`ProbVLM`: Probabilistic Adapter for Frozen Vison-Language Models

A. Additional Theoretical Support

We discuss Equation 4 from the main paper and how we simplify the same to obtain a loss function suitable for training deep learning models. Given an image and text embedding pair $(\mathbf{z}_{\mathcal{V}}, \mathbf{z}_{\mathcal{T}})$ (from frozen model) representing similar concepts, the output distributions from $\mathbf{\Psi}(\cdot;\zeta)$, $\mathcal{G}(\mathbf{z};\hat{\mathbf{z}}_{\mathcal{V}},\hat{\alpha}_{\mathcal{V}},\hat{\beta}_{\mathcal{V}})$ and $\mathcal{G}(\mathbf{z};\hat{\mathbf{z}}_{\mathcal{T}},\hat{\alpha}_{\mathcal{T}},\hat{\beta}_{\mathcal{T}})$ (later referred to as $\mathcal{G}_{\mathcal{V}}(\mathbf{z})$) and $\mathcal{G}_{\mathcal{T}}(\mathbf{z})$) should match. This can be measured directly from the likelihood as, $p(\mathbf{z}_v = \mathbf{z}_u)$, where $\mathbf{z}_v \sim \mathcal{G}_{\mathcal{V}}(\mathbf{z})$ and $\mathbf{z}_u \sim \mathcal{G}_{\mathcal{T}}(\mathbf{z})$ as in [1] , i.e.,

$$p(\mathbf{z}_v = \mathbf{z}_u) := \iint \mathcal{G}_{\mathcal{V}}(\mathbf{z}_v)\mathcal{G}_{\mathcal{T}}(\mathbf{z}_u)\delta(\mathbf{z}_v - \mathbf{z}_u)d\mathbf{z}_v d\mathbf{z}_u \quad (1)$$

where $\delta(\cdot)$ refers to the *Dirac-delta distribution*. The above integral can be simplified further by defining $\Delta\mathbf{z} = \mathbf{z}_v - \mathbf{z}_u$ and seeking $p(\Delta\mathbf{z}) = 0$. As both $\mathbf{z}_v$ and $\mathbf{z}_u$ are GGD random variables, $\Delta\mathbf{z}$ follows the distribution based on the *Bivariate Fox H-function* [2] given by,

$$\Delta z \sim \frac{1}{2\Gamma(1/\hat{\beta}_{\mathcal{V}}),\Gamma(1/\hat{\beta}_{\mathcal{T}})} \times$$
$$\int \mathcal{H}^{1,1}_{1,2}\left[At^2 \middle| \begin{matrix}(1-\frac{1}{\hat{\mathbf{z}}_{\mathcal{V}}},\frac{1}{\hat{\mathbf{z}}_{\mathcal{T}}})\\(0,1)(\frac{1}{2},1)\end{matrix}\right] \mathcal{H}^{1,1}_{1,2}\left[Bt^2 \middle| \begin{matrix}(1-\frac{1}{\hat{\mathbf{z}}_{\mathcal{T}}},\frac{1}{\hat{\mathbf{z}}_{\mathcal{T}}})\\(0,1)(\frac{1}{2},1)\end{matrix}\right] \cos t(\mu - z)dt \quad (2)$$

Where $A = \frac{\hat{\alpha}^2_{\mathcal{V}}\Gamma(1/\hat{\beta}_{\mathcal{V}})}{4\Gamma(3/\hat{\beta}_{\mathcal{V}})}$, $B = \frac{\hat{\alpha}^2_{\mathcal{T}}\Gamma(1/\hat{\beta}_{\mathcal{T}})}{4\Gamma(3/\hat{\beta}_{\mathcal{T}})}$, $\mu = \hat{\mathbf{z}}_v - \hat{\mathbf{z}}_u$, and $\mathcal{H}$ is the *Fox H function* [2] given by,

$$H^{m,n}_{p,q}\left[z \,\middle|\, \begin{matrix}(a_1,A_1) & (a_2,A_2) & \ldots & (a_p,A_p)\\(b_1,B_1) & (b_2,B_2) & \ldots & (b_q,B_q)\end{matrix}\right]$$
$$= \frac{1}{2\pi i}\int_L \frac{\prod_{j=1}^{m}\Gamma(b_j+B_j s)\prod_{j=1}^{n}\Gamma(1-a_j-A_j s)}{\prod_{j=m+1}^{q}\Gamma(1-b_j-B_j s)\prod_{j=n+1}^{p}\Gamma(a_j+A_j s)} z^{-s} ds \quad (3)$$

