VITHAN: Visual-Textual and Hyper-Relational Numeric-Aware Knowledge Graph Representation Learning with Transformers.

2025 Winter/Spring Undergraduate Research Participation Report

Jeemin Kim, Seonghun Choi

School of Computing, KAIST

1 Introduction

Knowledge Graphs (KGs) have become a cornerstone of modern artificial intelligence, providing a structured graphical format for representing real-world facts and powering applications like semantic search, question answering, and recommender systems. Conventionally, KGs represent knowledge as simple triplets in the form of (head entity, relation, tail entity). However, this rigid structure often fails to capture the complexity and richness of real-world information, which is frequently multifaceted and context-dependent.

To overcome this limitation, recent research has advanced in two distinct yet complementary directions. The first focuses on enriching KGs with multimodal information. A pioneering model in this area, VISTA (Visual-Textual Knowledge Graph Representation Learning), integrates visual and textual features to enrich entities and relations with deeper semantic meaning, but its architecture is confined to the traditional triplet structure. The second direction addresses structural complexity. Models like HyNT (Hyper-relational knowledge graph embedding with Numeric literals using Transformers) have been developed to handle hyper-relational facts, where a primary triplet is described by additional qualifiers, and numeric values. However, HyNT is not capable of utilizing the multimodal context that VISTA leverages.

A significant gap remains at the intersection of these frontiers: to the best of our knowledge, no existing framework can effectively learn from knowledge graphs that are simultaneously multimodal (visual-textual), hyper-relational, and numeric-aware. To bridge this gap, we propose VITHAN (**VI**sual-**T**extual and **H**yper-rel**A**tional **N**umeric-aware knowledge graph representation learning with transformers). VITHAN is a unified model designed to synthesize the strengths of VISTA and HyNT. By leveraging a multi-stage Transformer architecture, it learns abundant, contextualized embeddings from knowledge graphs that integrate visual, textual, structural, and numeric information within a single, cohesive framework.

The primary contributions of our research are threefold:

- We design and implement VITHAN, a novel, unified model capable of representation learning on complex knowledge graphs featuring multimodal, hyper-relational, and numeric data.
- We construct and release VTHNKG, a new series of benchmark datasets created by extending VTKG-I to include hyper-relational and numeric facts, facilitating research in this new domain.
- We conduct extensive experiments demonstrating that VITHAN generally outperforms its predecessors, VISTA and HyNT, on our new complex datasets, and provide a detailed analysis of its performance under various conditions.

The remainder of this report is structured as follows. Section 2 reviews related works in knowledge graph embedding. Section 3 details the VITHAN model architecture. Section 4 describes the construction of our VTHNKG datasets. Sections 5 and 6 present our experimental results and a thorough discussion of the findings. Finally, Section 7 concludes the report and suggests directions for future work.

2 Related Works

2.1 Knowledge Graph Embedding

A Knowledge Graph (KG) is a way of representing diverse, real-world knowledge in a structured, graphical format. A KG is composed of facts, with each fact represented as a triplet: (head entity, relation, tail entity). For example, the knowledge that a person is the student of KAIST is structured as the triplet: (person, studentOf, KAIST). The primary goal of knowledge graph representation learning is to embed these entities and relations into low-dimensional vector spaces. These learned embeddings are then utilized for tasks such as knowledge graph completion, which involves predicting missing or unknown links within the graph.

While early methods focused mainly on the structural information, recent advancements have sought to enrich these representations by incorporating additional information. Our work builds upon two such pioneering efforts that expand KGs in complementary directions: VISTA, which integrates rich multimodal data and HyNT, which handles complex hyper-relational and numeric facts.

2.2 Multimodal Knowledge Graphs and VISTA

The multimodal knowledge graph is one of the KGs that augment information from other modalities. While some approaches incorporate images or text descriptions for entities, they often fall short of capturing the full scope of visual and textual commonsense knowledge.

A key contribution in this area is VISTA. It introduces the concept of a Visual-Textual Knowledge Graph (VTKG), a novel framework where both entities and relations are accompanied by text descriptions for deeper semantic meaning and entire triplets can be represented by images.

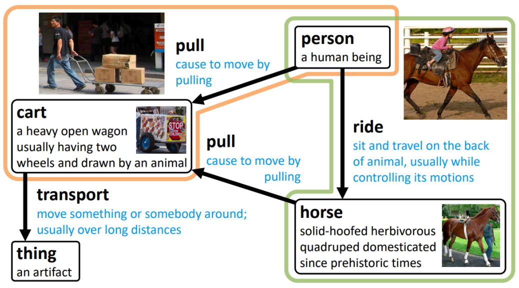

pull
cause to move by pulling
person
a human being
cart
a heavy open wagon usually having two wheels and drawn by an animal
pull
cause to move by pulling
ride
sit and travel on the back of animal, usually while controlling its motions
transport
move something or somebody around; usually over long distances
horse
solid-hoofed herbivorous quadruped domesticated since prehistoric times
thing
an artifact

Figure 1: Subgraph of VTKG - Multimodal Knowledge Graph with Visual/Textual Features[1]

Experiments have shown that VISTA significantly outperforms state-of-the-art methods in real-world VTKGs. However, VISTA is designed for the traditional triplet structure and could be extended to more complex structures like hyper-relational knowledge graphs.

2.3 Hyper-Relational and Numeric Knowledge Graphs and HyNT

Another limitation of traditional KGs is their reliance on the simple triplet format, which struggles to represent complex facts. To address this, hyper-relational knowledge graphs have been introduced, where a primary triplet can be associated with a set of qualifiers; auxiliary relation-entity pairs that provide additional context. For instance, the triplet (Nubjuki, studentOf, KAIST) can be enriched with the qualifier (liveIn, Daejeon).

[1] Jaejun Lee et al., "VISTA: Visual-Textual Knowledge Graph Representation Learning," Findings of the Association for Computational Linguistics: EMNLP 2023, 2023, p. 7314, Figure 1.

While several hyper-relational models exist, they often fail to properly handle numeric literals such as dates or counts, treating them as simple discrete objects or removing them entirely. To solve this, HyNT was proposed as a unified framework to learn representations from hyper-relational KGs that contain numeric values.

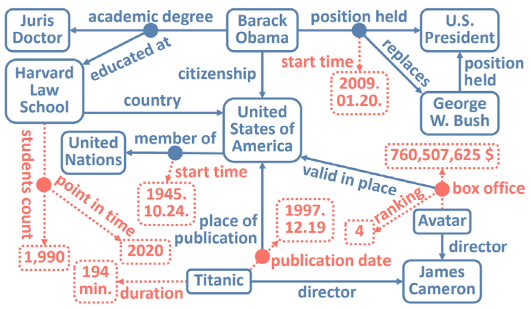


Figure 2: Subgraph of HN-WK[2]

Experiment results demonstrate that HyNT outperforms 12 different baseline models on real-world datasets. Even though its strength in handling structural complexity and numeric data, HyNT can be extended to consider additional multimodal information.

Our research is positioned at the confluence of these two state-of-the-art models. We aim to synthesize the strengths of VISTA and HyNT to create a single, unified model that can learn from knowledge graphs that are simultaneously multimodal, hyper-relational, and numeric-aware.

3 Model Architecture

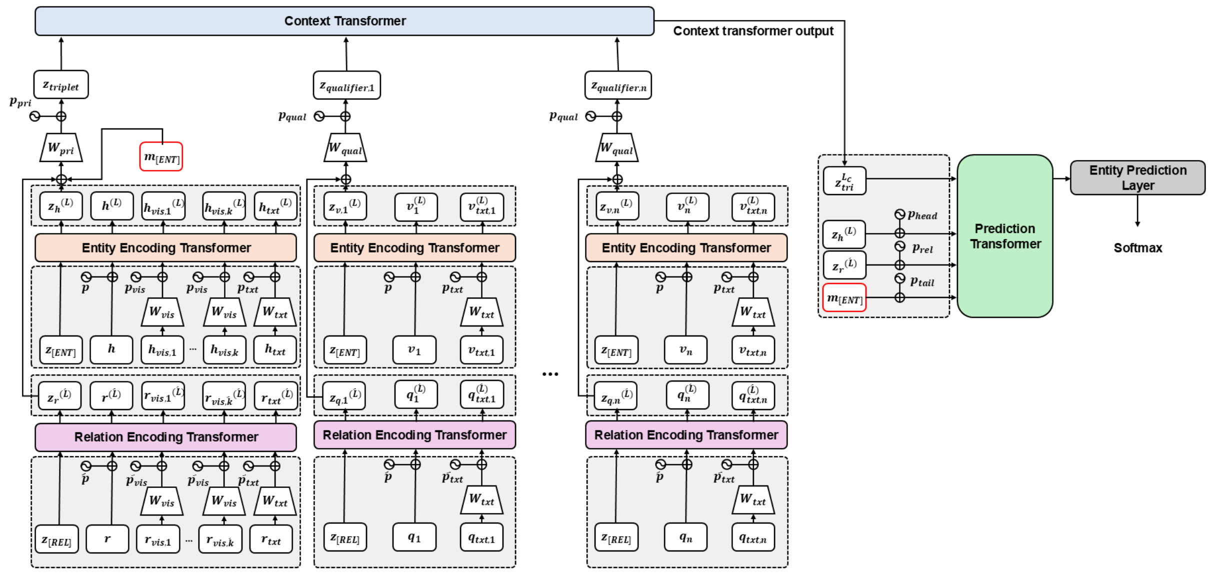
3.1 Overview

VITHAN (**VI**sual-**T**extual and **H**yper-rel**A**tional **N**umeric-aware knowledge graph representation learning with transformers) is a unified model designed to synthesize the distinct capabilities of VISTA and HyNT. Its architecture is engineered to learn from complex KGs that are simultaneously multimodal, hyper-relational, and numeric-aware. The model processes the dataset through three specialized encoding transformers before making predictions with a decoding prediction transformer.

