



Knowledge-aware Zero-shot Learning (K-ZSL): Concepts, Methods and Resources

Yuxia Geng¹, Zhuo Chen², Jiaoyan Chen³, Wen Zhang² and Jeff Z. Pan⁴

1. Hangzhou Dianzi University, China

2. Zhejiang University, China

3. The University of Manchester & University of Oxford, UK

4. The University of Edinburgh, UK

<https://china-uk-zsl.github.io/kg-zsl-tutorial-ijcai-2023/>

Tutorial of The 32nd International Joint Conference on Artificial Intelligence (19th August, 2023, Macao, S.A.R)

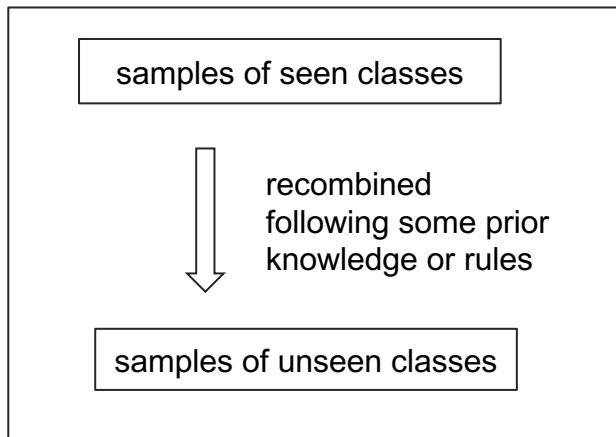
Part II – Knowledge-aware ZSL Methods

T3

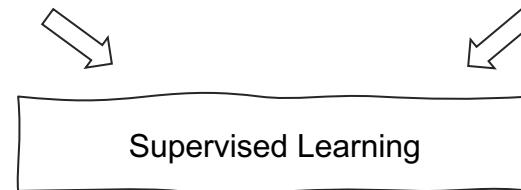
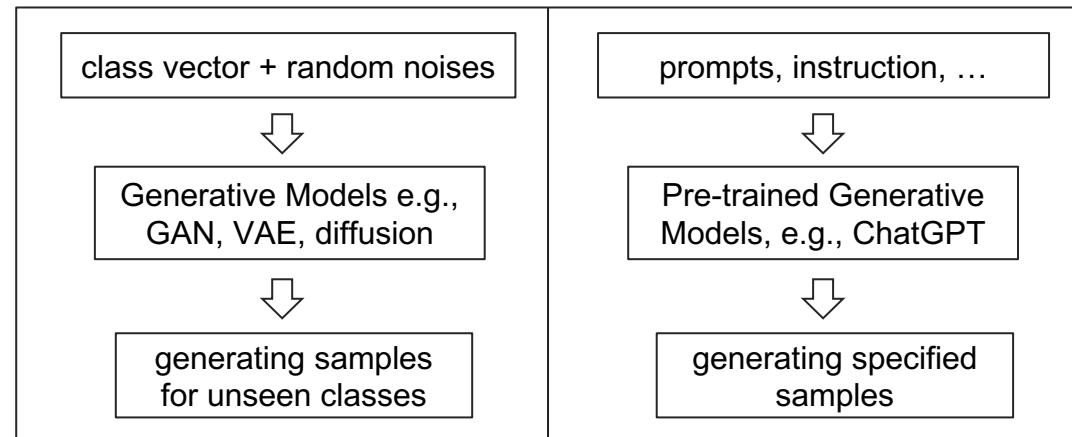
KGZSL: Knowledge Graph-based Sample Generation and ZSL Enhancement

Data Augmentation Strategy

Pseudo Samples by Priors

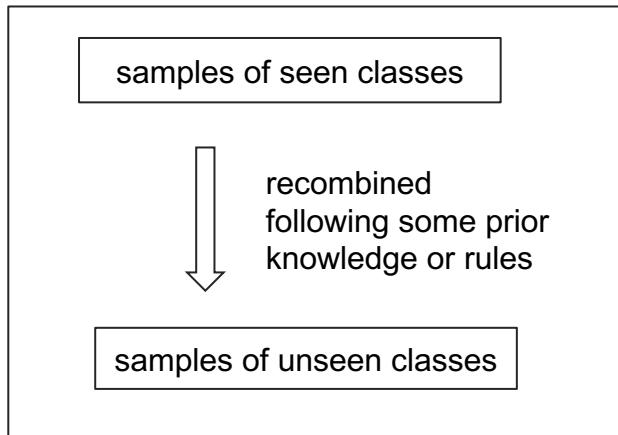


Generated Samples by Neural Networks

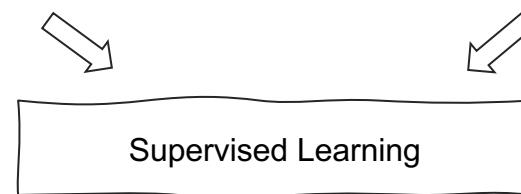
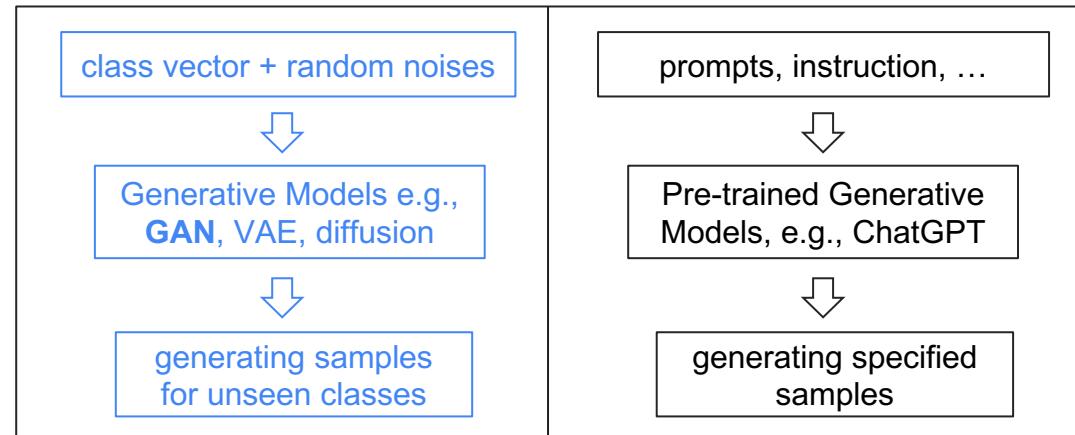


Data Augmentation Strategy

Pseudo Samples by Priors



Generated Samples by Neural Networks

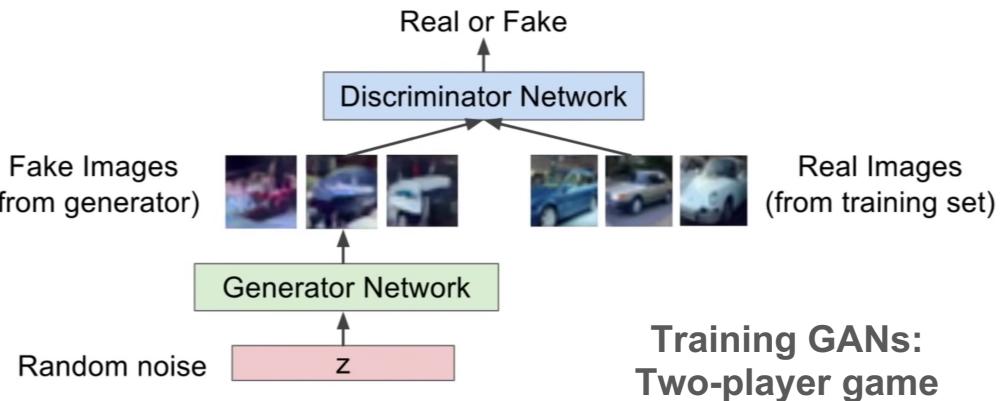


Generative Adversarial Networks (GANs)

- Generative models: Generative Adversarial Networks (GANs)



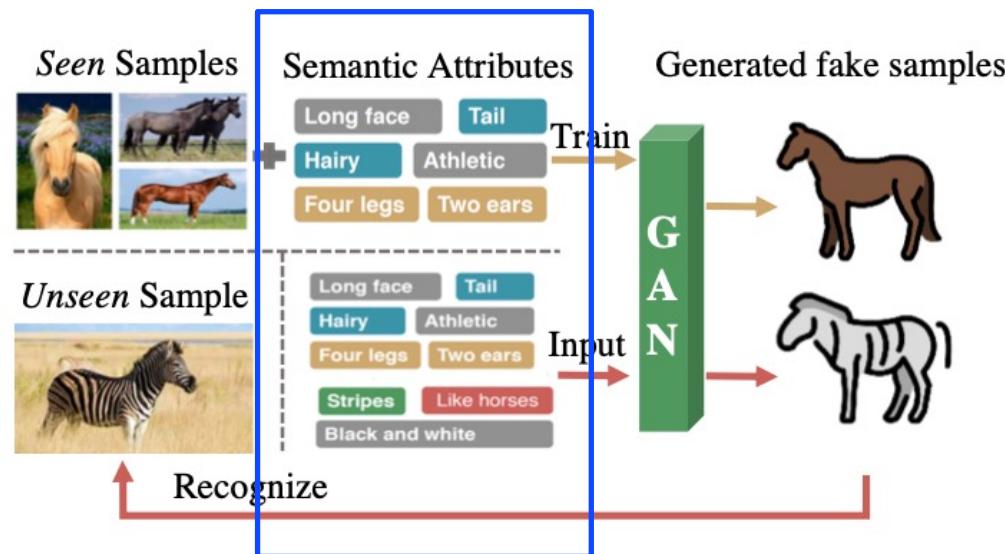
Generator Network: try to fool the discriminator by generating real-looking images;
Discriminator Network: try to distinguish between real and fake images



A human face synthesized by Deepfake

GANs for ZSL

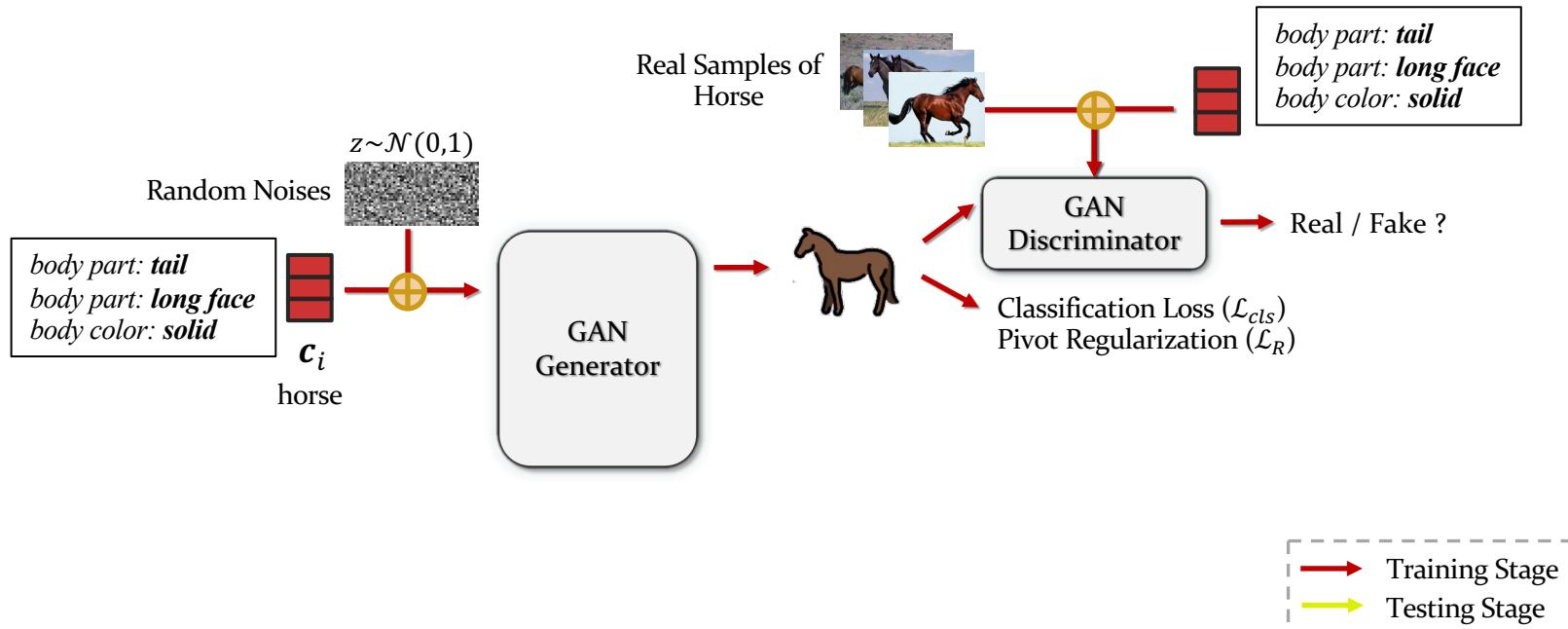
- Generative Adversarial Networks (GANs) & Zero-shot Learning



Conditional
GANs

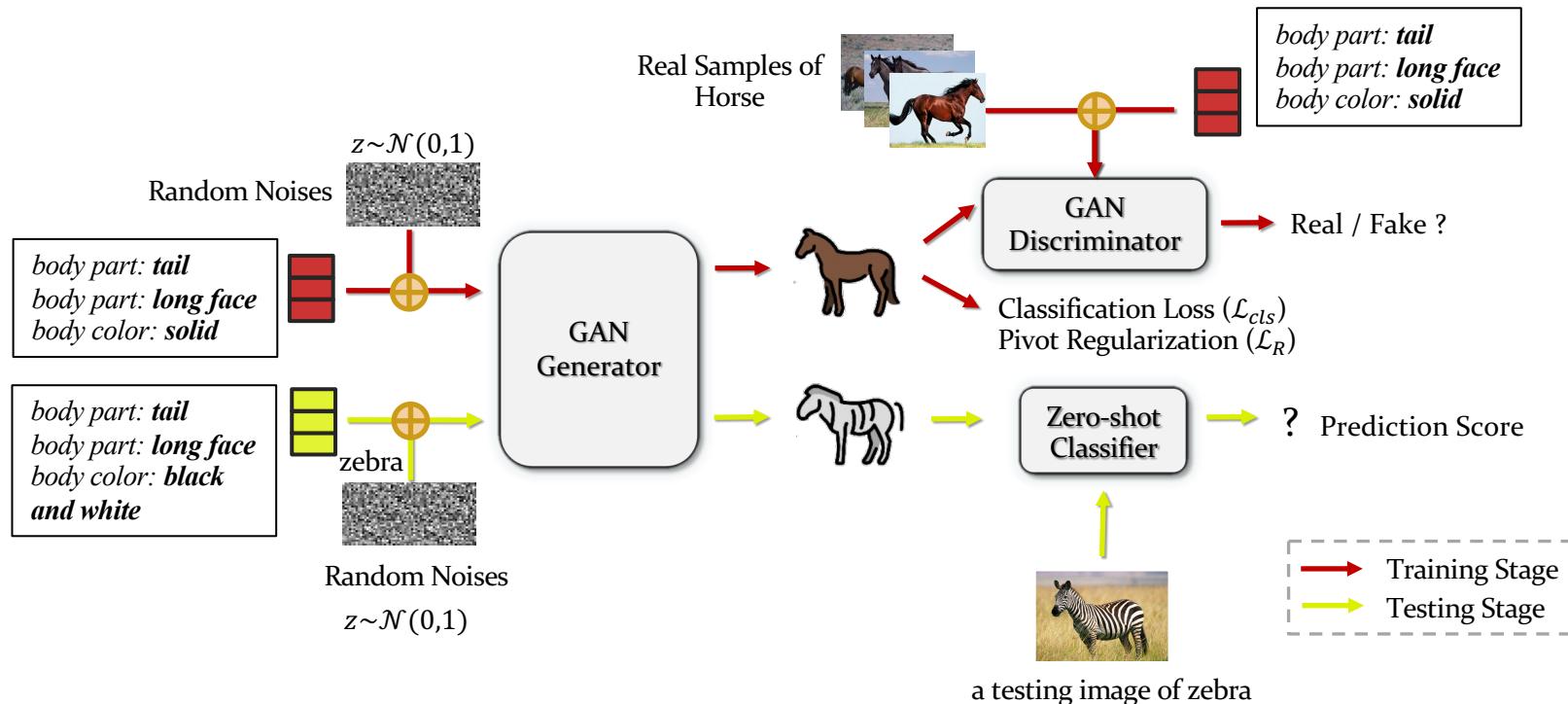
GANs for ZSL

- A running example with zero-shot animal classification



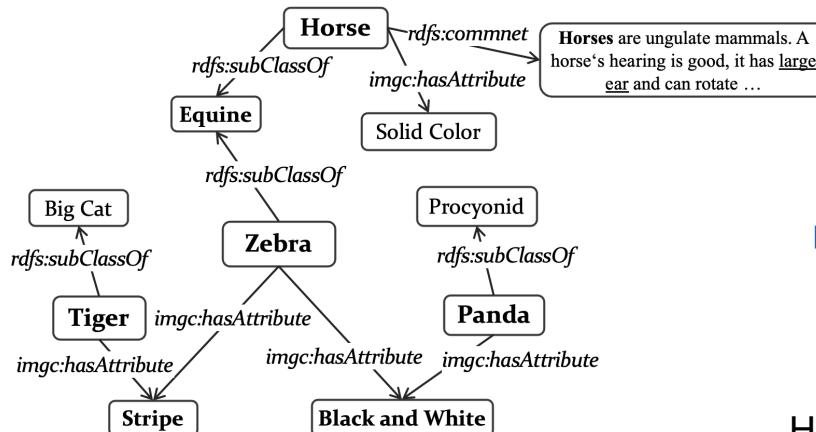
GANs for ZSL

- A running example with zero-shot animal classification



GANs for ZSL

- Knowledge Graphs have high compatibility in representing and integrating **diverse and discriminative** knowledge about seen and unseen classes



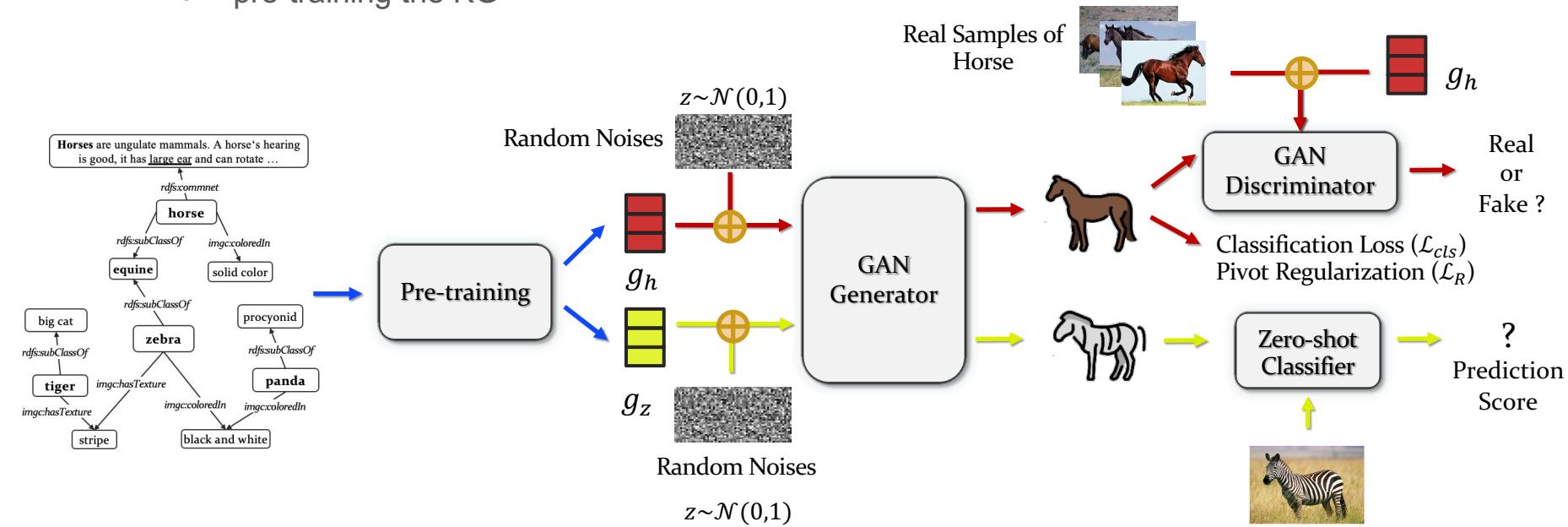
GAN

How do we incorporate a **symbolic KG** with GAN for ZSL problem?

a fragment of Knowledge Graph (KG)

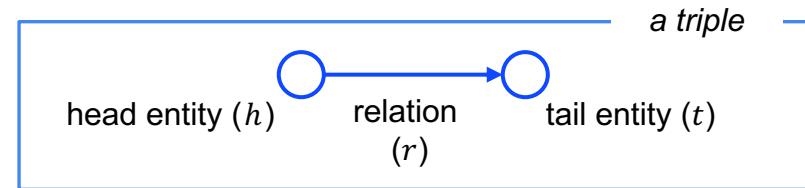
KGZSL: KG-enhanced Zero-shot Learning

- How do we incorporate a KG with GAN for ZSL problem?
 - pre-training the KG



KGZSL: KG-enhanced Zero-shot Learning

- How to pre-train a KG?
 - self-supervised training with triple supervision
 - scoring a triple under different assumptions



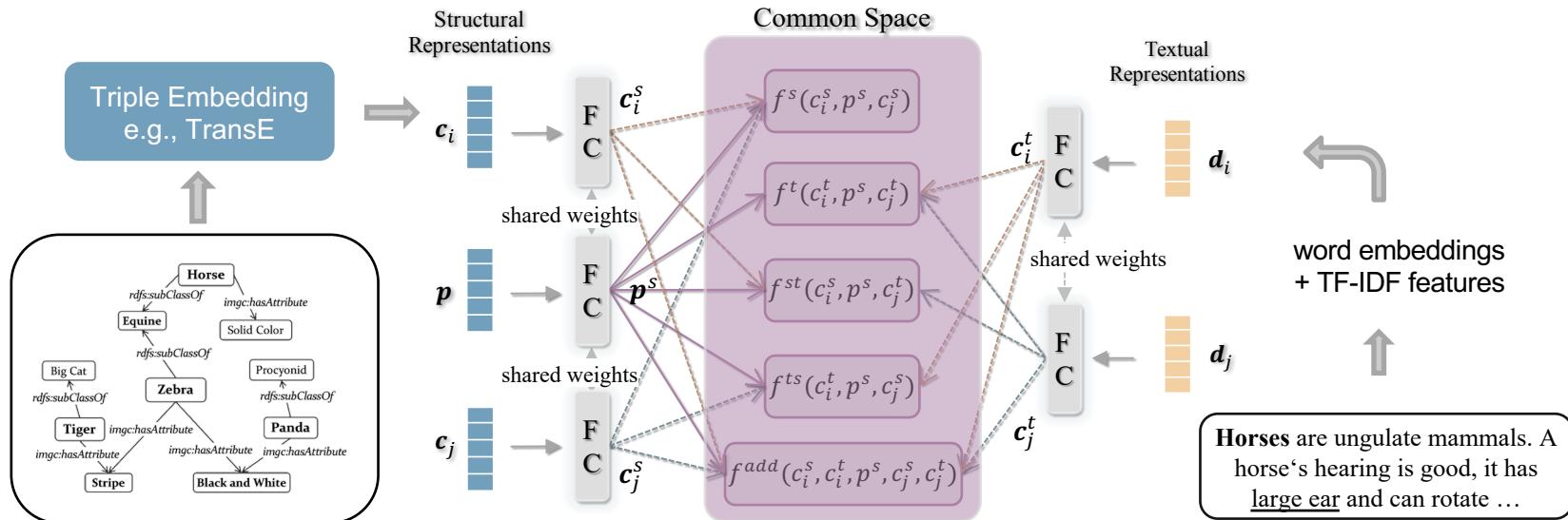
$$\min_{\Theta} \sum_{\tau^+ \in \mathbb{D}^+} \sum_{\tau^- \in \mathbb{D}^-} \max(0, \gamma - f_r(h, t) + f_{r'}(h', t'))$$

positive triples are expected to receive higher score than negative ones

vector space			
method	TransE	DistMult	RotatE
assumption	$h + r = t$	$h * M_r = t$	$h \circ r = t$
triple score	$f_r(h, t) = - h + r - t $	$f_r(h, t) = -(h^T \cdot \text{diag}(r) \cdot t)$	$f_{r(h,t)} = - h \circ r - t $

