5	7	0.01	0.001
WDsinger	0.001	0.01	512	1.0	0.6	8	0.0	0.01
ConceptNet100k	0.0005	0.01	512	1.0	0.4	6	0.01	0.001
CoDEx-S	0.0005	0.01	512	1.0	0.7	6	0.0	0.001
CoDEx-L	0.0004	0.01	256	1.0	0.5	6	0.0	0.0001
YAGO3-10	0.0005	0.01	512	1.0	0.7	4	0.0	0.01
AristoV4	0.001	0.01	512	10.0	0.7	9	0.1	0.001
DBpedia100k	0.0001	0.01	512	10.0	0.7	10	0.0	0.0001
Hetionet	0.0002	0.01	1024	1.0	0.7	10	0.1	0.01
FB15K237(10)	0.001	0.01	256	1.0	0.5	9	0.0	0.0001
FB15K237(20)	0.001	0.01	512	1.0	0.5	8	0.0	0.001
FB15K237(50)	0.002	0.01	2048	1.0	0.5	9	0.0	0.001
FB15K237	0.0005	0.01	2048	1.0	0.55	10	0.0	0.01
WN18RR	0.0005	0.01	256	1.0	0.65	6	0.0	0.1
CoDEx-M	0.0004	0.01	256	1.0	0.55	7	0.01	0.001

C Detailed Experimental Results

Tables 13 and 14 report per-dataset performance comparisons between MUZE and existing KGFMs on the entity prediction task under zero-shot and finetuned settings, respectively. MUZE demonstrates comparable or state-of-the-art performance on most datasets. Similarly, Tables 15 and 16 present the per-dataset results for the relation prediction task under zero-shot and finetuned settings, respectively, showing that MUZE achieves state-of-the-art results across the majority of the datasets.

Table 13: Zero-shot entity prediction results on 54 KG datasets.

