Appendix

A. Architecture Details

A.1. Multi-head Attention

By following the instruction from Transformer [44], the Q, K, V are computed from input hidden states $H \in \mathbb{R}^{T\times D}$ and $H' \in \mathbb{R}^{T'\times D}$. The two input matrices consist of respectively T and T' tokens of d dimensions each. The transformation is as follows:

$$\begin{aligned} Q &= HW_Q \quad W_Q \in \mathbb{R}^{D\times d_k}, \\ K &= H'W_K \quad W_K \in \mathbb{R}^{D\times d_k}, \\ V &= H'W_V \quad W_V \in \mathbb{R}^{D\times d_k}. \end{aligned} \tag{15}$$

An attention map is computed by the pairwise similarity between two tokens from H and H'.

$$\text{Attention}(Q,K,V) = \text{softmax}\left(QK^\top/\sqrt{d_k}\right)V, \tag{16}$$

After splitting k heads from H and H', the Multi-Head Attention (MHA) is concatenated from the outputs by running k attention operations. The same calculations of Q, K, V are conducted in each $i \in [k]$ head to form $Q^{(i)}$, $K^{(i)}$, $V^{(i)}$.

$$\begin{aligned} Head^{(i)} &= \text{Attention}(Q^{(i)}, K^{(i)}, V^{(i)}), \\ \text{MHA}(Q,K,V) &= \text{concat}_{i\in[k]}\left[Head^{(i)}\right] W_O, \end{aligned} \tag{17}$$

where the weight $W_O \in \mathbb{R}^{kd_k\times D}$ projects the concatenation of k head results to the output space D with the same dimension of the inputs. In our models, we set $d_k = D/k$. The other contents in the transformer block, such as MLP Block and residual connection, follow the instructions of Transformer [44]. D is set to 768 and k is set to 12 in our experiments. The detailed experiment settings are presented in the open-sourced codes.

A.2. Multi-head operation in PDE

The operation is similar to the aforementioned multi-head attention in A.1. In this operation, the input hidden states $H \in \mathbb{R}^{T\times D}$ are split into k heads, where T is sequence length and D is hidden size. In each head, we split the features and send them to two paths (μ, σ^2). The operation in the σ^2 path is followed:

$$\begin{aligned} &[Q^{(i)}_{\sigma^2}, K^{(i)}_{\sigma^2}, V^{(i)}_{\sigma^2}] = HW_{qkv}, \\ &Head^{(i)}_{\sigma^2} = \text{Act}\left(Q^{(i)}_{\sigma^2}{K^{(i)}_{\sigma^2}}^\top/\sqrt{d_k}\right)V^{(i)}_{\sigma^2}, \\ &\text{MH}_{\sigma^2}(Q^{(i)}_{\sigma^2}, K^{(i)}_{\sigma^2}, V^{(i)}_{\sigma^2}) = \text{concat}_{i\in[k]}\left[Head^{(i)}_{\sigma^2}\right] W_O, \end{aligned} \tag{18}$$

where d_k is set to $D/(2k)$. The weight $W_{qkv} \in \mathbb{R}^{d_k\times 3d_k}$ projects the inputs to the sub-space in each head. The weight $W_O \in \mathbb{R}^{kd_k\times D}$ projects the concatenation of k head results to the output space. The "Act" is an activation function and normalization function for considering sequence-level interaction. Moreover, the "MH" is the multi-head operation. On the σ^2 path, since the predicted vector has negative values from the activation function, PDE is expected to predict $\log\sigma$. After a simple exp operation, variance vectors are obtained. There are some candidate activation functions are considered: ReLU, ReLU^2, Sigmoid, and Softmax. Unless otherwise specified, the function Softmax is employed in PDE.

Dataset	#Images	#Text
Flickr30K [35]	29K	145K
GQA [17]	79K	1M
MSCOCO [30]	113K	567K
VG [25]	108K	5.4M
SBU [34]	875K	875K
CC-3M [38]	3.1M	3.1M
CC-12M [5]	12M	12M
ALIGN [18]	1.8B	1.8B

Table 7. Details of pre-training datasets in Table 8.