KGZSL: KG-enhanced Zero-shot Learning

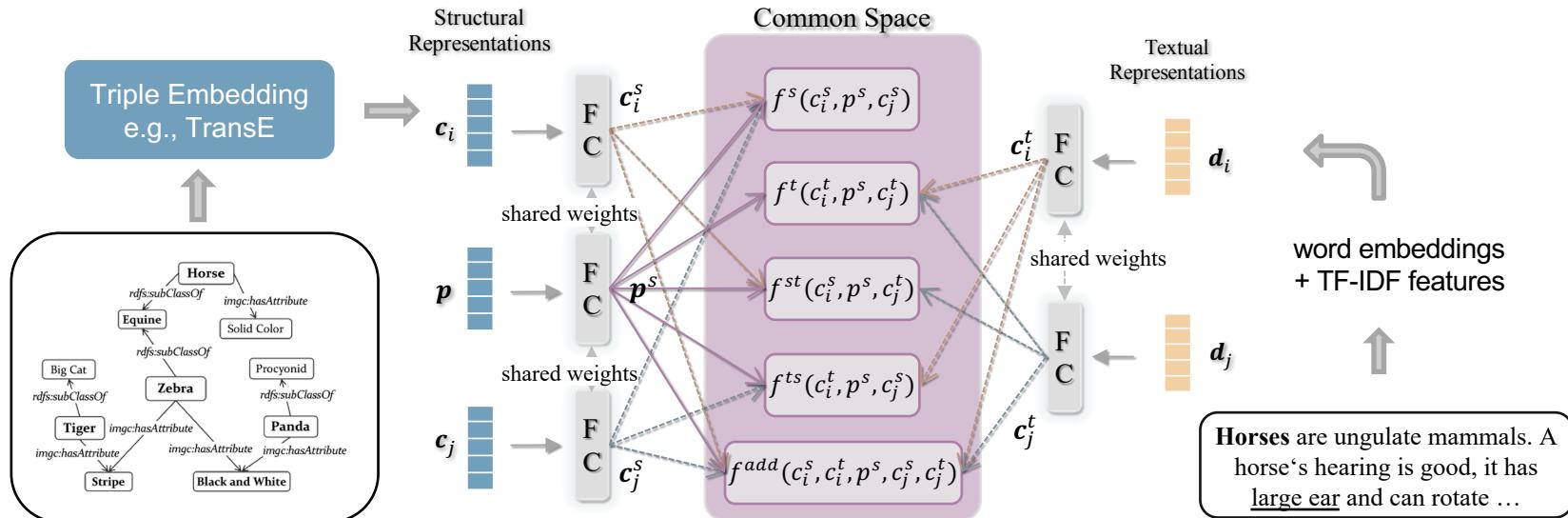
- text-aware pre-training



KGZSL: KG-enhanced Zero-shot Learning

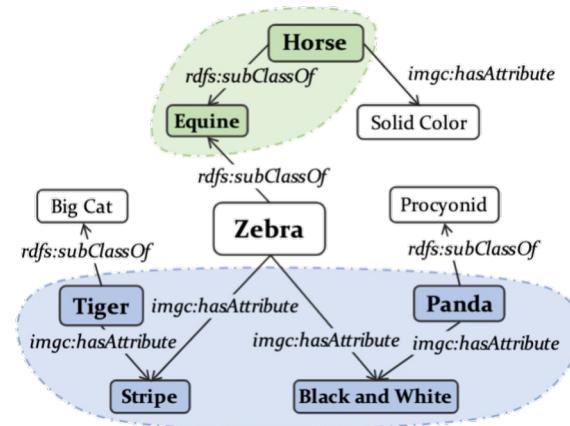
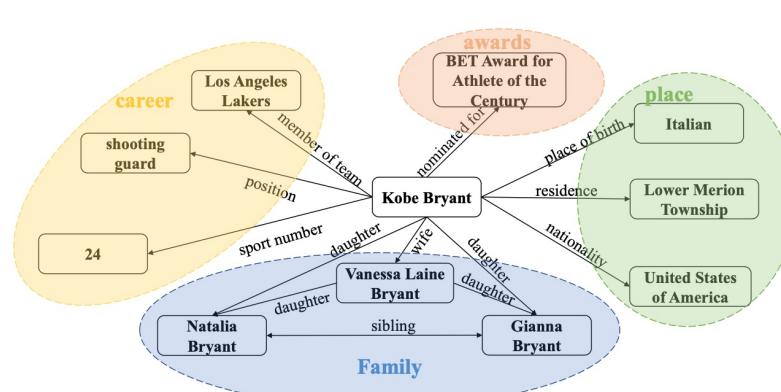
- text-aware pre-training

$$\mathbf{c}_i = [c_i^s; c_i^t]$$



DKZSL: KGZSL with Disentangled KG Embedding

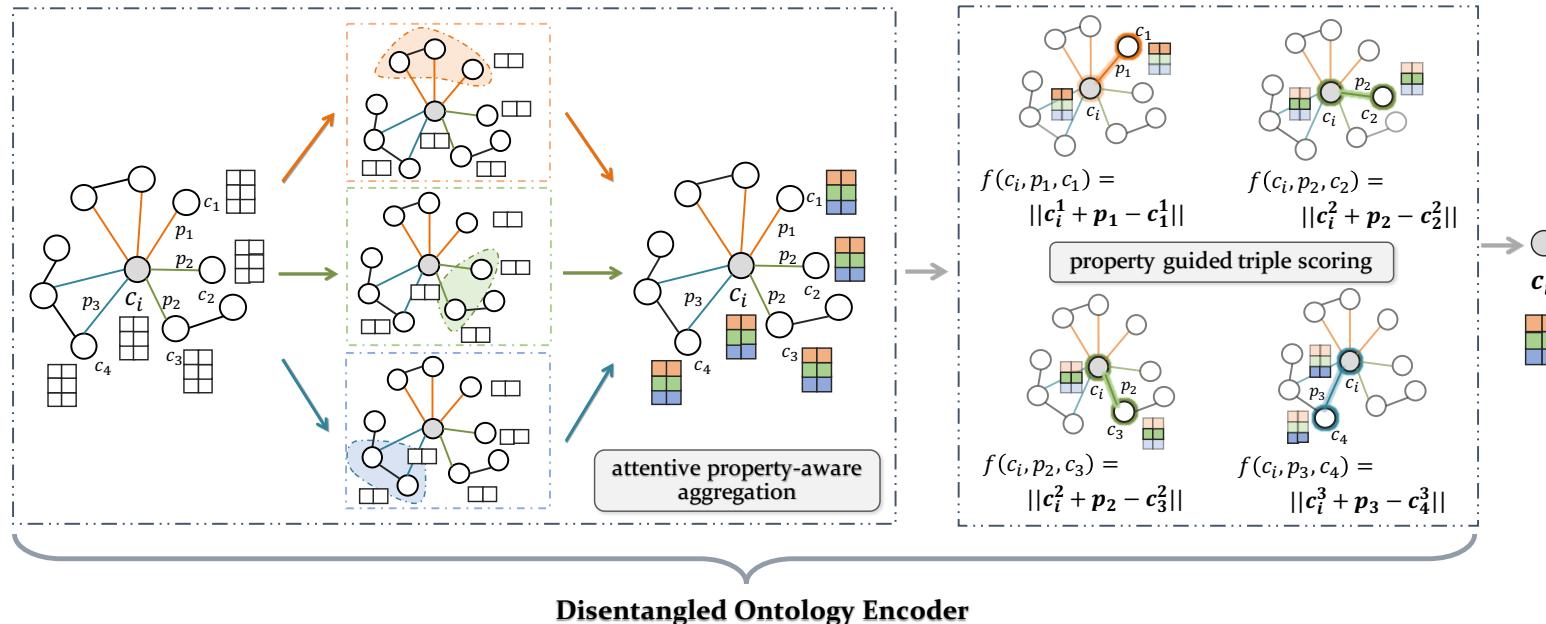
- different kinds of neighbors for a center entity
- disentanglement-aware pre-training



The **entanglement** characteristics of KGs: a class (entity) is often related to other classes (entities) in different semantic aspects.

DKZSL: KGZSL with Disentangled KG Embedding

- disentanglement-aware pre-training



DKZSL: KGZSL with Disentangled KG Embedding

- Learning disentangled embeddings for each class according to its semantics of different aspects
 - a) split the initial embedding of class i into K components, $\mathbf{c}_i = [\mathbf{c}_i^1, \mathbf{c}_i^2, \dots, \mathbf{c}_i^K]$;

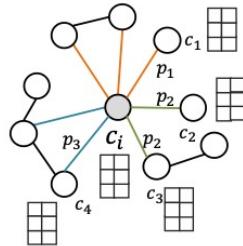


Illustration of DOZSL with $K = 3, d = 6$.

DKZSL: KGZSL with Disentangled KG Embedding

- Learning disentangled embeddings for each class according to its semantics of different aspects
 - split the initial embedding of class i into K components, $\mathbf{c}_i = [\mathbf{c}_i^1, \mathbf{c}_i^2, \dots, \mathbf{c}_i^K]$;
 - attentively aggregate the graph neighborhood information for each component;

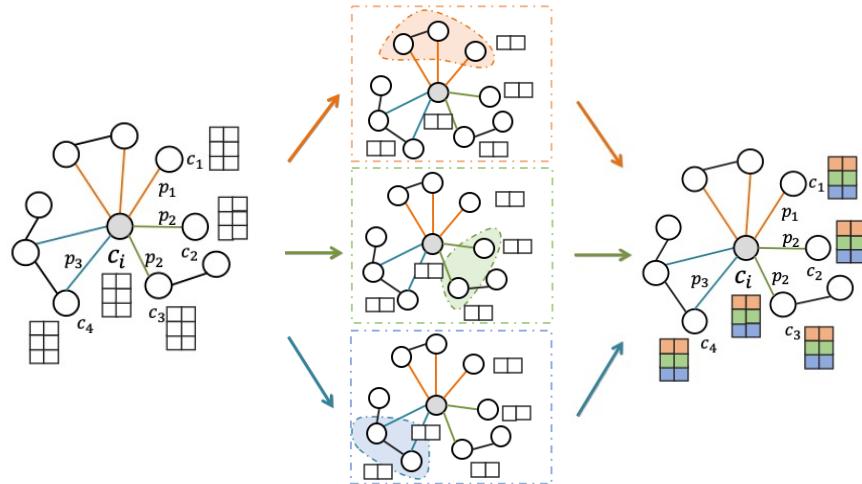


Illustration of DOZSL with $K = 3$, $d = 6$.

DKZSL: KGZSL with Disentangled KG Embedding

- Learning disentangled embeddings for each class according to its semantics of different aspects
 - split the initial embedding of class i into K components, $\mathbf{c}_i = [\mathbf{c}_i^1, \mathbf{c}_i^2, \dots, \mathbf{c}_i^K]$;
 - attentively aggregate the graph neighborhood information for each component;
 - extract **relation**-specific components to compute the triple scores to further refine the semantics of each component

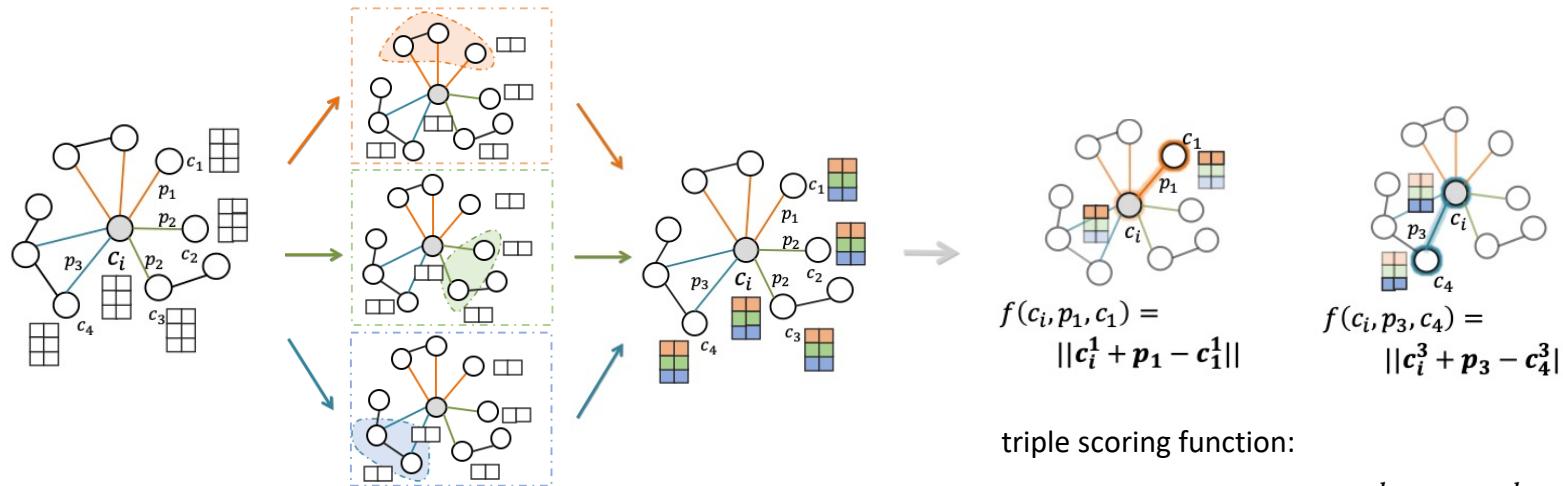
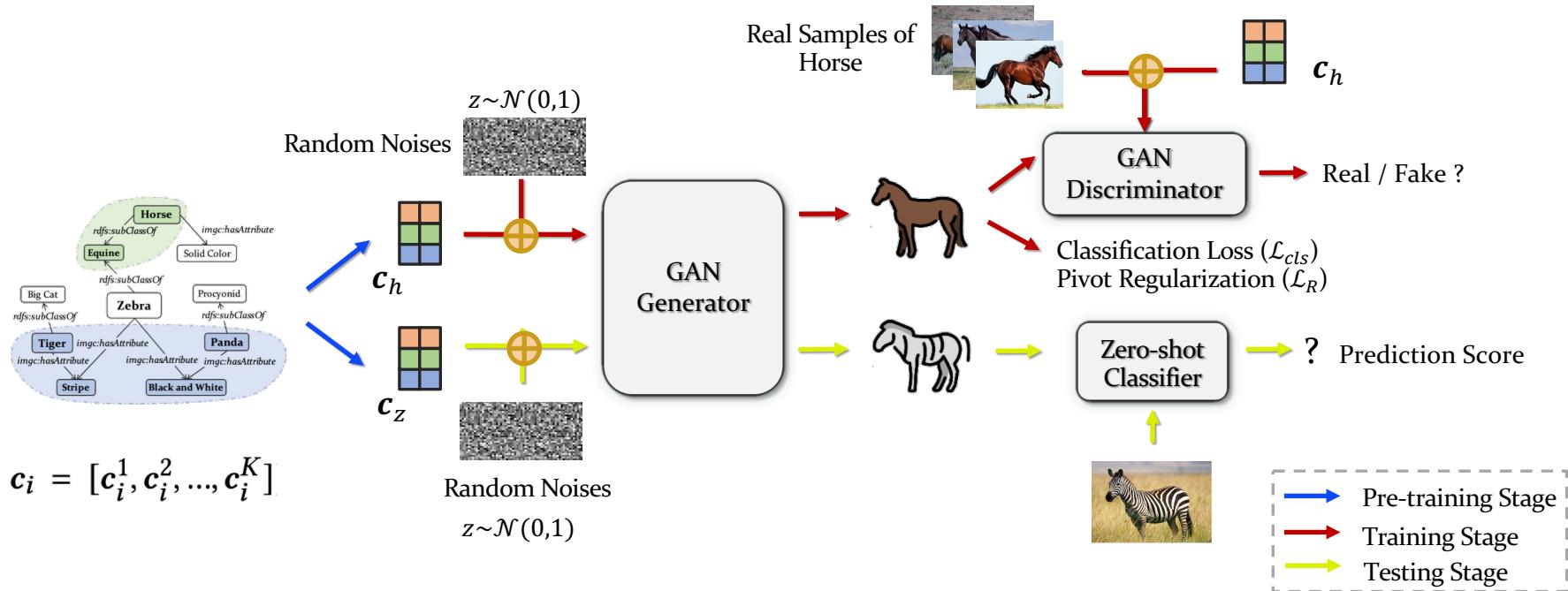


Illustration of DOZSL with $K = 3, d = 6$.

DKZSL: KGZSL with Disentangled KG Embedding

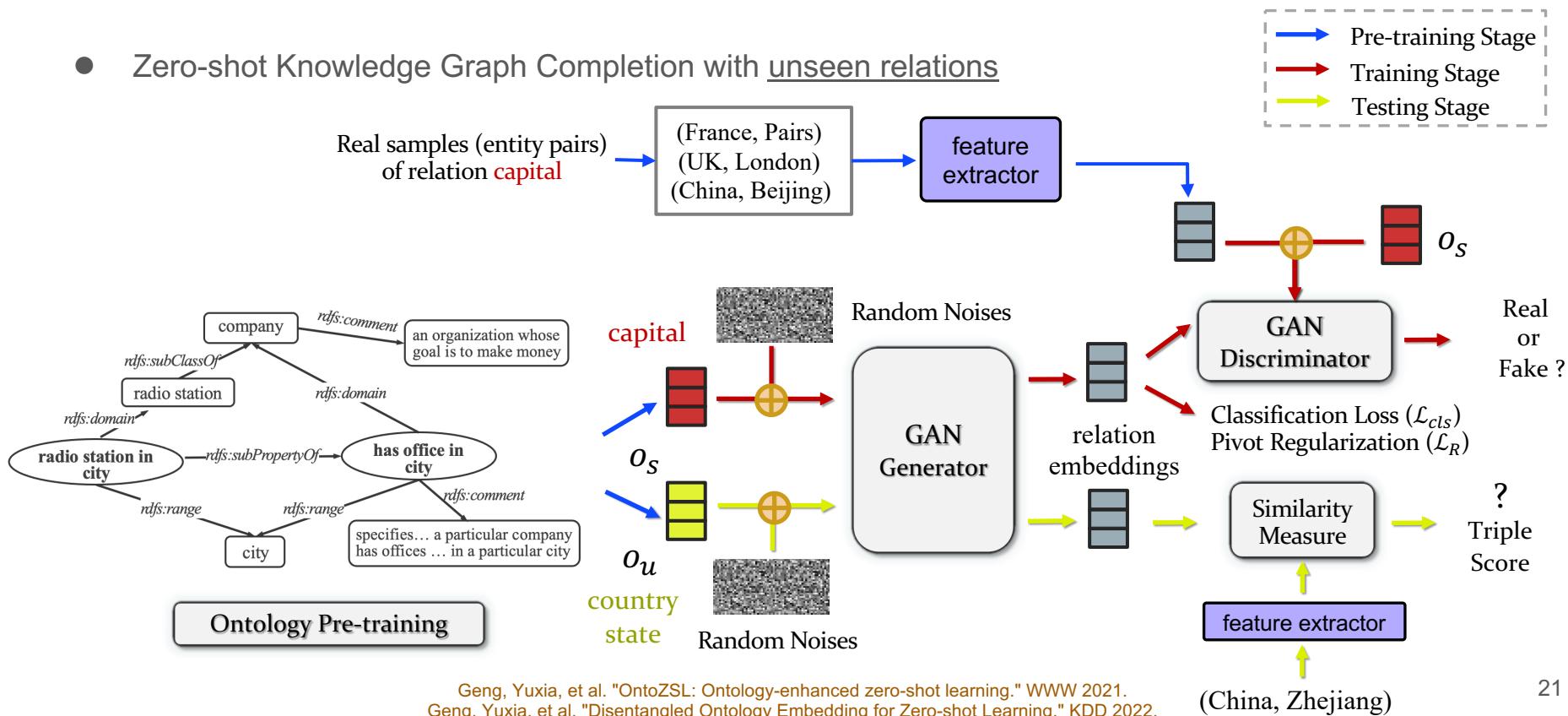
- disentanglement-aware pre-training



$$\mathbf{c}_i = [c_i^1, c_i^2, \dots, c_i^K]$$

OntoZSL & DOZSL

- Zero-shot Knowledge Graph Completion with unseen relations



More Reading

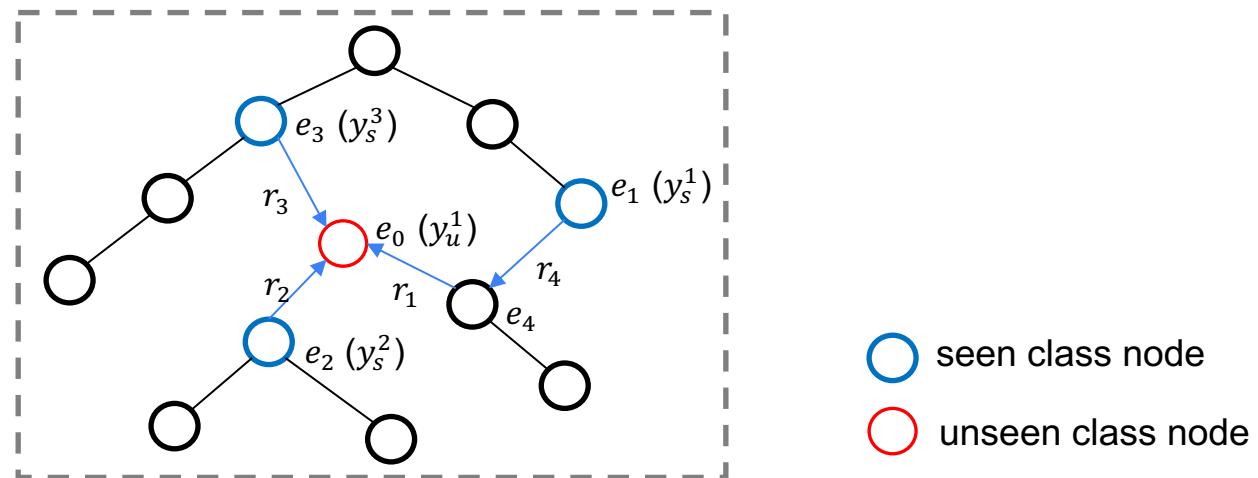
- Prior-based
 - Zero and Few Shot Learning with Semantic Feature Synthesis and Competitive Learning. TPAMI 2020
 - Iteratively learning embeddings and rules for knowledge graph reasoning. WWW 2019
 - Injecting Logical Background Knowledge into Embeddings for Relation Extraction. NAACL 2015
- VAE-based
 - Mishra et al., A Generative Model For Zero Shot Learning Using Conditional Variational Autoencoders, CVPR 2018
 - Wang et al., Zero-Shot Learning via Class-Conditioned Deep Generative Models, AAAI 2018
- Large Generative Model-based
 - Is synthetic data from generative models ready for image recognition? ICLR 2023