	ULTRA		MOTIF		TRIX		FLOCK		MUZE	
	MRR	Hit10	MRR	Hit10	MRR	Hit10	MRR	Hit10	MRR	Hit10
FB-25	0.388	0.640	0.384	0.640	0.393	0.650	**0.404**	**0.664**	0.392	0.662
FB-50	0.338	0.543	0.338	0.546	0.334	0.547	**0.352**	**0.566**	0.344	0.563
FB-75	0.403	0.604	0.399	0.614	0.401	0.611	**0.418**	**0.622**	0.406	0.621
FB-100	0.449	0.642	0.428	0.628	0.436	0.635	0.452	**0.663**	**0.453**	0.660
WK-25	**0.316**	**0.532**	0.311	0.493	0.305	0.496	0.280	0.491	0.292	0.507
WK-50	**0.166**	**0.324**	0.163	0.314	**0.166**	0.313	0.136	0.278	0.165	0.323
WK-75	0.365	0.537	0.366	0.540	0.368	0.513	0.382	0.538	**0.388**	**0.566**
WK-100	0.164	0.286	0.164	0.282	0.188	0.299	0.187	0.304	**0.190**	**0.311**
NL-0	0.342	0.523	0.324	0.497	0.385	0.549	0.381	0.606	**0.399**	**0.619**
NL-25	**0.395**	0.569	0.348	0.498	0.377	0.589	0.345	0.590	0.389	**0.628**
NL-50	0.407	0.570	0.373	0.532	0.404	0.548	0.366	0.565	**0.412**	**0.600**
NL-75	0.368	0.547	0.314	0.512	0.351	0.525	0.311	0.524	**0.373**	**0.584**
NL-100	0.471	0.651	0.438	0.647	0.486	0.676	0.452	0.692	**0.490**	**0.722**
MT1-tax	0.224	0.305	0.325	0.448	0.358	0.452	0.282	0.383	**0.364**	**0.457**
MT1-health	0.298	0.374	0.326	0.398	0.376	0.457	**0.385**	**0.481**	0.374	0.458
MT2-org	0.095	0.159	0.092	0.154	0.091	0.156	**0.100**	0.163	0.098	**0.170**
MT2-sci	0.258	0.354	0.286	0.435	0.323	0.465	0.318	0.458	**0.351**	**0.486**
MT3-art	0.259	0.402	0.269	0.414	0.284	0.441	**0.301**	**0.466**	0.277	0.442
MT3-infra	0.619	0.755	0.658	0.786	0.655	0.797	**0.684**	**0.821**	0.626	0.771
MT4-sci	0.274	0.449	0.283	0.451	0.290	0.460	0.301	0.463	**0.318**	**0.488**
MT4-health	0.624	0.737	0.627	0.762	0.677	0.775	**0.680**	**0.780**	0.625	0.776
Metafam	0.238	0.644	0.344	0.829	0.341	0.815	0.476	0.935	**0.964**	**1.000**
FBNELL	0.485	0.652	0.468	0.664	0.473	0.660	0.502	**0.700**	**0.503**	0.684
FB-v1	0.498	0.656	0.503	0.692	0.515	0.682	0.500	0.697	**0.534**	**0.724**
FB-v2	0.512	0.700	0.511	0.716	0.525	0.730	**0.535**	**0.737**	0.528	**0.737**
FB-v3	0.491	0.654	0.500	**0.692**	0.501	0.669	**0.511**	0.685	0.494	0.679
FB-v4	0.486	0.677	0.487	0.677	0.493	0.687	**0.505**	**0.702**	0.480	0.698
WN-v1	0.648	0.768	0.682	0.778	**0.699**	0.791	0.698	**0.803**	0.680	0.780
WN-v2	0.663	0.765	0.663	0.771	0.678	0.781	**0.696**	**0.790**	0.677	0.775
WN-v3	0.376	0.476	0.420	0.538	0.418	0.541	**0.467**	**0.608**	0.451	0.585
WN-v4	0.611	0.705	0.640	0.718	0.648	0.723	**0.653**	**0.729**	0.630	0.715
NL-v1	0.785	**0.913**	0.674	0.871	**0.806**	0.898	0.658	0.863	0.799	0.883
NL-v2	0.526	0.707	0.564	0.769	0.569	0.768	**0.588**	**0.797**	0.572	0.790
NL-v3	0.515	0.702	0.533	0.724	0.558	0.743	**0.590**	**0.783**	0.573	0.782
NL-v4	0.479	0.712	0.503	0.711	0.538	0.765	**0.555**	**0.786**	0.518	0.761
HM 1k	0.059	0.092	0.063	0.097	**0.072**	**0.128**	0.069	0.119	0.059	0.119
HM 3k	0.037	0.077	0.055	0.084	**0.069**	**0.119**	0.067	0.118	0.059	0.111
HM 5k	0.034	0.071	0.050	0.073	0.062	0.110	**0.064**	**0.116**	0.059	0.115
Indigo-BM	**0.440**	**0.648**	0.426	0.637	0.436	0.645	0.423	0.638	0.409	0.612
ILPC Small	0.302	0.443	0.295	0.444	0.303	0.455	**0.309**	0.459	0.305	**0.462**
ILPC Large	0.290	0.424	0.285	0.415	0.307	0.428	**0.318**	**0.438**	0.308	0.430
NELL995	0.406	0.543	0.491	0.623	0.472	0.629	0.494	**0.655**	**0.525**	0.649
NELL23k	0.239	0.408	0.220	0.384	**0.290**	**0.497**	0.233	0.398	0.266	0.438
WDsinger	0.382	0.498	0.397	0.514	**0.511**	**0.609**	0.410	0.528	0.400	0.526
ConceptNet100k	0.082	0.162	0.019	0.037	0.193	0.345	0.248	0.453	**0.282**	**0.463**
CoDEx-S	0.472	0.667	**0.474**	**0.670**	0.472	**0.670**	0.441	0.644	0.386	0.629
CoDEx-L	0.338	**0.469**	0.339	**0.469**	0.335	**0.469**	**0.342**	0.464	0.309	0.435
YAGO3-10	0.451	0.615	0.441	0.607	0.409	0.627	0.414	**0.674**	**0.504**	0.670
AristoV4	0.182	0.282	OOM	OOM	0.181	0.286	**0.308**	**0.443**	0.222	0.350
DBpedia100k	0.398	0.576	OOM	OOM	0.426	0.603	**0.450**	**0.627**	0.371	0.558
Hetionet	0.257	0.379	0.256	0.383	**0.279**	**0.420**	0.246	0.371	0.182	0.308
FB15K237(10)	0.248	0.398	0.236	0.384	0.246	0.393	0.246	0.402	**0.261**	**0.419**
FB15K237(20)	0.272	0.436	0.259	0.422	0.269	0.430	0.273	0.444	**0.280**	**0.452**
FB15K237(50)	**0.324**	**0.526**	0.312	0.508	0.321	0.521	0.319	0.518	0.322	0.520

Table 14: Finetuned entity prediction results on 57 KG datasets.

