Supporting Vision-Language Model Inference with Causality-pruning Knowledge Prompt

Jiangmeng Li  
jiangmeng2019@iscas.ac.cn  
Institute of Software Chinese Academy of Sciences  
Beijing, China

Wenyi Mo  
1462959772@qq.com  
Renmin University of China  
Beijing, China

Wenwen Qiang  
wenwen2018@iscas.ac.cn  
Institute of Software Chinese Academy of Sciences  
Beijing, China

Bing Su  
subingats@gmail.com  
Renmin University of China  
Beijing, China

Changwen Zheng  
changwen@iscas.ac.cn  
Institute of Software Chinese Academy of Sciences  
Beijing, China

ABSTRACT
Vision-language models are pre-trained by aligning image-text pairs in a common space so that the models can deal with open-set visual concepts by learning semantic information from textual labels. To boost the transferability of these models on downstream tasks in a zero-shot manner, recent works explore generating fixed or learnable prompts, i.e., classification weights are synthesized from natural language describing task-relevant categories, to reduce the gap between tasks in the training and test phases. However, how and what prompts can improve inference performance remains unclear. In this paper, we explicitly provide exploration and clarify the importance of including semantic information in prompts, while existing prompt methods generate prompts without exploring the semantic information of textual labels. A challenging issue is that manually constructing prompts, with rich semantic information, requires domain expertise and is extremely time-consuming. To this end, we propose Causality-pruning Knowledge Prompt (CapKP) for adapting pre-trained vision-language models to downstream image recognition. CapKP retrieves an ontological knowledge graph by treating the textual label as a query to explore task-relevant semantic information. To further refine the derived semantic information, CapKP introduces causality-pruning by following the first principle of Granger causality. Empirically, we conduct extensive evaluations to demonstrate the effectiveness of CapKP, e.g., with 8 shots, CapKP outperforms the manual-prompt method by 12.51% and the learnable-prompt method by 1.39% on average, respectively. Experimental analyses prove the superiority of CapKP in domain generalization compared to benchmark approaches.

CCS CONCEPTS
• Computing methodologies → Ontology engineering; Causal reasoning and diagnostics; Image representations.

KEYWORDS
multi-modal, vision-language model, prompt engineering, causality, knowledge graph, ontology

ACM Reference Format:
Jiangmeng Li, Wenyi Mo, Wenwen Qiang, Bing Su, and Changwen Zheng. 2022. Supporting Vision-Language Model Inference with Causality-pruning Knowledge Prompt. In Proceedings of Proceedings of ACM Multimedia conference (Conference ’22). ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION
As a promising alternative for visual representation learning, vision-language pre-training methods, e.g., CLIP [50] and ALIGN [31], jointly learn image and text representations with two modality-specific encoders by aligning the corresponding image-text pairs, which is achieved by adopting contrastive loss in pre-training. Benefiting from pre-training on large-scale data, models learn numerous visual concepts so that the learned representations have a strong generalization and can be transferred to various downstream tasks. [72] observes that the zero-shot generalization performance of the pre-trained vision-language model heavily relies on the form of the text input. Feeding pure labels, i.e., textual names of categories, into the text encoder leads to degenerate performance. To tackle this issue, recent works adopt various prompts to augment the textual labels [31, 32, 50, 53, 55, 72]. In the inference stage, the classification weights, i.e., textual label features, are obtained by providing the text encoder with prompts describing candidate categories. The image feature generated by the image encoder is compared with these label features for zero-shot classification.

We conclude the typical prompt generation paradigms in Figure 1. The paradigm, shown in Figure 1 a, rigidly applies the fixed prompt template, which suffers from a dilemma that a specific prompt template has inconsistent boosts for different tasks. A motivating example, proposed by [72], shows that using "a photo of a
2 demonstrates that prompts with additional semantic information boost the performance of CLIP on all downstream tasks.

To this end, we propose an innovative knowledge-aware prompt learning approach for pre-trained vision-language models, namely, Causality-pruning Knowledge Prompt (CapKP). As illustrated in Figure 1c, CapKP explores the semantic information associated with the label text by using labels as queries to retrieve an ontological knowledge graph. In practice, we observe that some derived knowledge is redundant for downstream tasks, which may degenerate the performance of our method. Therefore, CapKP introduces causality-pruning to refine the derived label-related knowledge subgraph following the first principle of Granger causality [23]. Empirically, CapKP outperforms the state-of-the-art methods, and the transferability comparison supports that CapKP demonstrates stronger robustness than the benchmark methods to domain shifts.

Contributions. The contributions of this paper are four-fold:

- We present a motivating study on the prompt engineering approaches for pre-trained vision-language models in downstream applications and identify the importance of exploring the semantic information of label texts.
- For effectively mining semantic information from the label text, we propose Causality-pruning Knowledge Prompt, which derives label-related semantic information by retrieving an ontological knowledge graph.
- Following the first principle of Granger causality, we propose a causality-pruning method to remove task-redundant information from the label-related knowledge subgraph.
- Empirically, we impose comprehensive comparisons to prove the effectiveness and generalization of our method.