T4

Feature Propagation-based Methods for KG-aware ZSL

Feature Propagation Strategy

- Regarding the structural class knowledge contained in KGs or ontologies, how about directly performing computation (feature transfer) on graph to tackle ZSL problems?
- For example, propagate model parameters or features from seen class nodes to unseen class nodes



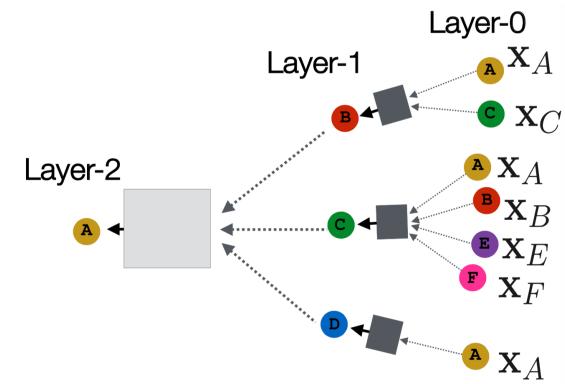
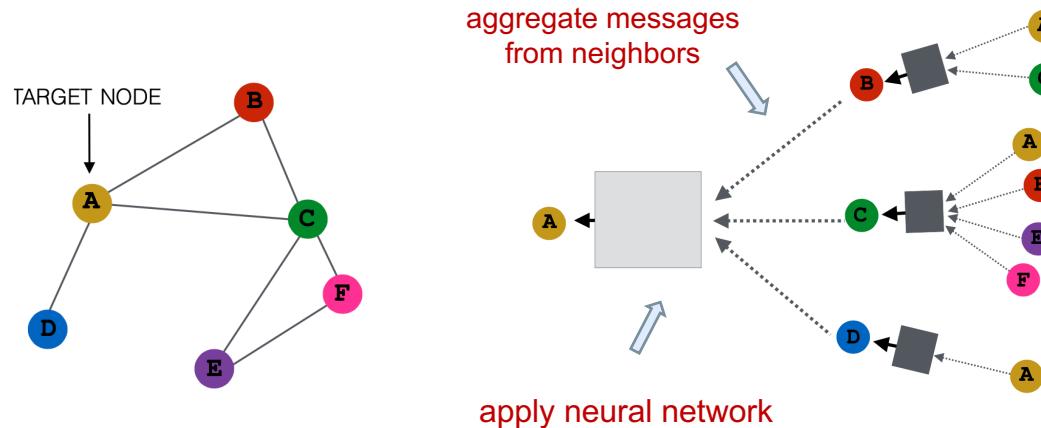
Graph Neural Networks (GNNs)

- Objective

- learn and output a meaningful embeddings for each node u , given the input graph $G = (V, E)$

- Important Steps

- neighbourhood aggregation: aggregating neighbour information and pass into a neural network
- self updating: update the node's hidden states using another network



Graph Neural Networks (GNNs)

- Representative GNNs

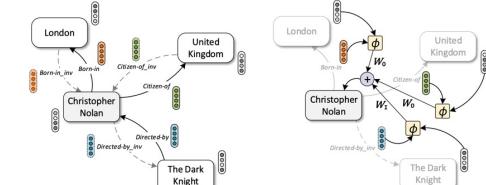
Composition-based Multi-Relational Graph Convolutional Networks (**CompGCN**)
Vashishth et al., ICLR'20

Modeling Relational Data with Graph Convolutional Networks (**RGCN**)
Schlichtkrull et al., ESWC'18

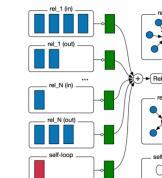
Graph Attention Networks (**GAT**)
Velickovic et al., ICLR'18

Inductive Representation Learning on Large Graphs (**GraphSAGE**)
Hamilton et al., NIPS'17

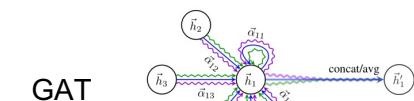
Semi-Supervised Classification with Graph Convolutional Networks (**GCN**)
Kipf and Welling, ICLR'17



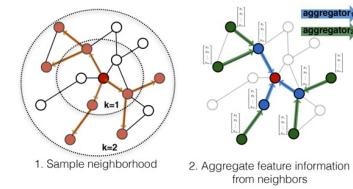
CompGCN



RGCN



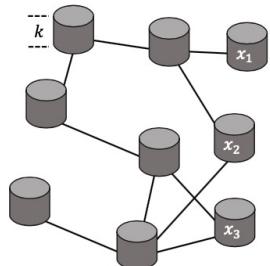
GAT



GraphSAGE

Propagate Classifiers on Graph: GCNZ

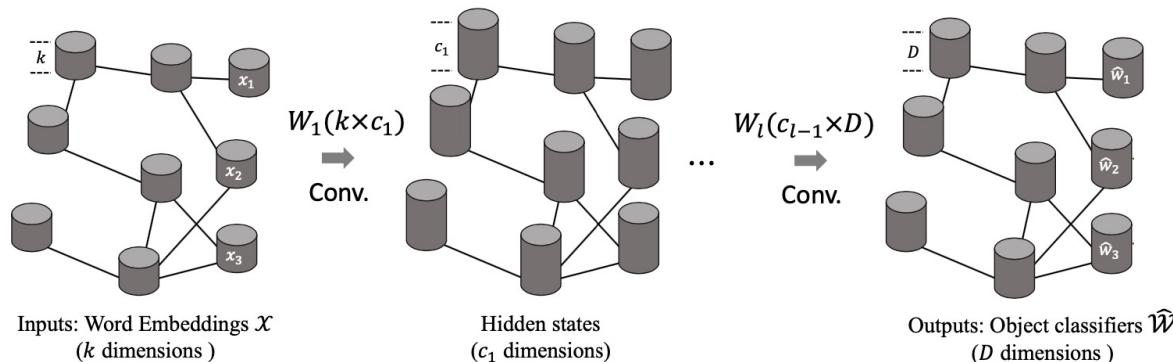
- Input: a graph $G = (V, E)$ with $|V|$ class nodes and hierarchical relationship edges from WordNet
 - Adjacency matrix
$$A_{i,j} = \begin{cases} a_{i,j} = 1 & (i,j) \in E \\ 0 & (i,j) \notin E \end{cases}$$
 - Node feature matrix: $X \in \mathbb{R}^{n \times k}$ for k -dim word embeddings of n class nodes



Inputs: Word Embeddings X
(k dimensions)

Propagate Classifiers on Graph: GCNZ

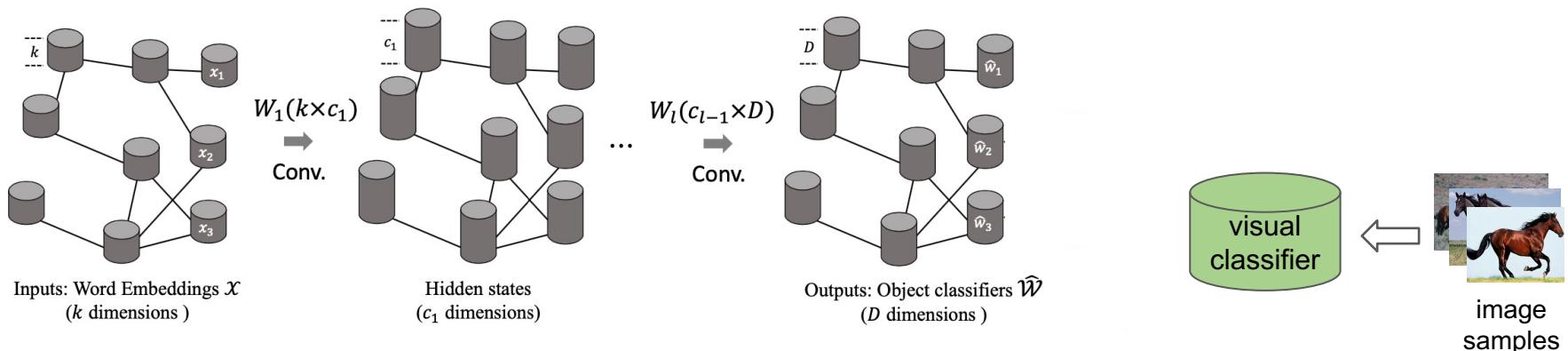
- Input: a graph $G = (V, E)$ with $|V|$ class nodes and hierarchical relationship edges from WordNet
 - Adjacency matrix
 - $A_{i,j} = \begin{cases} 1 & (i,j) \in E \\ 0 & (i,j) \notin E \end{cases}$
 - Node feature matrix: $X \in \mathbb{R}^{n \times k}$ for k -dim word embeddings of n class nodes



GCN: propagate features among class nodes and output classifiers

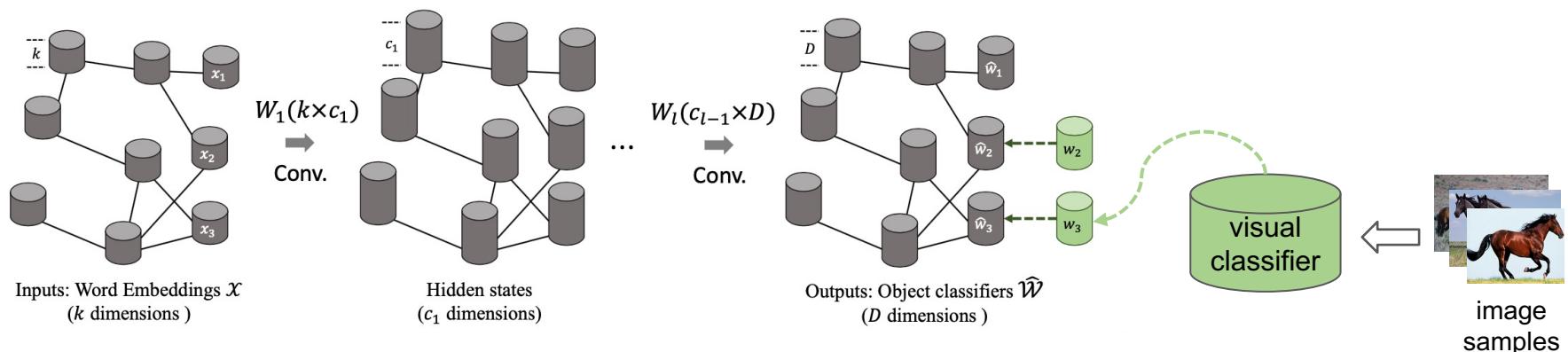
Propagate Classifiers on Graph: GCNZ

- Output: sample classifiers
 - each seen class node is supervised by a classifier, which is often a D -dim vector for class-specific sample features
 - each unseen class node is inferred to output its corresponding classifier for classification



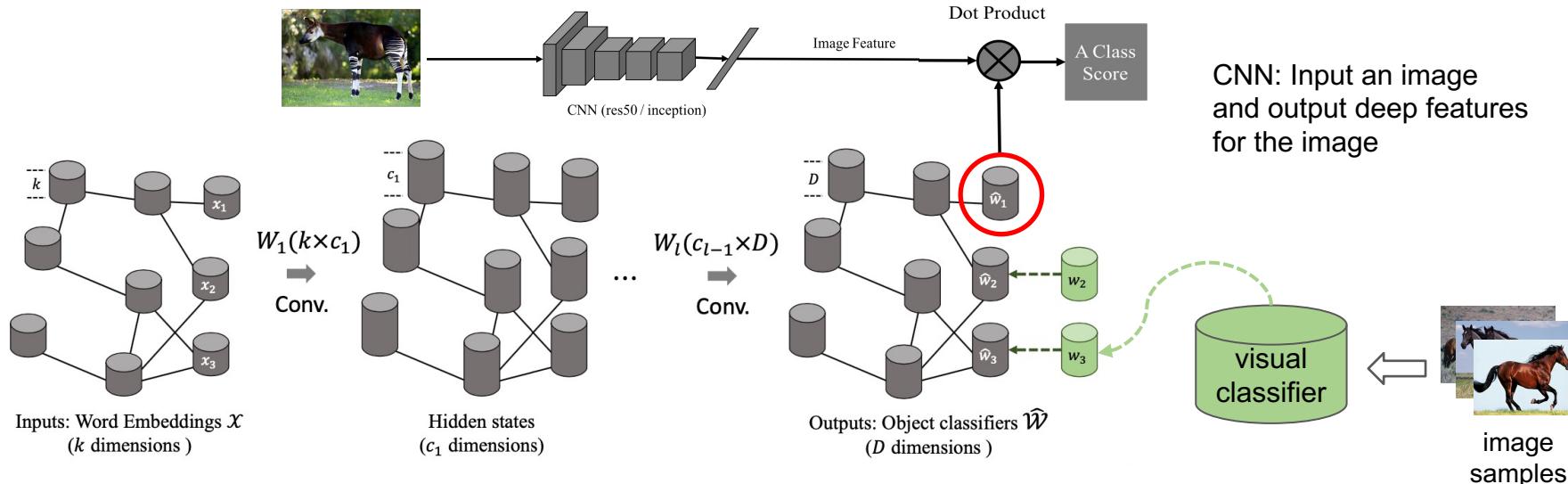
Propagate Classifiers on Graph: GCNZ

- Output: sample classifiers
 - each seen class node is supervised by a classifier, which is often a D -dim vector for class-specific sample features
 - each unseen class node is inferred to output its corresponding classifier for classification



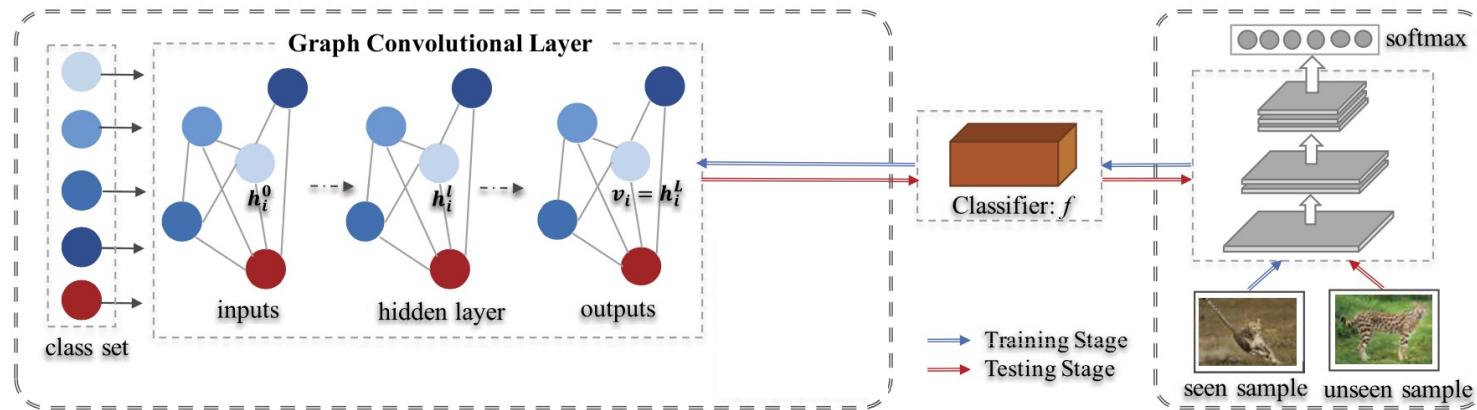
Propagate Classifiers on Graph: GCNZ

- Output: sample classifiers
 - each seen class node is supervised by a classifier, which is often a D -dim vector for class-specific sample features
 - each unseen class node is inferred to output its corresponding classifier for classification



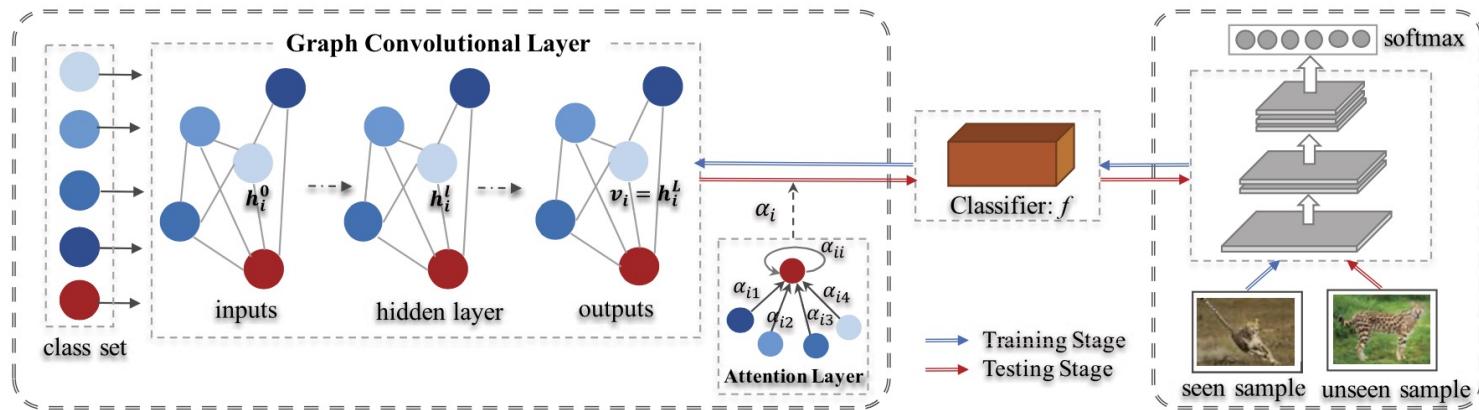
Propagate Classifiers on Graph: Attentive GCN

- Optimized Graph Propagation with **attention, to highlight significant neighbors**



Propagate Classifiers on Graph: Attentive GCN

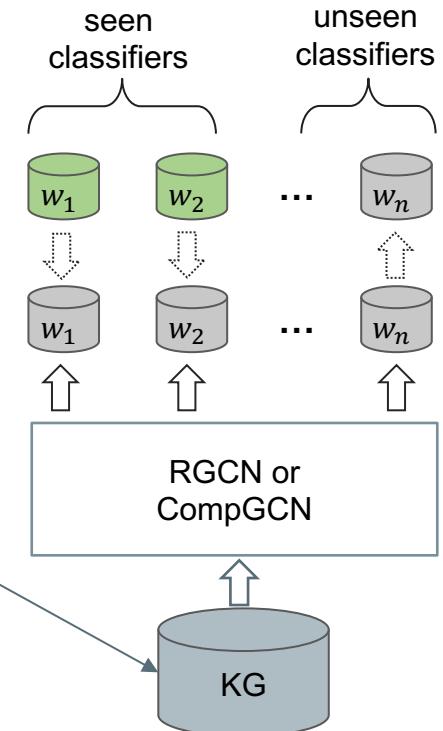
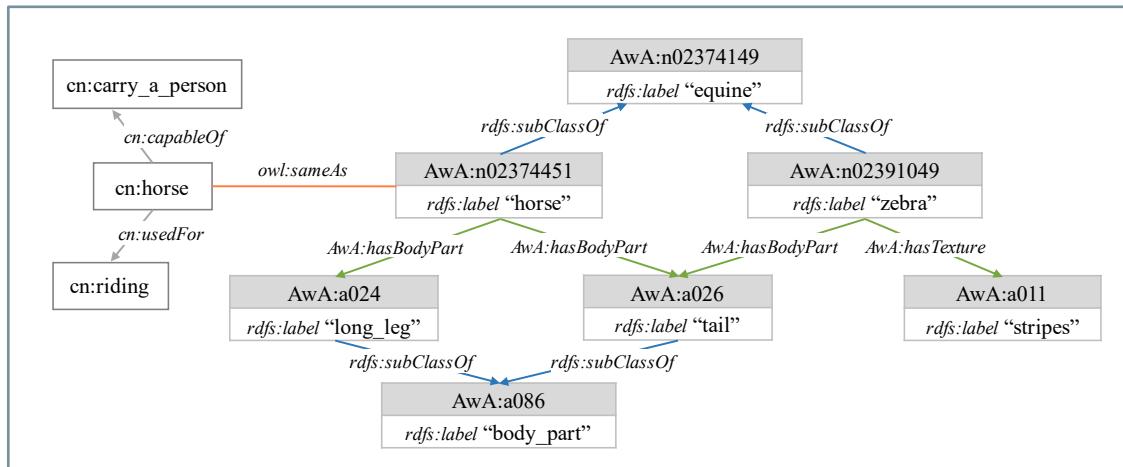
- Optimized Graph Propagation with **attention**, to highlight significant neighbors



$$\alpha_{ij} = \frac{\exp(\cos(v_i, v_j))}{\sum_{k \in \mathcal{N}_i} \exp(\cos(v_i, v_k))} \quad \bar{v}_i = \sum_{j \in \mathcal{N}_i} \alpha_{ij} \cdot v_j$$