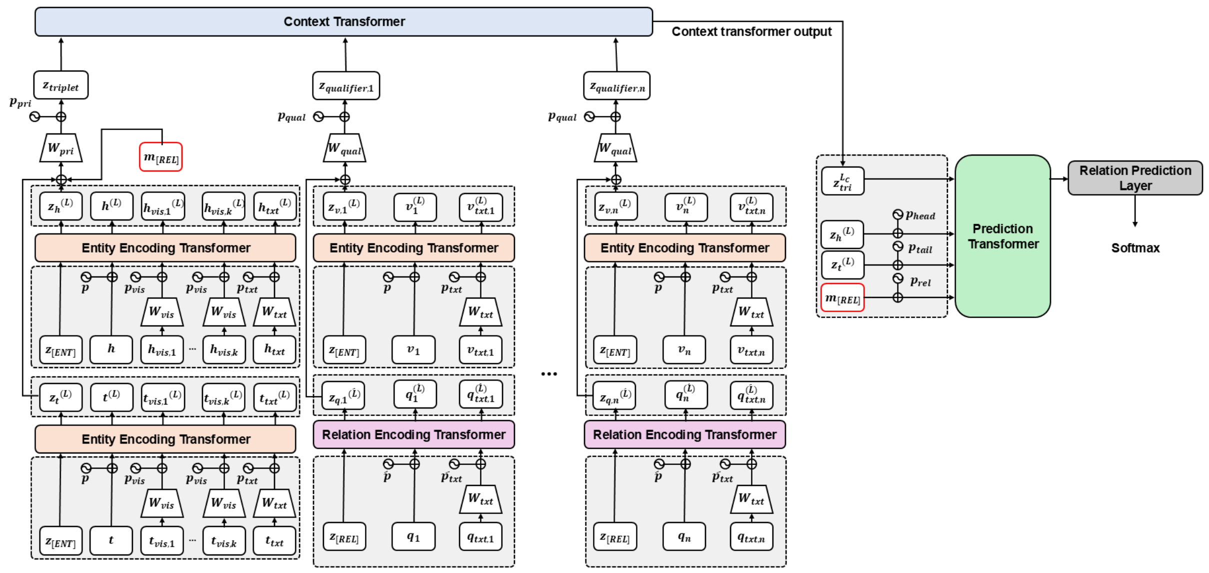
There are three main stages in VITHAN. **Entity Encoding Transformer and Relation Encoding Transformer** take a sequence of inputs for each item, including the base structural embedding, textual features from BERT, and visual features from ViT. This modality-aware encoding process allows the model to generate embeddings to capture semantic and visual understandings for further interpretation. Also, if there's no visual feature for entity or relation in qualifier, the encoding transformer can recognize it and operate only on existing features. For our datasets, every entity and relation in both triplets and qualifiers have visual features. **Context Transformer** takes a full hyper-relational fact that consists of a primary triplet and a set of qualifiers as a single sequence. By processing this sequence, the transformer learns the interactions and dependencies between the main fact and its associated qualifiers. This step is important for understanding how qualifiers modify the meaning of the primary triplet. Finally, the **Triplet Decoding Transformer** performs the knowledge graph completion task. It takes the contextualized representations from the Context Transformer and predicts the missing or masked component of facts. This component can be a discrete entity, a relation, or a continuous

[2] Chanyoung Chung et al., "Representation Learning on Hyper-Relational and Numeric Knowledge Graphs with Transformers," Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2023, p. 310, Figure 1.

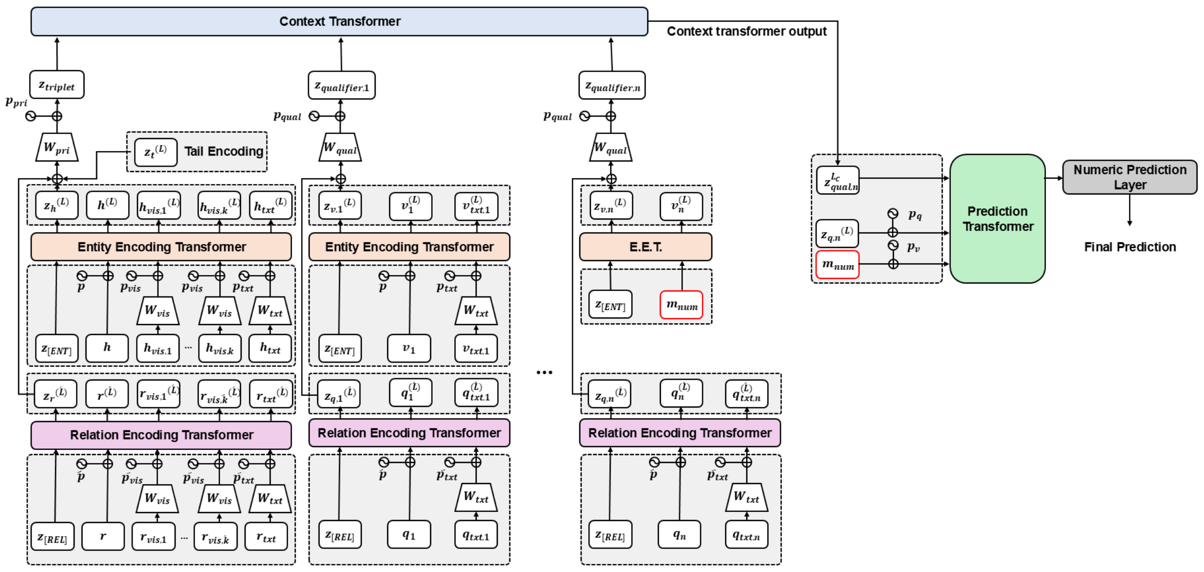
numeric value. This multi-stage architecture enables VITHAN to build a comprehensive understanding of the knowledge graph, leveraging visual, textual, structural, and numeric information to make accurate predictions.



(a) Predicting discrete tail entity in primary triplet



(b) Predicting relation in primary triplet



(c) Predicting numeric value in qualifier

Figure 3. Example of training procedure of VITHAN. We replace the missing component into a mask, then train the model to recover the ground-truth one.

3.2 Encoding Modules

3.2.1 Entity Encoding Transformer

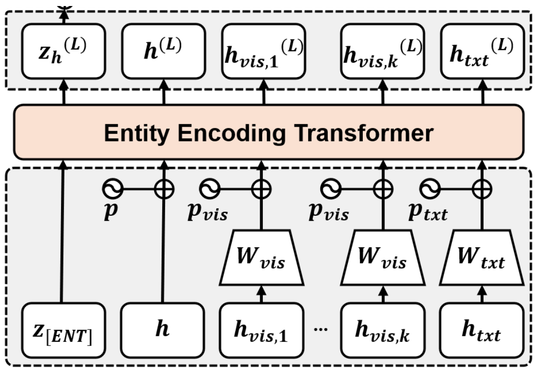


Figure 4. Architecture of Entity Encoding Transformer

The primary function of the Entity Encoding Transformer is to generate a single, powerful representation for each entity by fusing the structural, visual, and textual information. For any given entity in the knowledge graph, there are four key components. **Entity Token** ($z_{[ENT]}$) is a learnable embedding vector that is shared across all entities. Its final output state from the transformer serves as the aggregated representation for the entity. **Structural Embedding** (h) is a unique, learnable vector for each entity that captures its role and position within the graph's structure. **Visual Embedding** (h_{vis}) represents the visual features for entities with associated images. They are first extracted using a pre-trained Vision Transformer (ViT-Base), and these high-dimensional features are then projected into the model's working dimension via a linear transformation. This component can be ignored if there is no image information associated with the entity. **Textual Embedding** (h_{txt}) represents the textual descriptions for each entity, and is collected through a pre-trained BERT model. These features are subsequently projected into the model's dimension, similar to the visual embeddings.

These four components are concatenated to form the input sequence for the entity encoding transformer. To help the model distinguish between the different modalities, unique learnable positional encodings (p, p_{vis}, p_{txt}) are introduced to their respective embeddings. The full input sequence $X^{(0)}$ for an entity h with structural embedding h, visual features h_{vis}, and textual features h_{txt} can be represented as:

$$X^{(0)} = [z_{[ENT]} \,||\, (h + p) \,||\, (W_{vis} h_{vis,1} + p_{vis}) \,||\, ... \,||\, (W_{vis} h_{vis,k} + p_{vis}) \,||\, (W_{txt} h_{txt} + p_{txt})]$$
$$= [z_h^{(0)} \,||\, h^{(0)} \,||\, h_{vis,1}^{(0)} \,||\, ... \,||\, h_{vis,k}^{(0)} \,||\, h_{txt}^{(0)}]$$

Here, || denotes concatenation, and W_{vis} and W_{txt} are learnable projection matrices.

This sequence is then processed by a L-layer Transformer encoder. The self-attention mechanism within the encoder allows the different modality representations to interact, allowing the model to learn a combined, holistic understanding:

$$X^{(L)} = [z_h^{(L)} \,||\, h^{(L)} \,||\, h_{vis,1}^{(L)} \,||\, ... \,||\, h_{vis,k}^{(L)} \,||\, h_{txt}^{(L)}]$$

The final presentation for entity h, denoted $z_h^{(L)}$, is the output vector corresponding to the position of the special token $z_{[ENT]}$. This integrated embedding that contains all information from available modalities is then used in the subsequent modules of VITHAN.

3.2.2 Relation Encoding Transformer

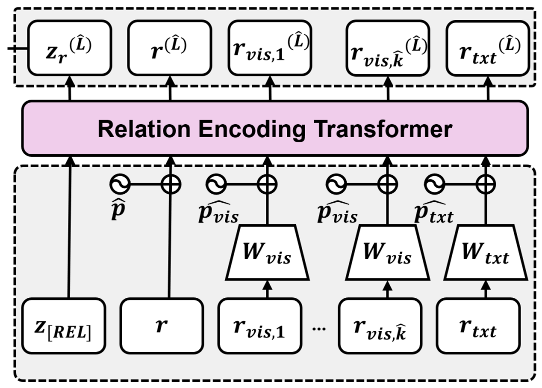


Figure 4. Architecture of Relation Encoding Transformer

Parallel to the entity encoder, the Relation Encoding Transformer is designed to generate representation for each relation in the knowledge graph. For the input sequence of a given relation r, there are four primary components. **Relation Token** ($z_{[REL]}$) is a learnable embedding vector shared across all relations. Similar to entity embeddings, there are **Structural Embedding** (r), **Visual Embedding** (r_{vis}), and **Textual Embedding** (r_{txt}). It has a difference in the visual feature extraction by utilizing the bounding box of the subject and the object entity. Three visual feature vectors from two entity image boxes and one bounding box are then concatenated to form a single feature vector of r_{vis}.