2 RELATED WORK

2.1 Vision-Language Models

Recent development of joint learning on vision and language representations achieves impressive success in various fields, including Visual Question Answering [1, 2, 17, 34], Image Captioning [30, 69], etc. A critical issue is that few high-quality annotated multimodal data is available. Therefore, state-of-the-art vision-language models are designed to be pre-trained on massive unannotated modal data is available. Therefore, state-of-the-art vision-language models are designed to be pre-trained on massive unannotated data by taking advantage of Transformer [61], e.g., ViLBERT [42], LXMERT [59], UNITER [7] and Oscar [38]. Such large-scale pre-trained vision-language models have great potential for learning universal representations and transferring them to various downstream tasks via prompting [31, 71]. A representative approach is CLIP [50], which pre-trains modality-specific encoders from 400 million image-text pairs and achieves impressive performance in zero-shot reference to multitudinous downstream tasks.

2.2 Prompt Design

Since directly applying pre-trained models to downstream tasks often leads to degenerate performance, CLIP [50] and PET [53] convert the labels of the downstream task into a batch of manual prompt templates. AutoPrompt [55] proposes to automatically search prompts from a template library. [32] proposes two approaches for building the prompt templates, including mining-based and paraphrasing-based approaches. However, such template-based...
prompting has a critical issue that despite the large-scale candidate template library, the optimal prompt may be excluded.

To perform effective and data-efficient improvement on downstream tasks, simple yet effective adapter-based approaches are proposed, which insert the extra learnable neural network, i.e., adapter, into the large pre-trained models and then train the adapter on downstream tasks under the premise of freezing the weights of the backbone, e.g., Adapters [29], CLIP-Adapter [16] and Tip-Adapter [70]. The adapter-based approach can be treated as a post-model prompting, which focuses on improving the performance in the inference stage by re-training adapters, but such an approach does not explore the latent visual concept knowledge learned by the vision-language model in the pre-training stage, which is contrary to the fundamental idea behind prompting, i.e., making the vision-language model recall the pre-trained knowledge relevant to the current downstream task. CoOp [72] and DenseCLIP [51] are proposed to automatically learn prompts without the template library, which aim to generate prompts that can make the vision-language model recall the task-relevant knowledge. These methods do not explore the semantic information of the label text in the inference stage, while in this paper we prove the importance of including the label-relevant semantic information in prompting and hence propose to derive such information by leveraging an ontological knowledge graph.

2.3 Knowledge Graph

Knowledge graph abstracts the knowledge in the real world into triples, e.g., <entity, relationship, entity>, to form a multilateral network of relationships, where nodes represent entities and the edges connecting nodes represent the relationships between entities. Knowledge graphs include general domain knowledge graphs, e.g., Wikidata [62], NELL [5], CN-dbpedia [67], ConceptNet [58], etc., and specific domain knowledge graphs, e.g., Open PHACTS [24], Watson [15], AMiner [60], etc. Specifically, ontological knowledge graphs [18] only have the ontology entities, i.e., conceptual types, for instance, Wikidata-ZS and NELL-ZS [49]. To understand the graph-based information from knowledge graphs, Graph embedding [19, 21, 56, 65] is proposed, which maps the high-dimensional graph data into the low-dimensional vector, e.g., TransE [3], TransR [41], RESCAL [45], KG-BERT [68]. To mine graph structure information, Graph Neural Network (GNN) based methods are proposed, e.g., KGCN [64]. For our approach, we refine the knowledge graph by considering graph causality to eliminate redundant information.

2.4 Graph Causality

In system identification, clarifying the causal relationship between variables by observing the data is a crucial research field. Granger causality [23] is widely used in many fields, e.g., neural network [33], financial economy [36], and medicine [44]. Graph structure has a strong ability to incorporate prior knowledge so graph-based approaches have become crucial tools [9, 47] for analyzing the complex relationships of various interactions among system variables. For understanding the high-dimensional and heterogeneous graph system [39, 73, 74], recent works adopt GNN to learn a graph representation. However, due to the lack of explicit declarative knowledge representation, such methods are regarded as black boxes. Obtaining the graph causality improves the model to mine the latent semantic information of the graph. [10] proposes to study the Granger causality among variables in a graph, which is extended by [13, 14]. From the perspective of causality, Gem [40] understands the behavior of GNN by following Granger causality and describes the causal relationship between each node and the output by splitting local subgraphs.
3 PRELIMINARIES

3.1 Vision-Language Pre-training

CLIP [50] introduces a pre-training approach to learn semantic knowledge from large amounts of image-text data.

3.1.1 Architecture. CLIP consists of an image encoder and a text encoder. The image encoder aims to learn a high-dimensional representation from an image, which can be implemented by a ResNet [25] or a ViT [12]. The text encoder aims to learn a text representation from a sequence of words, which is implemented by a Transformer [61].

3.1.2 Training. For texts, all tokens (words and punctuations) are mapped into lower-cased byte pair encoding representations [54]. They are further projected into vectors with 512 dimensions, which are then fed to the text encoder. The input sequence of the text encoder is capped at a fixed length of 77. The input images are encoded into the embedding space by the image encoder. CLIP is trained downstream tasks by measuring the similarity of image features with label-related semantic information rather than adopting a fixed prompt template, which is achieved by introducing refined knowledge from an external knowledge graph. To this end, CapKP consists of two stages: 1) ontology-enhanced knowledge embedding derives the label-related subgraph from an ontological knowledge graph by using the label token as a query; 2) causality-pruned graph representation removes causally irrelevant edges and nodes.