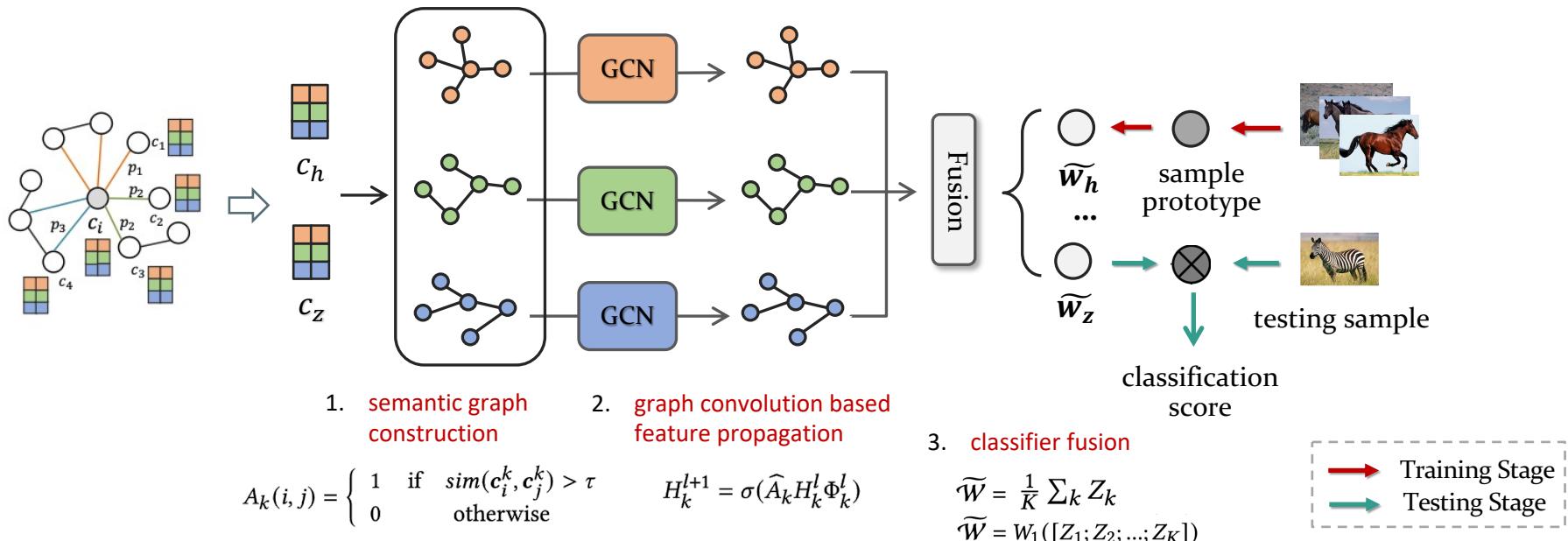
Propagate Classifiers on Graph: RGCN-ZSL, CompGCN-ZSL

- Optimized Graph Propagation with **relation-aware GCN**



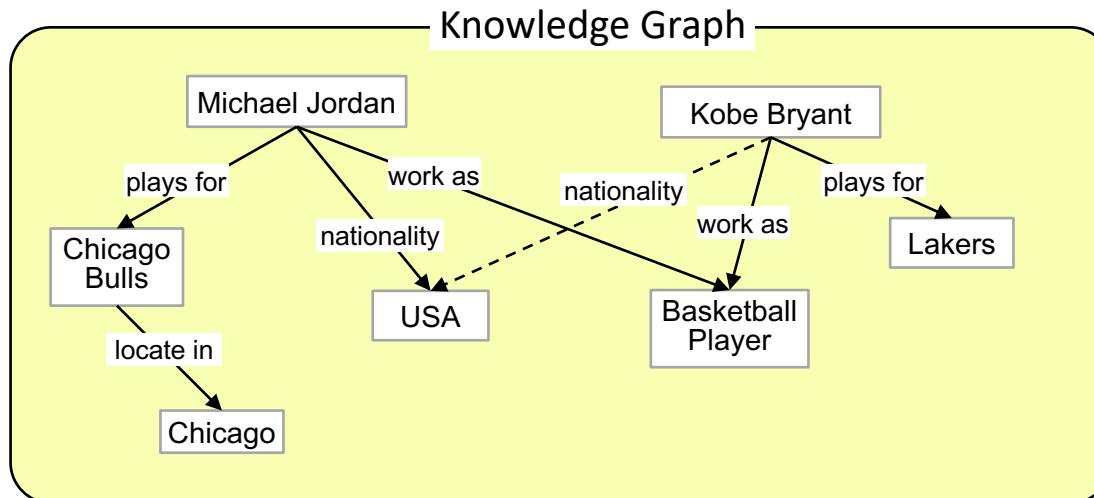
Propagate Classifiers on Graph: DKZSL

- Relational Graph Propagation with **disentangled semantic graphs**



Propagate Embeddings on Graph

- Knowledge Graph Completion

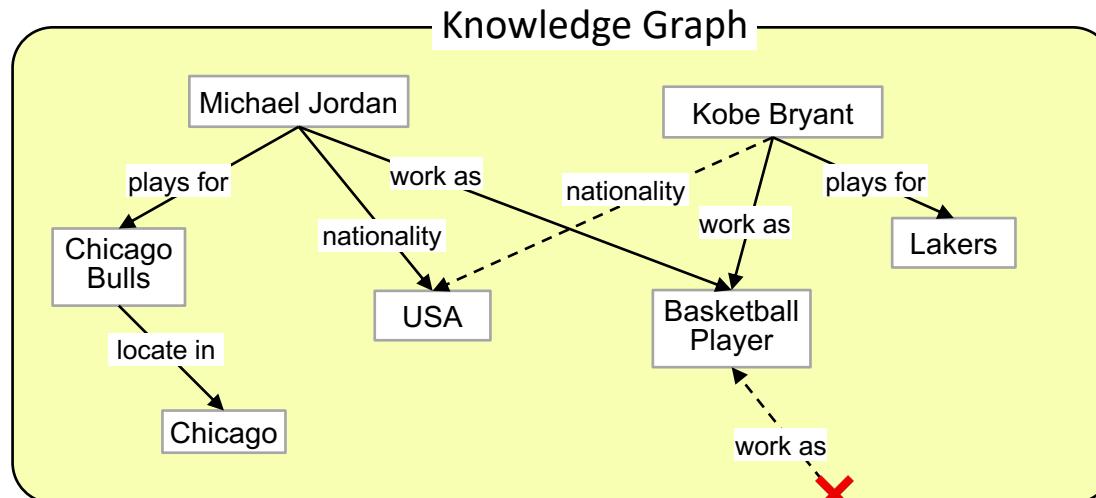


- (a) Training: learn embeddings for entities and relations
- (b) Traditional Completion: perform vector computation on the learned embeddings

👍 *TransE, DistMult, RotatE, ...*

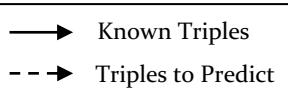
Propagate Embeddings on Graph

- Zero-shot Knowledge Graph Completion



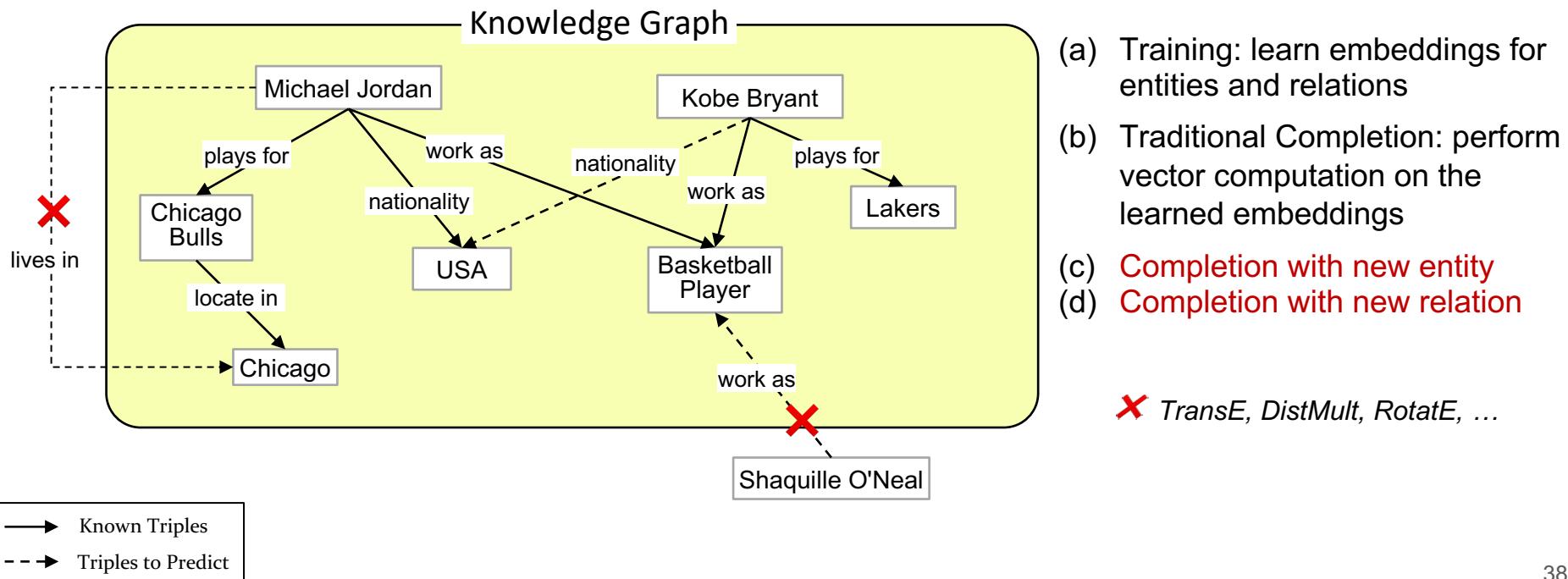
- Training: learn embeddings for entities and relations
- Traditional Completion: perform vector computation on the learned embeddings
- Completion with new entity

✗ *TransE, DistMult, RotatE, ...*



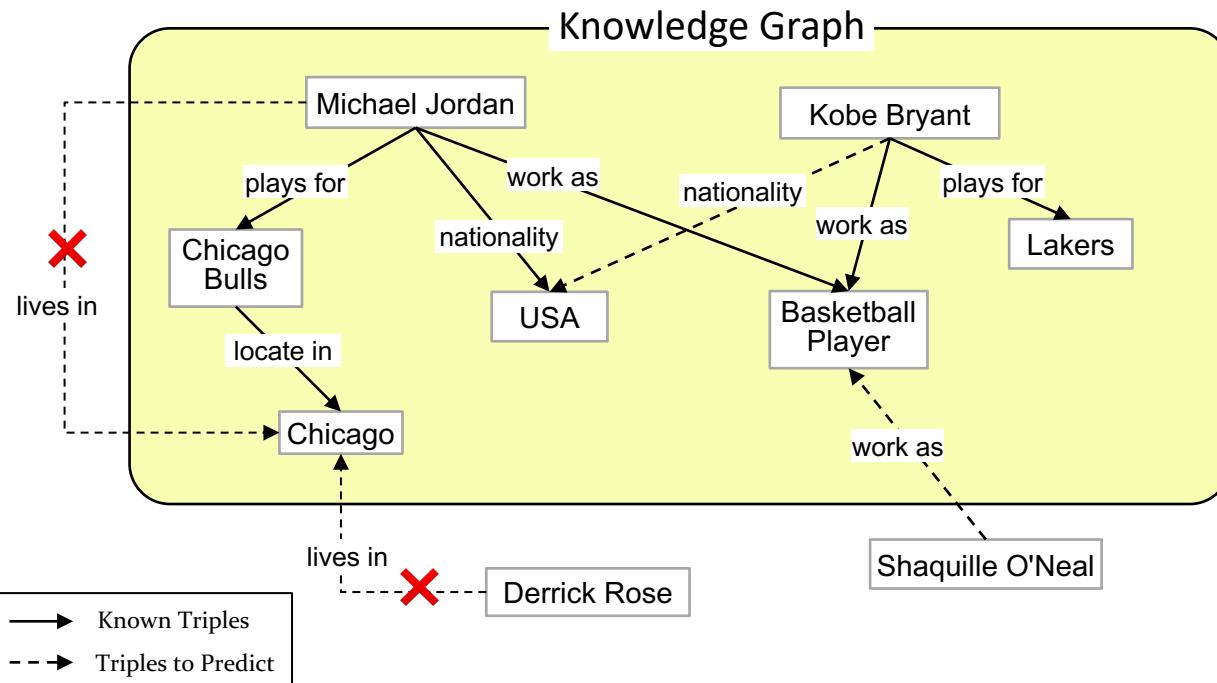
Propagate Embeddings on Graph

- Zero-shot Knowledge Graph Completion



Propagate Embeddings on Graph

- Zero-shot Knowledge Graph Completion

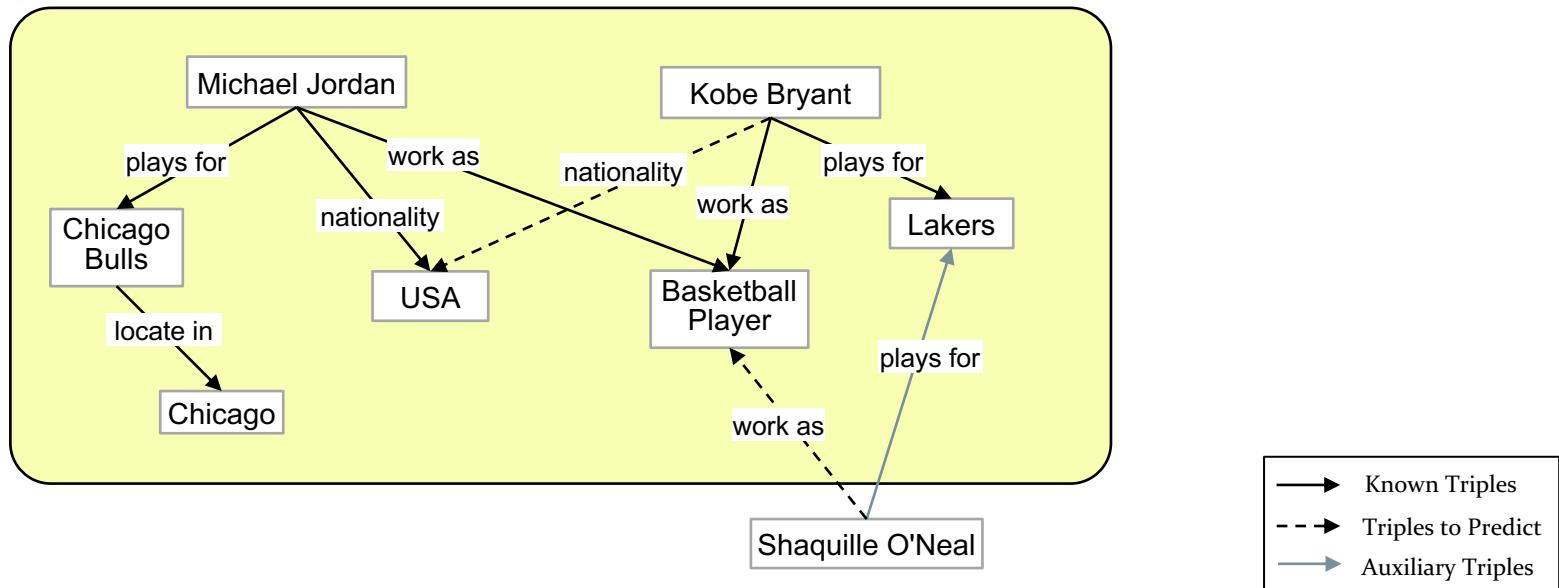


- Training: learn embeddings for entities and relations
- Traditional Completion: perform vector computation on the learned embeddings
- Completion with new entity
- Completion with new relation
- Completion with new entity and new relation

✗ *TransE, DistMult, RotatE, ...*

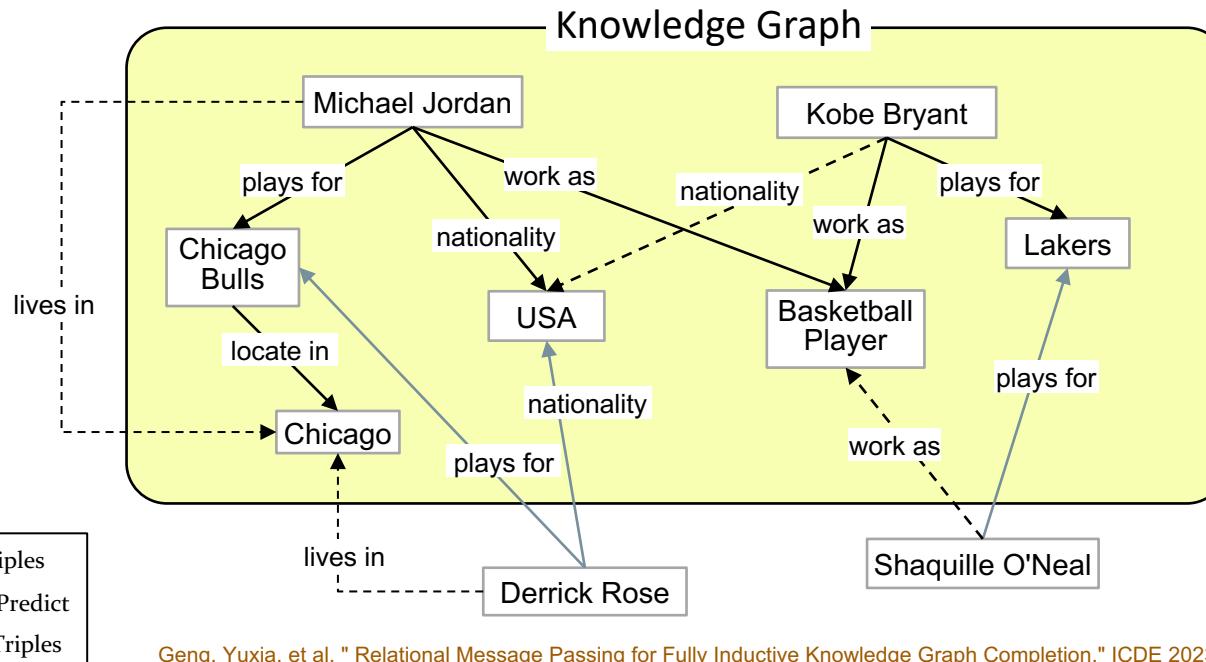
Propagate Entity Embeddings on Graph

- New (unseen) entities are represented by aggregating neighboring trained entities and relations

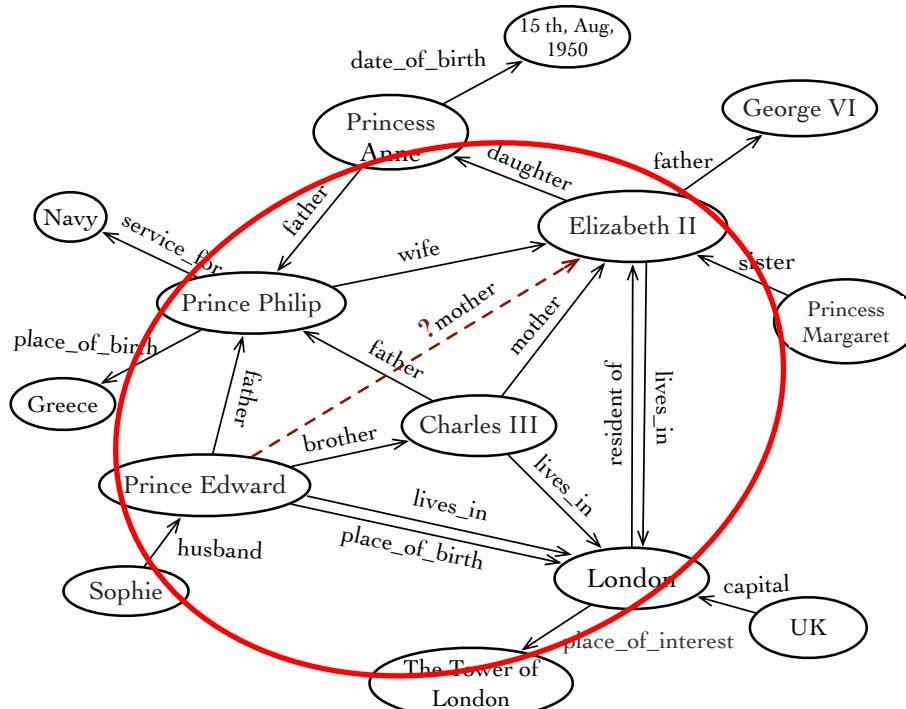


Propagate Relation Embeddings on Graph: RMPI

- generalize to unseen relations and unseen entities simultaneously



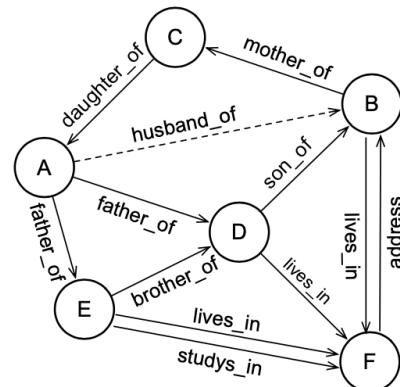
Propagate Relation Embeddings on Graph: RMPI



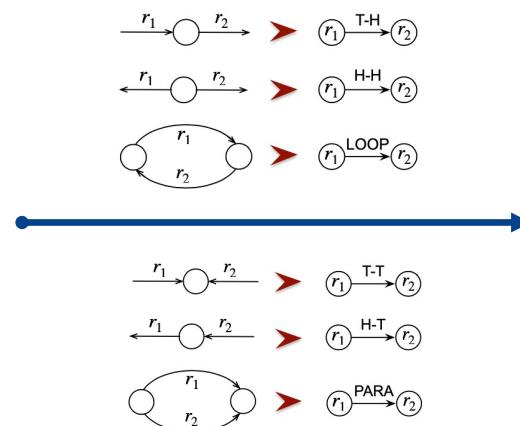
- New (unseen) relations are represented by aggregating neighboring trained relations, can generalize to unseen relations
- RMPI infers the plausibility of a triple without knowing entities, can generalize to unseen entities

Propagate Relation Embeddings on Graph: RMPI

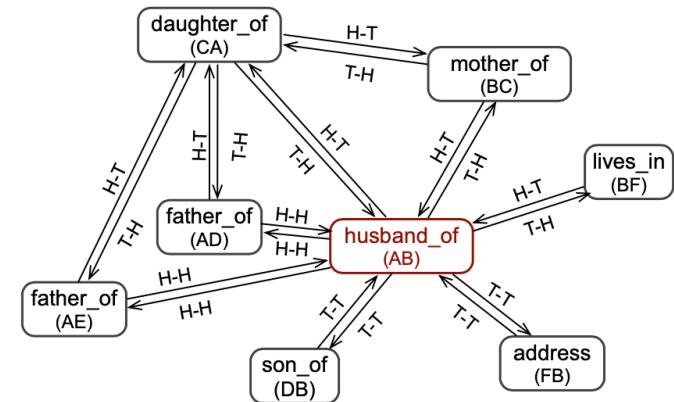
- Subgraph Extraction and Transformation (by 6 meta-relations)



Graph in Entity View



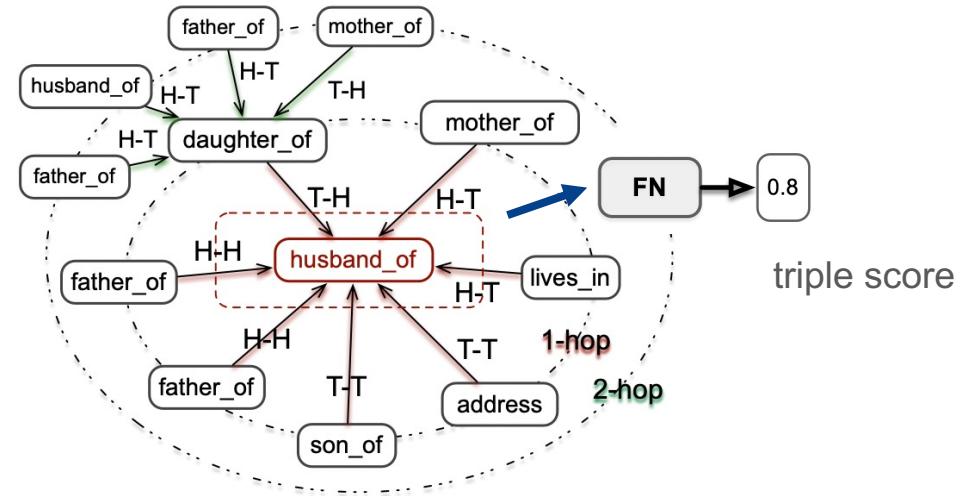
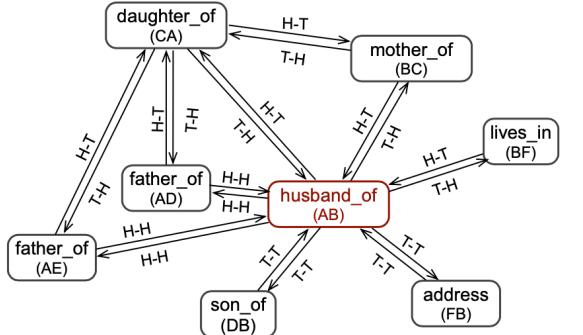
Meta Relations
connection patterns of relations in the original graph