These four components of relations are concatenated to form the input sequence for the relation encoding transformer. There are unique learnable positional encodings ($\hat{p}$, $\hat{p}_{vis}$, $\hat{p}_{txt}$) are introduced to their respective embeddings. The full input sequence $\hat{X}_r^{(0)}$ for an relation r with structural embedding r, visual features r_{vis}, and textual features r_{txt} can be represented as:

$$\hat{X}^{(0)} = [z_{[REL]} \,\|\, (r + \hat{p}) \,\|\, (\widehat{W}_{vis} r_{vis,1} + \hat{p}_{vis}) \,\|\, \dots \,\|\, (\widehat{W}_{vis} h_{vis,\hat{k}} + \hat{p}_{vis}) \,\|\, (\widehat{W}_{txt} h_{txt} + \hat{p}_{txt})]$$
$$= [z_r^{(0)} \,\|\, r^{(0)} \,\|\, r^{(0)}_{vis,1} \,\|\, \dots \,\|\, r^{(0)}_{vis,\hat{k}} \,\|\, r^{(0)}_{txt}]$$

Here, $\widehat{W}_{vis}$ and $\widehat{W}_{txt}$ are learnable projection matrices.

After $\hat{L}$ transformer encoder layers, the following representation matrix is derived:

$$\hat{X}^{(\hat{L})} = [z_r^{(\hat{L})} \,\|\, r^{(\hat{L})} \,\|\, r^{(\hat{L})}_{vis,1} \,\|\, \dots \,\|\, r^{(\hat{L})}_{vis,\hat{k}} \,\|\, r^{(\hat{L})}_{txt}]$$

The final, fused representation for relation *r*, denoted $z_r^{(\hat{L})}$, is the output vector corresponding to the $z_{[REL]}$ token. This process yields a strong relation embedding that is aware of both its semantic definition and its visual-textual context.

3.2.3 Context Transformer

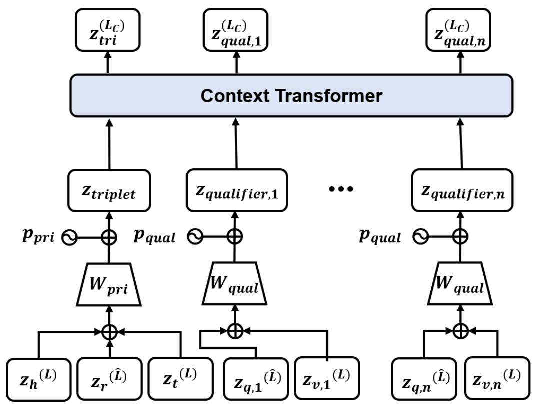


Figure 5. Architecture of Context Transformer

After the Entity and Relation Encoding Transformers produce modality-aware embeddings for individual components, the **Context Transformer** is responsible for analyzing how these components form a complete, complex fact. This module processes an entire hyper-relational fact to learn the interplay between a primary triplet and its associated qualifiers. The input to this transformer is a sequence where each element represents a part of the hyper-relational fact. First, **Primary Triplet Representation** ($z_{triplet}$) is generated by concatenating embeddings for the head, relation, and tail (e.g. $z_h^{(L)}$, $z_r^{(\hat{L})}$, $z_t^{(L)}$). This combined vector is then passed through a linear projection layer to create a compact representation for the primary triplet. The same process is applied to each qualifier pair to produce **Qualifier Representation** ($z_{qualifier,i}$). The fused embeddings for the qualifier's relation and value are concatenated and projected to form a single vector representing that qualifier. If the tail of the primary triplet or the value of a qualifier is a numeric literal, its numeric value is incorporated by scaling the corresponding entity embedding before the projection step. After the series of processing, these individual representations are assembled into a sequence. To differentiate the primary triplet from the qualifiers, distinct positional encoding vectors p_{tri} and p_{qual} are added. The final input sequence with n qualifiers is:

$$Y^{(0)} = [(z'_{triplet} + p_{pri}) \,\|\, (z'_{qualifier,1} + p_{qual}) \,\|\, \dots \,\|\, z'_{qualifier,n} + p_{qual})]$$
$$= [z_{triplet} \,\|\, z_{qualifier,1} \,\|\, \dots \,\|\, z_{qualifier,n}] = [z^{(0)}_{tri} \,\|\, z^{(0)}_{qual,1} \,\|\, \dots \,\|\, z^{(0)}_{qual,n}]$$

This sequence is passed through the Context Transformer, and the self attention mechanism allows the primary triplet and every qualifier to exchange information with each other. This process enables the VITHAN model to learn the relative importance of each component and create context-aware representations. After L_C layers, we get the final representation of a primary triplet and its qualifiers: $Y^{(L_C)} = [z_{tri}^{(L_C)} \,||\, z_{qual,1}^{(L_C)} \,||\, ... \,||\, z_{qual,n}^{(L_C)}]$

3.3 Prediction Transformer

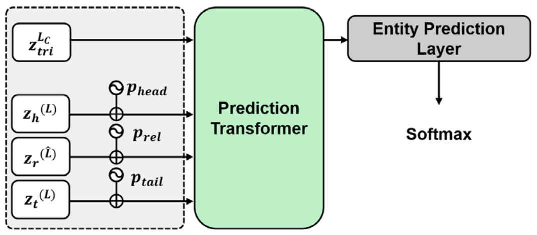


Figure 6. Architecture of Prediction Transformer for Entity Prediction

The final module in VITHAN architecture is the **Prediction Transformer**, which performs the knowledge graph completion task. After the Context Transformer generates representations for hyper-relational knowledge graphs, this decoder utilizes those to predict missing components. During training, a single component within a hyper-relational fact such as the head entity or a relation is masked. The Prediction Transformer is asked to catch this masked component based on the remaining unmasked information. For instance, to predict a primary triplet (*h*, *r*, *t*), the input sequence to the Prediction Transformer, $Z^{(0)}$ would be formed like:

$$\begin{aligned} Z^{(0)} &= [z_{tri}^{(L_C)} \,||\, (z_h^{(L)} + p_{head}) \,||\, (z_r^{(\hat{L})} + p_{rel}) \,||\, (z_t^{(L)} + p_{tail})] \\ &= [\bar{z}_{tri}^{(0)} \,||\, \bar{z}_h^{(0)} \,||\, \bar{z}_r^{(0)} \,||\, \bar{z}_t^{(0)}] \end{aligned}$$

Similarly, to make a prediction on a qualifier (q_i, v_i), the input of the prediction transformer is defined by:

$$\begin{aligned} Z^{(0)} &= [z_{qual_i}^{(L_C)} \,||\, (z_{q_i}^{(\hat{L})} + p_q) \,||\, (z_{v_i}^{(L)} + p_v)] \\ &= [\bar{z}_{qual_i}^{(0)} \,||\, \bar{z}_{q_i}^{(0)} \,||\, \bar{z}_{v_i}^{(0)}] \end{aligned}$$

Using the multi-head attention mechanism with n_p heads, the following representation after L_p layer is driven for the case of a primary triplet:

$$Z^{(L_p)} = [\bar{z}_{tri}^{(L_p)} \,||\, \bar{z}_h^{(L_p)} \,||\, \bar{z}_r^{(L_p)} \,||\, \bar{z}_t^{(L_p)}]$$

for the case of a qualifier:

$$Z^{(L_p)} = [\bar{z}_{qual_i}^{(L_p)} \,||\, \bar{z}_{q_i}^{(L_p)} \,||\, \bar{z}_{v_i}^{(L_p)}]$$

3.4 Training Phase

We trained our model on the VTHNKG datasets using a unified training framework that integrates elements from both VISTA and HyNT. The model is initialized with structural, visual, and textual features for all entities and relations, and uses a Transformer-based architecture with distinct context and prediction modules. During training, the model receives masked hyper-relational inputs, where the target for prediction may be a discrete entity, a relation, or a numeric value.

$$L_{total} = \lambda_1 L_{CE}(Entity) + \lambda_2 L_{CE}(Relation) + \lambda_3 L_{RMSE}(Numeric\ Value)$$

Each training batch is processed through the model to compute three types of outputs: entity scores, relation scores, and numeric predictions. The corresponding loss functions are applied like above: cross-entropy loss for entity and relation predictions, and mean squared error loss for numeric value predictions. In our experiment, we set all weights to 1. Also, to stabilize training, we apply gradient clipping and use the Adam optimizer with a cosine annealing learning rate scheduler with warm restarts.

Validation is conducted periodically during training. For each validation step, we evaluate the model on three tasks: link prediction (predicting missing entities), relation prediction (predicting missing relations), and numeric value prediction (predicting continuous values). These tasks are performed on both primary triplets and full hyper-relational facts including qualifiers. Evaluation metrics include MRR, Hit@1, Hit@3, and Hit@10 for classification tasks, and MSE for numeric ression. Also, to ensure robustness and traceability, we log detailed training results after each validation epoch. These include total loss, component-wise losses, and evaluation metrics. The model is checkpointed at every validation step, allowing for model selection and reproducibility. Overall, this training procedure enables effective learning from multimodal, symbolic, and numeric signals embedded in the VTHNKG datasets.

4 Visual-Textual and Hyper-Relational Numeric-Aware Knowledge Graphs

We construct a new dataset series, VTHNKG(**V**isual-**T**extual and **H**yper-relational **N**umeric-aware **K**nowledge **G**raph), by extending VTKG-I, the visual-textual knowledge graph introduced in the VISTA paper (J. Lee & Whang, 2023). VTKG-I consists of primary triplets (head, relation, tail), with each entity and relation accompanied by visual features (extracted from images and embedded with ViT) and textual features (derived from WordNet-based descriptions and embedded with BERT). We chose VTKG-I as the foundation for our work due to its relatively small size, which makes it easy to preprocess and modify, and its rich multimodal representations, which are highly suitable for visual- and textual-based link prediction tasks.

