Learning to Prompt CLIP for Monocular Depth Estimation: Exploring the Limits of Human Language

Imperial College London

Dylan Auty Krystian Mikolajczyk

Imperial College London

ICCV23 PARIS

OpenSUN 3D
1st Workshop on Open-Vocabulary Scene Understanding

Introduction and Motivation

This work focuses on using CLIP to perform monocular depth estimation.

- CLIP has broad and open-ended knowledge of many kinds of concepts
- Using this relies on correlation between prompt features and image features, allowing zero- or few-shot performance on a variety of tasks
- DepthCLIP [1] showed that the same 0-shot approach can be used for monocular depth estimation
- However, **prompting CLIP introduces bias by the use of human language, selected by humans**
- **We circumvent this bias by learning the tokens directly**, removing the biases arising from human choice of words and the discrete nature of human language.

We build on DepthCLIP using **learnable depth tokens** as part of the prompts, massively increasing performance with only a few thousand learnable parameters. Analysis of these learned tokens shows that they are surprisingly different to "sensible" human-language words, indicating that the optimal "words" for representing complex concepts to CLIP may not be words at all. We hope that this will inspire further work in the field that explores the role that non-linguistic tokens can play in open-ended scene understanding tasks.

Contributions

Our main contributions are:

1. **We introduce learnable prompt tokens to prompt CLIP for Monocular Depth Estimation (MDE).** Our system builds on and improves the work of DepthCLIP [1] by removing human biases caused by the use of human language.
2. **We perform extensive experiments to find optimal prompting strategies and templates.** We experiment with different numbers of depth-bin prompts, the use of learnable context tokens, and different depth-bin distributions.
3. **We analyse and interpret the learned prompts,** providing insight into the nature of the CLIP latent space and the suitability of language for prompting large language models in general. Our work gives compelling evidence that **human language is inefficient in precisely explaining the concept of depth to CLIP**, with consequences for future works that seek to better exploit its understanding of an open set of concepts.