A.3. D-MLM settings

Masked Language Modeling (MLM) is first utilized as a pre-training strategy of BERT [8] to predict masked words, which enhances the ability of contextual modeling. In multimodal pre-training, the missing words are reconstructed with retained text and information from another modality. The model can correctly identify the entity relationships between text and images, learning cross-model semantic alignment. Following the settings from several multimodal models [9, 23], the model randomly covers the text tokens with a probability of 15%, where 80% tokens are replaced with [MASK] token, 10% tokens are replaced with other random words, and 10% tokens remain unchanged.

B. Experiment Details

B.1. Experimental settings

Our experiments are conducted on 8 NVIDIA A100 GPUs. For usual settings in all experiments, we adopt the AdamW optimizer. The learning rate is warmed up first and then decayed linearly. When sampling point vectors from distribution representations, the sample number K is set to 5. In the pre-training phase, the model is trained for 100K steps with a batch size of $4,096$. The learning rate of feature extractors is set to $1e-5$. Cross-modal transformer and PDE's learning rates are both $5e-5$.

For pre-training details, We pre-train our model with D-MLM, D-ITM and D-VLC. In Equation 5 of D-VLC, a

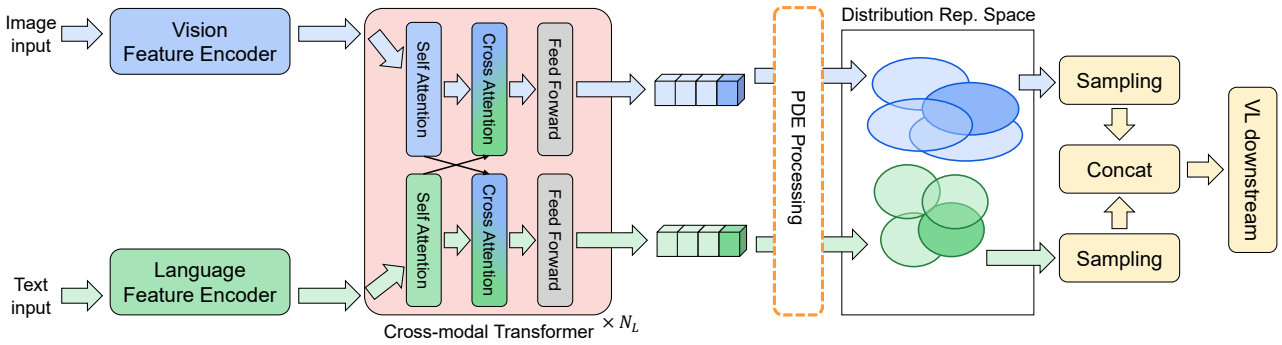

Figure 6. Fine-tuning MAP on different VL downstream tasks.

Model	Paper	Pre-training Datasets	Model size
Pre-training datasets include $>$ 10*M images*			
ALBEF (14M)	[26]	MSCOCO, VG, CC-3M, SBU, CC-12M	Base
SimVLM-base	[48]	ALIGN	Base
Pre-training datasets include $<$ 10*M images*			
UNITER-Large	[6]	MSCOCO, VG, CC-3M, SBU	Large
VILLA-Large	[10]	MSCOCO, VG, CC-3M, SBU	Large
UNIMO-Large	[27]	MSCOCO, VG, CC-3M, SBU	Large
VinVL-large	[56]	MSCOCO, CC-3M, SBU, F30k, GQA	Large
ViLT	[23]	MSCOCO, VG, CC-3M, SBU	Base
UNITER -Base	[6]	MSCOCO, VG, CC-3M, SBU	Base
OSCAR-Base	[29]	MSCOCO, VG, CC-3M, SBU	Base
UNIMO-Base	[27]	MSCOCO, VG, CC-3M, SBU	Base
ALBEF (4M)	[26]	MSCOCO, VG, CC-3M, SBU	Base
VLMo-Base	[47]	MSCOCO, VG, CC-3M, SBU	Base
TCL	[53]	MSCOCO, VG, CC-3M, SBU	Base
METER	[9]	MSCOCO, VG, CC-3M, SBU	Base

Table 8. Details of all models in Table 1 and 2.

is set to -0.005 and b is set to 6. In the full loss formula Equation 14, α is equal to 0.01. For the regularization loss of distributions in Eq. 10, the threshold $\gamma = 300$.