3.2 Graph Representation Learning

3.2.1 Graph Setup. We recap necessary preliminaries of graph representation learning. Let $\mathcal{G} = (V, E)$ be an attributed graph, where $V$ is the node set and $E$ is the edge set. Given a graph dataset $\mathcal{G} = G_i$, $i \in [1, N^G]$, where $G_i$ is sampled i.i.d. from the distribution $\mathcal{P}(\mathcal{G})$, the objective of graph representation learning is to learn an encoder $f^G(\cdot) : \mathcal{G} \rightarrow \mathbb{R}^{d^G}$, where $\mathbb{R}^{d^G}$ denotes a $d^G$-dimensional embedding space and $f^G(G_i)$ is the representation of $G_i$.

3.2.2 Graph Neural Network. Most benchmark methods employ GNN as the encoder. GNN encodes each node in $G_i = (V_i, E_i)$ into a representation vector, where $H_o$ denotes the representation vector

\[ \mathcal{P}(y = i|H) = \frac{\exp \left( \langle l_i, h \rangle \right)}{\sum_{j=1}^{K} \exp \left( \langle l_j, h \rangle \right)} \]

where $y$ denotes the semantically correct category for $x$, $\tau$ is the temperature hyper-parameter in CLIP, and $\langle \cdot, \cdot \rangle$ denotes the cosine similarity. Compared with the conventional classifier learning approach where only closed-set visual concepts can be classified, the zero-shot inference paradigm of vision-language pre-training models can explore open-set concepts with the text encoder.
We determine the Granger causality by iteratively removing the edges related to a relation-type (Semantic information in prompts). Introducing Assumption 4.1.

where \( \mathbf{v} \) is a set of learnable feature vectors, which are randomly initialized by Gaussian distributions. \( \varphi (\cdot) \) denotes the function of our proposed CapKP, and \( \lambda \) is the coefficient that controls the balance between \( \mu \) and \( \varphi (\cdot) \). \( b (\{Y_i\}) \) denotes the lower-cased byte pair encoding representation of label \( Y_i \), and \( \oplus \) is a concatenation function. Note that the output of \( \varphi (\cdot) \) is a vector with the same dimension as \( b (\{Y_i\}) \), e.g., 512 for CLIP. Feeding prompts \( \{p_i\}_{i=1}^K \) to the text encoder \( f^T (\cdot) \), we obtain the classification weights \( \{\mathbf{t}_i\}_{i=1}^K \), and the prediction probability is computed by Equation 1.

4.1.2 Label-shared prompt. From the perspective of revisiting the training data for the vision-language model, we observe that the input text does not focus on describing the label-specific and discriminative semantic information; on the contrary, words with semantic information shared by different labels appear in a large body of descriptive text. For the examples "a [golden retriever] runs on the grass with its tail wagging" and "an [Alaskan] sits on a couch with a floppy tail", there only exists the label-shared information, i.e., "tail", but no label-specific information. Such a phenomenon is
common in the description of fine-grained labels, and we thus hold an extended assumption:

**Assumption 4.2.** (Generalized semantic information in prompts). Label-specific semantic information could be task-redundant to prompt pre-trained vision-language models, while generalized label-shared semantic information is crucial for generating effective prompts.

See Section 5 for the experimental proof for Assumption 4.2. We thus propose a label-shared prompt form by

\[ p_i = \left( \mu + \lambda \cdot \psi \left( \left\{ \varphi \left( [Y_j] \right) \right\}_{j=1}^K \right) \right) \oplus b \left( [Y_i] \right). \]

where \( \varphi \) presents a cascade concatenation function, detailed by \( \left\{ \varphi \left( [Y_j] \right) \right\}_{j=1}^K = \varphi \left( [Y_1] \right) \oplus \varphi \left( [Y_2] \right) \oplus \ldots \oplus \varphi \left( [Y_K] \right) \), and \( \psi \) presents a linear mapping function in CapKP.

### 4.2 Ontology-enhanced Knowledge Embedding

We propose to retrieve an ontological knowledge graph by treating

\[ \text{where } \lfloor \cdot \rfloor \text{ presents a non-linear network, } \theta \text{ denotes the process of quantitatively computing H, and } \gamma \text{ is a inner-product function.} \]

\[ \varphi \left( [Y_j] \right) = \sum_{u \in \mathcal{N}(v)} \exp \left( < H_u, H_{e_{v,u}} > \right) \sum_{u' \in \mathcal{N}(v)} \exp \left( < H_u, H_{e_{v,u'}} > \right) \]

\[ \text{where } \mathcal{N}(v) \text{ denotes the corresponding representation for a node or an edge, } < \cdot, \cdot > \text{ is an inner-product function, and } \mathcal{N}(v) \text{ is the neighborhood set of } v \text{ in the causality-pruned graph } \kappa \left( G_i \right). \]

### 4.3 Causality-pruned Graph Representation

To describe our intuition of causality-pruning, we reform the definition of Granger causality [23] in the field of knowledge graph:

**Definition 4.3.** (Granger causality in knowledge graph). Granger causality [22, 23] describes the relationships between two (or more) variables when one is causing the other. In the field of the knowledge graph, if we are better able to predict variable \( Y \), e.g., higher score computed by a specific graph rule, using all available information \( \Omega \) than if the information apart from a variable \( X \), e.g., a type of relation, had been used, we say that \( X \) Granger-causes \( Y \) [22].

