A Addition Statement for Our New Dataset

A.1 Dataset Documentation and Intended Use

We offer a detailed overview of our dataset statistics in Sec. 3.3. To facilitate better understanding and ease of access, we have made our dataset project available on ModelScope at: *https://www.modelscope.cn/datasets/yutong/UKnow/summary*, which includes dataset summary, data preview, quickstart and data files.

The detailed data organization and corresponding download links are listed below:

- Original data: We gather our data from publicly available international news sources, accumulating a substantial volume of images and text. Subsequently, we compress the collected data into several zip archives and store them in original_data: `UKnow/raw_data/*`.
- Processed data:
  - Pre-node N_p: Building upon Phase-1, we leverage pre-trained deep learning models to extract valuable information from various domains. The resultant output from Phase-1 is structured as a dictionary and is then stored and saved to pre_node: `UKnow/processed_data/pre_node*`.
  - Node index N_n and Edge index N_e: As the outcomes acquired in Phase-1 (e.g., N_p) are not directly applicable for graph construction, we employ an information symbolization strategy to organize them into indices, namely N_n and N_e, which are subsequently saved to index: `UKnow/processed_data/*_index*.pickle`.
  - Knowledge graph G_m: Finally, we consolidate two types of internal knowledge (I_{in}, T_{in}) and three types of associative knowledge ($I_{cross}, T_{cross}, IT_{cross}$) into into one knowledge graph ($\mathbf{G}_m$), which is stored as a dictionary in graph: `UKnow/processed_data/graph*.pickle`.

Our dataset is intended for academic use and the corresponding license is based on: https://www.contributor-covenant.org/zh-cn/version/1/4/code-of-conduct.html, which was created by Coraline Ada Ehmke in 2014 and is released under the CC BY-NC-ND 4.0.

A.2 Author statement

We confirm the data licenses and that we bear all responsibility in case of violation of rights.

A.3 Hosting, licensing, and maintenance plan

Hosting and Licensing. Our dataset is hosted on ModelScope. Moreover, we furnish the relevant licenses in accordance with ModelScope at: https://www.contributor-covenant.org/zh-cn/version/1/4/code-of-conduct.html, which was created by Coraline Ada Ehmke in 2014 and is released under the CC BY 4.0 License.

Introduction to ModelScope. ModelScope is a platform designed for managing and optimizing machine learning models. It provides various tools and features to streamline the model development process, including version control, performance monitoring, and collaboration capabilities. As for managing datasets, ModelScope offers robust functionality for organizing, storing, and accessing data. Users can upload datasets to the platform, where they are securely stored and can be easily accessed by authorized team members. ModelScope also supports versioning of datasets, allowing users to track changes over time and ensure reproducibility in their experiments. Additionally, the platform provides tools for data preprocessing, visualization, and analysis, helping users to efficiently prepare their data for model training and evaluation. Overall, ModelScope offers comprehensive support for managing datasets throughout the machine learning lifecycle. Therefore, we choose ModelScope as our hosting platform.

Usage of ModelScope. To enable users to directly utilize all models on the ModelScope platform without configuring the environment, ModelScope integrates an online Notebook programming environment on its website and offers official mirrors for developers. These official mirrors allow users to bypass all installation and configuration steps, providing immediate access to the models. Currently the latest version of the CPU mirror and GPU mirror can be obtained from the office ModelScope repository.

Users also can setup local python environment using following commands:

```
conda create -n modelscope python=3.8
conda activate modelscope
pip install modelscope
```

Then, users can access and enjoy our dataset by:

```
from modelscope.msdatasets import MsDataset
ds =  MsDataset.load(‘yutong/UKnow’,␣subset_name=‘default’, split=‘train’)
```

Besides, we strongly recommend that users read the official documents for optimal use.

Maintenance Plan. In future work, we will persistently augment the dataset across various scales following the *UKnow* protocol. This endeavor aims to furnish a comprehensive, diverse, and resilient multimodal knowledge graph, thereby facilitating subsequent research endeavors.