Table 7 reports the statistic of images and text of the pre-training datasets in Table 8, which includes the pre-training datasets of all referenced models. Those datasets are constructed by combining public datasets. However, a substantial portion of the image URLs in datasets might be inaccessible now, which makes the number of images less than the statistic.

B.2. Fine-tuning details

The illustration of fine-tuning MAP on the VL downstream tasks is shown in Figure 6. For different downstream tasks, we just design a simple classifier for understanding tasks. We first sample the point vectors from distribution representations of [CLS]. Then we concatenate point representations from different modalities as global features to conduct classification and apply average pooling operation to all samples' results. The model MAP is trained for 10 epochs. The learning rates of feature extractors, Cross-modal transformer and PDE are $5e-6$, $2.5e-5$, and $2e-4$. In future work, we would like to try applying MAP to do several generation tasks by designing a simple decoder.

B.3. Comparison details

We summarize all referenced modes with model size and pre-training datasets in Table 8. The reported scores in Table 1 and 2 come from their papers. As described in Section 4.2, we introduce the definition of model size [56]. In detail, considering model parameter efficiency, the model size of Vision Language Pre-training (VLP) models can be categorized into at least 3 size: Small, Base, and Large. (1) "Small" indicates the small models prior to the transformer-based VLP models. (2) "Base" indicates the VLP models with similar size to BERT-Base [8]. (3) "Large" is the VLP

Run1	Run2	VQA2.0	SNLI-VE	NLVR2
rand_point	rand_MAP	$p < 0.001$ (-0.193)	$p < 0.001$ (-0.183)	$p < 0.001$ (0.267)
rand_point	pt_point	$p < 0.001$ (-0.211)	$p < 0.001$ (0.028)	$p < 0.001$ (-0.017)
rand_point	pt_MAP	$p < 0.001$ (-0.052)	$p < 0.001$ (0.098)	$p < 0.001$ (0.367)
rand_MAP	pt_point	$p < 0.001$ (-0.018)	$p < 0.001$ (0.211)	$p < 0.001$ (-0.284)
rand_MAP	pt_MAP	$p < 0.001$ (0.141)	$p < 0.001$ (0.280)	$p < 0.001$ (0.100)
pt_point	pt_MAP	$p < 0.001$ (0.159)	$p < 0.001$ (0.070)	$p < 0.001$ (0.384)

Table 9. Statistical significance calculated by Randomized Tukey HSD tests for Table 3 after 1,000 trials. p-value and (effect size) for different tasks.

	MLP	ReLU	ReLU^2	Sigmod
ReLU	$p < 0.001$ (0.385)	-	-	-
ReLU^2	$p < 0.001$ (0.287)	$p < 0.001$ (-0.098)	-	-
Sigmod	$p < 0.001$ (0.162)	$p < 0.001$ (-0.223)	$p < 0.001$ (-0.125)	-
PDE	$p < 0.001$ (-0.151)	$p < 0.001$ (0.234)	$p < 0.001$ (0.136)	$p < 0.001$ (0.011)

Table 10. Statistical significance calculated by Randomized Tukey HSD tests for Table 4 after 1,000 trials. p-value and (effect size).

model with a similar size to BERT-Large. Furthermore, the details of pre-training datasets are presented in Table 7.

B.4. VL downstream tasks

B.4.1 Visual Question Answering

Given an image and a corresponding question, VQA2.0 [12] is the task of providing a correct answer to the question.

B.4.2 NLVR2

The NLVR2 [41] task requires the system to judge whether the corresponding relationship between the description and two images is consistent.

B.4.3 SNLI-VE

SNLI-VE [49] task requires understanding three categories of relationships between images and text, which are entailment, neutral or contradiction.