	ULTRA		MOTIF		TRIX		FLOCK		MUZE	
	MRR	Hit10	MRR	Hit10	MRR	Hit10	MRR	Hit10	MRR	Hit10
FB-25	0.383	0.635	0.388	0.635	0.393	0.650	**0.405**	**0.666**	0.397	**0.666**
FB-50	0.334	0.538	0.340	0.544	0.334	0.547	**0.357**	**0.570**	0.349	0.569
FB-75	0.400	0.598	0.399	0.607	0.401	0.611	**0.425**	**0.630**	0.414	0.629
FB-100	0.444	0.643	0.439	0.642	0.436	0.633	**0.460**	**0.668**	0.452	0.662
WK-25	**0.321**	**0.535**	0.317	0.505	0.300	0.493	0.298	0.506	0.313	0.533
WK-50	0.140	0.280	0.160	0.304	0.166	0.313	0.127	0.260	**0.167**	**0.319**
WK-75	0.380	0.530	0.371	0.535	0.368	0.513	**0.405**	0.556	0.403	**0.562**
WK-100	0.168	0.286	0.173	0.284	0.188	0.299	0.187	0.306	**0.192**	**0.324**
NL-0	0.329	0.551	0.328	0.556	0.385	0.549	0.418	0.619	**0.423**	**0.655**
NL-25	0.407	0.596	0.390	0.580	0.377	0.589	0.405	**0.626**	**0.415**	**0.626**
NL-50	0.418	0.595	0.414	0.573	0.405	0.555	0.391	0.562	**0.443**	**0.623**
NL-75	0.374	0.570	0.360	0.548	0.351	0.525	0.344	0.544	**0.385**	**0.598**
NL-100	0.458	0.684	0.464	0.682	0.482	0.691	0.486	0.714	**0.507**	**0.721**
MT1-tax	0.330	0.459	0.416	0.522	0.397	0.508	0.413	0.497	**0.448**	**0.527**
MT1-health	0.380	0.467	0.385	0.473	0.376	0.457	**0.394**	**0.493**	0.389	0.484
MT2-org	0.104	0.170	0.106	0.170	0.098	0.162	**0.107**	0.174	0.104	**0.175**
MT2-sci	0.311	0.451	0.326	0.520	0.331	0.526	0.366	0.525	**0.409**	**0.551**
MT3-art	0.306	0.473	0.315	0.469	0.289	0.441	**0.330**	**0.483**	0.291	0.454
MT3-infra	0.657	0.807	0.683	0.827	0.672	0.810	**0.709**	**0.838**	0.675	0.820
MT4-sci	0.303	0.478	0.309	0.483	0.305	0.482	0.324	**0.509**	**0.331**	**0.509**
MT4-health	0.704	0.785	0.703	0.787	0.702	0.785	**0.711**	**0.790**	0.687	0.777
Metafam	0.997	**1.000**	**1.000**	**1.000**	0.997	**1.000**	0.992	**1.000**	**1.000**	**1.000**
FBNELL	0.481	0.661	0.481	0.664	0.478	0.655	**0.531**	**0.714**	0.508	0.708
FB-v1	0.509	0.670	0.530	0.702	0.515	0.682	0.549	**0.721**	**0.552**	**0.721**
FB-v2	0.524	0.710	**0.557**	0.744	0.525	0.730	0.553	**0.754**	0.548	0.750
FB-v3	0.504	0.663	0.519	0.684	0.501	0.669	**0.528**	**0.696**	0.517	0.689
FB-v4	0.496	0.684	0.508	**0.702**	0.493	0.687	**0.510**	**0.702**	0.495	**0.702**
WN-v1	0.685	0.793	0.703	0.806	0.705	0.798	**0.715**	**0.811**	0.706	0.803