B Preliminaries

Multimodal Knowledge Graph. An intuitive interpretation of multimodal knowledge graph is that the ordinary knowledge graph only consists of <head, relation, tail> triples like <(“Jony”), Citizen, (“New York”)> , but the multimodal knowledge graph consists of the following:
<(“Jony”), Citizen, (“NewYork”)>,
<(“Jony”), Appearance, (“[Face]”)>,
<(“NewYork”), Landmark, (“[Statueofliberty]”)>,
<(“[AirForceOne]”), Similarity, (“[AirForceTwo]”)>,
where $(\cdot)$ means a text node and $[\cdot]$ means a image node. The machine cannot understand what *“An old man with white hair”* is without establishing the connection between each word and its physical world meaning. However, with the help of multimodal knowledge graph, as a simple example, it is possible to generate a more informative entity-level sentence (*e.g.*, *“Biden is making a speech”*) instead of a vague concept-level description (*e.g.*, *“An old man with white hair is making a speech”*). To evaluate the effectiveness of multimodal knowledge graph (MMKG), several downstream tasks are often performed on the MMKGs, including common-sense reasoning, vision-language pre-training.

Common-sense Reasoning. Common-sense reasoning means answering queries by logic permutations. The specific task in this work is the link prediction. In the inference phase, feeding <("America"), Capital> to a reasoning model, the output should be <("Washington")>. Various works [3, 70, 68, 60, 94, 53] achieve reasoning by embedding entities and relations in knowledge graph into low-dimensional vector space. For instance, GQE [15] encodes queries through a computation graph with relational projection and conjunction ($\wedge$) as operators. Path-based methods [27, 82, 63, 51] start from anchor entities and determine the answer set by traversing the intermediate entities via relational path. There are also GCN [25] based methods [61, 16] pass message to iterate graph representation for reasoning. Common-sense reasoning is an extremely popular task in the field of knowledge graph. Since our dataset is based on the knowledge graph, the performance validation on common-sense reasoning is indispensable.

Vision-Language Pre-training Vision-language pre-training (VLP) can be divided into three categories based on how they encode images [10]: OD-based region features [5, 31, 34, 41, 66, 69], CNN-based grid feature [62, 19, 20] and ViT-based patch features [84, 30, 24]. Pre-training objectives are usually: masked language/image modeling (MLM/MIM) [2, 9, 39], image-text matching (ITM) [34, 19, 10], and image-text contrastive learning (ITC) [30, 50, 35]. In this work, we concentrate on the study of the how to introduce our UKnow into ITC method based on ViT-based patch features.

Image-Text Contrastive Learning. The recent CLIP [50] and ALIGN [21] perform pre-training using a crossmodal contrastive loss on millions of image-text pairs, which achieves remarkable performance on various downstream tasks [42, 62, 64]. MDETR [23] trains on multi-modal datasets which have explicit alignment between phrases and objects. GLIP [32] generates grounding boxes in a self-training fashion, and makes the learned representations semantic-rich. We implement these mainstream methods on our dataset, and also design a basic knowledge-based ITC method with UKnow.

C Experimental Details

In this section, we give more details about the computation complexity, training, fine-tuning hyperparameters and evaluation for reference.

C.1 Common-sense Reasoning

Datasets. Since our dataset is a knowledge graph, we benchmark the performance of KG-reasoning models on our dataset by completing KG-triples. The partitioning of the dataset is illustrated in the upper segment of Tab. 4.

Evaluation. The specific task of common-sense reasoning in this work is the link prediction. Given a test query q (*e.g.*„ <(“Jony”), Citizen, (?)>), we are interested in discovering non-trivial answers (*e.g.*„ “New York”). That is, answer entities where at least one edge needs to be imputed in order to create an answer path to that entity. Each entity in our multimodal knowledge graph is not limited to a text entity but a multimodal node. Following [56], for each non-trivial answer t of test query q, we rank it against non-answer entities $\mathcal{E}\setminus[\![q]\!]_{\text{test}}$ [3]. Then the rank of each answer is labeled as r. We use Mean Reciprocal Rank(MRR): $\frac{1}{r}$ and Hits-at-N $(\mathbf{H}@N)$: $1[r \leq N]$ as quantitative metrics.