\[ \Delta_{r_m} = \varepsilon_{r_m} - \varepsilon_{G_i} \]

\[ \begin{cases} r_m \text{ Granger - causes } Y, & \Delta_{r,m} > 0 \\ r_m \text{ NOT Granger - causes } Y, & \Delta_{r,m} \leq 0 \end{cases} \]
50
55
60
75
50
60
54
11 benchmark datasets within 8 shots. CapKP outperforms CoOp on most image classification datasets: ImageNet [11], Caltech101 [37], Stand-
ardCars [35], FGVC Aircraft [43], Flowers102 [46], OxfordPets [48], Food101 [4], SUN397 [66], UCF101 [57], DTD [8], and EuroSAT [26].

Table 1: Comparisons of CapKP with CoOp. Both models are trained on 11 benchmark datasets within 8 shots. CapKP outperforms CoOp on most datasets. \(\Delta\) denotes CapKP’s gain over CoOp.

|          | CoOp | CapKP | \(\Delta\) |
|----------|------|-------|-----------|
| OxfordPets | 85.32 | 91.18 | 5.86      |
| Flowers102 | 26.13 | 59.97 | 33.84     |
| FGVC Aircraft | 76.73 | 68.43 | -8.29     |
| DTD       | 71.82 | 71.22 | -0.60     |
| EuroSAT   | 71.94 | 61.56 | -10.38    |
| ImageNet  | 61.63 | 61.28 | -0.35     |
| Average   | 69.89 | 71.28 | 1.39      |

4.4 Variants of Our Method

Our complete method is called CapKP. We perform an ablation study by eliminating the module of causality-pruning and deriving a variant KP. Considering two forms of prompts that are discussed in Section 4.1.1 and Section 4.1.2, we abbreviate label-specific prompt and label-shared prompt as SPE and SHR, respectively.

4.5 Algorithm pipeline

In the inference stage, we train and evaluate our method by following the benchmark-setting [72]. It is worth noting that we only adopt causality-pruned graph representation in the test phase of few-shot learning. We take CapKP(SPE) as an example to demonstrate the pipeline in Algorithm 1.

5 EXPERIMENTS

5.1 Few-Shot Learning

5.1.1 Datasets. We conduct experiments on 11 publicly available image classification datasets: ImageNet [11], Caltech101 [37], StandardCars [35], FGVC Aircraft [43], Flowers102 [46], OxfordPets [48], Food101 [4], SUN397 [66], UCF101 [57], DTD [8], and EuroSAT [26].

5.1.2 Baselines. We compare our approach with two major baseline models: 1) CLIP [50], which is based on manual prompts, and we follow the instructions for prompt ensembling in [50] and input seven corresponding prompt templates into the CLIP text encoder; 2) CoOp [72], which automatically designs the prompt templates, and for fair comparisons, we adopt the best variants of CoOp.

5.1.3 Training Details. The set of learnable feature vectors \(\mu\) is randomly initialized by zero-mean Gaussian distributions with a standard deviation of 0.02. According to the parameter study in Appendix A, we assign the coefficient \(\lambda\) to \(10^{-2}\). We set the maximum epoch to 200, 100, and 50 for 16/8 shots, 4/2 shots, and 1 shot, respectively, while the maximum epoch on ImageNet is fixed to 50 for all shots. Unless otherwise specified, ResNet-50 [25] and Transformer [61] are used as the corresponding image and text...
encoders. We initially adopt Wikidata-ZS [49] as the target ontological knowledge graph, while we also conduct experiments to evaluate our method using Nell-ZS [49] in Appendix B.

5.1.4 Comparison with Baselines. The experimental results on 11 benchmark datasets are demonstrated in Figure 5, and the average results are shown in the top-left subfigure. We observe that CapKP achieves state-of-the-art results under settings of different shots. With fewer shots, e.g., 1/2/4/8 shots, CapKP improves the baselines by a significant margin. With the increase of shots, each compared method achieves better performance and the performance gap becomes smaller, while CapKP still outperforms benchmark methods. Figure 6 shows the gains obtained by CapKP at 16 shots over the hand-crafted prompt method, i.e., zero-shot CLIP. In specific tasks, e.g., Eurosat, CapKP beats zero-shot CLIP by nearly 50%. In Table 1, we observe that the improvements of CapKP over CoOp reach 4.18%, 2.40%, and 1.41% on fine-grained image classification datasets, including Food101, OxfordPets, and FGVC Aircraft, respectively. CapKP also outperforms CoOp by a significant margin (more than 1%) on scene and action recognition tasks, e.g., UCF101 and SUN397. The effectiveness of KP and CapKP further verifies the proposed Assumption 4.1 and Assumption 4.2, respectively.

5.2 Domain Generalization

5.2.1 Datasets. For domain generalization experiments, we use ImageNet as the source dataset and four variants of ImageNet, i.e., ImageNetV2 [52], ImageNet-Sketch [63], ImageNet-A [28] and ImageNet-R [27], as the target datasets. The classes of the variants are subsets of the 1,000 classes of ImageNet, allowing seamless transfer for the prompts learned by CoOp or CapKP.