Graph in Relation View

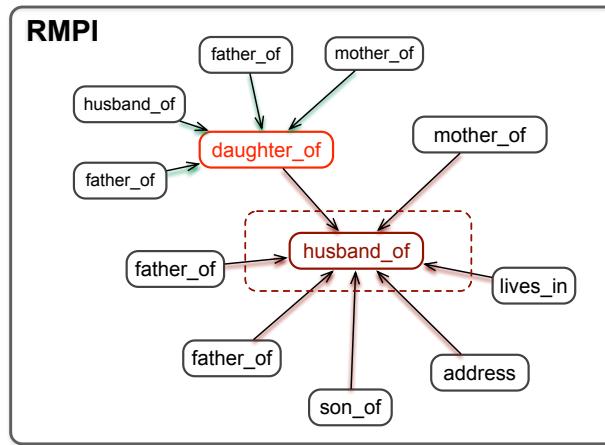
Propagate Relation Embeddings on Graph: RMPI

- Relational Message Passing (Feature Propagation) Network for directly propagating features between relations
 - a target relation-guided graph pruning strategy
 - a target relation-aware neighborhood aggregation mechanism



Propagate Relation Embeddings on Graph: RMPI

- Infer embeddings for unseen relations

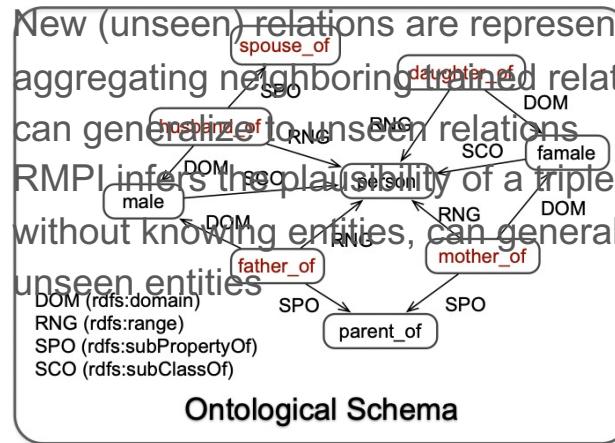


- aggregate features of their neighboring trained seen relations

- explore more relation correlations from KG's Ontology

1) pretrain the ontology and initialize relation embeddings used for relation message passing

- New (unseen) relations are represented by aggregating neighboring trained relations, can generalize to unseen relations
- RMPI infers the plausibility of a triple without knowing entities, can generalize to unseen entities



More Reading

- Propagate Classifiers
 - Kampffmeyer, Michael, et al. "Rethinking knowledge graph propagation for zero-shot learning." CVPR 2019.
 - Geng, Yuxia, et al. "Explainable zero-shot learning via attentive graph convolutional network and knowledge graphs." Semantic Web Journal 2021.
 - Wang et al., Zero-Shot Learning via Contrastive Learning on Dual Knowledge Graphs, ICCV 2021
 - Chen et al., Zero-shot Ingredient Recognition by Multi-Relational Graph Convolutional Network, AAAI 2020
- Propagate Embeddings (entity or image)
 - Lee, Chung-Wei, et al. "Multi-label zero-shot learning with structured knowledge graphs." CVPR 2018.
 - Wang, Peifeng, et al. "Logic attention based neighborhood aggregation for inductive knowledge graph embedding." AAAI 2019.
- Subgraph Reasoning
 - Teru, Komal, et al. "Inductive Relation Prediction by Subgraph Reasoning". ICML 2020.
 - Chen, Jiajun, et al. "Topology-aware correlations between relations for inductive link prediction in knowledge graphs." AAAI 2021.



Knowledge-aware Zero-shot Learning (K-ZSL): Concepts, Methods and Resources

Yuxia Geng¹, Zhuo Chen², Jiaoyan Chen³, Wen Zhang² and Jeff Z. Pan⁴

1. Hangzhou Dianzi University, China

2. Zhejiang University, China

3. The University of Manchester & University of Oxford, UK

4. The University of Edinburgh, UK

<https://china-uk-zsl.github.io/kg-zsl-tutorial-ijcai-2023/>

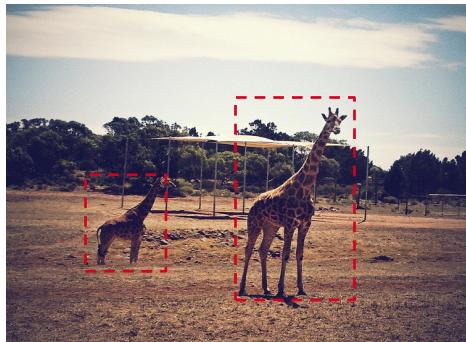
Tutorial of The 32nd International Joint Conference on Artificial Intelligence (19th August, 2023, Macao, S.A.R)

T5

KG Augmented Zero-shot Visual Question Answering

Knowledge-aware Visual Question Answering (K-VQA)

- Traditional VQA (E.g., VQA 2.0)
 - Multi-modal feature fusion



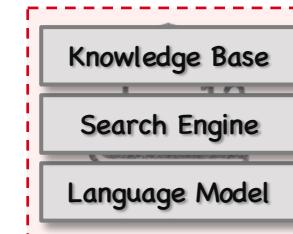
+ Q1: How many giraffes are there? = Ans: two

Knowledge-aware Visual Question Answering (K-VQA)

- **Knowledge-based VQA** (E.g., F-VQA)
 - Querying constructed sub-KG through SPARQL
 - Get unstructured knowledge via using search engine
 - Large-scale language model



+ **Q2: What object in this image is a toy?**



= Ans: toy

Zero-shot Visual Question Answering (ZS-VQA)

- Open-ended (Top-K) VQA: Answer Pool



What object in this image is a toy?

Answer: frisbee ?



What object has engine?

Answer: motorcycle ?

- Candidate Answer Pool:
person, bus, wall, desk, water, bicycle, tree, street, chair



Zero-shot Visual Question Answering (ZS-VQA)

- Zero-shot open-ended (Top-K) VQA: **Unseen Answer**



What object in this image is a toy?



What object has engine?

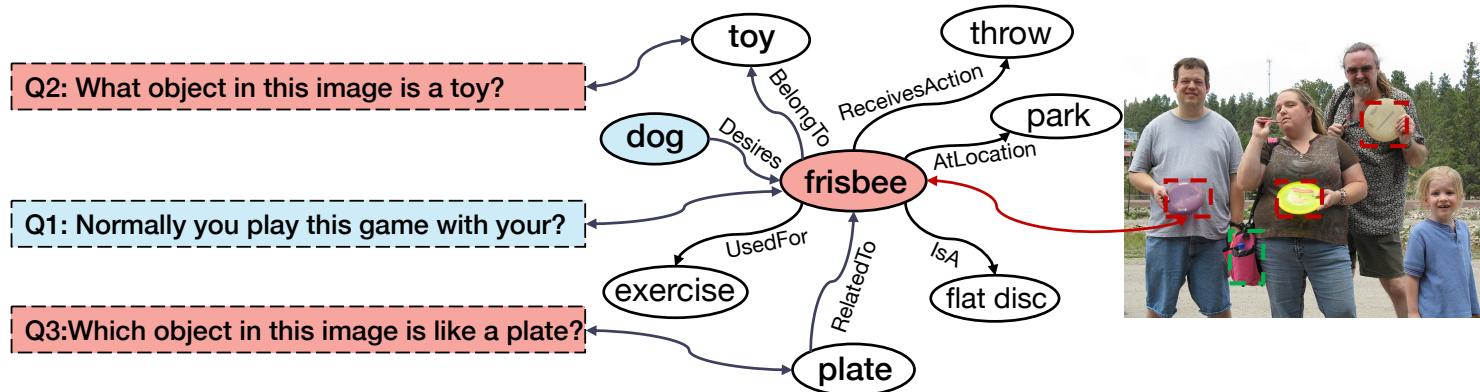


- Candidate Answer Pool:
person, bus, wall, desk, water, bicycle, tree, street, chair → **frisbee, luggage, hotdog, motorcycle**

KG Augmented Zero-shot Visual Question Answering

IJCAI 2023 K-ZSL Tutorial (Part II)

T5: KG ZS-VQA

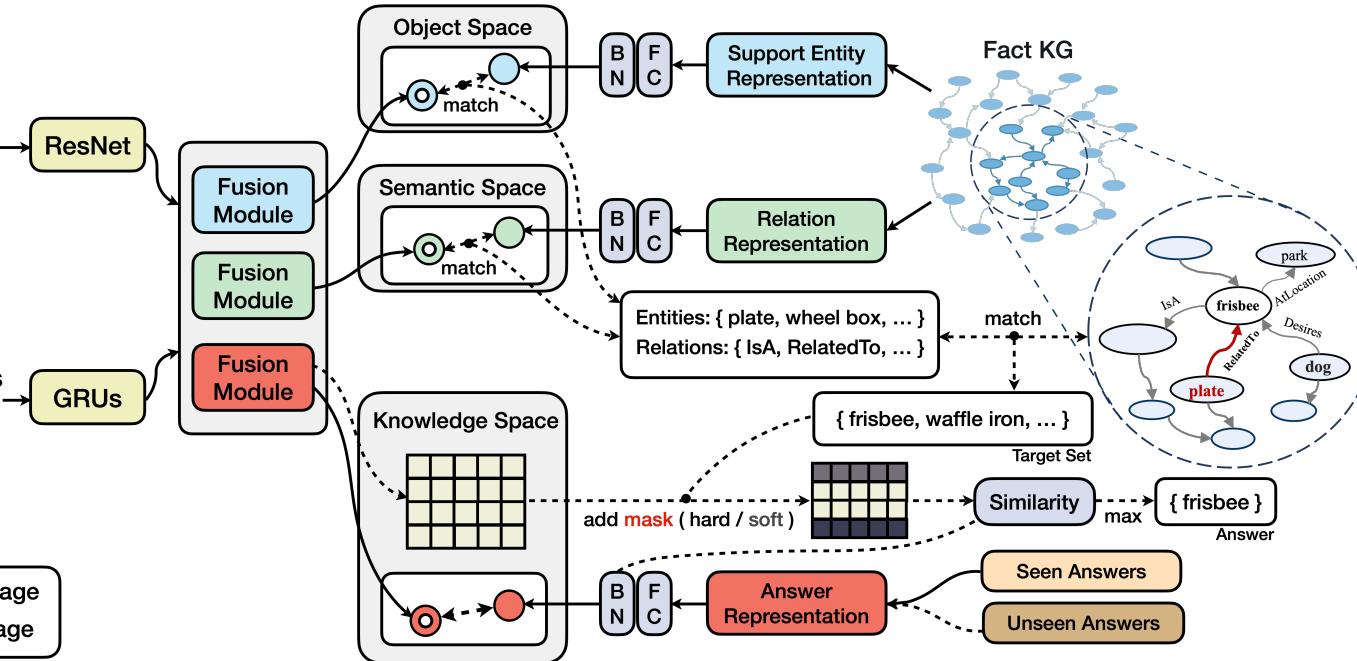


- A Zero-shot VQA algorithm using knowledge graphs and a mask-based learning mechanism for better incorporating external knowledge.
- A new answer-based Zero-shot VQA split for the F-VQA dataset.

KG Augmented Zero-shot Visual Question Answering



Q: Which object in this image is like a plate?

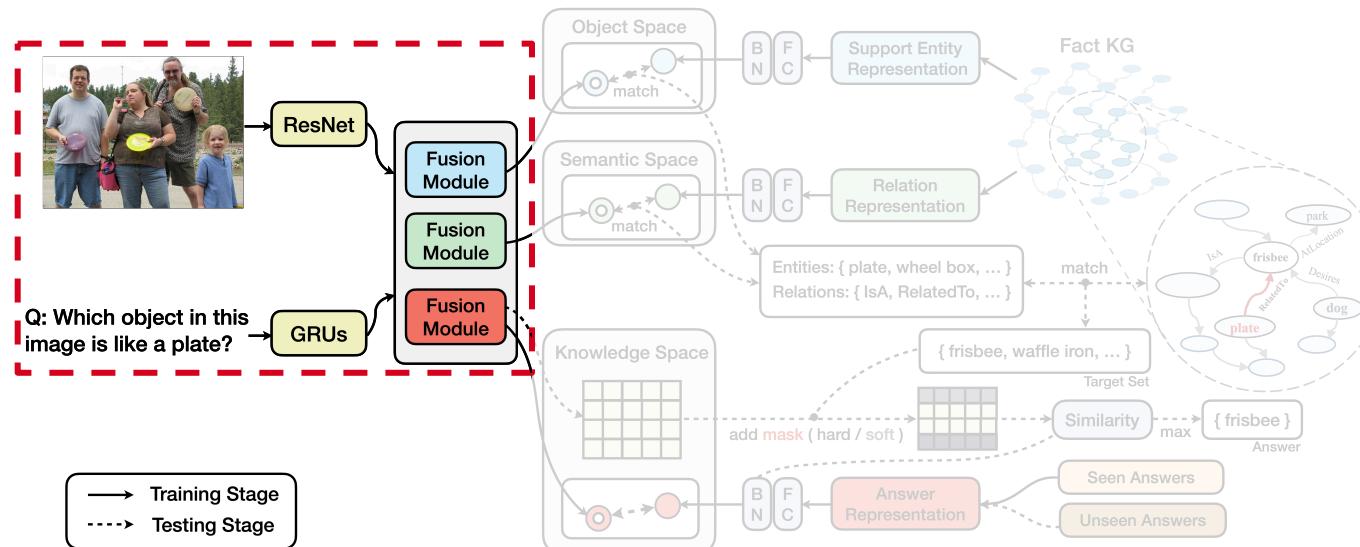


KG Augmented Zero-shot Visual Question Answering

IJCAI 2023 K-ZSL Tutorial (Part II)

T5: KG ZS-VQA

- **Multiple Feature Spaces :**
 - Convert VQA from a classification task to a mapping task

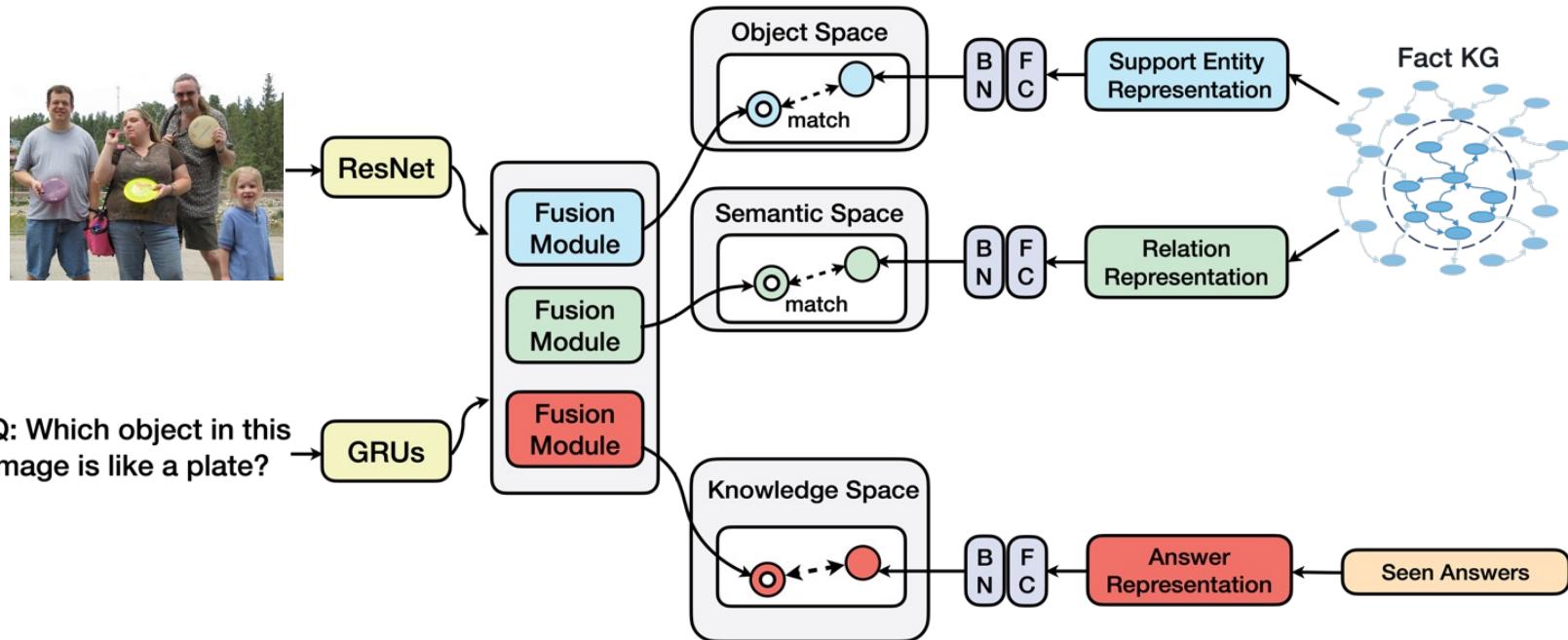


KG Augmented Zero-shot Visual Question Answering

IJCAI 2023 K-ZSL Tutorial (Part II)

T5: KG ZS-VQA

- **Multiple Feature Spaces :**
 - Convert VQA from a classification task to a mapping task

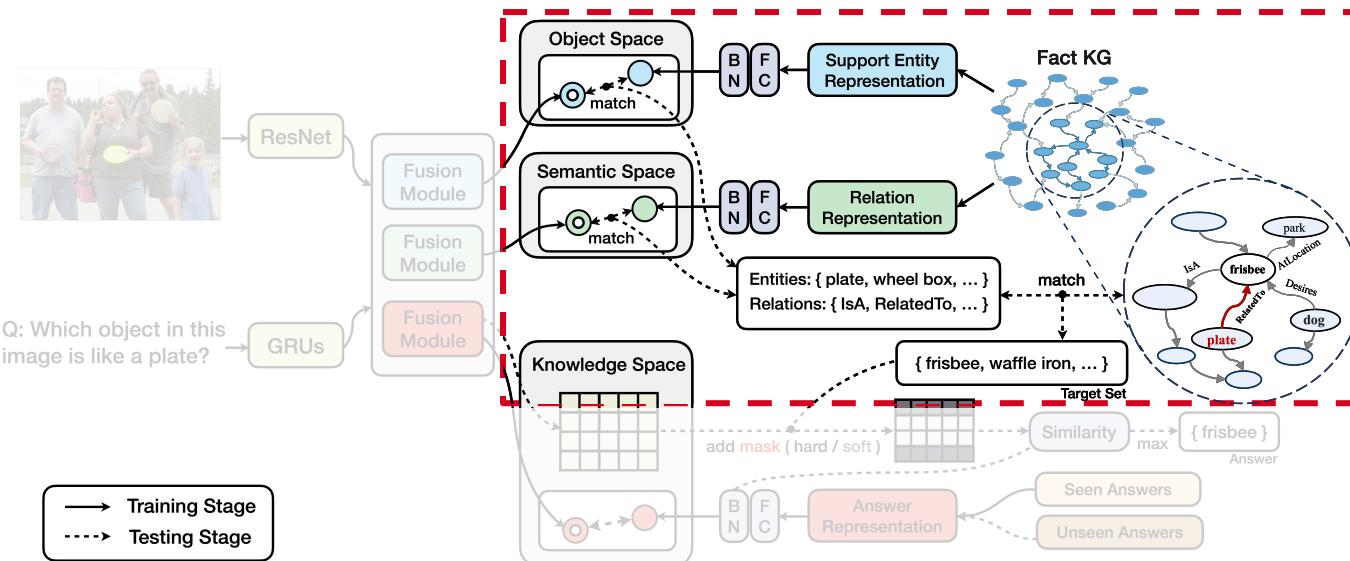


KG Augmented Zero-shot Visual Question Answering

IJCAI 2023 K-ZSL Tutorial (Part II)

T5: KG ZS-VQA

- **Answer Mask via Knowledge**
 - Enhance alignment process meanwhile alleviating error cascading