B.4.4 Image-Text Retrieval

MSCOCO [30] and Filkr30K [35] includes two tasks: Image-to-Text retrieval task and Text-to-Image retrieval task. Both tasks require the model to rank the images or text by computing the image-text similarity scores. In detail, we utilize the Karpathy & Fei-Fei 5K MSCOCO test set and Filkr30K test set and then report the top-K retrieval results.

B.5. Additional results for random initialized MAP

To examine the effectiveness of MAP without extra data, we compare MAP on the popular VL understanding task VQA2.0, with the existing reported methods. As shown in Table 13, MAP achieves a SOTA performance on VQA2.0 among the existing methods without extra data. It shows that PDE can bring multimodal uncertainty knowledge to the models without transferring from large-scale pre-training datasets.

B.6. Comparison between MAP and PCME

PCME [7] is a dual-tower architecture for retrieval, which uses soft contrastive loss with sampled points from distributions. In contrast, Our contrastive loss is based on 2W distance, which directly measures multiple distributions. From a quantization perspective, thanks to pre-training, MAP has a significant boost over PCME. On COCO5k test set, PCME's scores are 44.2/31.9 on I2T/T2I, MAP's scores are 79.3/60.9.

B.7. P-value based on Randomized Tukey HSD tests

In all experiments for our implemented models, p-values were obtained using the randomized Tukey HSD test [37]. The names of runs refer to the related tables. In the experiments, we evaluate the test split in all tasks. Table 10 reports the Randomized Tukey HSD tests for Table 4. In details of Table 9, the name of runs follows the rule: W_M, where W $\in \{\text{rand}, \text{pt}\}$ is random initialization or pre-training and M $\in \{\text{point}, \text{MAP}\}$ is utilizing "MAP w/o PDE" or "MAP". Similarly, Table 11 is the conducted tests for Table 6 with name rules: W_L, where W $\in \{\text{rand}, \text{pt}\}$ and L $\in \{2, 4, 6, 8\}$ is the number of layers of cross-modal transformer in MAP. The tests for Table 5 are shown in Table 12.

B.8. Details and additional examples of visualization

After exacting the distribution representations from PDE, we conduct several 2D toy experiments by using clustering

	rand_2	rand_4	rand_6	rand_8	pt_2	pt_4	pt_6
rand_4	$p < 0.001$ (-0.103)	-	-	-	-	-	-
rand_6	$p < 0.001$ (-0.134)	$p < 0.001$ (-0.031)	-	-	-	-	-
rand_8	$p < 0.001$ (-0.150)	$p < 0.001$ (-0.047)	$p < 0.001$ (-0.016)	-	-	-	-
pt_2	$p < 0.001$ (-0.007)	$p < 0.001$ (0.035)	$p < 0.001$ (0.066)	$p < 0.001$ (0.082)	-	-	-
pt_4	$p = 0.005$ (0.004)	$p < 0.001$ (0.107)	$p < 0.001$ (0.138)	$p < 0.001$ (0.154)	$p < 0.001$ (0.072)	-	-
pt_6	$p < 0.001$ (0.059)	$p < 0.001$ (0.161)	$p < 0.001$ (0.192)	$p < 0.001$ (0.208)	$p < 0.001$ (0.126)	$p < 0.001$ (0.054)	-
pt_8	$p < 0.001$ (0.070)	$p < 0.001$ (0.173)	$p < 0.001$ (0.204)	$p < 0.001$ (0.220)	$p < 0.001$ (0.138)	$p < 0.001$ (0.066)	$p < 0.001$ (0.012)

Table 11. Statistical significance calculated by Randomized Tukey HSD tests for Table 6 after 1,000 trials. p-value and (effect size).