To move beyond the limitations of basic triplet structures, we enhanced VTKG-I by adding multiple qualifiers and numeric values to each triplet, thereby creating a new hyper-relational knowledge graph, VTHNKG. Specifically, we introduced four types of numeric relations (lengthAverage, weightAverage, lifespan, legAmount) and assigned appropriate numeric values to relevant entities. These numeric attributes provide additional quantitative context that complements the original relational information. In addition to numeric values, we used various GPT-based APIs to generate meaningful and commonsense aligned qualifiers for each triplet. These qualifiers were designed to reflect plausible real-world facts in the form of (relation, entity) or (relation, value) pairs, enriching the contextual information associated with each knowledge unit.

As a result, VTHNKG integrates visual, textual, numerical, and logical components within a unified structure, offering a more expressive and comprehensive dataset for training and evaluating multimodal and hyper-relational knowledge graph models. This expanded representation enables the joint use of VISTA's multimodal learning capabilities and HyNT's numeric-aware hyper-relational modeling.

4.1 VTHNKG-O

Example for VTHNKG-O
`(dog.n.01 wear.v.01 jacket.n.01), (wear.v.02, coat.n.03), (wear.v.02, shirt.n.01), (wear.v.02, hat.n.01)`

VTHNKG-O, short for VTHNKG-Original_dictionary, is a variant of our extended dataset in which all qualifiers are constructed strictly using the original set of entities and relations from VTKG-I. In other words, no new entities or relations are introduced; instead, only those that already exist in the base dataset are reused to form qualifier pairs. This constraint ensures consistency with the original knowledge base while enabling the model to learn from additional contextual information. The qualifier generation process was conducted using the GPT-4o-mini API, which allowed us to generate plausible and contextually relevant qualifiers while adhering to the original dictionary.

4.2 VTHNKG-R (Rejected)

Example for VTHNKG-R
`(person.n.01, keep.v.01, cat.n.01), (care_for.v.02, animal.n.01), (feed.v.02, pet.n.01), (train.v.01, companion.n.01)`

VTHNKG-R, short for VTHNKG-Realtime_dictionary_updating, is a variant of the dataset that allows the introduction of new entities and relations for qualifier construction, beyond those originally present in VTKG-I. In this setting, the GPT-4o-mini API was used to generate qualifiers more flexibly, without being restricted to the existing dictionary. To integrate these newly generated elements into the knowledge graph, we additionally performed WordNet synset mapping and BERT-based embedding generation for each new entity and relation.

However, during dataset analysis, we found that a significant number of these newly introduced entities and relations appeared only once throughout the entire dataset. This extreme sparsity led to poor generalization and made the dataset unsuitable for training robust models. As a result, we decided not to use VTHNKG-R in our final experiments, despite its potential for expanding the knowledge base.

4.3 VTHNKG-NT

Example for VTHNKG-NT

`ex1. (wheel.n.03, weightAverage, 0.009505)`
`ex2. (person.n.01, park.v.01, bicycle.n.01), (ride.v.01, bicycle.n.01), (sit.v.01, bench.n.01),(rest.v.01, grass.n.01)`

VTHNKG-NT, which stands for VTHNKG-Numeric_Tail, is an extension of VTHNKG-O that incorporates numeric triplets into the dataset. These triplets were manually constructed by assigning multiple numeric values to entities using four predefined numeric relations: legAmount, weightAverage, lengthAverage, and lifespan. Each numeric triplet takes the form of a primary triple, where the tail is a numeric value representing a measurable attribute of the entity. This extension enriches the dataset with quantitative information, enabling the training and evaluation of models that are capable of numeric value prediction in addition to discrete entity and relation prediction.

4.4 VTHNKG-NQ

Example for VTHNKG-NQ
`(motorcycle.n.01, have.v.02, box.n.01), (weightAverage, 0.599332), (lengthAverage, 0.057627)`

VTHNKG-NQ, short for VTHNKG-Numeric_Qualifier, differs from VTHNKG-NT in the way numeric information is integrated. Rather than treating numeric triplets as independent triples, this dataset aligns each numeric triplet with an existing primary triplet from VTKG-I based on a shared head entity. When a match is found, the numeric triplet is transformed into a numeric qualifier and attached to the corresponding primary triplet. In this format, numeric values are represented as contextual qualifiers that provide additional quantitative information about the main fact. Notably, this dataset includes only numeric qualifiers and does not contain any discrete (non-numeric) qualifiers, focusing solely on the modeling of numeric context in a hyper-relational setting.

4.5 VTHNKG-CQI

Example for VTHNKG-CQI

```
(dog.n.01, wear.v.01, jacket.n.01), (wear.v.02, coat.n.03), (wear.v.02,
shirt.n.01), (wear.v.02, hat.n.01), (legAmount, 0.157895), (lengthAverage,
0.013559), (lifespan, 0.060302), (weightAverage, 0.019510)
```

VTHNKG-CQ, short for VTHNKG-Complex_Qualifier, is a dataset that incorporates both discrete and numeric qualifiers into the hyper-relational structure. It extends primary triplets from VTKG-I by appending multiple qualifiers, allowing each fact to be enriched with a mixture of symbolic (e.g., categorical) and quantitative context. This configuration enables more comprehensive modeling of real-world knowledge that is both structured and measurable.

However, during the construction process, we observed a significant imbalance between discrete and numeric qualifiers, numeric qualifiers were underrepresented compared to their discrete counterparts. To address this issue, we developed an improved version called VTHNKG-CQI (Complex Qualifier - Improved), which includes a broader range of numeric data to enhance balance and diversity. Model training and evaluation were conducted on VTHNKG-CQI, making it our primary dataset for assessing the performance of multimodal and hyper-relational models in realistic, heterogeneous knowledge settings.

4.6 VTHNKG-OA (and numerics)

Example for VTHNKG-OA

```
(person.n.01, keep.v.01, can.n.01), (contain.v.05, beer.n.01)
```

As our research progressed, we recognized the value of incorporating more commonsense-aligned knowledge into our dataset. While earlier variants relied on the GPT-4o-mini API for qualifier generation, we found that the GPT-4o API produced more generalized and semantically consistent qualifiers that better reflected everyday knowledge and reasoning. Based on this insight, we constructed VTHNKG-OA (Original dictionary Advanced), a dataset in which qualifiers are generated using GPT-4o instead of GPT-4o-mini, resulting in more realistic and meaningful contextual information. Although using GPT-4o incurred a higher cost, the improvement in qualifier quality justified this decision. Like previous datasets, VTHNKG-OA adheres to the original vocabulary of entities and relations from VTKG-I but enriches the dataset with more natural and commonsense-driven qualifiers.

Building upon this foundation, we also developed two extended versions of this dataset. VTHNKG-OA_NT includes numeric triplets in the tail position, allowing for explicit numeric reasoning tasks. VTHNKG-OA_CQI follows the same design as the previously described CQI variant, combining discrete and numeric qualifiers in a more balanced way. These variants offer a richer and more diverse foundation for training and evaluating multimodal models capable of both symbolic and numeric reasoning.

5 Experimental Results

5.1 Environment Setting

All experiments were conducted in a Colab Pro environment using T4 and A100 GPUs, depending on availability. To ensure reproducibility, we fixed all random seeds (NumPy, PyTorch, and Python) to 0 throughout the experiments. The training was carried out for a total of 1,050 epochs, with validation performed every 150 epochs. We set the number of layers for each Transformer module, entity encoder, relation encoder, context encoder, and prediction decoder to 4, keeping the architecture consistent across all model components. For visual feature inputs, we allowed up to 3 images per entity/relation. Although we empirically observed that some datasets performed better with only one visual input, we standardized the setting to 3 visual features to maintain uniformity across experiments. The hidden dimension of each Transformer block was set to 2048, while the embedding dimension for structural, visual, and textual features was set to 256. We used the Adam optimizer with a learning rate of 0.0004, and trained with a batch size of 1024. We applied a dropout rate of 0.15 independently to structural embeddings, visual features, and textual features. This setting was chosen to regularize the multimodal inputs and reduce the risk of overfitting in the Transformer-based architecture.

5.2 Link Prediction

Table 1: Link prediction results for VITHAN

Link Prediction(Triplet/All)				
Dataset	MRR	Hit@1	Hit@3	Hit@10
VTKG-I	0.3807/-	0.2885/-	0.4231/-	0.5538/-
VTHNKG-O	0.6233/0.5638	0.5341/0.4514	0.6629/0.6207s	0.8068/0.7868
VTHNKG-OT	0.3556/-	0.2921/-	0.3643/-	0.4777/-
VTHNKG-CQI	0.4139/0.2891	0.3447/0.1881	0.4015/0.3088	0.5947/0.5063
VTHNKG-NQ	0.4405/-	0.3712/-	0.4432/-	0.6023/-
VTHNKG-NT	0.5905/0.5374	0.517/0.4373	0.6395/0.5955	0.7245/0.7179
VTHNKG-OA	0.5119/0.5063	0.4242/0.3929	0.572/0.5801	0.678/0.7066
VTHNKG-OA_NT	0.4604/0.4819	0.3872/0.385	0.4983/0.5431	0.5892/0.6534
VTHNKG-OA_CQI	0.4968/0.4463	0.428/0.3474	0.5038/0.4772	0.6439/0.6594

5.3 Relation Prediction

Table 2: Relation prediction result for VITHAN

Relation Prediction(Triplet/All)				
Dataset	MRR	Hit@1	Hit@3	Hit@10
VTKG-I	0.2783/-	0.1846/-	0.3154/-	0.4846/-
VTHNKG-O	0.559/5094	0.4773/0.3992	0.5985/0.583	0.697/0.7055

VTHNKG-OT	0.297/-	0.1667/-	0.3827/-	0.4877/-
VTHNKG-CQI	0.297/0.5954	0.1742/0.4989	0.3409/0.6411	0.5682/0.7768
VTHNKG-NQ	0.3241/0.6325	0.2273/0.572	0.3485/0.656	0.5227/0.748
VTHNKG-NT	0.6305/0.5233	0.5528/0.419	0.6832/0.594	0.7516/0.6965
VTHNKG-OA	0.295/0.4654	0.1818/0.3471	0.3485/0.538	0.4773/0.6746
VTHNKG-OA NT	0.3322/0.4899	0.1879/0.3664	0.4242/0.5749	0.5394/0.6883
VTHNKG-OA CQI	0.2416/0.6896	0.1212/0.6066	0.2803/0.7392	0.5/0.8387

5.4 Numeric Prediction

Table 3: Numeric prediction result for VITHAN

Numeric Value Prediction(Only for Numeric Dataset)	
Dataset	Mean SE
VTHNKG-OT	0.0335
VTHNKG-NT	0.0068
VTHNKG-OA NT	0.0052
VTHNKG-NQ	0.017
VTHNKG-CQI	0.0342
VTHNKG-OA CQI	0.0022

6 Discussion

6.1 Discussion about Link Prediction

VTKG-I: Our analysis begins with the baseline dataset, VTKG-I, where the model achieved an MRR of 0.3807. This result serves as a benchmark, representing the VITHAN model's foundational ability to perform link prediction using only the core visual and textual features of entities and relations. All subsequent performance gains are measured against this initial result.