Table 9: **A new benchmark of the common-sense reasoning task.** We report four metrics of each model on the validation and test sets. All experiments were repeated five times and the variance is shown in the table.

Model	Val-H@1	Val-H@3	Val-H@10	Val-MRR	Test-H@1	Test-H@3	Test-H@10	Test-MRR
TransE [3]	11.75 ± 0.113	29.04 ± 0.112	31.76 ± 0.143	14.77 ± 0.153	11.26 ± 0.114	21.68 ± 0.115	31.57 ± 0.127	14.66 ± 0.123
Q2B [54]	14.99 ± 0.118	25.78 ± 0.135	36.76 ± 0.169	18.80 ± 0.166	14.48 ± 0.119	25.17 ± 0.135	36.32 ± 0.163	18.46 ± 0.134
Q2B*	16.84 ± 0.115	29.00 ± 0.166	38.85 ± 0.169	19.66 ± 0.158	16.35 ± 0.122	28.67 ± 0.174	38.45 ± 0.184	19.27 ± 0.146
BETAE [55]	18.04 ± 0.129	33.02 ± 0.161	41.97 ± 0.179	21.16 ± 0.167	17.65 ± 0.129	32.75 ± 0.160	41.67 ± 0.177	20.75 ± 0.140
BETAE*	19.02 ± 0.125	33.97 ± 0.173	43.17 ± 0.199	21.64 ± 0.173	18.22 ± 0.135	33.52 ± 0.187	42.68 ± 0.198	21.23 ± 0.154
QA-GNN [87]	21.69 ± 0.124	38.11 ± 0.167	45.97 ± 0.180	22.83 ± 0.179	21.05 ± 0.128	37.26 ± 0.164	44.32 ± 0.175	22.06 ± 0.165

Table 10: **A new benchmark of the novel event classification task.** All models are fine-tuned in the training set.

Model	IMG	TXT	Event-11		Event-9185	
			ACC@1	ACC@5	ACC@1	ACC@5
CLIP [50]	✓		65.77	76.82	54.62	63.19
DeCLIP [35]	✓		66.43	78.32	54.86	63.82
ALBEF [30]	✓		66.29	77.84	55.03	63.47
TCL [85]	✓		66.80	78.91	55.87	64.33
CLIP		✓	64.32	75.92	57.48	65.78
DeCLIP		✓	65.89	77.51	59.76	67.81
ALBEF		✓	65.31	76.97	58.43	66.32
TCL		✓	66.03	78.14	59.94	68.23
CLIP	✓	✓	66.08	72.88	57.42	65.65
DeCLIP	✓	✓	67.16	72.96	58.64	66.49
ALBEF	✓	✓	68.03	74.26	60.04	68.13
TCL	✓	✓	68.69	75.02	60.89	69.17

Baselines. We consider four baselines: TransE [3], Q2B [54] and BETAE [56]. Since the *UKnow* based plug-in module can be attached to any reasoning models, we implement the Q2B* with our module based on Q2B and BETAE* based on BETAE. As shown in Tab. 9, BETAE* achieves on average **21.64%** and **21.23%** MRR on the validation and testing set of our dataset, respectively. For a fair comparison (*e.g.*, TransE), our dataset does not construct complex logic such as FOL [14] to evaluate the performance of multi-hop logical reasoning.

C.2 Multimodal Event Classification

We propose a novel task called multimodal event classification, leveraging event annotations (Tab. 3) from both Wiki's event categories and our own manual tagging. The event annotation helps intelligent machines understand human activities and history, offering the possibility to identify which *type of event* or which *real historical event* a picture or a text is relevant to. As shown in Tab. 10, TCL [85] achieves on **66.80%** and **55.87%** on ACC@1 when using the image-input on the *Event-11* and *Event-9185*, respectively. We simply modify all the baseline methods and add a late-fusion module after the image/text encoder to support multimodal classification. Results show that TCL with multimodal inputs obtains gains of **1.89%** and **5.02%** compared with the singlemodal, which demonstrates that multimodal pre-training is more helpful for downstream multimodal tasks.