Run1	Run2	VQA2.0	SNLI-VE	NLVR2
rand	MLM_ITM	$p < 0.001$ (0.142)	$p < 0.001$ (0.301)	$p < 0.001$ (0.099)
rand	MLM_VLC	$p < 0.001$ (0.149)	$p < 0.001$ (0.264)	$p < 0.001$ (0.047)
rand	ITM_VLC	$p < 0.001$ (-0.624)	$p < 0.001$ (0.588)	$p < 0.001$ (0.122)
rand	MLM_ITM_VLC	$p < 0.001$ (0.195)	$p < 0.001$ (0.079)	$p < 0.001$ (0.202)
MLM_ITM	MLM_VLC	$p < 0.001$ (0.007)	$p < 0.001$ (-0.038)	$p < 0.001$ (-0.052)
MLM_ITM	ITM_VLC	$p < 0.001$ (-0.765)	$p < 0.001$ (0.286)	$p < 0.001$ (0.023)
MLM_ITM	MLM_ITM_VLC	$p < 0.001$ (0.138)	$p < 0.001$ (-0.222)	$p < 0.001$ (0.103)
MLM_VLC	ITM_VLC	$p < 0.001$ (0.053)	$p < 0.001$ (0.324)	$p < 0.001$ (0.075)
MLM_VLC	MLM_ITM_VLC	$p < 0.001$ (-0.772)	$p < 0.001$ (-0.185)	$p < 0.001$ (0.155)
MLM_VLC	MLM_ITM_VLC	$p < 0.001$ (0.818)	$p < 0.001$ (-0.509)	$p < 0.001$ (0.080)

Table 12. Statistical significance calculated by Randomized Tukey HSD tests for Table 5 after 1,000 trials. p-value and (effect size). MLM, ITM, VLC and rand indicate D-MLM, D-ITM, D-VLC and random initialization respectively.

Model	VQA2.0 (test-dev)
ViLBERT [32]	68.93
MCAN [55]	70.63
UNITER [6]	67.03
METER-swin [9]	72.38
METER-clip-vit [9]	71.75
MAP (ours)	**73.35**

Table 13. Evaluation on VQA2.0 of models with random initialization.

algorithms in machine learning. We utilize the pre-trained MAP with PDE to embed images and text onto distribution representations first. Then, the toy experiments are deployed to find non-linear connections from the input high-dimensional data. In detail, we consider the μ and σ^2 representations in the experiments separately and each experiment calculates more than a thousand image-text pairs. Figures 7, 8, 9 shows several additional visualization examples of the distribution representations in different scenarios[4].

B.9. Visualization between point representations and distribution representations

To explore the differences between representations, we compare our method with ALBEF. For ALBEF (4M), we follow the same method and visualize the features of the same image-sentence pairs (see Fig. 10). Compared to ALBEF, our method takes advantages in capturing rich semantics and concepts in these pairs.

C. Ethical Considerations

Multimodal representation learning is a widely used technique that can have ethical effects. Social bias seems to be rooted in the data due to accumulated biases on the web, such as gender bias in MSCOCO [3]. We believe that our framework could be corrupted, leading to bias concerns, such as having preferences towards certain groups or features. Given that the above problems cover a wide range of issues, such as privacy, fairness, and bias [1, 14], we suggest applying our models to specific contextualization examples. Users should also provide open discussions in their specific research areas and industrial environments.

[4] All images and related captions come from MSCOCO dataset [30].