VTHNKG-O & VTHNKG-OA: The introduction of discrete qualifiers in the VTHNKG-O & VTHNKG-OA datasets led to the most significant performance gains. On an "All" basis, VITHAN scored a higher MRR on VTHNKG-O (0.5638) compared to VTHNKG-OA (0.5063). This dramatic improvement over the baseline is clear evidence that the model effectively leverages discrete qualifiers for link prediction. While it might seem counterintuitive that VTHNKG-O outperforms OA, this can be attributed to the datasets' characteristics. VTHNKG-O contains numerous qualifiers that are either duplicates or close paraphrases of the primary triplet, which may have provided strong, direct signals for the link prediction task. In contrast, VTHNKG-OA features more general, commonsense information with little direct overlap. Nonetheless, achieving such a high score on VTHNKG-OA is still a commendable result.

VTHNKG-NQ: The VTHNKG-NQ dataset isolated the impact of numeric information by adding only numeric qualifiers. With an MRR of 0.4405, this approach improved performance over the baseline, suggesting that numeric data does provide a helpful signal. However, this gain was substantially lower than that from the discrete qualifier group, indicating that for link prediction, discrete qualifiers are far more impactful than numeric ones.

VTHNKG-CQI & VTHNKG-OA_CQI: In the VTHNKG-CQI & VTHNKG-OA_CQI datasets, we explored the effect of mixing both discrete and numeric qualifiers within a single triplet. This approach, however, led to a notable drop in performance, with an MRR of 0.2891 for CQI and 0.4463 for OA_CQI. The poor result on CQI, performing worse than even the numeric-only NQ and the baseline VTKG-I, suggests the model likely struggled with information overload or had difficulty weighing the importance of the two different data types. Interestingly, the higher-quality qualifiers in VTHNKG-OA_CQI proved to be more robust, holding up better than CQI's qualifiers in the "noisy" environment created by mixing data types.

VTHNKG-NT & VTHNKG-OA_NT: In contrast, the VTHNKG-NT & VTHNKG-OA_NT datasets took a different approach, treating numeric information as separate, independent triplets rather than mixed-in qualifiers. This strategy was far more successful, yielding MRRs of 0.5374 for NT and 0.4819 for OA_NT. The performance of this group is significantly higher than that of the mixed-qualifier (CQI) group. This demonstrates unequivocally that separating information types is a far more effective strategy for the model than blending them.

In conclusion, these findings suggest an optimal structure for training the VITHAN model. For link prediction, it is more effective to attach discrete facts to the original triplet as qualifiers, while numeric information should be maintained in separate triplets to avoid being diluted. The results also show that the potential for confusion between data types can be so detrimental that having no numeric information at all can be more effective than integrating it improperly.

6.2 Discussion about Relation Prediction

VTKG-I: The performance on VTKG-I serves as the baseline, representing a model’s ability to predict relations in a standard knowledge graph with only triplets and no qualifiers. The MRR of 0.2783 reflects the inherent difficulty of the task when only multimodal features are available.

VTHNKG-O & VTHNKG-OA: The introduction of symbolic, non-numeric qualifiers in VTHNKG-O improves relation prediction for primary triplets (MRR from 0.2783 to 0.05590). This suggests that the contextual information provided by the qualifiers helps the model better understand and disambiguate the primary relation. The performance for predicting all relations (triplet + qualifier), with an MRR of 0.5094, is also strong. A similar trend is seen in VTHNKG-OA, where the higher-quality and generalized qualifiers also improve performance over the baseline as MRR of 0.4654. VTHNKG-O shows better performance than VTHNKG-OA, and this suggests that qualifiers in VTHNKG-O generated by the smaller GPT-4o-mini may be simpler and more direct to learn.

VTHNKG-NQ: In VTHNKG-NQ, which contains only numeric qualifiers without discrete ones, the model shows a strong ability to predict the qualifier relations with MRR of 0.9774. However, the performance on the triplet relations (MRR 0.3241) shows only a modest improvement over the baseline, suggesting that numeric qualifiers provide a little level of semantic context for the primary relation.

VTHNKG-CQI & VTHNKG-OA_CQI: The datasets with both complex discrete and numeric qualifiers show the most improvement in predicting qualifier relations. In VTHNKG-OA CQI, the “All” MRR reaches 0.6896, the highest among all datasets. This is driven by the model’s ability to accurately predict both symbolic and highly-predictable numeric relations. Interestingly, the performance on primary triplets in these datasets (MRR 0.2970 for CQI, 0.2416 for OA CQI) does not have the same benefit of VTHNKG-O, possibly because the increased complexity and diversity of qualifiers make it harder for the model to focus on the primary relation.

VTHNKG-NT & VTHNKG-OA_NT: When numeric values are introduced as the tail of primary triplets, the model’s ability to predict the relation within triplets improves significantly (MRR 0.6305). This is likely because the prediction on numeric relations is simpler than discrete relations. A similar trend is observed in VTHNKG-OA NT.

In conclusion, a notable trade-off emerges when comparing the datasets generated by GPT-4o-mini (O-base) with those generated by the more advanced GPT-4o (OA-base). The OA-series consistently achieves superior performance in predicting the relations within qualifiers, yet this comes at the cost of slightly lower accuracy when predicting the primary triplet's relation. This means that the qualifiers in the OA datasets are more complex, general, and reflective of real-world commonsense.

6.3 Discussion about Numeric Value Prediction

Table n presents the mean squared error (MSE) of numeric value predictions made by the VITHAN model across various datasets that contain numeric literals. The results show that the accuracy of numeric predictions significantly varies depending on how the dataset is constructed and where the numeric information is structurally positioned.

VTHNKG-OT, NT, and OA_NT are datasets in which numeric literals appear as the tail of triplets. In the case of VTHNKG-OT, the MSE is relatively high at 0.0335, which may be due to the dataset containing only triplet-form data without qualifiers. This suggests that the model may have failed to generalize the numeric information semantically. In contrast, VTHNKG-NT and VTHNKG-OA_NT show a notably lower MSE, despite having the same numeric triplet structure. The key difference is that, unlike OT, these datasets include discrete qualifiers alongside the numeric triplets, which likely helped the model learn a more semantically grounded context, resulting in improved performance. The further improvement seen in OA_NT, where the qualifiers are more semantically generalized, supports this interpretation.

On the other hand, VTHNKG-NQ, CQI, and OA_CQI are datasets where numeric literals are presented as qualifiers. Although NQ contains the same types of information as OT, the format is shifted from triplets to qualifiers. In this case, the MSE decreases significantly from 0.0335 to 0.017, suggesting that the model is capable of learning meaningful representations of numeric literals when they are presented as structured qualifiers in a consistent way. However, in the case of VTHNKG-CQI, the MSE rises again to 0.0342, indicating that the model struggled with generalization due to the mixed and possibly ambiguous semantic signals in the dataset. Despite sharing similar numeric content with NQ, the inconsistent context in CQI appears to have hindered the model's learning. Finally, VTHNKG-OA_CQI achieves the best numeric prediction performance (MSE = 0.0022), suggesting that the use of more generalized, commonsense-aware qualifiers, derived from the GPT-4o generated OA base, helping the model semantically generalize more effectively. This result underscores the importance of both structural clarity and semantic quality in learning numeric representations.

6.4 Comparison with VISTA

To specifically evaluate the advantage of integrating hyper-relational and numeric processing, we conducted a direct comparison between our proposed model, VITHAN, and its multimodal predecessor, VISTA. This analysis was performed on VTHNKG-O dataset, which contains the multimodal information that VISTA can utilize, but also includes the qualifier and numeric data that VISTA's architecture is not designed to process.

Table 4: Link Prediction Comparison between VISTA and VITHAN using VTHNKG-O dataset

Model	Type	MRR
VISTA	Head	0.6593
	Tail	0.2278
	All	0.4435

VITHAN	Head	**0.6747**
	Tail	**0.5719**
	All	**0.5638**

As the results indicate, VITHAN outperforms VISTA across all link prediction metrics. Significant improvement is in the prediction of tail entities, where VITHAN achieves an MRR of 0.5719, more than VISTA's score of 0.2278. This substantial improvement can be directly attributed to VITHAN's ability to leverage qualifier information. VISTA, when faced with a query like (Seoul, capitalOf, ?), must make a prediction solely based on the head entity and the relation. In contrast, VITHAN can process additional qualifiers associated with the fact.