5.2.2 Results. The results, with various vision backbones, are shown in Table 3. CapKP achieves the best performance on most datasets, which demonstrates that our method is generally more robust to distribution shifts than baselines. CapKP(M=4) has better performance than CapKP(M=16), which is tenable and consistent with [72], i.e., using fewer context tokens leads to better robustness.

5.3 Further Analysis

5.3.1 Interpreting the Learned Prompts. We interpret the learned prompt by transforming the learned feature vector into the word closest to the corresponding vector in the hidden space. Table 2 shows the visualized feature vectors of μ on benchmark datasets. We observe that there exist words that are task-relevant, e.g., “winery,” “grain” and “ente” for Food101, “compliant” and “dog” for OxfordPets. From the experimental results demonstrated by [50], we observe that CoOp hardly learns task-relevant lexical features, since its training is only based on gradient back-propagation. Concretely, our proposed CapKP empowers the model to learn task-relevant feature vectors with rich semantic information. See Appendix C for further study, which demonstrates that the vectors learned by CapKP(SPE) capture more task-related words.

5.3.2 Visualization of Causality-pruning. We visualize the process of the proposed causality-pruning. Figure 7 illustrates two examples on Food101 and StanfordCars, which show that different relation-types Granger-cause predicting the graph on different datasets. Following the first principle of Granger causality, CapKP can remove several task-irrelevant relations. The results in Figure 5 further support the effectiveness of the proposed causality-pruning.

6 CONCLUSION

In this paper, we find out the importance of the textual label’s semantic information for prompting the pre-trained vision-language model through empirical observation. To explore such semantic information, we propose CapKP, which complements semantic information for the input label text by leveraging an ontological knowledge graph and further refining the derived label-relevant knowledge subgraph by the proposed causality pruning. We conduct extensive comparisons to prove the superiority of CapKP over benchmark manual prompt methods and learnable prompt methods in few-shot classification and domain generalization.

REFERENCES

[1] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering. In CVPR. Computer Vision Foundation / IEEE Computer Society, 6077–6086.

[2] Stanislav Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. 2015. VQA: Visual Question Answering. In ECCV. IEEE Computer Society, 2448–2454.

[3] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakunkino. 2013. Translating Embeddings for Modeling Multi-relational Data. In NIPS. 2787–2795.

[4] Lukas Bussard, Matthieu Guillaumin, and Luc Van Gool. 2014. Food-101: Mining Discriminative Components with Random Forests. In ECCV (6) (Lecture Notes in Computer Science, Vol. 8694). Springer, 446–461.

[5] Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R. Hruschka Jr., and Tom M. Mitchell. 2010. Toward an Architecture for Never-Ending Language Learning. In AAAI. AAAI Press.

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020. A Simple Framework for Contrastive Learning of Visual Representations. In Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event (Proceedings of Machine Learning Research). PMLR.

[7] Yun-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2019. UNITER: Learning Universal Image-TExt Representations. CoRR abs/1909.11740 (2019).

[8] Mircea Cimpoi, Subhransu Maji, Jasonos Kokkinos, Sammy Mohamed, and Andrea Vedaldi. 2014. Describing Textures in the Wild. In CVPR. IEEE Computer Society, 3606–3613.

[9] Rainer Dahlhaus. 2000. Graphical interaction models for multivariate time series. Metrika 51 (08 2000), 157–172. https://doi.org/10.1007/s001840000055

[10] Rainer Dahlhaus and Michael Eichler. 2003. Causality and graphical models in time series analysis. Oxford Stat. Sci. Ser 27 (01 2003).

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. In CVPR. IEEE Computer Society, 248–255.

[12] Aleksey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafo Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021.

[13] Michael Eichler. 2007. Granger causality and path diagrams for multivariate time series. Journal of Econometrics 137 (04 2007), 354–353. https://doi.org/10.1016/j.jeconom.2005.06.032

[14] Michael Eichler. 2008. Causal inference from time series. Proceedings from the 13th International Congress of Logic, Methodology and Philosophy of Science (01 2008).

[15] David A. Ferrucci, Eric W. Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya Kalyanpur, Adam Lally, J. William Murdock, Eric Nyberg, John M. Prager, Adria Vedaldi, and Tom M. Mitchell. 2010. Toward an Architecture for Never-Ending Language Learning. In AAAI. AAAI Press.

[16] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021.

[17] Michael Eichler. 2007. Granger causality and path diagrams for multivariate time series. Journal of Econometrics 137 (04 2007), 354–353. https://doi.org/10.1016/j.jeconom.2005.06.032

[18] Michael Eichler. 2008. Causal inference from time series: What can be learned from granger causality? Proceedings from the 13th International Congress of Logic, Methodology and Philosophy of Science (01 2008).

[19] David A. Ferrucci, Eric W. Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya Kalyanpur, Adam Lally, J. William Murdock, Eric Nyberg, John M. Prager, Adria Vedaldi, and Tom M. Mitchell. 2010. Toward an Architecture for Never-Ending Language Learning. In AAAI. AAAI Press.

[20] Michael Eichler. 2008. Causal inference from time series: What can be learned from granger causality? Proceedings from the 13th International Congress of Logic, Methodology and Philosophy of Science (01 2008).

[21] Peng Gao, Shijie Geng, Renru Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. 2021. CLIP-Adapter: Better Vision-Language Models with Feature Adapters. CoRR (2021).