C.3 Single- & Cross-Modal Retrieval

We design four kinds of single- & cross-modal retrieval tasks: image-to-image, text-to-text, image-to-text, and text-to-image. The construction of GT is based on the event annotations in G_m (Fig. 4). We treat images or texts belonging to the same news event as a similar semantic cluster, and the goal of retrieval is to recall the nearest neighbors within this cluster. The features used for retrieval are derived from the output of the previous layer of the classifier.

As shown in Tab. 11, TCL [85] achieves on **33.24%**, **43.37%** and **45.22%** R@1, R@5, R@10 on the zero-shot setting of image retrieval. The results are **58.89%**, **68.47%** and **73.91%** when fine-tuning the pre-trained parameters, which means the pre-training→fine-tuning strategy is extremely beneficial for downstream retrieval. We provide more details about hyperparameters in Sec. C.5.

C.4 Visual Task Adaptation

Visual Task Adaptation Benchmark (VTAB) [89] is a diverse, realistic, and challenging vision representation benchmark, containing 19 tasks and covering a broad spectrum of domains and semantics. These tasks are grouped into three sets: NATURAL, SPECIALIZED, and STRUCTURED which utilize natural world, professional technology and artificial environment images respectively. We benchmark models on VTAB with ACC@1. We fine-tune models for 10 epoch in each task and compute the inner product between outputs of

Table 11: **A new benchmark of the retrieval task.** Zero-shot means freezing the pre-trained parameters then transfer to the test set for inference. Fine-tune means tuning the pre-trained parameters in the training set before inference.

Model	Retrieval	Zero-Shot			Fine-Tune		
		R@1	**R@5**	**R@10**	**R@1**	**R@5**	**R@10**
CLIP [50]	IMAGE	32.41	41.96	43.92	55.97	67.44	71.28
DeCLIP [35]	IMAGE	32.75	42.36	44.38	56.96	66.59	70.95
ALBEF [30]	IMAGE	32.88	42.76	44.79	58.56	67.83	72.24
TCL [85]	IMAGE	33.24	43.37	45.22	58.89	68.47	73.91
CLIP	TEXT	33.02	42.56	46.03	56.50	65.12	70.20
DeCLIP	TEXT	34.00	43.97	47.11	55.87	65.20	70.35
ALBEF	TEXT	33.87	43.86	46.82	56.77	65.91	71.15
TCL	TEXT	34.67	44.25	47.67	56.60	65.50	70.54
CLIP	IMG-to-TXT	32.73	42.64	44.72	56.32	66.93	70.61
DeCLIP	IMG-to-TXT	32.96	42.84	45.17	57.21	66.80	71.26
ALBEF	IMG-to-TXT	33.20	42.97	45.32	58.43	67.59	71.95
TCL	IMG-to-TXT	33.37	43.25	46.04	58.70	67.88	72.33
CLIP	TXT-to-IMG	31.78	41.04	42.51	55.74	64.38	69.56
DeCLIP	TXT-to-IMG	32.13	41.55	42.99	55.84	65.12	70.32
ALBEF	TXT-to-IMG	31.95	41.32	42.85	57.21	66.04	71.50
TCL	TXT-to-IMG	32.56	42.04	43.74	57.17	65.92	71.47

Table 12: **The comparison of *w/* and *w/o UKnow* pre-training.** Zero means the model is initialized with all-zero parameters *w/o* pre-training. CLIP* means pre-training with origin CLIP contrast loss on our dataset. Ours means *UKnow* pre-training.