Method Overview

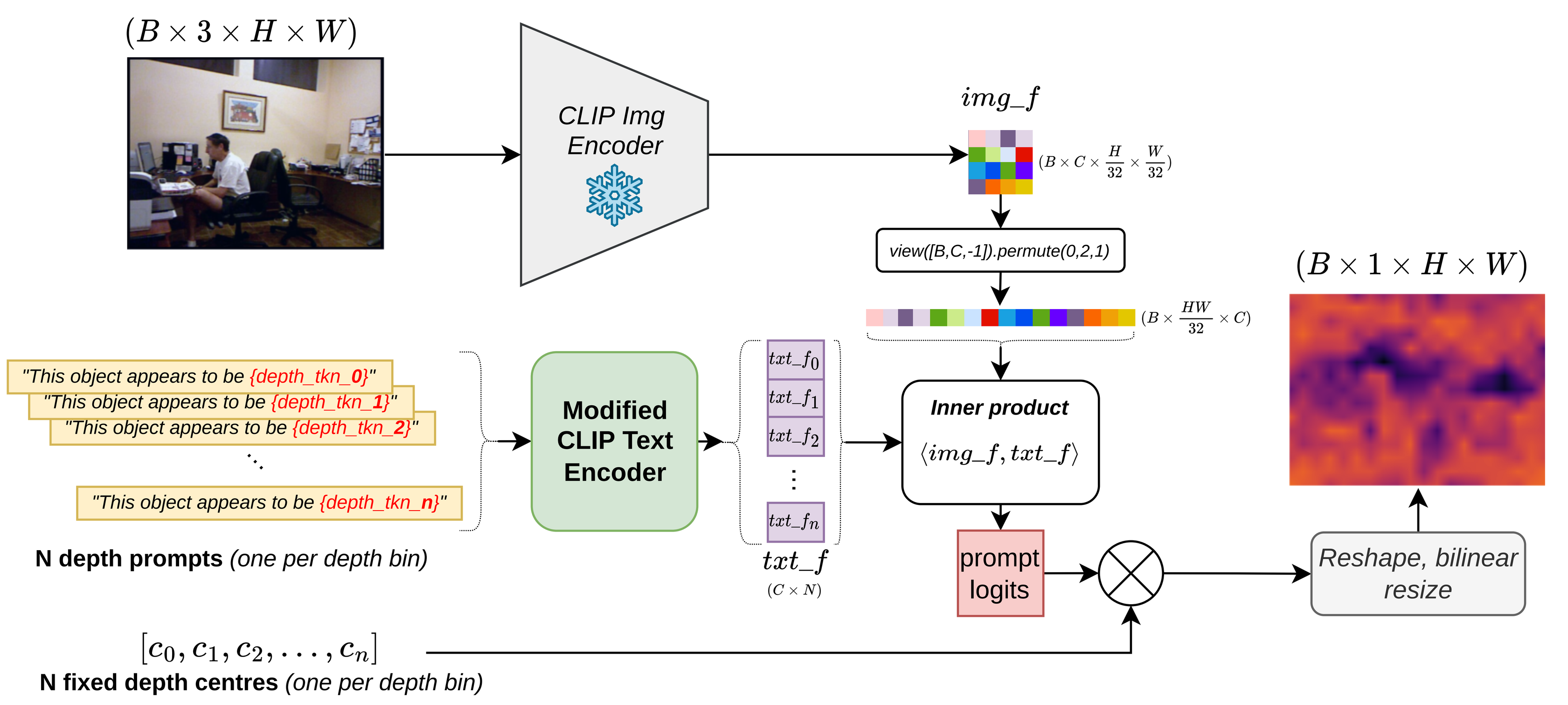

Figure 1. **An overview of the pipeline used**. The basic structure of the pipeline is the same as [1], but with our **modified CLIP encoder** used instead to allow the use of **learnable tokens in place of human words** in the prompts. The modified CLIP text encoder is detailed in figure 2. The pretrained CLIP model is completely frozen; the only parameters that we train are those in the learnable tokens. **Note that the output prediction is low-resolution by design:** our aim is to probe the limits of CLIP's understanding without the confounding factor of a specialised learned decoder.

Our method is shown in figure 1. Similar to DepthCLIP [1], the range of possible depth values is divided in to N bins, and each bin is assigned a text prompt, e.g. *"This object appears to be {very near/near/far/very far etc.}"*. These prompts are encoded with CLIP's text encoder and correlated with the CLIP image features for a given patch of the input image. The final depth value is the sum of the bin centers c_i (in metres) multiplied by that bin's prompt's correlation with the image patch features.