[22] Peng Gao, Zhengkai Jiang, Haoxuan You, Pan Lu, Steven C. H. Hoi, Xiaogang Wang, and Hongsheng Li. 2019. Dynamic Fusion With Intra- and Inter-Modality
Table 2: Visualization of feature vectors $\mu$ with the length of 16 learned by CapKP(SHR). We derive the words by measuring the Euclidean distances between word embeddings and each specific feature vector of $\mu$, and the quantified distances are shown in parentheses. N/A represents non-Latin characters. The task-relevant words are marked in BOLD.

| # | ImageNet Food101 OxfordPets DTID UCF101 |
|---|---|---|---|---|---|
| 1 | where (1.4992) | winery (1.0086) | vac (1.2409) | himiss (1.1484) | **ditch** (1.2119) |
| 2 | N/A (1.2287) | grain (1.2845) | o (1.2407) | essential (0.9013) | **peek** (1.2222) |
| 3 | a (1.4260) | gra (0.9716) | sav (1.1540) | dw (1.5142) | tolerate (0.9314) |
| 4 | allow (1.5391) | N/A (0.9463) | rous (0.9295) | thats (1.1830) | photo (1.3124) |
| 5 | thepersonnetwork (5:642) | wonder(1.4191) | mo(0.8267) | **** (1.6619) | thinkin(1.4635) |
| 6 | inadequate (1.9244) | go(0.8285) | goe(0.8335) | serious(0.6558) | siveto(0.9611) |
| 7 | humans (1.4764) | artsy (1.2367) | zar (1.2723) | daener (1.3578) | spl (1.8495) |
| 8 | hon (1.2362) | valen (1.0958) | autumn (1.4099) | 2 (0.7010) | believed (1.6348) |
| 9 | inn (1.0578) | ente (1.2312) | / (1.1853) | N/A (1.4208) | stooll (0.2023) |
| 10 | for (1.0963) | preparation (1.3788) | firmly (0.9633) | der (1.4271) | **swing** (1.3811) |
| 11 | gi (1.1065) | absol (1.4179) | bles (0.8048) | ett (1.8202) | braving (1.8438) |
| 12 | s (1.5250) | ardu (1.6931) | **owner** (1.0334) | gs (1.1643) | gent (1.5629) |
| 13 | about (1.4677) | struck (1.0167) | compliaint (1.2167) | **order** (1.4104) | visuals (1.4737) |
| 14 | oxy (1.7380) | pab (0.6368) | **dog** (1.1007) | il (1.7677) | shima (1.7922) |
| 15 | s (1.7768) | main (1.5293) | scrutiny (1.3651) | phase (1.1731) | south (1.5168) |
| 16 | nage (1.6341) | er (0.9222) | enabled (1.1544) | death (0.9549) | par (1.6291) |

Figure 7: Visualization of Causality-pruning, demonstrating that CapKP removes task-irrelevant relations.

Table 3: Comparisons of CapKP with baselines on robustness to distribution shift using different vision backbones. Both CoOp and CapKP use the shared label prompt, i.e., SHR. $M$ denotes the length of learnable feature vectors $\mu$.

| Method | Target | Average |
|---|---|---|
| ResNet-50 | -V2 -Sketch -A -R |
| Zero-Shot CLIP | 51.34 | 33.32 | 21.65 | 56.00 | 40.58 |
| CoOp (M=16) | 55.11 | 37.24 | 22.12 | 54.96 | 41.23 |
| CoOp (M=4) | 55.40 | 34.67 | 23.06 | 56.60 | 42.43 |
| CapKP(M=16) | 55.48 | 33.10 | 21.57 | 54.49 | 41.16 |
| CapKP(M=4) | 55.14 | 34.75 | 23.43 | 57.44 | 42.69 |
| ResNet-101 | -V2 -Sketch -A -R |
| Zero-Shot CLIP | 54.81 | 38.71 | 28.05 | 64.38 | 46.49 |
| CoOp (M=16) | 58.66 | 39.08 | 28.89 | 63.00 | 47.41 |
| CoOp (M=4) | 58.60 | 40.40 | 29.60 | 64.49 | 48.39 |
| CapKP(M=16) | 58.03 | 39.78 | 28.87 | 63.46 | 47.54 |
| CapKP(M=4) | 59.06 | 40.80 | 29.91 | 65.29 | 48.77 |
| ViT-B/32 | -V2 -Sketch -A -R |
| Zero-Shot CLIP | 54.79 | 40.82 | 29.57 | 65.99 | 47.79 |
| CoOp (M=16) | 58.08 | 40.44 | 30.62 | 64.45 | 48.40 |
| CoOp (M=4) | 58.24 | 41.48 | 31.34 | 65.78 | 49.21 |
| CapKP(M=16) | 58.56 | 40.81 | 30.55 | 65.83 | 48.94 |
| CapKP(M=4) | **58.75** | **41.48** | **31.97** | **66.66** | **49.71** |
| ViT-B/16 | -V2 -Sketch -A -R |
| Zero-Shot CLIP | 60.83 | 46.15 | 47.77 | 73.96 | 57.18 |
| CoOp (M=16) | 64.18 | 46.71 | 48.41 | 74.32 | 58.41 |
| CoOp (M=4) | **64.56** | **47.89** | **49.93** | **75.14** | **59.38** |
| CapKP(M=16) | 64.32 | 46.99 | 49.13 | 74.40 | 58.71 |
| CapKP(M=4) | 64.28 | **48.19** | **49.29** | **75.59** | **59.34** |

Attention Flow for Visual Question Answering. In CVPR Computer Vision Foundation / IEEE, 6639–6648.