	CIFAR100	Caltech101	DTD	Flowers102	Pets	SVHN	Sun397	Camelyon	EuroSAT	Resisc45	Retinopathy	ClevrCount	ClevrDist	DMLab	KITTIDist	dSprLoc	dSprOri	sNORBAzim	NORBElev	**VTAB (avg.)**
Zero	58.39	53.54	49.26	52.51	58.93	64.24	48.96	52.44	63.95	60.03	58.62	62.78	62.59	44.27	45.87	75.89	74.48	67.54	60.89	58.69
CLIP*	75.25	71.74	58.39	77.54	74.40	79.42	61.72	70.42	81.56	76.43	67.85	81.25	80.48	60.03	63.98	84.33	82.66	83.68	76.57	74.09
Ours	76.79	72.73	60.44	78.48	76.33	80.56	62.37	72.23	83.27	77.26	65.91	82.46	81.34	63.37	65.74	85.61	82.79	85.12	76.64	**75.23**

images and label texts with prompts [50] through pre-trained image encoders and text encoders as the similarity score. As shown in Tab. 12, our approach obtains gains of avg. **1.14%** compared with the origin CLIP when fairly using the same *UKnow*'s data for the upstream pre-training. For the suboptimal performance on the Retinopathy and NORBElev datasets, we carefully examine the composition of both dataset. The Diabetic Retinopathy dataset consists of image-label pairs with high-resolution retinal images labeled to indicate the presence of diabetic retinopathy (DR) on a scale from 0 to 4. Similarly, the NORBElev dataset contains jittered texture images. It is evident that these data significantly differ from the natural images collected in UKnow. In constract, commonly used general image datasets in practical applications, such as CIFAR-10, tend to show greater improvements when utilizing UKnow. This observation suggests that researchers, when designing advanced knowledge-based pre-training methods with UKnow, should carefully consider balancing data domains according to specific downstream tasks. Additionally, accurate node construction is essential for building a robust multimodal knowledge graph to fully leverage the advantages of UKnow. This underscores the importance of designing effective pre-processing functions P, particularly in specialized subfields such as the Retinopathy dataset. In these domains, more dedicated data pre-processing models, such as medical image segmentation and detection models, can be employed to enhance feature extraction.

The backbone of CLIP is ViT-B/32. The cost of pre-train is 26h / 30epoch. The key hyperparameters are *bs: 512, lr: 0.001, warmup: 1e4, eps: 1e-8, beta1: 0.9, beta2: 0.999, dim: 512, AdamW*. The detailed setting can be found in Sec. C.5. It is essential to highlight that the image-text PAIR constitutes only one type of data in our protocol. By leveraging the capabilities of *UKnow* , our pre-trained CLIP model can effectively comprehend the inherent knowledge ingrained within the data, resulting in superior performance than the original CLIP model (as observed in Tab. 12, Row2, utilizing image-text PAIR only).

C.5 Hyperparameters

Tab. 13 and Tab. 14 list the hyperparameters that differ on each models and are determined with the validation performance on our dataset. In particular, Tab. 13 lists 7 common hyperparameters, such as learning rate, batch size, warmup, epoch number, *etc.*, employed during pre-training. The pre-trained model is evaluated using a standard pipeline consisting of pre-training on Dataset1, fine-tuning on Dataset2-Train, and testing on either Dataset2-Test/Val. Therefore, we list the hyperparameters used during fine-tuning in Tab. 14, which are slightly different from Tab. 13. We omit some of the model results, since ALBEF and TCL share the same set of hyperparameters, and the original CLIP and CLIP-UKnow share the same set of parameters.

Table 13: **Hyperparameters for models of pre-training.**

Hyperparameter	**ALBEF**	**DeCLIP**	**CLIP-UKnow**
Learning Rate	0.0001	0.001	0.001
Batch Size	128	128	512
Number of Epochs	30	30	30
Weight Decay	0.02	0.1	0.1
Optimizer	`AdamW`	`AdamW`	`AdamW`
Feature Dim	256	512	512
Warmup	20epc	5000	10000

Table 14: **Hyperparameters for models of fine-tuning.**

Hyperparameter	**ALBEF**	**DeCLIP**	**CLIP-UKnow**
Learning Rate	0.0001	$5e$-5	$5e$-5
Batch Size	128	256	256
Number of Epochs	128	20	20
Weight Decay	0.02	0.02	0.02
Optimizer	`AdamW`	`AdamW`	`AdamW`
Feature Dim	256	512	512
Warmup	4epc	6epc	6epc

C.6 Computation Complexity

Here we detail the time cost of pre-training and fine-tuning. The GPU is NVIDIA(R) A100, the memory of GPU is 81,251MiB, driver version is 470.154, CUDA version is 11.4. The CPU is Intel(R) Xeon(R) Platinum 8369B @ 2.90GHz with 15 physical computation cores. The environment is Python 3.6.12 with Torch 1.10.1. Results are as shown in Tab. 15 and Tab. 16.