Using Learnable Depth Tokens

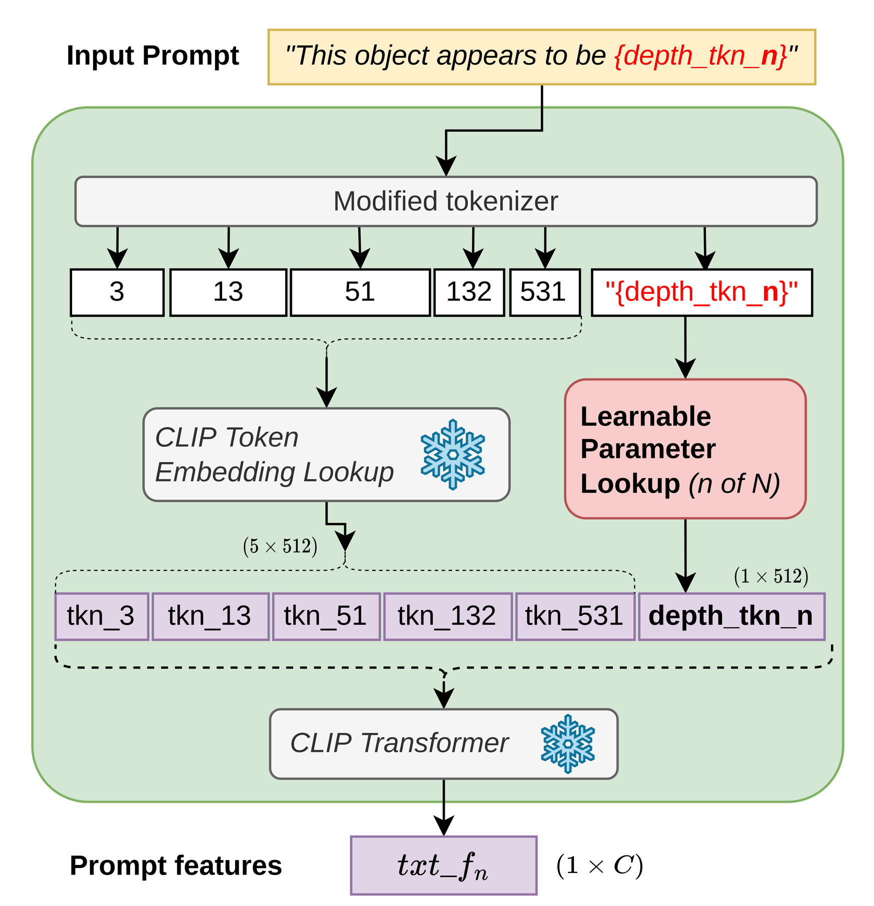

Figure 2. **Our modified CLIP text encoder**, used to allow the insertion of learnable tokens in place of the human-word tokens. Special words in the prompt are replaced by the tokenizer with learnable parameters, while all other words are replaced with the frozen and pretrained CLIP tokens.

The stock CLIP encoder replaces each word with a pretrained and frozen 512d token that is used as the input to the transformer. We **modify the CLIP text encoder** (figure 2), replacing the distance-related words (*"near/far" etc.*) with special tokens which are mapped to **learnable 512d tokens** instead of the pretrained ones. This means the total number of additional parameters is only $512 \times N$.

This allows learning of **non-linguistic depth tokens** to represent the concept of depth, removing the natural bias inherent in language. Analysis of the learned tokens shows that they do not always map to depth-related English words.

The same setup is also used to add learnable **context tokens**; these are constant between depth bin prompts and replace the priming text *"This object appears to be..."*.

Experiments are done that confirm the efficacy of these learnable depth tokens compared to depth-relevant, depth-irrelevant and random prompts. We also experiment with learnable context tokens and varying the binning strategy (see paper).

Depth Token Ablation: Learnable Vs. Human-Language Vs. Random Controls

Depth tkns	Dset	Abs	RMS	RMSL	log10	δ_1	δ_2	δ_3
colour-7	nyu	1.381	3.010	0.786	0.331	0.131	0.277	0.449
size-7	nyu	2.130	4.431	1.281	0.446	0.048	0.119	0.235
depth-7	nyu	1.014	2.413	0.566	0.265	0.227	0.431	0.606
random-7	nyu	1.593	3.308	0.929	0.372	0.081	0.190	0.354
rand-txt-7	nyu	1.335	2.875	0.754	0.324	0.129	0.287	0.471
learned-7	nyu	**0.319**	**0.970**	**0.139**	**0.128**	**0.465**	**0.776**	**0.922**
colour-7	kitti	2.177	23.470	1.328	0.446	0.077	0.163	0.267
size-7	kitti	3.363	33.978	2.067	0.568	0.048	0.103	0.171
depth-7	kitti	2.353	25.518	1.433	0.454	0.094	0.193	0.297
random-7	kitti	1.664	19.279	0.994	0.370	0.119	0.251	0.400
rand-txt-7	kitti	2.887	29.553	1.789	0.535	0.046	0.098	0.161
learned-7	kitti	**0.303**	**6.322**	**0.119**	**0.112**	**0.550**	**0.830**	**0.938**

Table 1. **Comparison of human-word, random, and learned tokens** across a 7-bin scale, on the NYUv2 and KITTI datasets. Best results in bold, second best underlined. The use of only a single learned token in each prompt improves performance significantly across every metric. We also see that the geometrically-related human-language tokens are not always the best, particularly for KITTI where size-7 is outperformed by colour-7.

We compare our learnable non-linguistic depth tokens to several control prompts for a 7-bin setup, both linguistic and non-linguistic. We use relevant-human-language ordinal scales of depth and size (*"very near/small"* to *"very far/large"*), and an irrelevant linguistic control of colour to control for the effect of the ordinal language itself (*"very/slightly"* etc.) with a non-geometric concept (redness/greenness). We also use 7 randomly selected tokens from the CLIP token vocabulary, and 7 frozen and random 512d vectors in place of the pretrained vectors that would normally be used to replace words.