WN-v2	0.679	0.779	0.680	0.781	0.682	0.780	**0.702**	0.795	0.701	**0.806**
WN-v3	0.411	0.546	0.466	0.590	0.425	0.543	**0.494**	**0.627**	0.454	0.588
WN-v4	0.614	0.720	0.659	0.733	0.650	0.722	**0.665**	0.741	0.662	**0.750**
NL-v1	0.757	0.878	0.712	0.873	**0.804**	0.899	0.762	**0.928**	**0.804**	0.871
NL-v2	0.575	0.761	0.566	0.765	0.571	0.764	**0.612**	**0.806**	0.593	0.801
NL-v3	0.563	0.755	0.580	0.764	0.571	0.759	**0.606**	**0.803**	0.587	0.791
NL-v4	0.469	0.733	0.507	0.740	0.551	0.772	**0.572**	**0.801**	0.514	0.775
HM 1k	0.042	0.100	0.067	0.107	**0.072**	0.128	0.071	**0.153**	0.057	0.113
HM 3k	0.030	0.090	0.054	0.103	**0.069**	0.119	0.067	**0.153**	0.048	0.093
HM 5k	0.025	0.068	0.049	0.091	**0.074**	0.118	0.061	**0.130**	0.053	0.103
Indigo-BM	0.432	0.639	0.426	0.635	**0.436**	**0.645**	0.418	0.633	0.400	0.600
ILPC Small	0.303	0.453	0.302	0.449	0.303	0.455	0.305	0.454	**0.311**	**0.467**
ILPC Large	0.308	0.431	0.307	0.432	0.310	0.431	**0.320**	**0.441**	0.319	**0.441**
NELL995	0.509	0.660	0.514	0.655	0.506	0.648	0.531	**0.665**	**0.542**	0.663
NELL23k	0.268	0.450	0.256	0.441	**0.306**	**0.536**	0.280	0.465	0.277	0.455
WDsinger	0.417	0.526	0.423	0.532	**0.502**	**0.620**	0.435	0.543	0.419	0.536
ConceptNet100k	0.310	0.529	N/A	N/A	0.340	0.564	**0.352**	**0.580**	0.319	0.517
CoDEx-S	**0.490**	**0.686**	**0.490**	0.680	0.484	0.676	0.463	0.644	0.406	0.648
CoDEx-L	0.343	0.478	**0.355**	**0.489**	0.348	0.481	0.342	0.467	0.327	0.456
YAGO3-10	0.557	0.710	**0.603**	**0.735**	0.541	0.702	0.552	0.700	0.571	0.708
AristoV4	0.343	0.496	OOM	OOM	0.345	0.499	**0.383**	**0.523**	0.262	0.404
DBpedia100k	0.436	0.603	OOM	OOM	0.457	0.619	**0.470**	**0.623**	0.431	0.598
Hetionet	0.399	0.538	**0.446**	**0.575**	0.394	0.534	0.314	0.465	0.261	0.403
FB15K237(10)	0.254	0.411	0.254	0.411	0.253	0.408	**0.260**	**0.420**	**0.260**	0.417
FB15K237(20)	0.274	0.445	0.273	0.444	0.273	0.441	**0.284**	**0.459**	0.282	0.453
FB15K237(50)	0.325	**0.528**	0.323	0.523	0.322	0.522	0.317	0.517	**0.328**	0.524
FB15K237	**0.368**	**0.564**	0.336	0.532	0.366	0.559	0.343	0.532	0.330	0.516
WN18RR	0.480	0.614	0.507	0.609	0.514	0.611	**0.550**	**0.656**	0.524	0.629
CoDEx-M	**0.372**	**0.525**	0.357	0.509	0.365	0.521	0.351	0.496	0.332	0.484