[18] Yuxia Geng, Jiaoyan Chen, Zhao Chen, Jeff Z. Pan, Zhiquan Ye, Zonggang Yuan, Yantao Jia, and Huajun Chen. 2021. OntoZSL: Ontology-enhanced Zero-shot Learning. In WWW: ACM / IW3C2, 3325–3336.
As shown in Figure 9, we report the results of the model trained on four datasets at 8 shots using Wikidata-ZS or NELL-ZS ontological knowledge graphs.

We detail the descriptions of the candidate knowledge graphs in Table 4. Nell-ZS is constructed based on NELL [5] and Wikidata-ZS is based on Wikidata [62]. Both Nell and Wikidata are large-scaled and another merit is the existence of official relation descriptions. The NELL and Wikidata are two well-configured knowledge graphs and the textual descriptions of Nell-ZS and Wikidata-ZS consist of multiple information.

We observe the results reported in Figure 9 and find that, generally, our model CapKP and the ablation model KP achieve better performance using the Wikidata-ZS knowledge graph compared to using NELL-ZS on several benchmark datasets, e.g., DTD and

![Parameter study on $\lambda$](image)

The value of the corresponding coefficient $\lambda$

Figure 8: Parameter study on $\lambda$. We choose the best coefficient value of $\lambda$ as $10^{-2}$ in benchmark experiments. The shade denotes the range of experimental results.

In this section, we provide several experimental analyses about the advantages of our proposed method. The experiments to find appropriate hyperparameters are conducted as well.

A APPENDIX: PARAMETER STUDY OF $\lambda$

As demonstrated in Figure 8, we report the results of the model with different $\lambda$ values based on Flowers102 at 1 shot. The parameter study is conducted on the validation set. To explore the influence of $\lambda$, we fix other experimental settings and select $\lambda$ from the range of $[10^{-4}, 10^{-3}, 10^{-2}, 0.1, 1]$. We can observe that the score reaches the maximum when the $\lambda$ is $10^{-2}$, which indicates that an appropriate tuning of the impact of the knowledge embedding to guide the training of learnable label features, i.e., $\mu$, can indeed promote the performance of CLIP on downstream tasks. While excessively emphasizing the impact of knowledge embedding on training may degenerate the ability of the learnable features $\mu$ to fit appropriate prompts needed for downstream tasks by using gradient back-propagation so that the performance of CLIP is weakened. The setting of $\lambda$ is shared among different downstream tasks.

B APPENDIX: PERFORMING CAPKP WITH DIFFERENT KNOWLEDGE GRAPHS

As shown in Figure 9, we report the results of the model trained on four datasets at 8 shots using Wikidata-ZS or NELL-ZS ontological knowledge graphs.

We detail the descriptions of the candidate knowledge graphs in Table 4. Nell-ZS is constructed based on NELL [5] and Wikidata-ZS is based on Wikidata [62]. Both Nell and Wikidata are large-scaled and another merit is the existence of official relation descriptions. The NELL and Wikidata are two well-configured knowledge graphs and the textual descriptions of Nell-ZS and Wikidata-ZS consist of multiple information.

We observe the results reported in Figure 9 and find that, generally, our model CapKP and the ablation model KP achieve better performance using the Wikidata-ZS knowledge graph compared to using NELL-ZS on several benchmark datasets, e.g., DTD and

![Wikidata-ZS vs. NELL-ZS](image)

Table 4: Statistics of the adopted ontological knowledge graphs. # Ent. denotes the number of unique entities. # Triples denotes the amount of relation triples. # Train/Dev/Test denotes the number of relations for training/validation/testing.

| Dataset  | # Ent. | # Triples | # Train/Dev/Test |
|----------|--------|-----------|-----------------|
| Nell-ZS  | 1,186  | 3,055     | 139/10/32       |
| Wiki-ZS  | 3,491  | 10,399    | 469/20/48       |

EuroSAT. We reckon the reason is that Wikidata-ZS has more detailed relations and entities, which empowers our method to locate label-related knowledge subgraphs, yet Nell-ZS does not have sufficient relations and entities so our method may not be able to find knowledge subgraphs corresponding to several specific labels. Such a conclusion is consistent with our proposed Assumption 4.1. However, we further observe that the difference between the performance of our method using Wikidata-ZS and using Nell-ZS is not extremely large on some benchmark datasets, e.g., Food101 and OxfordPets. According to Assumption 4.2, we speculate the reason is that Although Nell-ZS does not contain enough label-related knowledge, it contains sufficient generalized label-related knowledge for certain datasets, for instance, Nell-ZS does not contain the entities of "chocolate" and "potato", while it contains "concept:food" so that the knowledge subgraph of "concept:food" can be used for amounts of labels. According to Assumption 4.2, the important content of prompts may not contain label-specific and discriminative information, and generalized label-shared semantic information is crucial for generating effective prompts.

In general, both Assumption 4.1 and Assumption 4.2 can be further proved by the experiments in Figure 9.