6.5 Comparison with HyNT

Table 5: MRR value comparison between VITHAN and HyNT

Type	Link Prediction(Triplet/All)		Relation Prediction(Triplet/All)	
Model	VITHAN	HyNT	VITHAN	HyNT
VTKG-I	0.3807/-	**0.3847/-**	**0.2783/-**	0.2739/-
VTHNKG-O	**0.6233/0.5638**	0.607/0.5247	0.559/**0.5094**	**0.5969**/0.4816
VTHNKG-OT	**0.3556/-**	0.3354/-	**0.297/-**	0.2533/-
VTHNKG-CQI	0.4139/0.2891	**0.4654/0.4117**	0.297/0.5954	**0.3718/0.6725**
VTHNKG-NQ	0.4405/-	**0.4914/-**	**0.3241/0.6325**	0.2104/0.5756
VTHNKG-NT	**0.5905/0.5374**	0.5659/0.5246	**0.6305/0.5233**	0.6234/0.5059
VTHNKG-OA	**0.5119/0.5063**	0.4956/0.4899	**0.295/0.4654**	0.2503/0.4453
VTHNKG-OA NT	**0.4604/0.4819**	0.4346/0.4477	**0.3322/0.4899**	0.2938/0.4322
VTHNKG-OA CQI	**0.4968/0.4463**	0.3586/0.371	**0.2416/0.6896**	0.2002/0.6674

A clear pattern emerges from the experimental results in Table n: VITHAN often outperforms HyNT across the majority of datasets and tasks. This performance gap stems from a core architectural difference. VITHAN leverages both visual and textual embeddings to grasp the rich semantic context of entities and relations. In contrast, HyNT is limited to processing only structural and numerical information, leaving it blind to this crucial multimodal context. This section delves deeper into the results, analyzing performance in light of each dataset's unique characteristics.

6.5.1 The Power of Multimodal Context

VITHAN's fundamental advantage lies in its ability to understand what entities and relations actually are, not just how they are connected. While HyNT can navigate the structure of the knowledge graph, VITHAN interprets the visual and textual cues associated with it. This explains its lead in both Link and Relation Prediction. The model doesn't just predict a link; it predicts a semantically plausible link, informed by a deeper understanding that HyNT inherently lacks.

6.5.2 Performance on Text-based Qualifiers

VITHAN's advantage becomes particularly pronounced on datasets enriched with discrete, text-based qualifiers like VTHNKG-O and VTHNKG-OA. The reason is straightforward: VITHAN uses its BERT-based embeddings to interpret the meaning of a qualifier like (wear.v.02, coat.n.03), while HyNT can only perceive it as another structural link.

Notably, the performance boost was even greater on VTHNKG-OA, which used the more advanced GPT-4o for qualifier generation. This shows that VITHAN doesn't just benefit from any context; it thrives on high-quality, commonsense-aligned textual information, effectively turning richer data into better predictive accuracy.

6.5.3 The Nuances of Handling Numeric Data

Table 6: Mean SE value comparison between VITHAN and HyNT

Dataset	Mean SE of VITHAN	Mean SE of HyNT
VTHNKG-OT	0.0335	**0.0263**
VTHNKG-CQI	0.0342	**0.0101**
VTHNKG-NQ	0.017	**0.0013**
VTHNKG-NT	**0.0068**	0.0274
VTHNKG-OA NT	**0.0052**	0.0734
VTHNKG-OA CQI	**0.0022**	0.0449

The analysis of numerical prediction accuracy, measured by Mean Squared Error (SE), reveals a telling dependency on the structural representation of numeric data. In these tests, a lower SE value signifies a more accurate prediction. The results clearly delineate the distinct strengths of each model. Most notably, HyNT demonstrates a clear superiority on datasets where numeric values are structured as qualifiers, such as VTHNKG-NQ and VTHNKG-CQI. The substantial performance gap on VTHNKG-NQ (0.0013 for HyNT vs. 0.017 for VITHAN) provides strong evidence that HyNT's architecture is highly specialized for interpreting and predicting numerical information embedded within a hyper-relational structure. Conversely, VITHAN holds a decisive advantage on the VTHNKG-NT series of datasets, where numeric values are positioned as the tail in a standard triplet. VITHAN's success here stems from its ability to treat a number not as an isolated value, but as an attribute connected to an entity's core identity. It leverages its comprehensive understanding of an entity's visual and textual features to make a more accurate, context-aware inference of its numerical properties like weight or length. This capability is maximized in the VTHNKG-OA NT and VTHNKG-OA CQI datasets. On these, VITHAN's error rates (0.0052 and 0.0022, respectively) become exceptionally low, demonstrating a powerful synergy. The high-quality semantic context provided by the advanced qualifiers enhances the model's overall understanding of an entity, which in turn directly translates to more precise numerical predictions.

In conclusion, performance on numerical prediction is highly contingent on the method of data representation. While HyNT excels as a structural specialist for handling numeric qualifiers, VITHAN operates as a more robust context-based reasoner. Its ability to leverage an entity's fundamental semantic and visual identity to infer its numerical properties highlights its powerful potential in richer and more realistic data environments.

6.5.4 Complex, Heterogeneous Qualifiers

The most complex datasets provided the most revealing insights. Surprisingly, on VTHNKG-CQI, which mixes discrete and numeric qualifiers, HyNT outperformed VITHAN across the board. This anomaly suggests that the structural complexity of mixing qualifier types may have confused VITHAN, while HyNT's structure-focused approach gave it an edge. Also, we will explain about this issue in section 6.5.5.

However, this dynamic shifted dramatically with VTHNKG-OA_CQI. This dataset has the same complex structure as its predecessor but features the higher-quality, commonsense qualifiers from GPT-4o. Here, VITHAN reclaimed a decisive lead (e.g., LP: 0.4968 vs. 0.3586). This reversal is the most compelling evidence in our study: when provided with rich and meaningful semantic context, VITHAN's multimodal capabilities can overcome structural complexity and unlock a superior level of reasoning.

6.5.5 Overfitting Problem on VTHNKG-NQ & VTHNKG-CQI

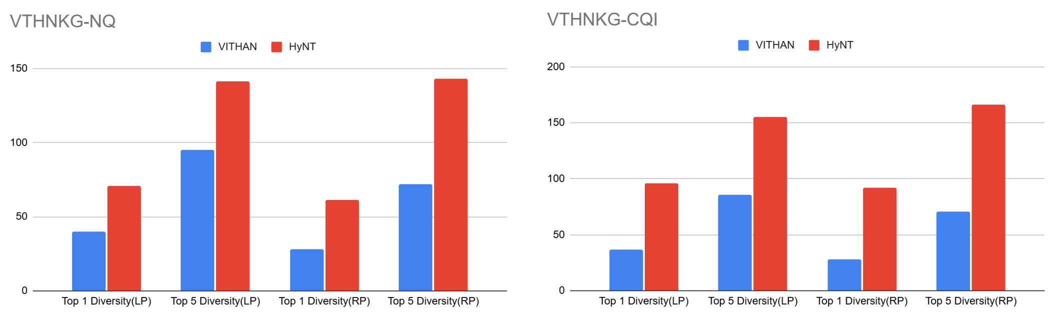


Figure 7: Difference in diversity of answer between VITHAN and HyNT

Figure n illustrates the difference in prediction diversity between VITHAN and HyNT on two datasets: VTHNKG-NQ and VTHNKG-CQI. The bars represent the number of unique predictions among the top-1 and top-5 ranked outputs for both link prediction (LP) and relation prediction (RP). Across all four metrics, HyNT consistently exhibits higher diversity than VITHAN, indicating that it explores a wider range of plausible answers rather than focusing narrowly on a small set of frequently predicted entities or relations.

This pattern suggests that HyNT has better generalization capability, particularly in settings where numeric qualifiers are prevalent. Both VTHNKG-NQ and VTHNKG-CQI emphasize numerical context as a key aspect of knowledge representation. Since HyNT is explicitly designed to model and infer from numeric values in hyper-relational graphs, it is better equipped to make context-sensitive predictions that vary according to the quantitative nuances of each fact.

In contrast, VITHAN relies heavily on visual and textual features, which can introduce unintended bias or redundancy, especially when those modalities dominate the learning process. The lower diversity observed in VITHAN's predictions may indicate that the model has overfitted to specific patterns that frequently appear in the training set, resulting in repetitive and less adaptive responses during inference.

Taken together, these findings suggest that although VITHAN performs well in structurally simpler datasets, it may lack the flexibility to respond accurately in scenarios requiring nuanced numeric reasoning. HyNT's ability to maintain higher prediction diversity reflects not only stronger generalization but also resilience against overfitting, particularly in complex hyper-relational settings like VTHNKG-CQI.

7 Conclusion

In this report, we proposed VITHAN, a novel knowledge graph representation learning method designed to effectively handle visual and textual features alongside hyper-relational and numeric data. To validate our approach, we benchmarked VITHAN against strong baselines, VISTA and HyNT, on nine datasets extended from VTKG-I. Our results demonstrate that VITHAN outperforms these existing models in most scenarios, particularly on complex datasets. This highlights its capability to successfully integrate heterogeneous information into rich, unified embeddings for downstream tasks like link, relation, and numeric value prediction. However, VITHAN's performance was not universally superior. Notably, on simpler datasets that lack hyper-relational facts, VISTA and HyNT achieved more competitive results. This suggests that the optimal model is contingent on data complexity, and that VITHAN's primary advantage lies in its ability to navigate intricate knowledge graphs where various data types are intertwined.

Building on this work, our future research will address several limitations. First, we plan to tackle the overfitting observed on the VTHNKG-NQ and CQI datasets to improve model stability. Second, we will expand our datasets to train a model that generalizes better to real-world applications. Finally, we aim to enhance the model architecture to better mitigate the performance drop when numeric values are introduced, thereby strengthening its numerical reasoning.

In conclusion, VITHAN is a significant step forward in learning from the complex, multi-modal knowledge graphs emerging today. It serves as a solid foundation for future research and more sophisticated, practical applications.

8 Reference

[1] C. Chung, J. Lee, and J. J. Whang, "Representation Learning on Hyper-Relational and Numeric Knowledge Graphs with Transformers," in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023, pp. 310–322. doi: 10.1145/3580305.3599490.

[2] J. Lee, C. Chung, H. Lee, S. Jo, and J. J. Whang, "VISTA: Visual-Textual Knowledge Graph Representation Learning," in *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023, pp. 7314–7328. doi: 10.18653/v1/2023.findings-emnlp.488.

9 Appendix

The following tables show details of the VITHAN test results for every dataset.