Table 15: Zero-shot relation prediction results on 54 KG datasets.

	ULTRA		TRIX		FLOCK		MUZE	
	MRR	Hit1	MRR	Hit1	MRR	Hit1	MRR	Hit1
FB-25	0.687	0.565	0.805	0.724	0.895	0.839	**0.944**	**0.910**
FB-50	0.696	0.575	0.780	0.699	0.880	0.820	**0.925**	**0.882**
FB-75	0.698	0.555	0.822	0.747	0.903	0.844	**0.948**	**0.915**
FB-100	0.830	0.728	0.921	0.880	0.962	0.938	**0.978**	**0.965**
WK-25	0.857	0.760	0.881	0.823	**0.952**	**0.929**	0.949	0.924
WK-50	0.865	0.793	0.868	0.818	0.921	0.882	**0.930**	**0.886**
WK-75	0.911	0.875	0.916	0.883	0.962	0.944	**0.966**	**0.948**
WK-100	0.887	0.812	0.907	0.869	**0.963**	**0.937**	0.956	0.926
NL-0	0.632	0.502	0.658	0.519	0.714	0.574	**0.793**	**0.705**
NL-25	0.688	0.562	0.742	0.614	0.729	0.632	**0.842**	**0.769**
NL-50	0.680	0.569	0.755	0.636	0.813	0.728	**0.868**	**0.808**
NL-75	0.795	0.692	0.788	0.699	0.833	0.756	**0.881**	**0.834**
NL-100	0.743	0.564	0.884	0.796	0.939	0.889	**0.966**	**0.939**
MT1-tax	0.985	0.976	0.975	0.958	**0.998**	**0.997**	**0.998**	**0.997**
MT1-health	0.721	0.561	0.973	0.949	0.991	0.983	**0.992**	**0.985**
MT2-org	0.974	0.951	0.986	0.973	**0.991**	**0.984**	**0.991**	0.983
MT2-sci	0.976	0.961	0.964	0.941	**0.995**	**0.992**	0.988	0.980
MT3-art	0.881	0.798	0.885	0.825	**0.944**	**0.907**	0.943	0.906
MT3-infra	0.962	0.935	0.940	0.905	0.989	0.980	**0.992**	**0.987**
MT4-sci	0.933	0.891	0.966	0.944	0.974	0.957	**0.976**	**0.960**
MT4-health	0.826	0.719	0.937	0.898	0.990	0.983	**0.993**	**0.989**
Metafam	0.124	0.000	0.291	0.011	0.490	0.223	**0.866**	**0.755**
FBNELL	0.700	0.564	0.726	0.605	0.833	0.737	**0.897**	**0.832**
FB-v1	0.646	0.523	0.705	0.599	0.814	0.723	**0.894**	**0.839**
FB-v2	0.695	0.570	0.713	0.590	0.847	0.761	**0.929**	**0.886**
FB-v3	0.679	0.553	0.742	0.644	0.860	0.780	**0.929**	**0.887**
FB-v4	0.638	0.488	0.766	0.665	0.873	0.799	**0.936**	**0.895**
WN-v1	0.836	0.740	0.792	0.613	0.924	0.858	**0.936**	**0.887**
WN-v2	0.853	0.790	0.764	0.572	0.924	0.863	**0.932**	**0.878**
WN-v3	0.707	0.577	0.741	0.568	**0.937**	**0.888**	0.931	0.881
WN-v4	0.860	0.803	0.764	0.570	**0.937**	**0.886**	0.936	0.883
NL-v1	0.636	0.358	0.657	0.453	0.862	0.731	**0.945**	**0.891**
NL-v2	0.742	0.652	0.780	0.696	0.893	0.855	**0.919**	**0.886**
NL-v3	0.669	0.544	0.725	0.612	0.815	0.731	**0.901**	**0.850**
NL-v4	0.606	0.489	0.794	0.691	0.868	0.807	**0.924**	**0.833**
HM 1k	0.626	0.447	0.663	0.414	0.687	0.500	**0.802**	**0.660**
HM 3k	0.592	0.439	0.664	0.418	0.714	0.549	**0.852**	**0.751**
HM 5k	0.605	0.452	0.672	0.428	0.746	0.593	**0.854**	**0.749**
Indigo-BM	0.681	0.559	0.852	0.765	0.956	0.921	**0.964**	**0.936**
ILPC Small	0.905	0.843	0.919	0.872	0.955	0.921	**0.958**	**0.925**
ILPC Large	0.875	0.799	0.894	0.829	0.948	0.908	**0.950**	**0.914**
NELL995	0.583	0.437	0.578	0.457	0.684	0.555	**0.821**	**0.733**
NELL23k	0.669	0.548	0.756	0.657	0.831	0.762	**0.877**	**0.820**
WDsinger	0.668	0.546	0.720	0.621	0.823	0.738	**0.855**	**0.770**
ConceptNet100k	0.181	0.083	0.650	0.469	0.795	0.658	**0.851**	**0.766**
CoDEx-S	0.900	0.820	0.961	0.935	0.982	0.970	**0.984**	**0.972**
CoDEx-L	0.892	0.824	0.902	0.837	**0.973**	**0.950**	0.960	0.928
YAGO3-10	0.646	0.403	0.783	0.598	**0.971**	**0.943**	0.968	0.940
AristoV4	0.254	0.201	0.389	0.265	**0.597**	**0.496**	0.481	0.383
DBpedia100k	0.650	0.509	0.717	0.582	**0.919**	**0.861**	0.893	0.828
Hetionet	0.634	0.524	0.809	0.707	0.940	0.890	**0.959**	**0.922**
FB15K237(10)	0.688	0.550	0.795	0.711	0.918	0.876	**0.939**	**0.905**
FB15K237(20)	0.695	0.558	0.834	0.758	0.952	0.923	**0.959**	**0.935**
FB15K237(50)	0.717	0.591	0.876	0.812	0.968	0.946	**0.974**	**0.956**

Table 16: Finetuned relation prediction results on 57 KG datasets.