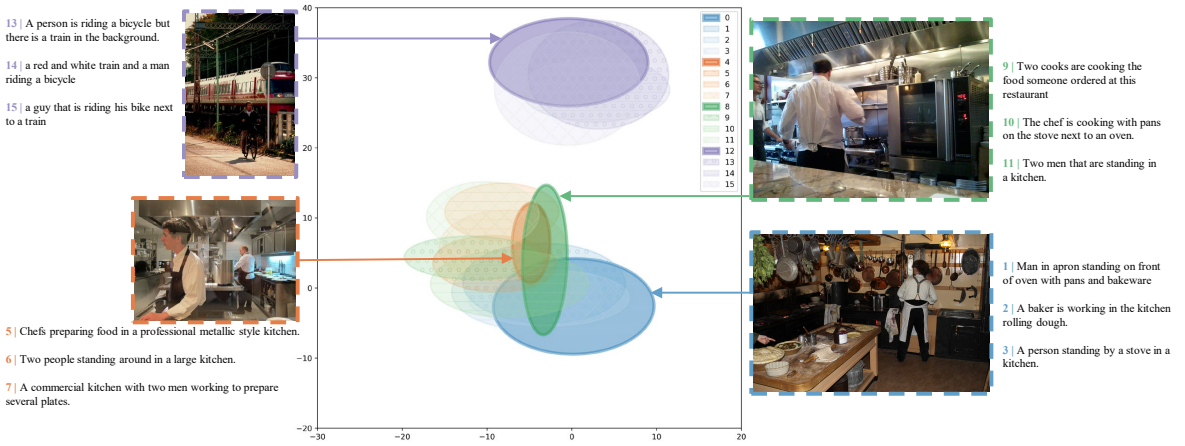

Figure 7. Additional example 1. There are some images and captions of “chef”, “kitchen”, “person”, “bike” and so on.

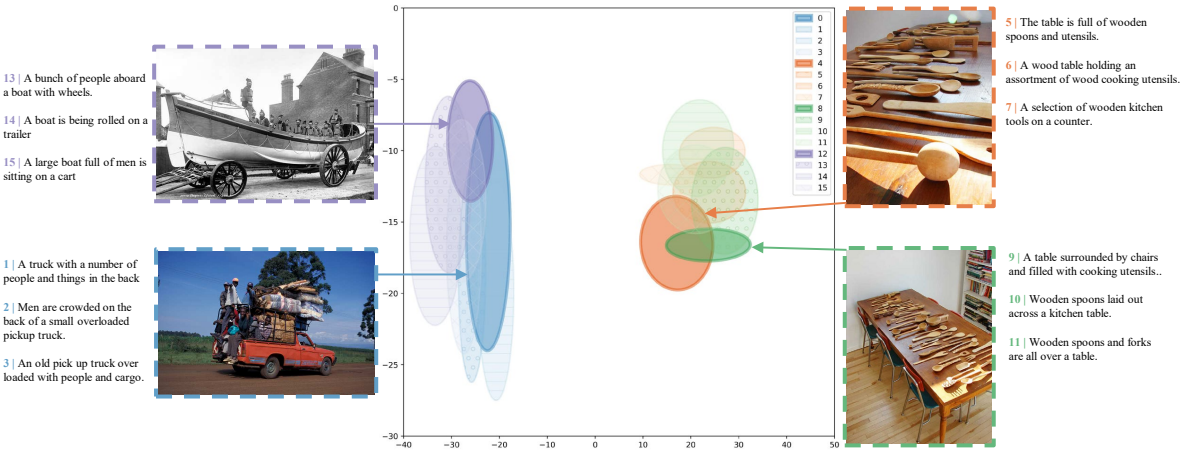

Figure 8. Additional example 2. There are some images and captions of “utensils”, “people”, “truck”, “table” and so on.

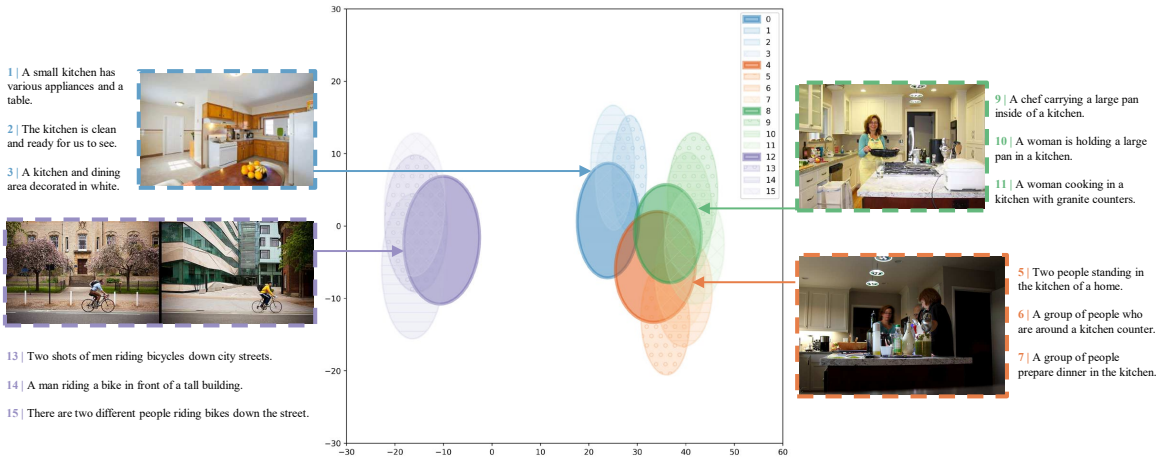

Figure 9. Additional example 3. There are some images and captions of “woman”, “kitchen”, “street”, “bike” and so on.