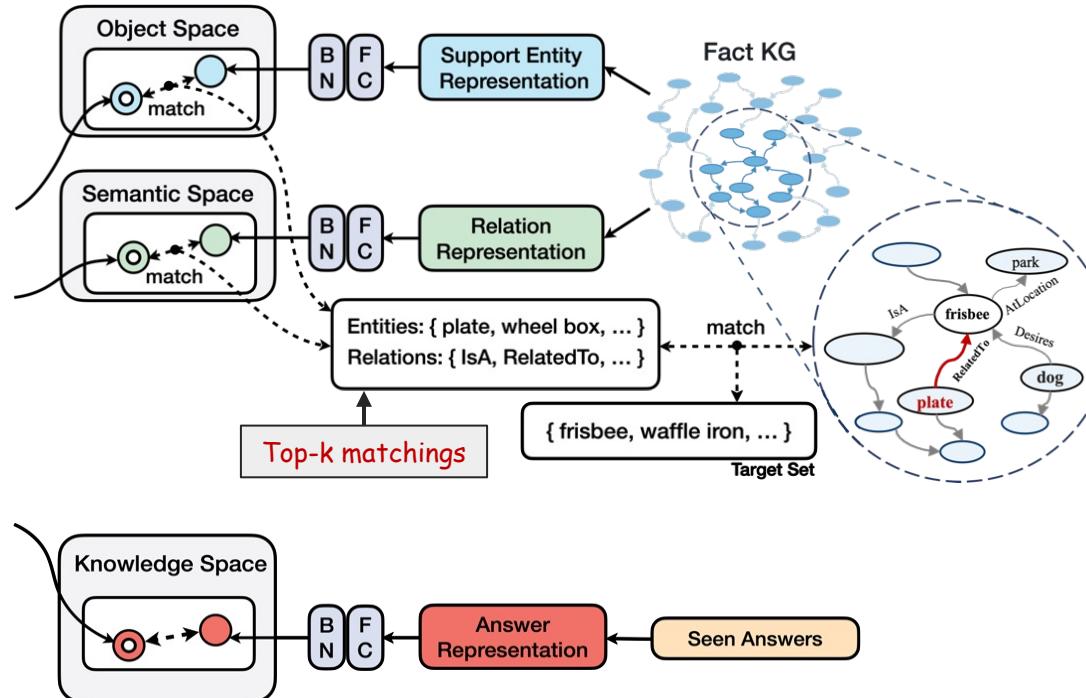


KG Augmented Zero-shot Visual Question Answering

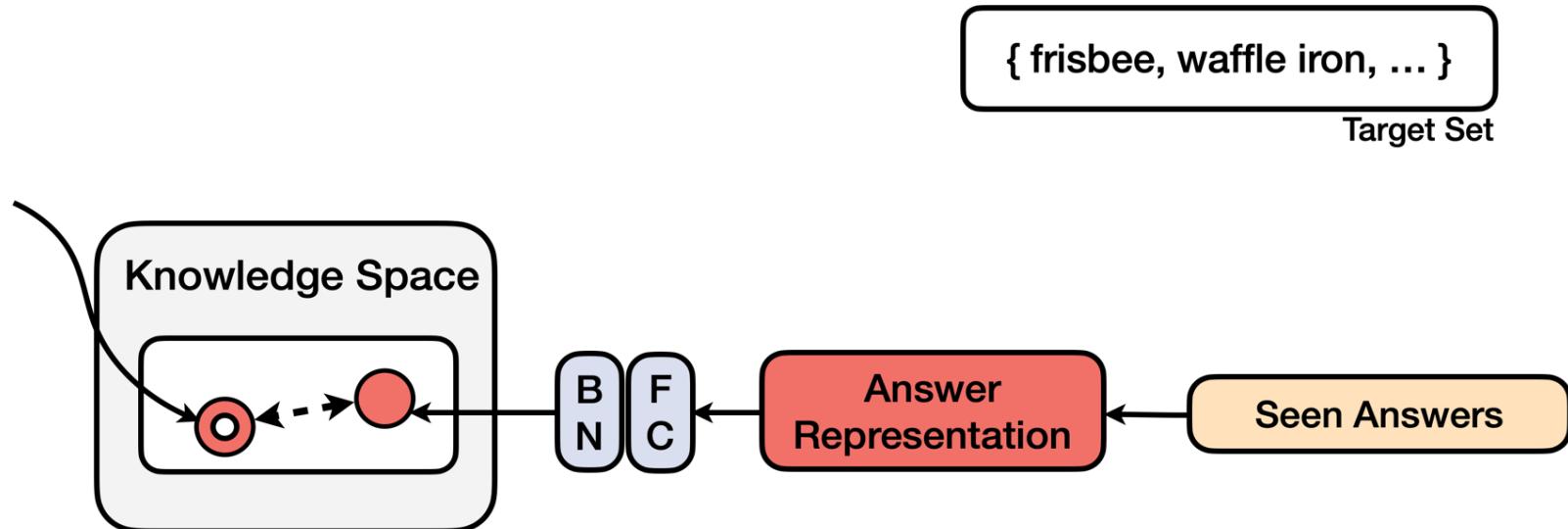
IJCAI 2023 K-ZSL Tutorial (Part II)

T5: KG ZS-VQA

- **Answer Mask via Knowledge**
 - Enhance alignment process meanwhile alleviating error cascading

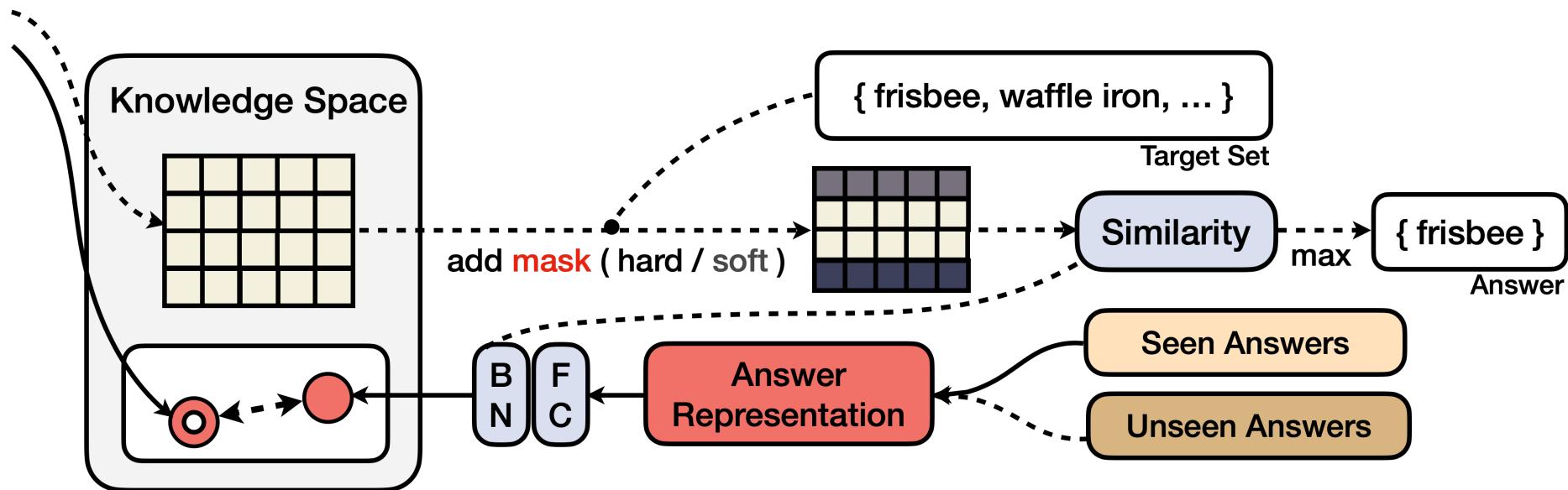


- **Answer Mask via Knowledge**
 - Filter a candidate target set



- **Answer Mask via Knowledge**

- soft mask score
- hard mask score



KG Augmented Zero-shot Visual Question Answering

IJCAI 2023 K-ZSL Tutorial (Part II)

T5: KG ZS-VQA

Question		What thing does the animal in this image have as a part?	Which object in this image could perform a screen	which object in this image is faster than bike?	Which object in this image is related to fly?
Image					
Our Model	Support Entity	zebra, elephant, zoo, giraffe	screen capture, screen, computer display, display image	fast than bicycle, bike, ride, fast than car	fly, catch fly, attempt to fly, learn to fly
	Relation	has a, part of, is a	capable of, used for, related to	slow, expensive, efficient	related to, belong to, specific
	Answer	stripe, horse, string ✓	computer, tv, keyboard ✓	car, bicycle, traffic light ✓	dragonfly, bird, airplane ✓
SAN [†]	Answer	string, water, ocean ✗	mouse, hand, lamp ✗	bicycle, traffic light, airplane ✗	airplane, kite, fly ✗
Ground Truth		stripe	computer	car	dragonfly

Question		Which object in this image is a cartilaginous fish?	Which object in this image is related to drive?	What is the place in this image used for?
Image				
Our Model	Support Entity	fish, eat fish, fish tank, carp, crab	drive, disk drive, drive on, drive lorry, drive only on track	ocean, sandy, lake, ocean beach, sand
	Relation	is a, belong to, related to	related to, specific, used for	used for, related to, capable of
	Answer	ray, jellyfish, lobster, turtle, sea, fish ✓	horse, cattle, car, cart, train, bicycle ✓	store boat, sail boat, swim, ski, swimming, life preserver ✓
SAN [†]	Answer	ray, turtle, fish, frog, jellyfish, lobster ✓	horse, cart, cow, sheep, cattle, camel ✓	life preserver, travel across water, sea, desert, ocean , store boat ✗
Ground Truth		grass	cattle	swim

More Reading

- Fvqa: Fact-based visual question answering. TPAMI 2018
- Out of the Box: Reasoning with Graph Convolution Nets for Factual Visual Question Answering. NeurIPS 2018
- Straight to the facts: Learning knowledge base retrieval for factual visual question answering. ECCV 2018
- OK-VQA: A visual question answering benchmark requiring external knowledge. CVPR 2019
- KVQA: Knowledge-aware visual question answering. AAAI 2019
- Mucko: Multi-Layer Cross-Modal Knowledge Reasoning for Fact-based Visual Question Answering. IJCAI 2020
- Conceptbert: Concept-aware representation for visual question answering. EMNLP 2020
- Krisp: Integrating implicit and symbolic knowledge for open-domain knowledge-based vqa. CVPR 2021
- Retrieval augmented visual question answering with outside knowledge. EMNLP 2022
- LaKo: Knowledge-driven Visual Question Answering via Late Knowledge-to-Text Injection. IJCKG 2022
- K-lite: Learning transferable visual models with external knowledge. NeurIPS 2022
- Multi-modal answer validation for knowledge-based vqa. AAAI 2022

T6

Cross-modal Semantic Grounding for Contrastive ZSL

Cross-modal Semantic Grounding for ZSL

IJCAI 2023 K-ZSL Tutorial (Part II)
T6: Cross-modal Semantic Grounding

Key for ZSL:

Transfer semantic knowledge from seen to unseen classes based on their semantic information

Limitation for ZSL:

Lack of **region-attribute data** makes it difficult for ZSL models to understand the relationship of different attribute-feature pairs during training



Chuck-will's-widow

primary_color: brown, buff | wing_pattern: spotted |
head_pattern: plain | belly_pattern: striped | ...
bill_shape: hooked | wing_shape: broad

V.S.



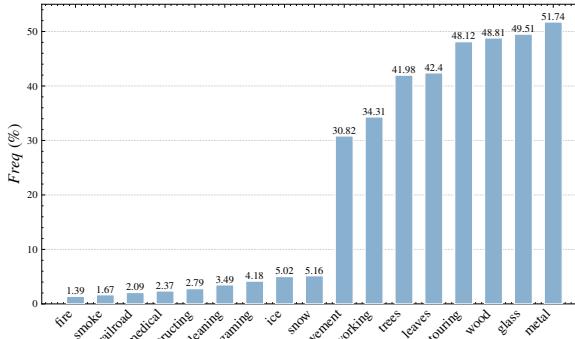
Black-and-white Warbler

primary_color: black, grey | wing_pattern: striped |
head_pattern: eyebrow, striped | eye_color: black | ...
belly_pattern: solid | back_pattern: multi-colored |

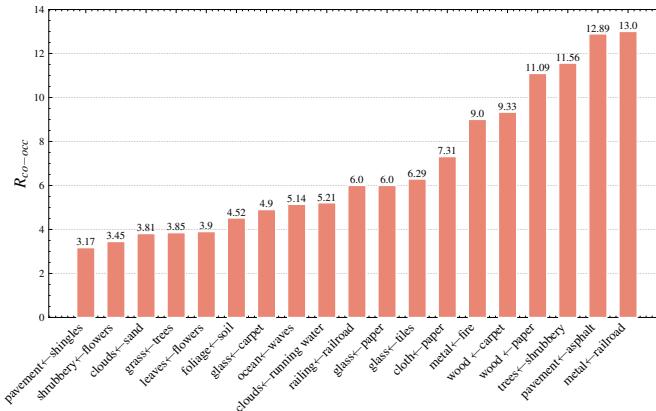
Cross-modal Semantic Grounding for ZSL



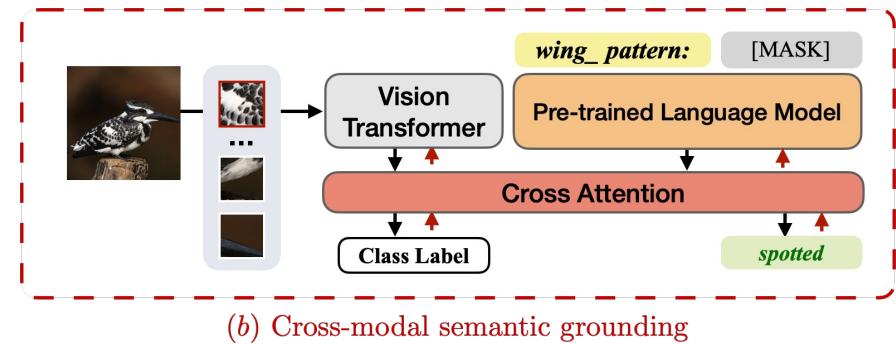
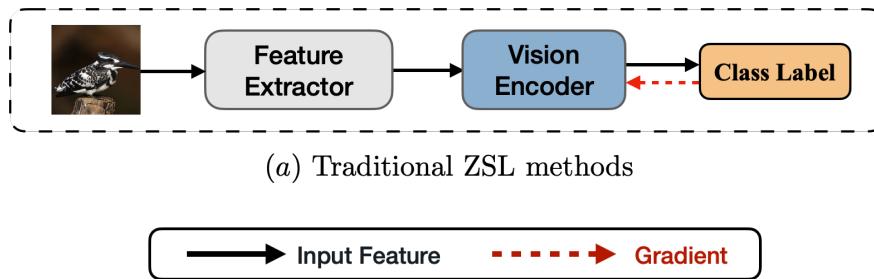
(a) Imbalance distribution of attributes



(b) Attribute co-occurrence

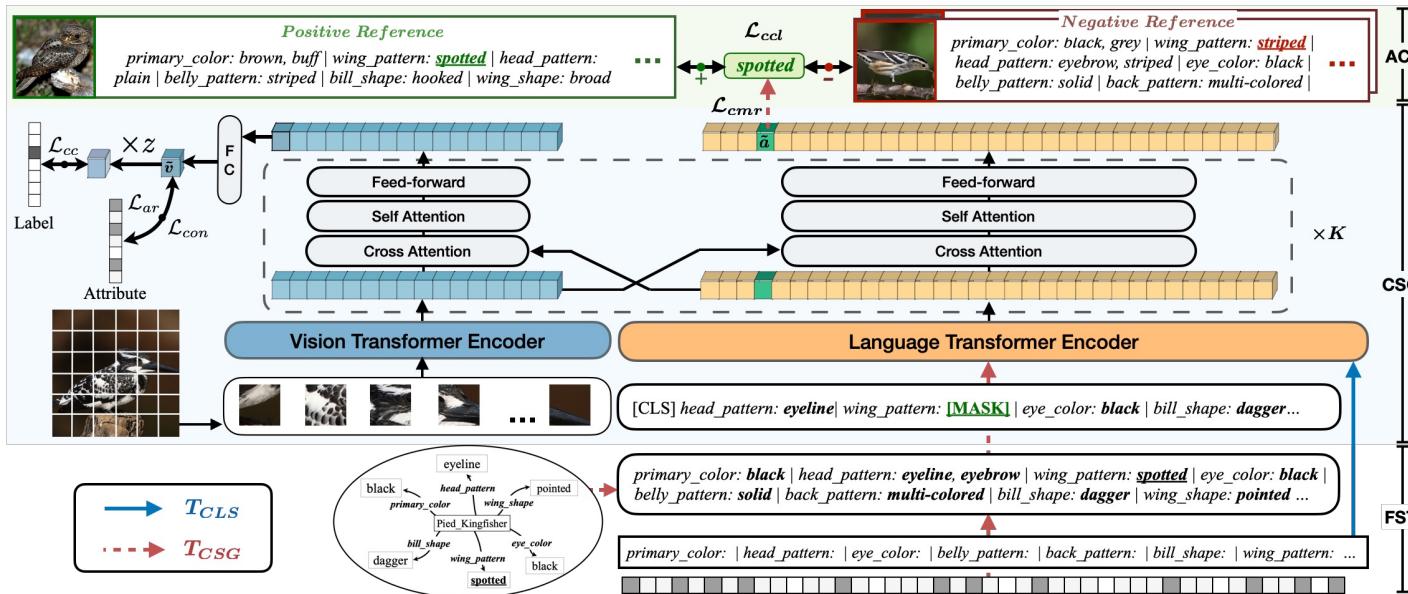


Cross-modal Semantic Grounding for ZSL



- (a) The paradigm of previous ZSL methods.
- (b) The paradigm of our method DUET which exploits the semantics of PLMs to augment the transformer-based vision encoder via reconstructing masked attributes with a cross-model attention mechanism.

Cross-modal Semantic Grounding for ZSL



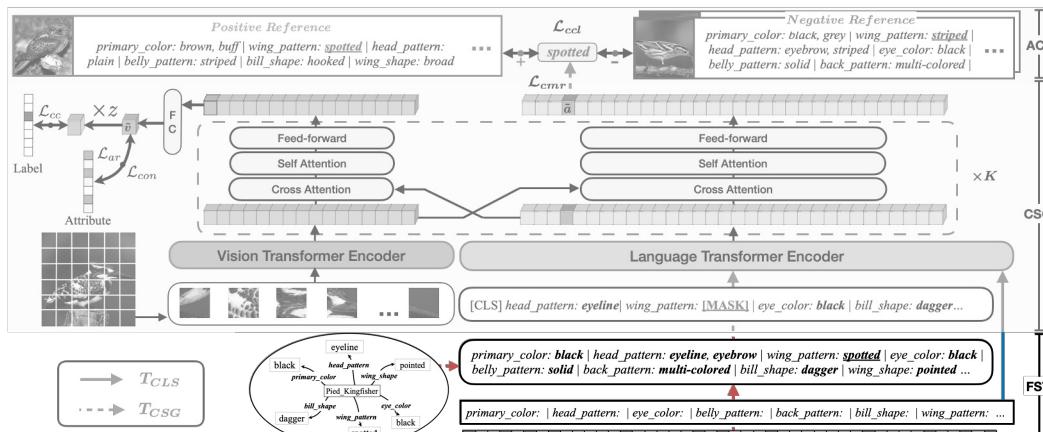
- Attribute-level Contrastive Learning (ACL)

- Cross-modal Semantic Grounding (CSG)

- Feature-to-Sequence Transformation (FST)

Cross-modal Semantic Grounding for ZSL

1. Transform different types of attributes into a **textual sequence**
2. Make our model **compatible** to multiple ZSL tasks with different side information



$$\mathcal{D}_s = \{(x^s, y^s) | x^s \in \mathcal{X}^s, y^s \in \mathcal{Y}^s\}$$

$$\mathcal{D}_u = \{(x^u, y^u) | x^u \in \mathcal{X}^u, y^u \in \mathcal{Y}^u\}$$

unseen classes (\mathcal{C}^u)
seen classes (\mathcal{C}^s)
 $c \in \mathcal{C} = \mathcal{C}^s \cup \mathcal{C}^u$

$$\mathcal{A} = \{a_1, \dots, a_{|\mathcal{A}|}\}$$

$$z^c = [z_1^c, \dots, z_{|\mathcal{A}|}^c]^\top$$

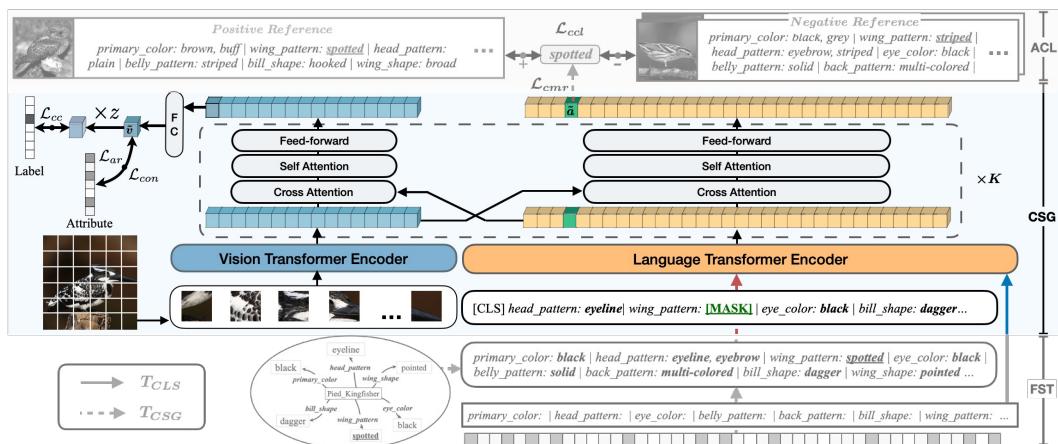
..|color: brown|haspart: tail,flippers,...|...
Prompt Attribute Prompt Attributes

Cross-modal Semantic Grounding for ZSL

- Attribute Phrase Masking (APM)
 - Linear weighted random sampling (LWRS)
 - Random attributes pruning (RAP)

$$a_t = LWRS(\mathcal{A})$$

$$\mathcal{A}_{rap} = RAP(r_{rap}, \mathcal{A})$$



- Cross-modal Mask Reconstruction (CMR)

$$\mathcal{L}_{cmr} = \mathbb{E}_{x \sim \mathcal{X}^s} [-\mathbf{z}_a \sum_{i=1}^{Len(w)} \log P(w_i | \hat{\mathcal{A}}_{rap \setminus t}, x)]$$

- Basic ZSL Classification Objective

- Attribute regression (\mathcal{L}_{cc})
- Class cross-entropy (\mathcal{L}_{ar})
- class-level supervised contrastive (\mathcal{L}_{con})

$$\begin{aligned} \mathcal{L}_{ar} &= \mathbb{E}_{x \sim \mathcal{X}^s} \|\tilde{v} - z\|_2^2 \\ \mathcal{L}_{cc} &= \mathbb{E}_{x \sim \mathcal{X}^s} [-\log \frac{\exp(\tilde{v} \cdot z)}{\sum_{\hat{c} \in \mathcal{C}^s} \exp(\tilde{v} \cdot z^{\hat{c}})}] \\ \mathcal{L}_{con} &= \mathbb{E}_{x \sim \mathcal{X}^s} [-\log f_{\theta}(\tilde{v} | s, x)] \end{aligned}$$

$$f_{\theta} = \frac{\exp(D(\tilde{v}, \tilde{v}^+)/\tau)}{\exp(D(\tilde{v}, \tilde{v}^+)/\tau) + \sum_{\tilde{v}' \in \mathcal{N}(\tilde{v})} \exp(D(\tilde{v}, \tilde{v}')/\tau)}$$

Cross-modal Semantic Grounding for ZSL

- Two Stage

- T_{CSG} : cross-modal semantic grounding
- T_{CLS} : simple image classification without textual attributes

..|color:|haspart:|pattern:|shape:|...