	ULTRA		TRIX		FLOCK		MUZE	
	MRR	Hit1	MRR	Hit1	MRR	Hit1	MRR	Hit1
FB-25	0.684	0.563	0.805	0.724	0.909	0.857	**0.947**	**0.915**
FB-50	0.696	0.575	0.780	0.699	0.881	0.820	**0.932**	**0.890**
FB-75	0.754	0.638	0.822	0.699	0.911	0.854	**0.953**	**0.924**
FB-100	0.851	0.769	0.921	0.880	0.965	0.939	**0.981**	**0.969**
WK-25	0.897	0.834	0.905	0.860	**0.968**	**0.954**	0.948	0.914
WK-50	0.865	0.793	0.881	0.840	0.925	0.876	**0.940**	**0.902**
WK-75	0.911	0.875	0.937	0.910	0.965	0.948	**0.967**	**0.949**
WK-100	0.924	0.879	0.916	0.885	**0.970**	**0.946**	0.962	0.936
NL-0	0.632	0.502	0.655	0.518	0.731	0.602	**0.801**	**0.712**
NL-25	0.737	0.622	0.709	0.606	0.757	0.634	**0.856**	**0.790**
NL-50	0.808	0.704	0.774	0.683	0.814	0.721	**0.879**	**0.823**
NL-75	0.795	0.678	0.790	0.671	0.848	0.774	**0.891**	**0.845**
NL-100	0.803	0.678	0.885	0.793	0.937	0.887	**0.968**	**0.946**
MT1-tax	0.990	0.984	0.995	0.990	**0.999**	**0.998**	0.998	0.996
MT1-health	0.929	0.867	0.973	0.949	**0.994**	0.988	**0.994**	**0.989**
MT2-org	0.981	0.963	0.987	0.978	**0.994**	**0.988**	0.992	0.985
MT2-sci	0.977	0.961	0.990	0.984	**0.995**	**0.992**	0.990	0.982
MT3-art	0.907	0.851	0.887	0.828	0.950	0.916	**0.951**	**0.917**
MT3-infra	0.966	0.947	0.970	0.952	**0.996**	**0.993**	0.994	0.989
MT4-sci	0.954	0.929	0.972	0.952	**0.983**	**0.968**	0.981	0.967
MT4-health	0.951	0.919	0.986	0.979	**0.995**	**0.991**	0.993	0.989
Metafam	0.368	0.112	0.265	0.024	**0.997**	**0.995**	0.995	0.989
FBNELL	0.720	0.576	0.766	0.639	0.879	0.801	**0.904**	**0.846**
FB-v1	0.650	0.513	0.705	0.599	0.855	0.766	**0.907**	**0.854**
FB-v2	0.675	0.547	0.713	0.590	0.887	0.812	**0.937**	**0.894**
FB-v3	0.677	0.556	0.742	0.644	0.879	0.810	**0.939**	**0.899**
FB-v4	0.690	0.560	0.766	0.665	0.884	0.807	**0.939**	**0.896**
WN-v1	0.844	0.754	0.776	0.591	0.926	0.879	**0.959**	**0.930**
WN-v2	0.834	0.766	0.765	0.574	0.927	0.869	**0.951**	**0.911**
WN-v3	0.707	0.577	0.756	0.594	**0.950**	**0.911**	0.936	0.887
WN-v4	0.861	0.795	0.804	0.651	0.943	0.898	**0.962**	**0.932**
NL-v1	0.719	0.504	0.590	0.341	0.883	0.766	**0.938**	**0.876**
NL-v2	0.668	0.549	0.811	0.740	0.911	0.870	**0.925**	**0.890**
NL-v3	0.646	0.484	0.757	0.643	0.868	0.795	**0.906**	**0.855**
NL-v4	0.570	0.412	0.822	0.735	0.906	0.849	**0.928**	**0.886**
HM 1k	0.626	0.447	0.663	0.414	0.756	0.561	**0.810**	**0.679**
HM 3k	0.592	0.439	0.664	0.418	0.790	0.623	**0.825**	**0.712**
HM 5k	0.605	0.452	0.672	0.428	0.744	0.591	**0.829**	**0.718**
Indigo-BM	0.726	0.614	0.835	0.746	0.946	0.903	**0.964**	**0.935**
ILPC Small	0.922	0.876	0.919	0.872	0.953	0.918	**0.960**	**0.928**
ILPC Large	0.875	0.799	0.894	0.829	0.953	0.915	**0.960**	**0.927**
NELL995	0.630	0.513	0.578	0.457	0.713	0.584	**0.842**	**0.758**
NELL23k	0.688	0.571	0.755	0.658	0.869	0.805	**0.889**	**0.827**
WDsinger	0.730	0.603	0.721	0.627	0.885	0.815	**0.912**	**0.857**
ConceptNet100k	0.612	0.488	0.712	0.551	0.885	0.813	**0.897**	**0.829**
CoDEx-S	0.942	0.900	0.964	0.943	0.981	0.967	**0.985**	**0.975**
CoDEx-L	0.907	0.850	0.908	0.845	**0.973**	**0.950**	0.967	0.940
YAGO3-10	0.930	0.891	0.826	0.666	0.970	0.942	**0.981**	**0.966**
AristoV4	0.254	0.201	0.498	0.381	**0.651**	**0.547**	0.602	0.493
DBpedia100k	0.650	0.509	0.780	0.665	0.923	0.869	**0.937**	**0.895**
Hetionet	0.737	0.646	0.922	0.862	0.942	0.897	**0.962**	**0.929**
FB15K237(10)	0.688	0.550	0.795	0.711	0.940	0.905	**0.953**	**0.923**
FB15K237(20)	0.695	0.558	0.846	0.778	0.958	0.931	**0.966**	**0.943**
FB15K237(50)	0.728	0.618	0.903	0.858	0.970	0.948	**0.977**	**0.960**
FB15K237	0.795	0.709	0.924	0.870	0.976	0.957	**0.979**	**0.964**
WN18RR	0.914	0.871	0.783	0.634	**0.982**	**0.968**	0.978	0.962
CoDEx-M	0.919	0.870	0.931	0.886	**0.974**	**0.952**	**0.974**	**0.952**