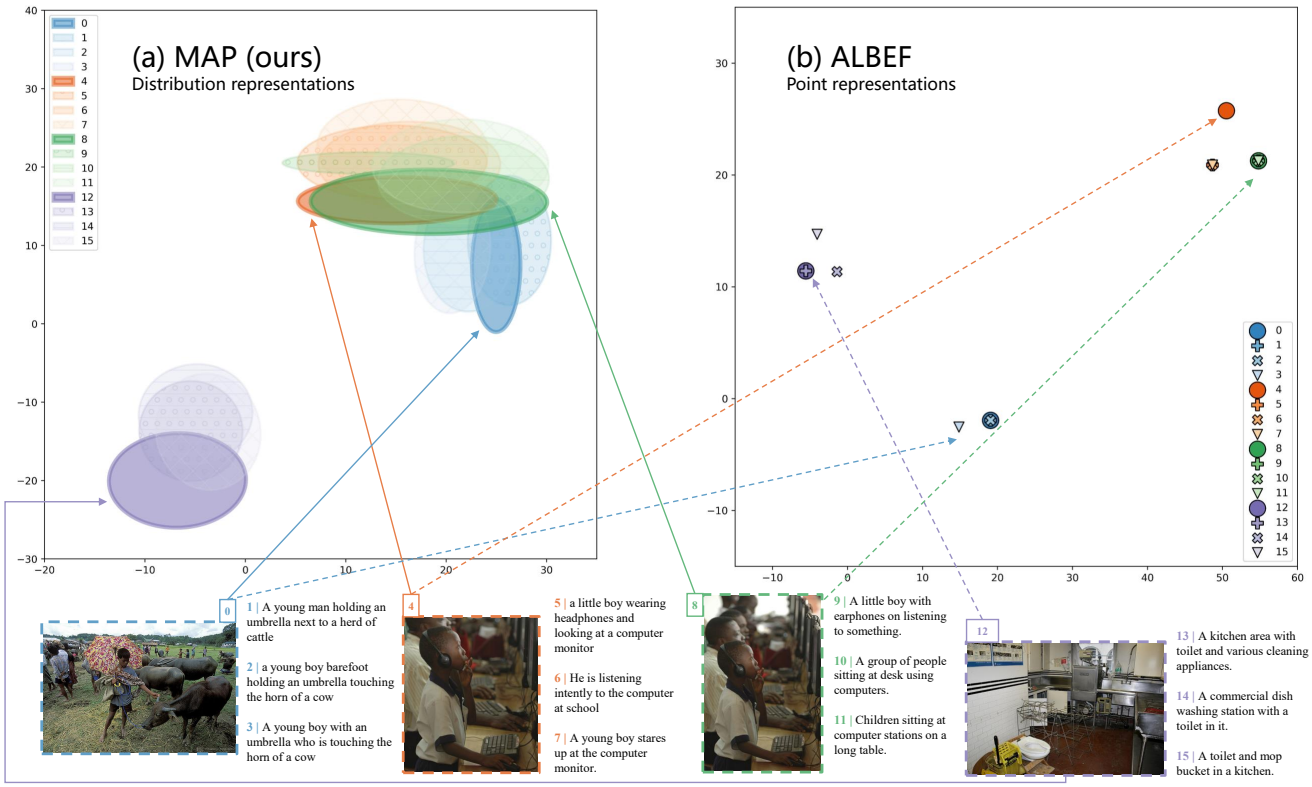

Figure 10. Visualization analysis on distribution representations and point representations.