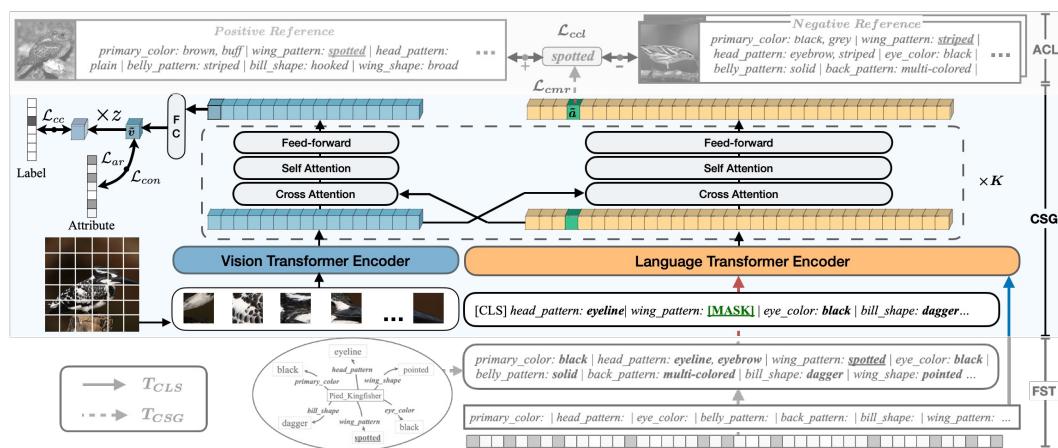
Prompt Prompt Prompt Prompt

- Multi-task Learning

$$L_{CLS} = \mathcal{L}_{zsl} + \lambda_{con}\mathcal{L}_{con}$$

$$L_{CSG} = \mathcal{L}_{zsl} + \lambda_{cmr}\mathcal{L}_{cmr}$$

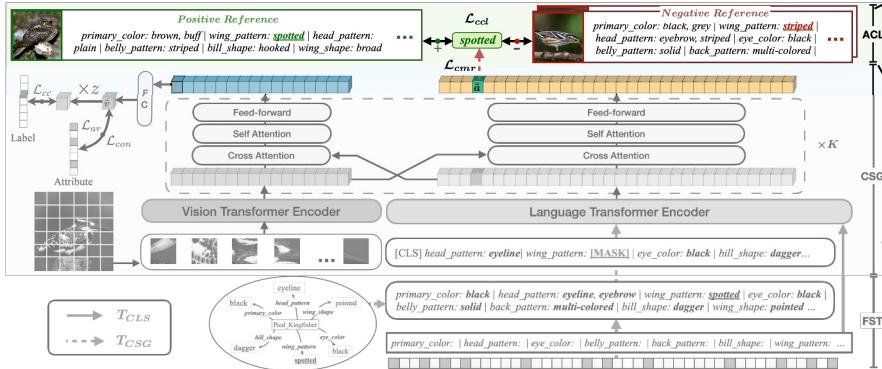
$$\mathcal{L}_{zsl} = \mathcal{L}_{cc} + \lambda_{ar}\mathcal{L}_{ar}$$



Cross-modal Semantic Grounding for ZSL

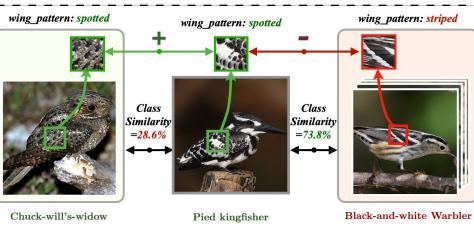
$$\mathcal{L}_{acl} = \mathbb{E}_{x \sim \mathcal{X}^s} [-\text{Min}(z_a, z_{a+}) \log f_\phi(\tilde{a} | s, x)]$$

$$f_\phi = \frac{\exp(D(\tilde{a}, \tilde{a}^+)/\tau)}{\exp(D(\tilde{a}, \tilde{a}^+)/\tau) + \sum_{\tilde{a}' \in \mathcal{N}(\tilde{a})} \exp(D(\tilde{a}, \tilde{a}')/\tau)}$$



- Attribute-based (negative) sampling strategy:

$$L_{CSG} \leftarrow L_{CSG} + \mathcal{L}_{acl}$$



Select

- those distinctive classes as positive references when they are associated with at least one common attribute (e.g., "spotted")
- those similar classes as negative references when they have mutually exclusive attributes (e.g., "striped") toward the same aspect (e.g., "wing pattern")

Cross-modal Semantic Grounding for ZSL

..| color: [MASK] | haspart: [MASK] | ..
 Prompt Prompt

Image Case	Prompt	Attribute
oilrig (S)	transportation function	DUET: boating (46.60%) swimming (34.39%) digging (19.01%) GT: boating (51.99%) swimming (12.71%)
	coarse material	DUET: metal (39.91%) ocean (37.55%) wire (22.54%) GT: ocean (48.52%) metal (43.90%) wire (6.93%)
	specific material	DUET: fire (40.75%) waves (37.78%) smoke (21.47%) GT: still water (24.26%) waves (16.17%) fire (15.08%)
living room (S)	environment function	DUET: reading (65.42%) eating (31.13%) working (13.45%) GT: reading (46.60%) eating (12.33%) socializing (6.85%)
	specific material	DUET: cloth (39.55%) flowers (38.47%) tiles (21.98%) GT: carpet (28.78%) cloth (23.30%) flowers (10.96%)
	feeling	DUET: soothng (59.97%) symmetrical (32.92%) cluttered (7.11%) GT: soothing (49.33%)
	environment function	DUET: sports (49.15%) playing (32.78%) socializing (18.07%) GT: congregating (47.31%) sports (43.48%) playing (11.51%)
basketball arena (U)	technical function	DUET: competing (40.00%) audience (36.19%) exercise (23.81%) GT: competing (52.43%) audience (52.43%) exercise (17.90%)
	surface	DUET: glossy (41.56%) rusty (32.74%) sterile (25.70%) GT: glossy (17.90%) dry (15.34%)
	transportation function	DUET: driving (42.95%) biking (37.98%) climbing (19.07%) GT: driving (41.53%) biking (17.96%)
bus depot (U)	coarse material	DUET: asphalt (34.28%) metal (33.71%) concrete (32.01%) GT: asphalt (38.94%) metal (22.06%) pavement (20.77%)
	light	DUET: natural (57.30%) indoor (33.55%) direct sun/sunny (9.15%) GT: natural (54.51%) direct sun/sunny (20.36%)

Figure 6: Attribute prediction for interpretation.

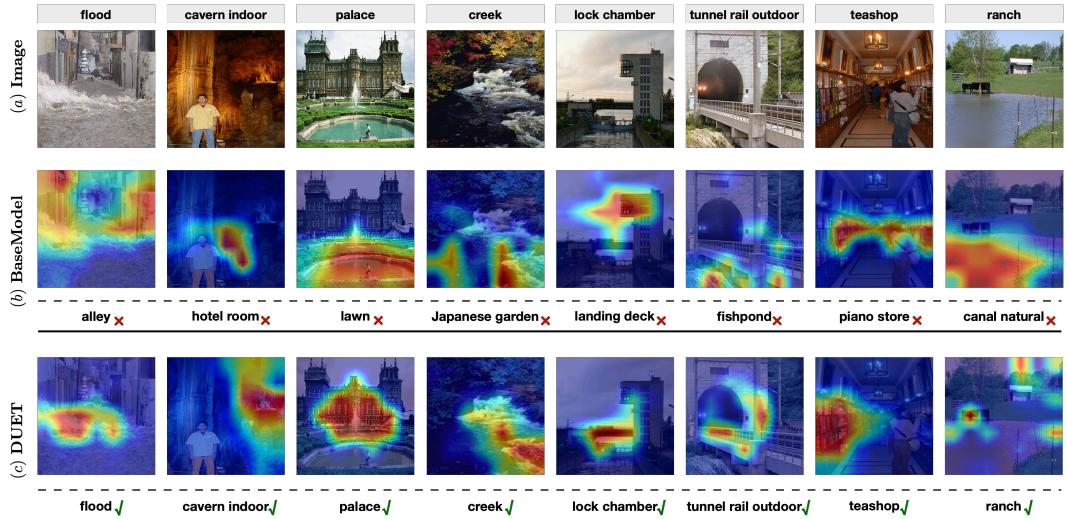


Figure 8: Visualization of attention maps together with attribute grounding results. (a) Original images. (b) BaseModel: Only using ENC_{att} without cross-modal semantic grounding (CSG) and attribute-level contrastive learning (ACL). (c) DUEL.

More Reading

- Adaptive and Generative Zero-Shot Learning. ICLR 2021
- TransZero: Attribute-guided Transformer for Zero-Shot Learning. AAAI 2022
- MSDN: Mutually Semantic Distillation Network for Zero-Shot Learning. CVPR 2022
- Progressive Semantic-Visual Mutual Adaption for Generalized Zero-Shot Learning. CVPR 2023
- Decomposed soft prompt guided fusion enhancing for compositional zero-shot learning . CVPR 2023

T7

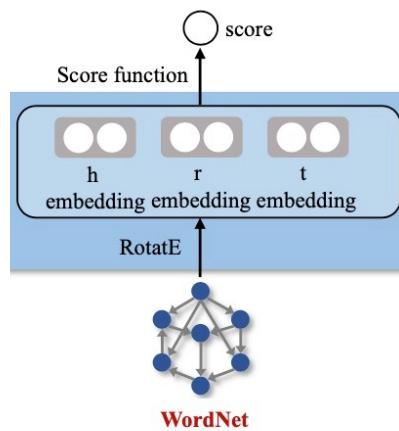
KG Structure Pretraining for KG-aware ZSL

KGTransformer

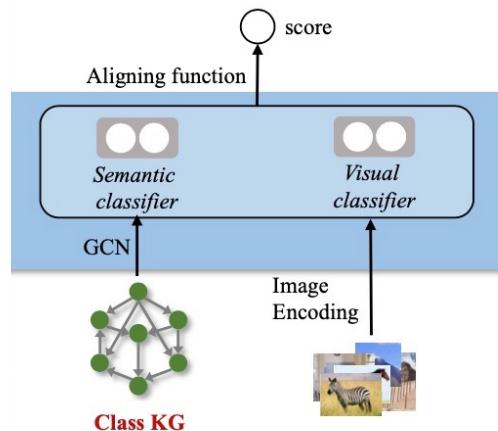
- Knowledge graphs are useful resource for many tasks
 - Knowledge management
 - Natural Language Understanding
 - Zero-shot Learning (of course ;))
 - ...
- Question:
 - How is knowledge graph used into those task models?

KGTransformer

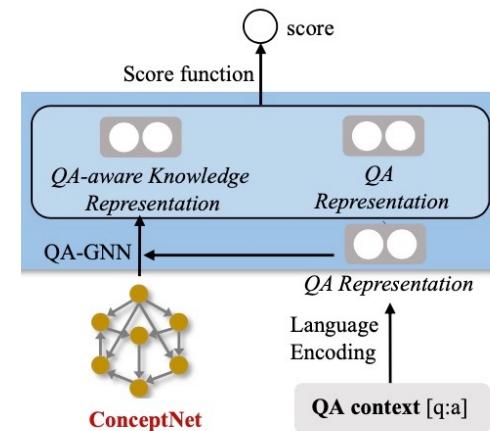
- Examples:



Triple Classification(RotateE)



Zero-shot Image Classification (GCNz)

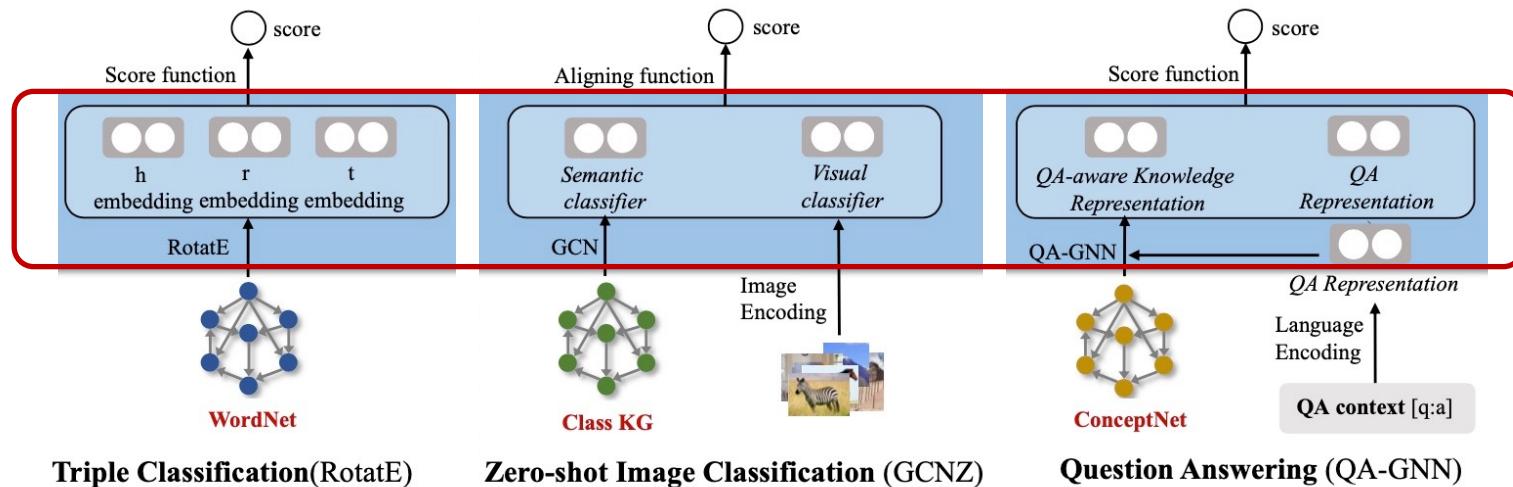


Question Answering (QA-GNN)

KGTransformer

- Examples:

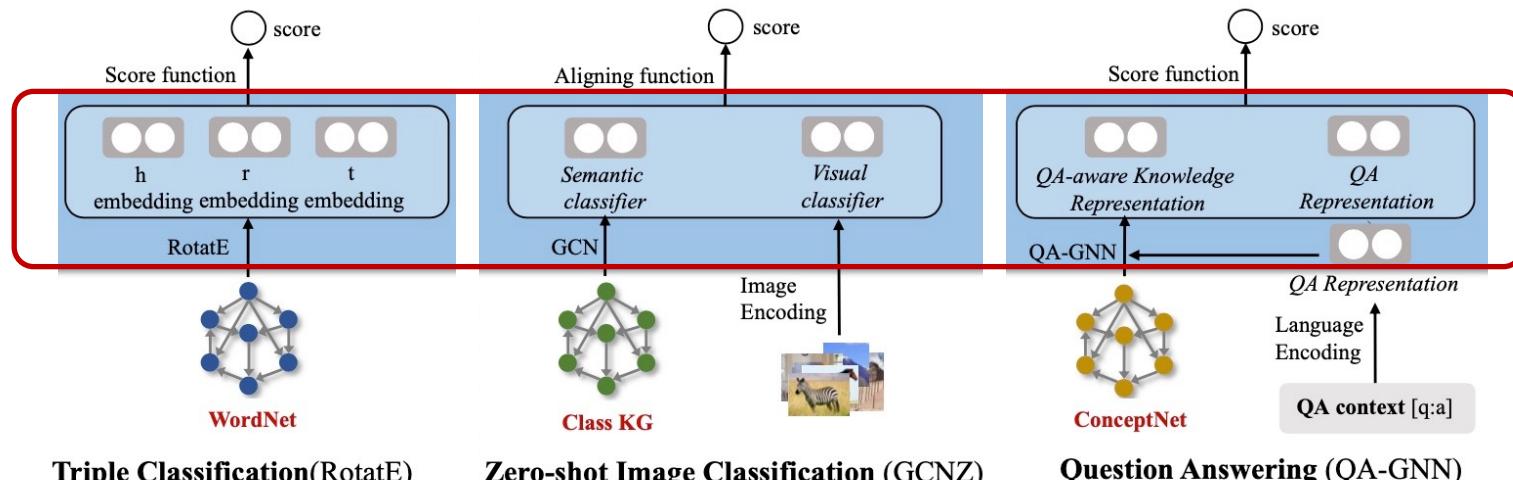
Knowledge Representation
 Representation knowledge graph in vector space
& Knowledge Fusion
 Fuse knowledge graph representation into task model



KGTransformer

- Examples

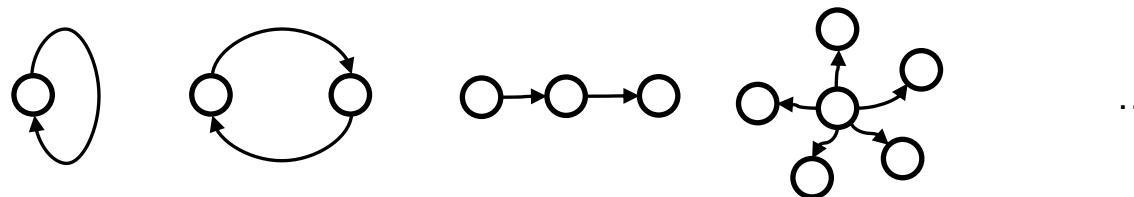
Knowledge Representation
 Representation knowledge graph in vector space
& Knowledge Fusion
 Fuse knowledge graph representation into task model



- They all have knowledge representation and fusion module
- The knowledge graphs used in the task are different

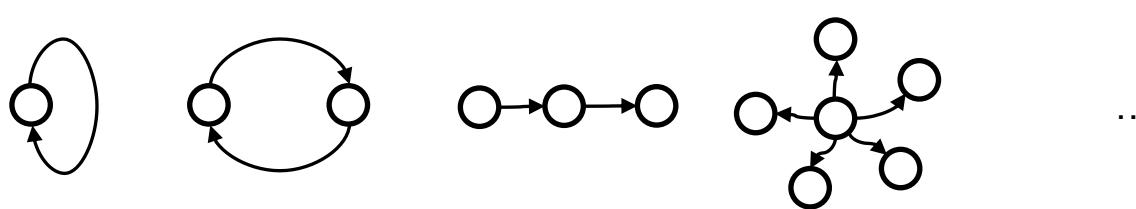
KGTransformer

- Knowledge graphs
 - YAGO, Wikidata, ConceptNet, ...
 - They have common structures
 - Circles
 - Paths
 - Star shapes
 - ...



KGTransformer

- Knowledge graphs
 - YAGO, Wikidata, ConceptNet, ...
 - They have common structures
 - Circles
 - Paths
 - Star shapes
 - ...



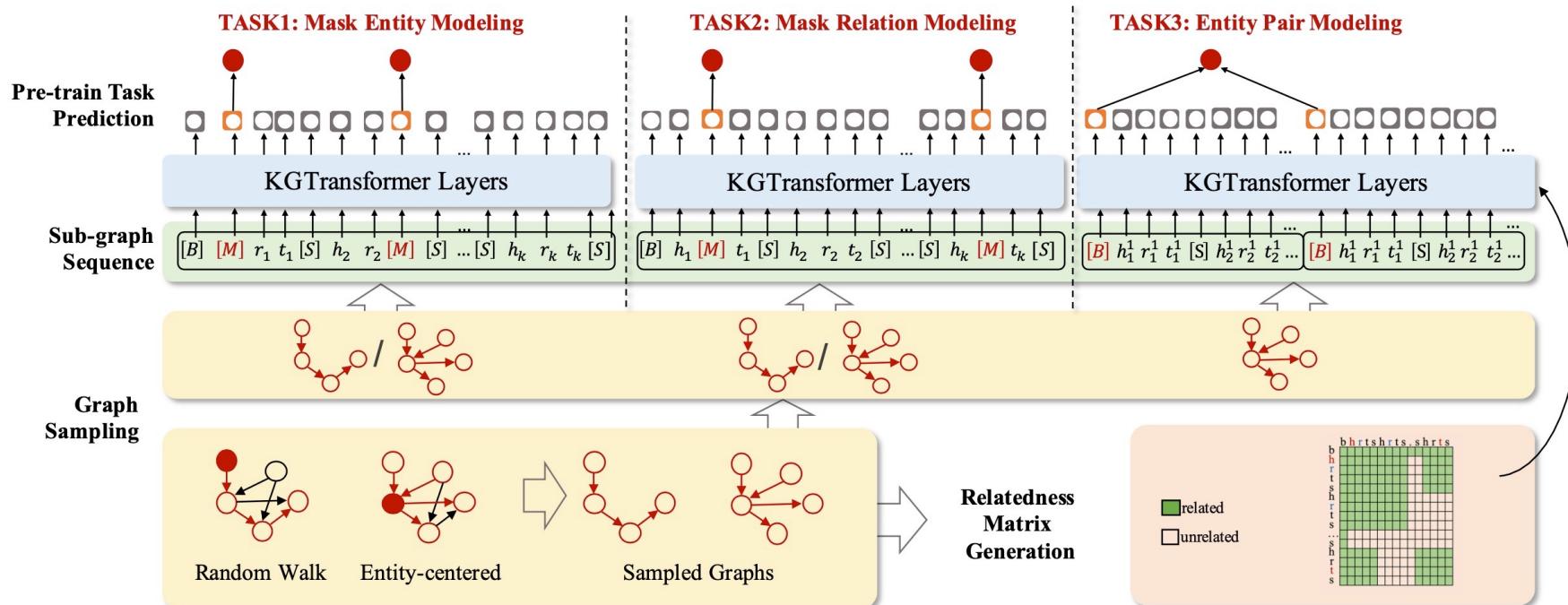
KGTransformer

- Pre-trained knowledge graph model (PKGM)
 - Learn universal embeddings of entities and relations that could be applied in many tasks
 - These embeddings are supposed to contain
 - Entity similarity
 - Hierarchies
 - Relationships
 - ...
 - Challenges
 1. What if the task KG and the pre-trained KG have different entities and relations?
 2. Essential interaction and fusion between KG and the task data is missing.
 - designing a fusion module as part of the work for downstream task model devising

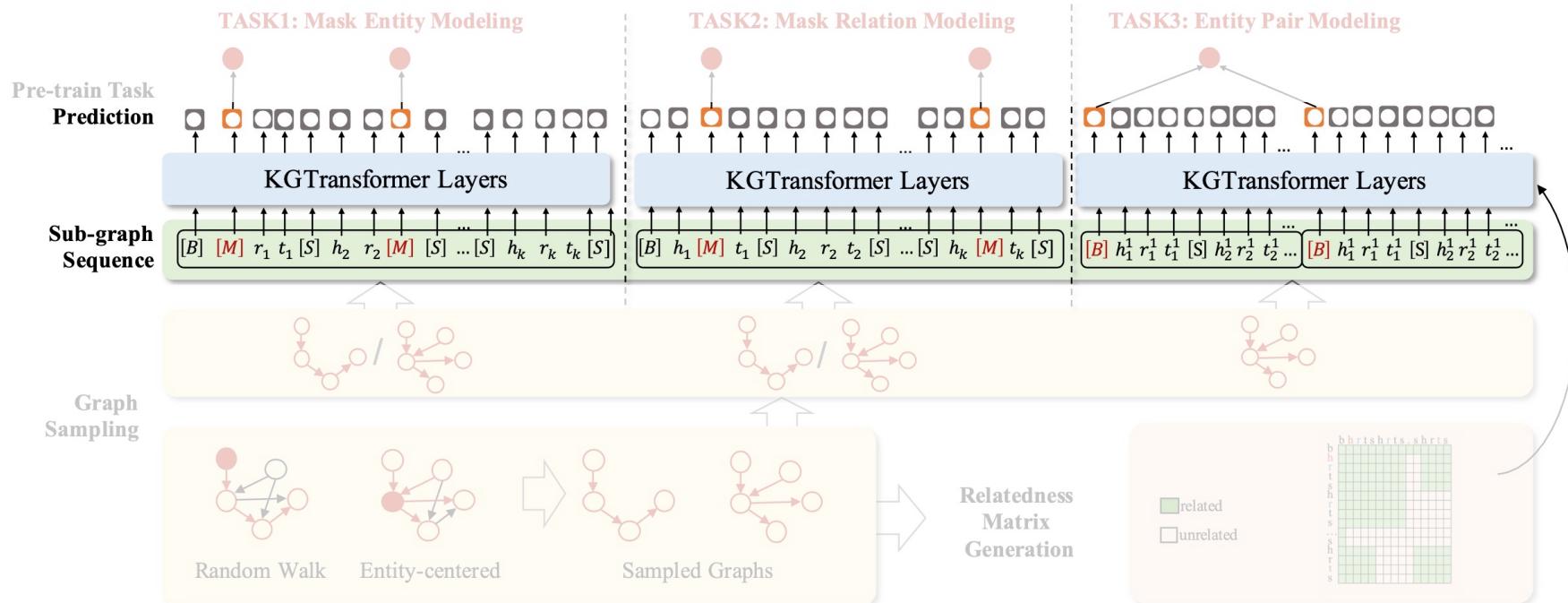
KGTransformer

- Pre-trained knowledge graph model (PKGM)
 - Learn universal embeddings of entities and relations that could be applied in many tasks
 - These embeddings are supposed to contain
 - Entity similarity
 - Hierarchies
 - Relationships
 - ...
 - Challenges
 1. What if the task KG and the pre-trained KG have different entities and relations?
 - Be unrelated to specific entities and relations
 2. Essential interaction and fusion between KG and the task data is missing.
 - Prompt tuning to enable uniform and flexible fusion