D ADDITIONAL ANALYSIS OF MUZE

Multi-Seed Variance Table 17 reports the zero-shot entity and relation prediction performance of MUZE across three independent random seeds for parameter initialization. We report the mean and standard deviation of MRR averaged over 54 KGs. The results show that MUZE exhibits low variability across random seeds for both entity and relation prediction.

Runtime by Varying Numbers of Queries Table 18 shows how inference time varies with the number of queries on YAGO3-10. The runtime of query-specific methods increases approximately linearly with the number of queries, whereas MUZE remains nearly constant. For a small number of queries, the query-specific methods do not exhibit a perfect linear trend, as all queries can be processed within a single batch. As the number of queries increases, the increasing trend becomes linear. The linearity occurs because these methods require a separate forward pass over the entire KG for each query. In contrast, MUZE requires only one-time query-independent computation over the KG, followed by lightweight per-query scoring. Note that the reported runtime of MUZE includes 32 Monte Carlo samples, whereas those of TRIX and ULTRA use a single inference run. With a single Monte Carlo sample, MUZE processes one query in 1 second and 5,000 queries in only 4 seconds, matching ULTRA's runtime for a single query while scaling substantially better.

Table 17: Stability of MUZE across three random seeds. Zero-shot MRRs for entity prediction (Ent.) and relation prediction (Rel.) on 54 KGs are reported.

	Seed #1	Seed #2	Seed #3	Mean±Std
Ent.	0.401	0.400	0.398	0.400±0.001
Rel.	0.919	0.921	0.924	0.921±0.002

Table 18: Inference time across varying numbers of queries and zero-shot entity prediction MRR on YAGO3-10.

	1	10	100	1,000	5,000 (Full)	MRR
ULTRA	1 second	1 second	4 seconds	29 seconds	143 seconds	0.451
TRIX	2 seconds	3 seconds	6 seconds	49 seconds	246 seconds	0.409
FLOCK	8 seconds	71 seconds	644 seconds	6,242 seconds	31,333 seconds	0.414
MUZE	22 seconds	22 seconds	22 seconds	24 seconds	28 seconds	0.504