Table 7. Detailed Metrics of every dataset result

Dataset	Task	Type	MRR	Hit@1	Hit@3	Hit@10
VTKG-I	Link Prediction	Head	0.5756	0.4692	0.6462	0.8
		Tail	0.1859	0.1077	0.2	0.3077
		Triplet	0.3807	0.2885	0.4231	0.5538
		All	0.3807	0.2885	0.4231	0.5538
	Relation Prediction	Qualifier	-	-	-	-
		Triplet	0.2783	0.1846	0.3154	0.4846
		All	0.2783	0.1846	0.3154	0.4846
VTHNKG-O	Link Prediction	Head	0.6747	0.6061	0.7045	0.8182
		Tail	0.5719	0.4621	0.6212	0.7955
		Triplet	0.6233	0.5341	0.6629	0.8068
		All	0.5638	0.4514	0.6207	0.7868
	Relation Prediction	Qualifier	0.4918	0.3717	0.5775	0.7086
		Triplet	0.559	0.4773	0.5985	0.697

		All	0.5094	0.3992	0.583	0.7055
VTHNKG-OT	Link Prediction	Head	0.4976	0.4383	0.5123	0.6049
		Tail	0.1773	0.1085	0.1783	0.3178
		Triplet	0.3556	0.2921	0.3643	0.4777
		All	0.3556	0.2921	0.3643	0.4777
	Relation Prediction	Qualifier	-	-	-	-
		Triplet	0.297	0.1667	0.3827	0.4877
		All	0.297	0.1667	0.3827	0.4877
	Numeric Prediction	Mean SE	0.0335			
VTHNKG-CQ	Link Prediction	Head	0.6596	0.6136	0.6742	0.7576
		Tail	0.1329	0.0682	0.1288	0.2576
		Triplet	0.3963	0.3409	0.4015	0.5076
		All	0.2605	0.1944	0.2618	0.384
	Relation Prediction	Qualifier	0.4226	0.3106	0.4689	0.6513
		Triplet	0.2769	0.1742	0.3106	0.5455
		All	0.3921	0.2821	0.4358	0.6292
	Numeric Prediction	Mean SE	0.0518			
VTHNKG-CQI	Link Prediction	Head	0.6666	0.6136	0.6667	0.7879
		Tail	0.1612	0.0758	0.1364	0.4015
		Triplet	0.4139	0.3447	0.4015	0.5947
		All	0.2891	0.1881	0.3088	0.5063
	Relation Prediction	Qualifier	0.6435	0.5513	0.6895	0.8105
		Triplet	0.297	0.1742	0.3409	0.5682
		All	0.5954	0.4989	0.6411	0.7768
	Numeric Prediction	Mean SE	0.0342			
VTHNKG-NQ	Link Prediction	Head	0.7529	0.6894	0.7652	0.9242
		Tail	0.1281	0.053	0.1212	0.2803
		Triplet	0.4405	0.3712	0.4432	0.6023
		All	0.4405	0.3712	0.4432	0.6023

	Relation Prediction	Qualifier	0.9774	0.9576	1	1
		Triplet	0.3241	0.2273	0.3485	0.5227
		All	0.6325	0.572	0.656	0.748
	Numeric Prediction	Mean SE	0.017			
VTHNKG-NT	Link Prediction	Head	0.5994	0.528	0.6273	0.7205
		Tail	0.5798	0.5038	0.6541	0.7293
		Triplet	0.5905	0.517	0.6395	0.7245
		All	0.5374	0.4373	0.5955	0.7179
	Relation Prediction	Qualifier	0.4774	0.3617	0.5559	0.6729
		Triplet	0.6305	0.5528	0.6832	0.7516
		All	0.5233	0.419	0.594	0.6965
	Numeric Prediction	Mean SE	0.0068			
VTHNKG-OA	Link Prediction	Head	0.6892	0.6136	0.7424	0.8409
		Tail	0.3347	0.2348	0.4015	0.5152
		Triplet	0.5119	0.4242	0.572	0.678
		All	0.5063	0.3929	0.5801	0.7066
	Relation Prediction	Qualifier	0.5338	0.4134	0.614	0.7538
		Triplet	0.295	0.1818	0.3485	0.4773
		All	0.4654	0.3471	0.538	0.6746
VTHNKG-OA_NT	Link Prediction	Head	0.5703	0.5212	0.5939	0.6364
		Tail	0.323	0.2197	0.3788	0.5303
		Triplet	0.4604	0.3872	0.4983	0.5892
		All	0.4819	0.385	0.5431	0.6534
	Relation Prediction	Qualifier	0.569	0.4559	0.6505	0.7629
		Triplet	0.3322	0.1879	0.4242	0.5394
		All	0.4899	0.3664	0.5749	0.6883
	Numeric Prediction	Mean SE	0.0052			
VTHNKG-OA_CQI	Link Prediction	Head	0.8147	0.7803	0.803	0.9091
		Tail	0.1789	0.0758	0.2045	0.3788
		Triplet	0.4968	0.428	0.5038	0.6439

		All	0.4463	0.3474	0.4772	0.6594
	Relation Prediction	Qualifier	0.7661	0.6895	0.8176	0.8965
		Triplet	0.2416	0.1212	0.2803	0.5
		All	0.6896	0.6066	0.7392	0.8387
	Numeric Prediction	Mean SE	0.0022			

Table 8. Answer Diversity, Distribution, and Correct Rate of every dataset result

Dataset	Task	Analysis	Value
VTKG-I	Link Prediction	Number of Test	260
		Top1 Answer Diversity	80
		Top1 Answer Distribution	[('person.n.01', 68), ('wheel.n.01', 11), ('street.n.01', 6), ('building.n.01', 6), ('bus.n.01', 6)]
		Top5 Answer Diversity	152
		Top5 Answer Distribution	[('person.n.01', 89), ('cat.n.01', 37), ('dog.n.01', 34), ('elephant.n.01', 32), ('boat.n.01', 30)]
		Top5 Correct Rate	[('plate.n.04', 1.0, 1), ('person.n.01', 0.6629, 89), ('table.n.02', 0.0952, 21), ('baggage.n.01', 0.0909, 11), ('sky.n.01', 0.0909, 11)]
	Relation Prediction	Number of Test	130
		Top1 Answer Diversity	61
		Top1 Answer Distribution	[('have.v.01', 17), ('stand.v.01', 7), ('wear.v.01', 6), ('transport.v.02', 5), ('touch.v.01', 5)]
		Top5 Answer Diversity	149
		Top5 Answer Distribution	[('have.v.01', 37), ('transport.v.02', 24), ('hold.v.02', 21), ('keep.v.01', 18), ('touch.v.01', 18)]
		Top5 Correct Rate	[('have.v.01', 0.3243, 37), ('wear.v.01', 0.2857, 14), ('pull.v.01', 0.0909, 11), ('sit.v.01', 0.0769, 13), ('have.v.02', 0.0769, 13)]
VTHNKG-O	Link Prediction	Number of Test	638
		Top1 Answer Diversity	111
		Top1 Answer Distribution	[('person.n.01', 105), ('person.n.02', 39), ('coat.n.03', 23), ('truck.n.01', 20), ('car.n.01', 15)]
		Top5 Answer Diversity	176

		Top5 Answer Distribution	[('person.n.01', 167), ('person.n.02', 107), ('truck.n.01', 81), ('bicycle.n.01', 74), ('dog.n.01', 70)]
		Top5 Correct Rate	[('person.n.01', 0.5329, 167), ('pillow.n.01', 0.5, 10), ('doughnut.n.02', 0.5, 8), ('cake.n.01', 0.4444, 9), ('hotdog.n.02', 0.375, 16)]
	Relation Prediction	Number of Test	506
		Top1 Answer Diversity	103
		Top1 Answer Distribution	[('have.v.01', 49), ('have.v.02', 46), ('hold.v.10', 39), ('ride.v.02', 18), ('inspect.v.01', 16)]
		Top5 Answer Diversity	192
		Top5 Answer Distribution	[('have.v.01', 107), ('have.v.02', 92), ('hold.v.10', 87), ('own.v.01', 64), ('hold.v.11', 62)]
		Top5 Correct Rate	[('tall.a.01', 0.4286, 14), ('have.v.02', 0.4239, 92), ('have.v.01', 0.3551, 107), ('look.v.02', 0.3462, 26), ('drive_on', 0.2941, 17)]
VTHNKG-OT	Link Prediction	Number of Test	291
		Top1 Answer Diversity	79
		Top1 Answer Distribution	[('person.n.01', 72), ('baggage.n.01', 8), ('boat.n.01', 8), ('horse.n.01', 8), ('pillow.n.01', 7)]
		Top5 Answer Diversity	153
		Top5 Answer Distribution	[('person.n.01', 101), ('motorcycle.n.01', 38), ('horse.n.01', 35), ('dog.n.01', 33), ('desk.n.01', 26)]
		Top5 Correct Rate	[('person.n.01', 0.6931, 101), ('plate.n.02', 0.2, 5), ('hat.n.01', 0.1667, 6), ('sky.n.01', 0.1, 10), ('necktie.n.01', 0.0909, 22)]
	Relation Prediction	Number of Test	162
		Top1 Answer Diversity	64
		Top1 Answer Distribution	[('have.v.01', 15), ('lengthAverage', 13), ('weightAverage', 12), ('keep.v.01', 11), ('touch.v.01', 7)]
		Top5 Answer Diversity	157
		Top5 Answer Distribution	[('have.v.01', 36), ('weightAverage', 35), ('lengthAverage', 29), ('hold.v.02', 24), ('touch.v.01', 24)]