The results (in table 1) show that our learnable non-linguistic depth prompts produce massive and immediate improvement over any of the controls in both the indoor and outdoor domains.

Ablation: Use of Learnable Context Tokens

Prompt format	Depth tkns	Dset	Abs	RMS	RMSL	log10	δ_1	δ_2	δ_3
Baseline	depth-7	nyu	1.014	2.413	0.566	0.265	0.227	0.431	0.606
1o1d	depth-7	nyu	0.323	0.975	0.142	0.129	0.461	0.772	0.920
1o2d	depth-7	nyu	0.323	0.974	0.141	0.129	0.462	0.773	0.920
4o4d	depth-7	nyu	**0.318**	**0.965**	**0.138**	**0.127**	**0.466**	**0.778**	**0.923**
Baseline	depth-7	kitti	2.353	25.518	1.433	0.454	0.094	0.193	0.297
1o1d	depth-7	kitti	0.331	6.528	0.132	0.120	0.511	0.809	0.929
1o2d	depth-7	kitti	0.321	6.420	0.127	0.117	0.527	0.817	0.932
4o4d	depth-7	kitti	**0.309**	**6.334**	**0.122**	**0.114**	**0.541**	**0.826**	**0.936**

Table 2. **Effect of adding learned context tokens to human-language depth tokens**; "**x**o**y**d" indicates **x** learnable context tokens, then the word "object", then **y** further learnable context tokens, then the depth token for that prompt (using 7-point human language ordinal depth scale for depth tokens).

Prompt format	Depth tkns	Dset	Abs	RMS	RMSL	log10	δ_1	δ_2	δ_3
Baseline	learned-7	nyu	0.319	0.970	0.139	0.128	0.465	0.776	0.922
ls4o4d	depth-7	nyu	0.318	0.965	0.138	0.127	0.466	0.778	0.923
ls4o4d	learned-7	nyu	**0.317**	**0.955**	**0.136**	**0.126**	**0.474**	**0.782**	**0.925**
Baseline	learned-7	kitti	**0.303**	6.322	**0.119**	**0.112**	**0.550**	**0.830**	**0.938**
ls4o4d	depth-7	kitti	0.309	6.334	0.122	0.114	0.541	0.826	0.936
ls4o4d	learned-7	kitti	0.307	**6.209**	**0.119**	0.113	0.546	**0.830**	**0.938**

Table 3. **Effect of combining both learned context and learned depth tokens.** Some improvement from the combined use of both learned depth tokens and learned context tokens may be seen, but in the case of KITTI the results indicate that the majority of the performance may be attributed to the learnable depth tokens.

Following the work of [3] and [2], tokens are learnt that aim to prime the model to retrieve the correct subset of the latent space. These remain the same between depth bins, but the learnable depth tokens still change as before. We see that using more learnable context tokens produces better performance (table 2) and that these tokens combine well with the learnable depth tokens to produce some performance improvement depending on the dataset used (table 3).