C APPENDIX: INTERPRETING THE LEARNED SPE PROMPTS

Table 5 shows that the vectors $\mu$ learned by CapKP (SPE) captured words. As we can see from the table, compared with CapKP (SHR), CapKP (SPE) gets more task-related words, such as, "pesto", "baguette", "pistachio", and "cereals", etc., for Food101. Additionally,
Figure 10: Real-world examples of input pairs for CLIP in the pre-training phase, including descriptive text and images.

Table 5: Visualization of feature vectors $\mu$ with the length of 101 learned by CapKP(SPE) on Food101, i.e., a category corresponds to a word. We derive the words by measuring the Euclidean distances between word embeddings and each specific feature vector of $\mu$, and the quantified distances are shown in parentheses. N/A represents non-Latin characters.

| No. | No. | No. | No. | No. |
|-----|-----|-----|-----|-----|
| 1   | chandelier (0.7779) | 2   | N/A (0.7436) | 3   | oured (0.8188) | 4   | val (0.7030) |
| 5   | mises (0.7793) | 6   | pesto (0.6852) | 7   | daily (0.6644) | 8   | vaz (0.5889) |
| 9   | ergon (0.7521) | 10  | watercolor (0.7230) | 11  | daz (0.6812) | 12  | baguette (0.6713) |
| 13  | pistachio (0.6607) | 14  | cto (0.6839) | 15  | sista (0.6383) | 16  | dips (0.6789) |
| 17  | cereals (0.6704) | 18  | france (0.7560) | 19  | frozen (0.7448) | 20  | aquaman (0.6924) |
| 21  | antibiotic (0.6608) | 22  | valued (0.6634) | 23  | pulgia (0.7410) | 24  | closures (0.6657) |
| 25  | jerusalem (0.6763) | 26  | tomorrows (0.6448) | 27  | exec (0.6715) | 28  | kkkk (0.6721) |
| 29  | bir (0.6609) | 30  | cheeses (0.6547) | 31  | almond (0.6828) | 32  | ole (0.6342) |
| 33  | ube (0.7104) | 34  | overview (0.5283) | 35  | backpacks (0.7365) | 36  | eminem (0.7356) |
| 37  | favor (0.7520) | 38  | relive (0.8267) | 39  | adele (0.6814) | 40  | thfc (0.7880) |
| 41  | ols (0.6711) | 42  | tgf (0.7312) | 43  | bluebells (0.6687) | 44  | riverfront (0.6762) |
| 45  | cant (0.6855) | 46  | sharkweek (0.6512) | 47  | historia (0.7014) | 48  | demedebate (0.7810) |
| 49  | sip (0.6500) | 50  | poses (0.7375) | 51  | prioritize (0.7455) | 52  | woodworking (0.6334) |
| 53  | theflash (0.6589) | 54  | southbank (0.7308) | 55  | seniors (0.6221) | 56  | gd (0.7075) |
| 57  | netneutrality (0.8200) | 58  | rhp (0.7581) | 59  | itate (0.7597) | 60  | wines (0.7842) |
| 61  | firework (0.6766) | 62  | played (0.6340) | 63  | beal (0.7019) | 64  | sett (0.5379) |
| 65  | preparations (0.6624) | 66  | nocturn (0.6469) | 67  | cellphone (0.7243) | 68  | psb (0.7516) |
| 69  | saturday (0.6831) | 70  | grinder (0.6256) | 71  | enjoy (0.6534) | 72  | arunjaitley (0.6464) |
| 73  | period (0.7466) | 74  | bilingual (0.7739) | 75  | ate (0.7057) | 76  | airs (0.6548) |
| 77  | serenawilliams (0.6726) | 78  | beans (0.7494) | 79  | glaze (0.7056) | 80  | - (0.7802) |
| 81  | sobbing (0.6839) | 82  | earring (0.6759) | 83  | youthful (0.7889) | 84  | ance (0.6806) |
| 85  | N/A (0.6594) | 86  | kiwis (0.7744) | 87  | sport (0.6623) | 88  | inktober (0.7291) |
| 89  | handsome (0.6672) | 90  | sundaymorning (0.6880) | 91  | flir (0.6305) | 92  | theopen (0.6199) |
| 93  | stana (0.7209) | 94  | louisvuitton (0.7560) | 95  | roast (0.7016) | 96  | pgtour (0.6807) |
| 97  | yxe (0.7106) | 98  | almost (0.6571) | 99  | royalwedding (0.7211) | 100 | nit (0.7473) |
| 101 | birdwatching (0.7417) |
the vectors learned by CoOp [72] are basically ambiguous words, such as, “lc”, “beh”, “matches”, “nytimes”, “prou”, “lower”, “minute”, “~”, “well”, “ends”, “mis”, “somethin”, and “seminar”, etc. Therefore, CapKP captures more meaningful and task-relevant words than CoOp.

As demonstrated in Figure 10, we observe that the original input text of CLIP indeed contains several words with rich semantic information. Such a fact proves that our proposed assumptions are reliable, and the visualization results shown in Table 2 and Table 5 further demonstrate that our proposed method can learn words with semantic information. Concretely, we conclude that our method can indeed learn several label-related words by leveraging an ontological knowledge graph, and such an approach can improve the performance of vision-language models on downstream tasks, which is proved by our conducted comparisons.