		Top5 Correct Rate	[('lengthAverage', 0.4138, 29), ('have.v.01', 0.3611, 36), ('weightAverage', 0.3143, 35), ('fly.v.01', 0.25, 4), ('wear.v.01', 0.2143, 14)]
VTHNKG-CQ	Link Prediction	Number of Test	638
		Top1 Answer Diversity	6
		Top1 Answer Distribution	[('person.n.02', 340), ('person.n.01', 281), ('road.n.01', 12), ('pizza.n.01', 2), ('grass.n.01', 2)]
		Top5 Answer Diversity	37
		Top5 Answer Distribution	[('person.n.01', 617), ('motorcycle.n.01', 497), ('person.n.02', 374), ('laptop.n.01', 366), ('ball.n.03', 366)]
		Top5 Correct Rate	[('person.n.01', 0.141, 617), ('person.n.02', 0.0909, 374), ('grass.n.01', 0.0385, 26), ('table.n.02', 0.0328, 61), ('hand.n.01', 0.0256, 39)]
	Relation Prediction	Number of Test	631
		Top1 Answer Diversity	20
		Top1 Answer Distribution	[('have.v.02', 148), ('legAmount', 97), ('hold.v.10', 67), ('have.v.01', 58), ('drive.v.01', 43)]
		Top5 Answer Diversity	65
		Top5 Answer Distribution	[('have.v.01', 302), ('have.v.02', 275), ('hold.v.10', 196), ('hold.v.11', 128), ('weightAverage', 126)]
		Top5 Correct Rate	[('legAmount', 0.8198, 111), ('wear.v.02', 0.3226, 31), ('wear.v.01', 0.1892, 37), ('weightAverage', 0.1349, 126), ('lengthAverage', 0.119, 126)]
VTHNKG-CQI	Link Prediction	Number of Test	638
		Top1 Answer Diversity	37
		Top1 Answer Distribution	[('person.n.01', 140), ('laptop.n.01', 95), ('person.n.02', 85), ('truck.n.01', 44), ('motorcycle.n.01', 36)]
		Top5 Answer Diversity	86
		Top5 Answer Distribution	[('person.n.01', 320), ('person.n.02', 207), ('dog.n.01', 159), ('horse.n.01', 150), ('motorcycle.n.01', 134)]
		Top5 Correct Rate	[('person.n.01', 0.275, 320), ('road.n.01', 0.1538, 13), ('coat.n.03', 0.0833, 48), ('person.n.02', 0.0628, 207), ('shirt.n.01', 0.0625, 48)]

	Relation Prediction	Number of Test	950
		Top1 Answer Diversity	28
		Top1 Answer Distribution	[('weightAverage', 134), ('lengthAverage', 125), ('hold.v.10', 109), ('legAmount', 104), ('lifespan', 82)]
		Top5 Answer Diversity	71
		Top5 Answer Distribution	[('weightAverage', 445), ('lengthAverage', 390), ('lifespan', 365), ('legAmount', 335), ('hold.v.10', 223)]
		Top5 Correct Rate	[('lengthAverage', 0.3205, 390), ('legAmount', 0.3045, 335), ('wear.v.02', 0.2778, 36), ('weightAverage', 0.2764, 445), ('lifespan', 0.2575, 365)]
VTHNKG-NQ	Link Prediction	Number of Test	264
		Top1 Answer Diversity	40
		Top1 Answer Distribution	[('person.n.01', 110), ('horse.n.01', 52), ('car.n.01', 17), ('cat.n.01', 8), ('jacket.n.01', 5)]
		Top5 Answer Diversity	95
		Top5 Answer Distribution	[('person.n.01', 169), ('car.n.01', 71), ('plate.n.02', 69), ('horse.n.01', 68), ('boat.n.01', 61)]
		Top5 Correct Rate	[('person.n.01', 0.4911, 169), ('truck.n.01', 0.4, 5), ('cart.n.01', 0.2857, 7), ('giraffe.n.01', 0.2857, 7), ('tree.n.01', 0.2, 10)]
	Relation Prediction	Number of Test	250
		Top1 Answer Diversity	28
		Top1 Answer Distribution	[('lifespan', 83), ('have.v.01', 31), ('weightAverage', 13), ('hold.v.02', 13), ('stand.v.01', 13)]
		Top5 Answer Diversity	72
		Top5 Answer Distribution	[('lifespan', 102), ('against', 91), ('drive.v.01', 88), ('cover.v.02', 81), ('eat.v.02', 81)]
		Top5 Correct Rate	[('lifespan', 0.8137, 102), ('legAmount', 0.6667, 15), ('lengthAverage', 0.4815, 27), ('weightAverage', 0.3243, 37), ('have.v.01', 0.2885, 52)]
VTHNKG-NT	Link Prediction	Number of Test	670
		Top1 Answer Diversity	114

		Top1 Answer Distribution	[('person.n.01', 117), ('person.n.02', 56), ('truck.n.01', 20), ('bicycle.n.01', 14), ('car.n.01', 13)]
		Top5 Answer Diversity	177
		Top5 Answer Distribution	[('person.n.01', 178), ('person.n.02', 99), ('bicycle.n.01', 75), ('truck.n.01', 65), ('car.n.01', 64)]
		Top5 Correct Rate	[('person.n.01', 0.5, 178), ('can.n.01', 0.4286, 7), ('hotdog.n.02', 0.4, 15), ('doughnut.n.02', 0.3636, 11), ('roof.n.02', 0.3333, 3)]
	Relation Prediction	Number of Test	537
		Top1 Answer Diversity	102
		Top1 Answer Distribution	[('have.v.01', 45), ('have.v.02', 37), ('hold.v.10', 23), ('hold.v.11', 23), ('ride.v.02', 18)]
		Top5 Answer Diversity	200
		Top5 Answer Distribution	[('have.v.01', 96), ('have.v.02', 84), ('hold.v.10', 74), ('hold.v.11', 71), ('carry.v.02', 65)]
		Top5 Correct Rate	[('lengthAverage', 0.4375, 16), ('have.v.01', 0.375, 96), ('have.v.02', 0.369, 84), ('wear.v.02', 0.3611, 36), ('weightAverage', 0.3556, 45)]
VTHNKG-OA	Link Prediction	Number of Test	593
		Top1 Answer Diversity	111
		Top1 Answer Distribution	[('person.n.01', 107), ('bag.n.06', 27), ('bench.n.01', 24), ('baggage.n.01', 20), ('backpack.n.01', 15)]
		Top5 Answer Diversity	163
		Top5 Answer Distribution	[('person.n.01', 140), ('baggage.n.01', 71), ('bag.n.06', 69), ('bicycle.n.01', 69), ('hand.n.01', 67)]
		Top5 Correct Rate	[('person.n.01', 0.6071, 140), ('backpack.n.01', 0.3167, 60), ('bench.n.01', 0.3, 60), ('surfboard.n.01', 0.2941, 17), ('sink.n.01', 0.2857, 7)]
	Relation Prediction	Number of Test	461
		Top1 Answer Diversity	105
		Top1 Answer Distribution	[('contain.v.05', 49), ('carry.v.02', 29), ('rest_on', 28), ('have.v.01', 26), ('sit_on', 26)]
		Top5 Answer Diversity	186

		Top5 Answer Distribution	[('contain.v.05', 95), ('hold.v.02', 86), ('use.v.01', 77), ('rest_on', 68), ('carry.v.02', 60)]
		Top5 Correct Rate	[('have.v.01', 0.4146, 41), ('carry.v.02', 0.4, 60), ('wear.v.01', 0.4, 40), ('drink.v.01', 0.4, 5), ('sit_on', 0.3036, 56)]
VTHNKG-OA _NT	Link Prediction	Number of Test	626
		Top1 Answer Diversity	117
		Top1 Answer Distribution	[('person.n.01', 109), ('bag.n.06', 26), ('backpack.n.01', 22), ('bench.n.01', 22), ('bottle.n.01', 17)]
		Top5 Answer Diversity	171
		Top5 Answer Distribution	[('person.n.01', 143), ('hand.n.01', 85), ('backpack.n.01', 71), ('baggage.n.01', 69), ('bag.n.06', 66)]
		Top5 Correct Rate	[('person.n.01', 0.6154, 143), ('beer.n.01', 0.3333, 6), ('doughnut.n.02', 0.3333, 6), ('bench.n.01', 0.2787, 61), ('bed.n.01', 0.2778, 18)]
	Relation Prediction	Number of Test	494
		Top1 Answer Diversity	102
		Top1 Answer Distribution	[('contain.v.05', 52), ('carry.v.02', 39), ('rest_on', 31), ('sit_on', 26), ('have.v.01', 25)]
		Top5 Answer Diversity	188
		Top5 Answer Distribution	[('contain.v.05', 99), ('rest_on', 75), ('carry.v.02', 70), ('hold.v.02', 70), ('use.v.01', 63)]
		Top5 Correct Rate	[('carry.v.02', 0.3714, 70), ('have.v.01', 0.3409, 44), ('wear.v.01', 0.3333, 45), ('park.v.01', 0.3333, 3), ('sit_on', 0.2727, 55)]
VTHNKG-OA _CQI	Link Prediction	Number of Test	593
		Top1 Answer Diversity	67
		Top1 Answer Distribution	[('person.n.01', 119), ('bag.n.06', 33), ('road.n.01', 21), ('baggage.n.01', 20), ('horse.n.01', 20)]
		Top5 Answer Diversity	120
		Top5 Answer Distribution	[('person.n.01', 207), ('car.n.01', 112), ('hand.n.01', 109), ('bag.n.06', 94), ('motorcycle.n.01', 76)]

		Top5 Correct Rate	[('plant.n.02', 0.5, 4), ('person.n.01', 0.43, 207), ('street.n.02', 0.2857, 7), ('bench.n.01', 0.25, 60), ('backpack.n.01', 0.2353, 68)]
	Relation Prediction	Number of Test	905
		Top1 Answer Diversity	39
		Top1 Answer Distribution	[('lengthAverage', 130), ('weightAverage', 121), ('legAmount', 102), ('lifespan', 91), ('contain.v.05', 63)]
		Top5 Answer Diversity	97
		Top5 Answer Distribution	[('legAmount', 245), ('ride.v.02', 207), ('lengthAverage', 201), ('weightAverage', 200), ('lifespan', 182)]
		Top5 Correct Rate	[('lengthAverage', 0.6219, 201), ('weightAverage', 0.615, 200), ('lifespan', 0.5165, 182), ('legAmount', 0.4163, 245), ('wear.v.01', 0.3333, 51)]

Details of our experimental results such as HyNT and VISTA are listed in the following link:
VITHAN Experimental Result (Details)

You can reproduce the test results in here: https://github.com/wall9-CS/VITHAN