Table 15: **The time cost of pre-training.**

Model	**Backbone**	**Epoch**	**Batch**	**Time/h**
DeCLIP	ViT-B/32	30	128	91
ALBEF	ViT-B/16	30	128	69
TCL	ViT-B/16	30	128	67
CLIP*	ViT-B/32	30	512	25
CLIP-UKnow	ViT-B/32	30	512	26

Table 16: **The time cost of downstream fine-tuning.**

Model	**Backbone**	*UKnow* Tasks			VTAB		
		Epoch	**Batch**	**Time/h**	**Epoch**	**Batch**	**Time/h**
DeCLIP	ViT-B/32	20	128	12	-	-	-
ALBEF	ViT-B/16	20	128	10	-	-	-
TCL	ViT-B/16	20	128	10	-	-	-
Zero*	ViT-B/32	-	-	-	15	128	3
CLIP*	ViT-B/32	20	256	8	15	128	3
CLIP-UKnow	ViT-B/32	20	256	8	15	128	3

D Discussion

D.1 Complexity

We notice that the detailed pipeline and protocol may appear complex and require effort to implement and understand fully. However, this complexity is necessary to ensure that the pipeline is robust, flexible, and capable of handling diverse and multimodal datasets.

To mitigate the implementation challenges, we have designed the pipeline to be modular, like Phase-1/2/3, allowing each phase to be independently replaced, added, or disabled based on specific needs. Moreover, we present an extra dataset documentation and construct a website in Sec. A.1. It provides a detailed data organization, corresponding download links, and an example code to guide users through the process, making the protocol more accessible and easier to adopt. Our goal is to balance complexity with practicality, ensuring that the benefits of a thorough and versatile approach outweigh the initial learning curve.

D.2 Correlation between the Knowledge View and Phase 1

In Phase 1, Content Extraction is designed to preprocess raw data (such as images and texts) using pre-trained deep learning models, which extract essential information that serves as the foundation for our knowledge view. The extracted content N_p provides a rich, structured collection of attributes and features that capture both global and semantic-level details from the input. It transforms raw data into a set of key-value pairs that represent various aspects of the input content. These key-value pairs encapsulate knowledge at different levels, which are critical for constructing meaningful nodes in the subsequent phases. This structured output essentially forms the knowledge view of our system, where each extracted piece of information is treated as a node attribute. These attributes are later symbolized and linked in Phase 2, leading to the construction of the multimodal knowledge graph in Phase 3. Thus, the content extracted in Phase 1 is directly correlated with the knowledge view, serving as the core data that the entire graph construction process relies upon.

D.3 Limitation and Future Work

Despite the strides made, our research bears certain limitations. First of all, our current dataset primarily centers on text and image modalities which serve as fundamental pillars for information storage and representation, but lack other useful modalities. In future work, we aim to diversify modalities by augmenting our dataset with a broader range of modalities (*e.g.,* audio, video, 3D, etc.) to facilitate exploration across various downstream tasks. Second, for each downstream task, we selected several basic yet most suitable methods for our work as our baseline, resulting in slight deviations with current state-of-the-art (SOTA) performance. Our primary objective lies in validating the efficacy of our proposed dataset and protocols, and demonstrating the most straightforward and intuitive approach for utilizing our dataset. Hence, we made certain trade-offs, sacrificing some performance by opting for a more rudimentary approach instead of pursuing the SOTA method to enhance understanding and usage. We anticipate that our simplified demonstration will stimulate the community to delve deeper into the potential enhancements that *UKnow* can offer in improving performance.

D.4 Societal Impact

As stated in Sec. 3.2, our dataset originates from publicly accessible international news sources via the Wikipedia API. These sources only contain events that are publicly available and do not include any sensitive information. Consequently, we confidently affirm that our research carries no potential negative societal impacts.