KGTransformer



KGTransformer



KGTransformer

- KGTransformer layer
 - Subgraph sequence as input

$$s_{in} = [[B], h_1, r_1, t_1, [S], h_2, r_2, t_2, [S], \dots, h_k, r_k, t_k, [S]]$$

- Modified transformer layer with relatedness matrix

$$M_{ij} = \begin{cases} 1 & \text{if } i = 1 \text{ or } j = 1 \\ 1 & \text{if } trp(i) \cap trp(j) \neq \emptyset, \\ 0 & \text{otherwise} \end{cases}$$

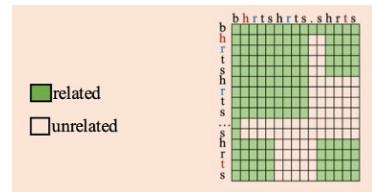
$$trp(n) = \{h_p, r_p, t_p\} \text{ where } p = \lfloor (n - 2)/4 \rfloor + 1$$

- Attention

$$Q = HW_Q, \quad K = HW_K, \quad V = HW_V,$$

$$A = \frac{QK^\top \odot M}{\sqrt{d_K}} + (1 - M) * \delta,$$

$$Attn(H) = \text{softmax}(A)V,$$

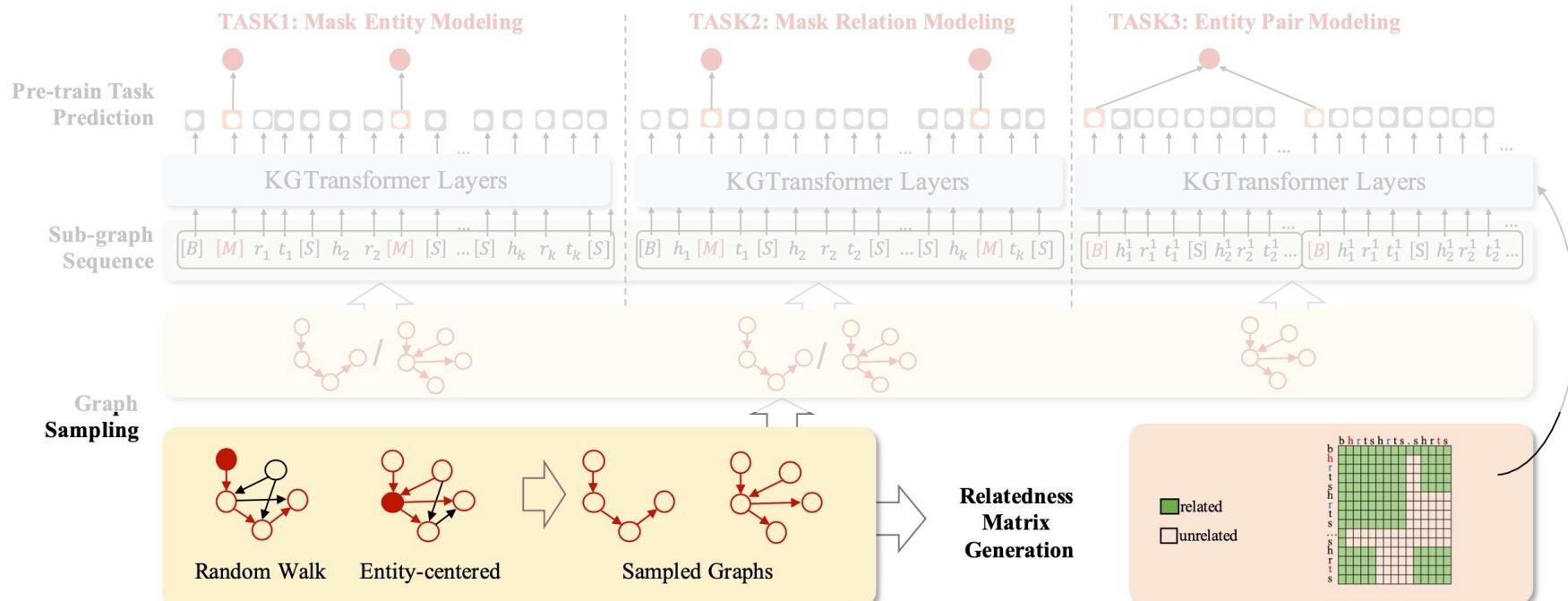


Compare to GNN & Conventional Transformer:

- (William Shakespeare, field of work, Fiction)
- (France, capital, Paris)
- (William Shakespeare, notable book, Hamlet)

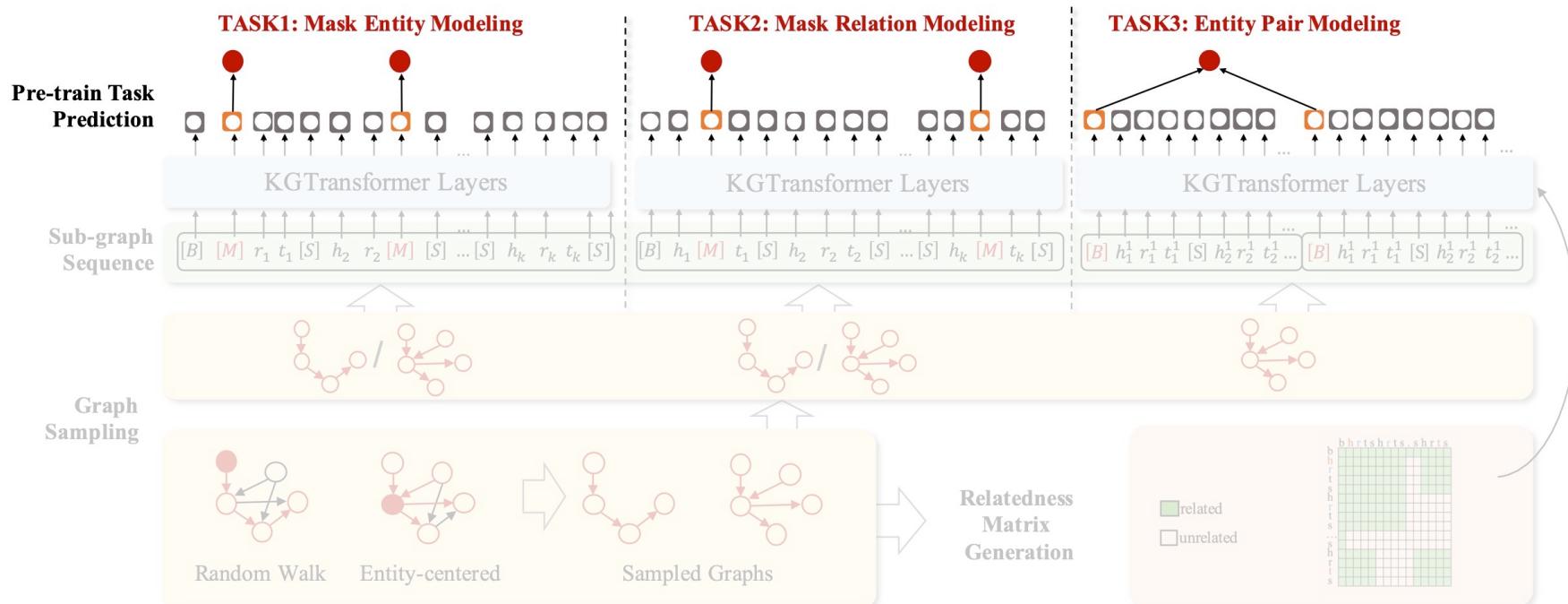
KGTransformer

- Graph Sampling



KGTransformer

- Pre-training Task



KGTransformer

- Masked Entity Modeling (MEM)

$$\begin{aligned} L_{MEM}(\mathcal{G}_e) = & \sum_{e \in \mathcal{M}_e} CE(s_e^m W_{MEM} \mathbf{E}(e)^\top, 1) \\ & + \sum_{e' \in \Delta} CE(s_e^m W_{MEM} \mathbf{E}(e')^\top, 0), \end{aligned}$$

- Masked Relation Modeling (MRM)

$$L_{MRM}(\mathcal{G}_e) = \sum_{r \in \mathcal{M}_r} CE(MLP(s_r^m W_{MRM}), l_r), \quad W_{MRM} \in \mathbb{R}^{d \times d}$$

- Entity Pair Modeling(EPM)

$$L_{EPM}(\mathcal{G}_{e_i}, \mathcal{G}_{e_j}) = CE(MLP([s_{[B]_{e_i}}^m || s_{[B]_{e_j}}^m]), l_{(e_i, e_j)})$$

- Training objective

$$L(\mathcal{G}) = \sum_{e \in \mathcal{E}} (L_{MEM}(\mathcal{G}_e^1) + L_{MRM}(\mathcal{G}_e^2) + L_{EPM}(\mathcal{G}_e^3, \mathcal{G}_{e'}))$$

KGTransformer

- Prompt Tuning

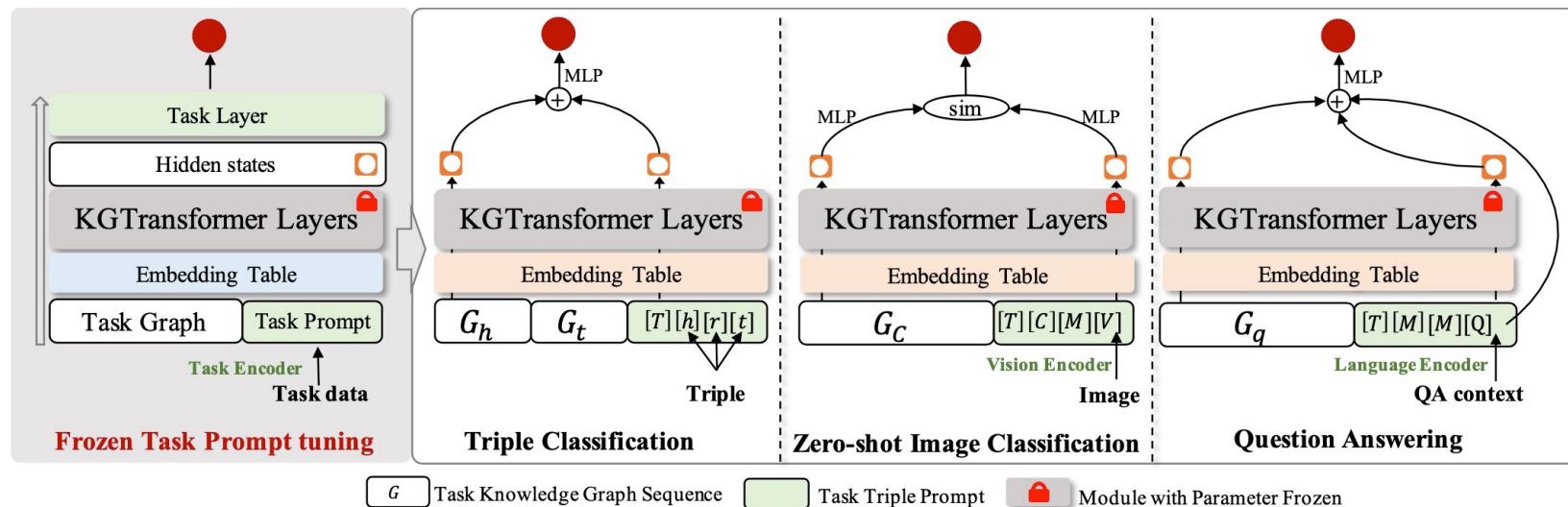


Figure 3: Overview of task prompt tuning (left) and examples of three specific tasks (right).

KGTransformer

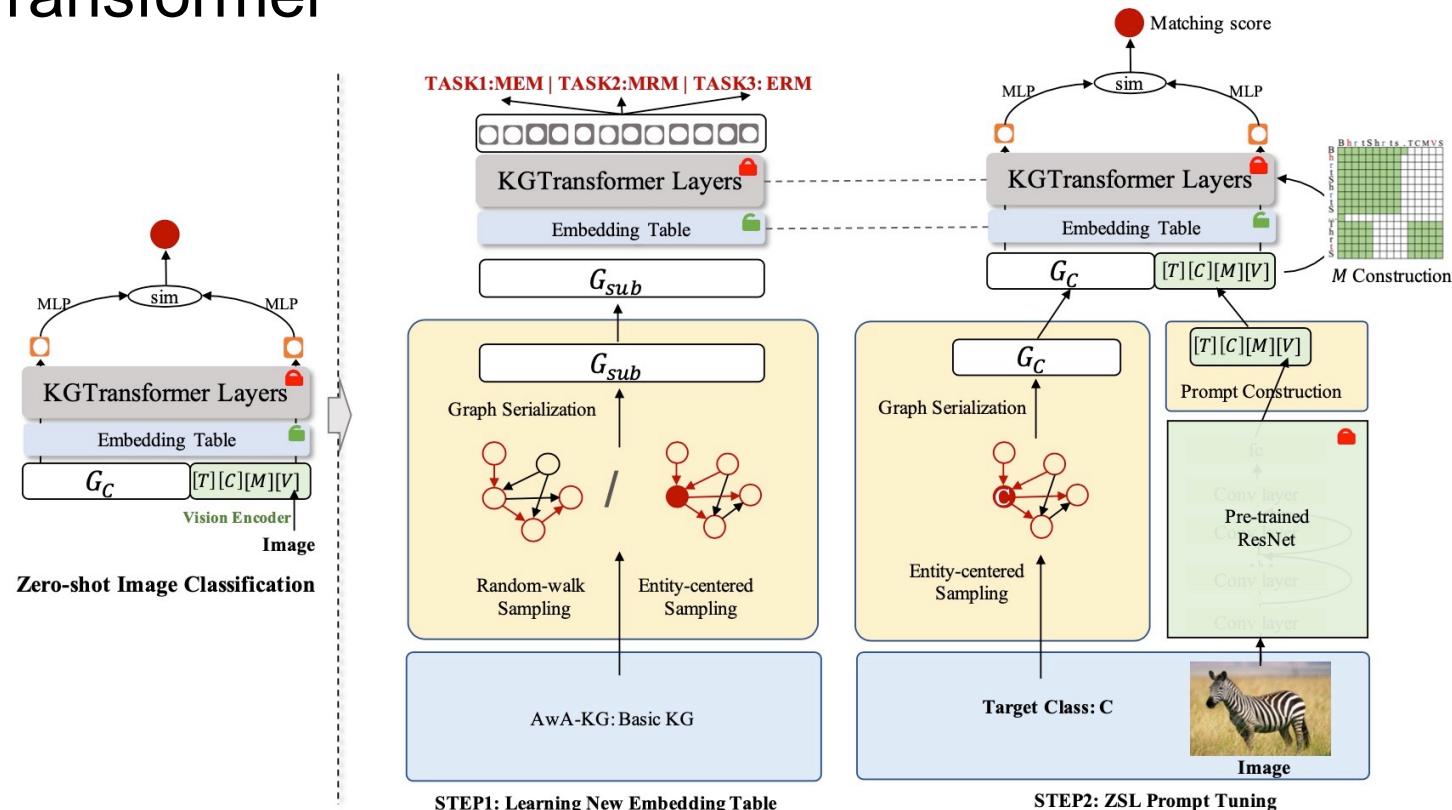
- Experiment
 - Pre-training dataset

	#R	#E	# T	Std _{rel}	Std _{ent}	Density($\times 10^{-6}$)
WFC	317	133435	1015556	13230	134	0.18
<i>wn18rr</i>	11	40943	93003	12540	9	5.04
<i>fb15k - 237</i>	237	14541	310115	2467	128	6.19
<i>codex</i>	69	77951	612437	24664	159	1.46

- Task datasets

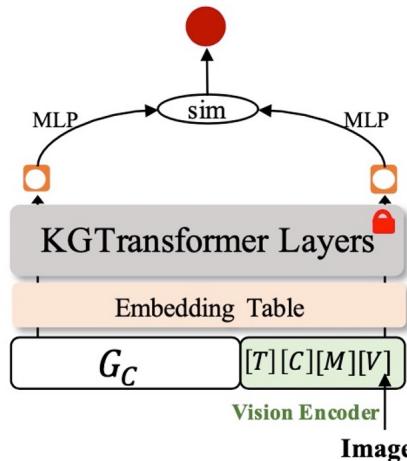
	Task KG				Task Data				Properties		
	KG	#E	#R	#T	Task Sample	# Train	#Valid	#Test	Task Type	Modality	Overlap to WFC
T1: Triple Classification	WN18RR	40943	11	-	Triple	86835	6068	6268	in-KG	KG	Yes
T2: ZSL Image Classification	AwA-KG	146	16	1595	Image	23527	-	13795	out-of-KG	Image+KG	No
T3: Question Answering	CommonsenseQA	64388	16	309444	QA pair	8500	1221	1241	out-of-KG	Language+KG	No

KGTransformer



KGTransformer

- Zero-shot image classification results on AwA-KG



Zero-shot Image Classification

	T1	S	U	H
DeViSE[10]	43.24	86.44	6.40	11.91
GCNZ[60]	62.98	75.59	20.28	31.98
OntoZSL[14]	62.65	59.59	50.58	54.71
<i>KGTransformer</i>	66.26	61.13	55.14	57.98
<i>KGT-finetune</i>	63.59	63.61	50.08	56.04
<i>KGT-scratch</i>	58.96	56.48	47.93	51.85

KGTransformer

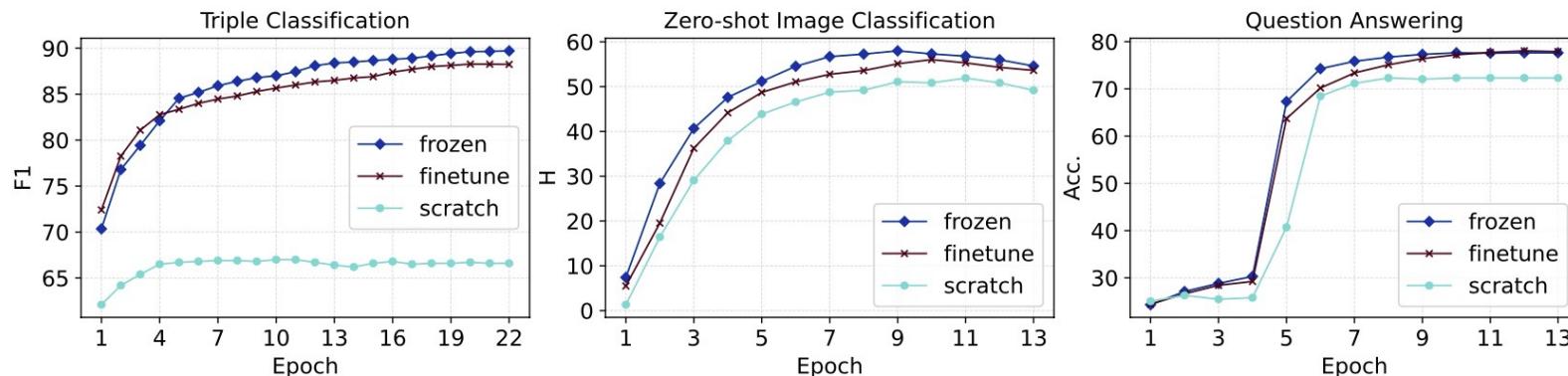
- Triple classification on WN18RR
- QA Accuracy on CommonsenseQA

	Acc.	Precision	Recall	F1
TransE [3]	88.35	93.45	82.48	87.62
RotatE [46]	88.26	93.03	82.71	87.57
ComplEx [50]	85.07	96.73	72.59	82.94
<i>KGTransformer</i>	89.21	85.56	94.32	89.73
<i>KGT-finetune</i>	87.48	83.02	94.22	88.27
<i>KGT-scratch</i>	67.02	67.91	64.55	66.19

	IHdev	IHtest
RoBERTa-Large [34]	73.07	68.69
RoBERTa-Large [34](ours)	72.24	68.49
+ GconAttn [59]	72.61	68.59
+ KagNet [28]	73.47	69.01
+ MHGRN [9]	74.45	71.11
+ QA-GNN [67]	76.54	73.41
+ <i>KGTransformer</i>	77.64	74.13
+ <i>KGT-finetune</i>	78.05	74.21
+ <i>KGT-scratch</i>	72.32	69.22

KGTransformer

	Triple Classification				Zero-shot Image Classification				Question Answering	
	Acc.	Precision	Recall	F1	T1	S	U	H	IHdev	IHtest
KGTransformer	89.20	85.56	94.32	89.73	66.26	61.13	55.14	57.98	77.64	74.13
- MEM	86.85	82.15	94.16	87.74	63.44	61.49	52.77	56.80	75.84	72.60
- MRM	84.91	79.42	94.22	86.19	62.36	66.41	47.97	55.70	74.45	71.47
- EPM	87.78	83.54	94.10	88.51	65.73	64.13	52.50	57.74	76.33	72.84
-M	74.15	67.43	93.42	78.33	61.47	59.98	36.33	45.25	74.12	69.70



Conclusion

- Data augmentation
 - OntoZSL [Geng et al. WWW'21], DOZSL [Geng et al. KDD'22]
- Feature Propagation
 - GCNZ [Wang et al. CVPR'18], X-ZSL [Geng et al. SWJ'21], DOZSL [Geng et al. KDD'22], RMPI [Geng et al. ICDE'23]
- KG for VQA
 - ZS-F-VQA [Chen et al. ISWC'21]
- Semantic Grounding
 - DUET [Chen et al. AAAI'23]
- Transformer
 - KGTransformer [Zhang et al. WWW'23]

See our hands-on session (Part III) for more details
of experimenting with representative KG-aware ZSL
methods

Thanks!

Contacts:

Jiaoyan Chen (jiaoyan.chen@manchester)

Yuxia Geng (yuxia.geng@hdu.edu.cn)

Zhuo Chen (zhuo.chen@zju.edu.cn)

Wen Zhang (zhang.wen@zju.edu.cn)

Jeff Z. Pan (j.z.pan@ed.ac.uk)