Analysis of Learned Tokens

{depth_0}		{depth_1}		{depth_2}		{depth_3}		{depth_4}		{depth_5}		{depth_6}	
Token	Dist.	Token	Dist.	Token	Dist.	Token	Dist.	Token	Dist.	Token	Dist.	Token	Dist.
{depth_2}	0.177	*close</w>*	0.000	*{depth_0}*	0.177	*{depth_2}*	0.313	*{depth_6}*	0.185	*distant</w>*	0.000	*{depth_4}*	0.185
{depth_3}	0.320	*closest</w>*	0.725	*{depth_3}*	0.313	*{depth_0}*	0.320	*{depth_2}*	0.780	*distance</w>*	0.656	*{depth_2}*	0.776
{depth_6}	0.781	*close*	0.746	*{depth_6}*	0.776	*{depth_6}*	0.813	*{depth_0}*	0.790	*dissi*	0.907	*{depth_0}*	0.781
{depth_4}	0.790	*closes</w>*	0.835	*{depth_4}*	0.780	*{depth_4}*	0.820	*{depth_3}*	0.820	*dist*	0.924	*{depth_3}*	0.813
·</w>	0.895	*clo*	0.872	*·</w>*	0.907	*coscino</w>*	0.915	*coscino</w>*	0.913	*thest</w>*	0.977	*coscino</w>*	0.894
coscino</w>	0.911	*glou*	0.883	*coscino</w>*	0.918	*·</w>*	0.918	*·</w>*	0.976	*ssian</w>*	1.006	*·</w>*	0.955
atility</w>	0.923	*closer</w>*	0.887	*atility</w>*	0.928	*atility</w>*	0.918	*atility</w>*	0.981	*pist*	1.010	*atility</w>*	0.971
mikequind	0.949	*closing</w>*	0.887	*mikequind*	0.956	*mikequind*	0.969	*mikequind*	0.992	*dian*	1.013	*mikequind*	0.980
arty	0.979	*chose</w>*	0.914	*arty*	0.980	*arty*	0.982	*laghate*	1.010	*drifting</w>*	1.015	*laghate*	0.992
kirstel</w>	0.987	*lose</w>*	0.926	*kirstel</w>*	0.993	*âł*	0.994	*ison</w>*	1.018	*distribu*	1.022	*ison</w>*	1.000
rhea</w>	0.993	*closed</w>*	0.942	*rhea</w>*	1.002	*ison</w>*	0.995	*kirstel</w>*	1.023	*distri*	1.022	*kirstel</w>*	1.015
laghate	1.001	*chosen</w>*	0.947	*ison</w>*	1.005	*kirstel</w>*	1.002	*rectan*	1.042	*titan*	1.028	*arty*	1.015
ison</w>	1.002	*choose*	0.953	*laghate*	1.009	*rhea</w>*	1.005	*arty*	1.042	*dis*	1.033	*rectan*	1.026
âł	1.005	*most</w>*	0.958	*pknot</w>*	1.014	*pknot</w>*	1.013	*soyuz</w>*	1.055	*disappear</w>*	1.033	*rhea</w>*	1.029
pknot</w>	1.005	*chooses</w>*	0.971	*âł*	1.022	*laghate*	1.017	*rhea</w>*	1.056	*dito</w>*	1.034	*soyuz</w>*	1.030

Table 4. **Nearest-neighbours to learned depth tokens in CLIP embedding space.** Learned tokens from 7 evenly-distributed bins on NYUv2. 'Distance' is Euclidean distance after normalisation of embeddings. Token 0 corresponds to a bin centre of approx. 0.714m, and token 6 to approx. 9.29m. We see that tokens 1 and 6 correspond with the tokens 'close</w>' and 'distant</w>' respectively, but that the remaining tokens are closest to other learned tokens.

Interestingly, while the learned tokens tend to be near to one another in CLIP embedding space, their nearest neighbours from the CLIP token space have seemingly nothing to do with depth or distance in all but two cases (shown in table 4). This may indicate that the nuance contained in the concept of depth is not trivially explained in words, and that unrelated words may contain more "meaning" relating to apparently unrelated concepts than would have been thought.

Conclusion

We improve on the CLIP prompting for monocular depth estimation technique of DepthCLIP [1] by introducing **learnable tokens to better represent the concept of depth**. We are able to produce significant improvements in performance with only a few thousand learnable parameters, and we find the learned tokens to be significantly different than human-language tokens that might be assumed sensible. These findings show that while CLIP contains surprising general knowledge, accessing it using human-chosen prompts may be sub-optimal, implying that its understanding of the world extends beyond the limits of what language can succinctly represent. While this work focuses on the concept of specific depths, it may be the case that other similarly abstract concepts could also require learnable, non-linguistic tokens to effectively describe them.

References

[1] Renrui Zhang, Ziyao Zeng, and Ziyu Guo. Can Language Understand Depth? In *ACM Multimedia 2022*. arXiv, July 2022. arXiv:2207.01077 [cs].

[2] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional Prompt Learning for Vision-Language Models, October 2022. arXiv:2203.05557 [cs].

[3] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to Prompt for Vision-Language Models. *International Journal of Computer Vision*, 130(9):2337–2348, September 2022. arXiv:2109.01134 [cs].