Equation 2 does not provide a scalable objective function suitable for training deep neural networks. Hence, we propose an approximation that is easily scalable for deep-learning models given by,

$$p(\mathbf{z}_v = \mathbf{z}_u) = \iint \mathcal{G}_{\mathcal{V}}(\mathbf{z}_v)\mathcal{G}_{\mathcal{T}}(\mathbf{z}_u)\delta(\mathbf{z}_v - \mathbf{z}_u)d\mathbf{z}_v d\mathbf{z}_u$$
$$\approx \int \frac{1}{2}\left(\mathcal{G}_{\mathcal{V}}(\mathbf{z})\delta(\mathbf{z}-\mathbf{z}_{\mathcal{T}}) + \mathcal{G}_{\mathcal{T}}(\mathbf{z})\delta(\mathbf{z}-\mathbf{z}_{\mathcal{V}})\right)d\mathbf{z} \quad (4)$$

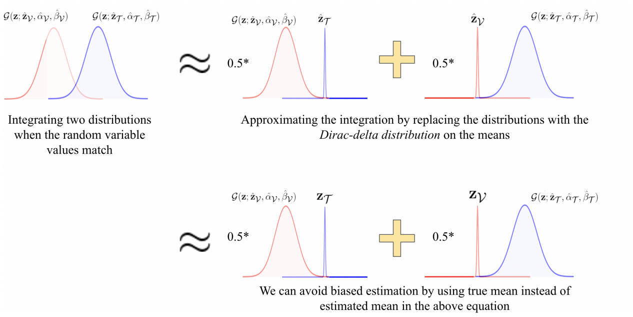
$\mathcal{G}(\mathbf{z};\hat{\mathbf{z}}_{\mathcal{V}},\hat{\alpha}_{\mathcal{V}},\hat{\beta}_{\mathcal{V}})$
$\mathcal{G}(\mathbf{z};\hat{\mathbf{z}}_{\mathcal{T}},\hat{\alpha}_{\mathcal{T}},\hat{\beta}_{\mathcal{T}})$
$\approx$
0.5*
$\mathcal{G}(\mathbf{z};\hat{\mathbf{z}}_{\mathcal{V}},\hat{\alpha}_{\mathcal{V}},\hat{\beta}_{\mathcal{V}})$
$\hat{\mathbf{z}}_{\mathcal{T}}$
0.5*
$\hat{\mathbf{z}}_{\mathcal{V}}$
$\mathcal{G}(\mathbf{z};\hat{\mathbf{z}}_{\mathcal{T}},\hat{\alpha}_{\mathcal{T}},\hat{\beta}_{\mathcal{T}})$
Integrating two distributions when the random variable values match
Approximating the integration by replacing the distributions with the *Dirac-delta distribution* on the means
$\approx$
0.5*
$\mathcal{G}(\mathbf{z};\hat{\mathbf{z}}_{\mathcal{V}},\hat{\alpha}_{\mathcal{V}},\hat{\beta}_{\mathcal{V}})$
$\mathbf{z}_{\mathcal{T}}$
0.5*
$\mathbf{z}_{\mathcal{V}}$
$\mathcal{G}(\mathbf{z};\hat{\mathbf{z}}_{\mathcal{T}},\hat{\alpha}_{\mathcal{T}},\hat{\beta}_{\mathcal{T}})$
We can avoid biased estimation by using true mean instead of estimated mean in the above equation

Figure 1: Visualizing the approximation in Equation 4.

To understand the above approximation, we refer to Figure 1. We notice that the integral in Equation 1 tries to convolve the two distribution, with an additional constraint of those distributions being equal in value. While convolving the two generalized gaussian distributions is hard, Figure 1 shows that a rough approximation for the same is to convolve a generalized gaussian distribution with the Dirac-delta distribution. Further, instead of using the estimated means from `ProbVLM` in the Dirac-delta distribution (that are to be near-perfect reconstructions of the embeddings obtained from the frozen network), we use the embeddings from the frozen encoders as shown in Figure 1. This finally leads to Equation 4. The first term in the integral, $\int \mathcal{G}_{\mathcal{V}}(\mathbf{z})\delta(\mathbf{z}-\mathbf{z}_{\mathcal{T}})d\mathbf{z}$, is the likelihood of the text embedding $\mathbf{z}_{\mathcal{T}}$ under the predicted distribution, $\mathcal{G}_{\mathcal{V}}(\mathbf{z})$, for the visual embedding. Similarly, the second term is the likelihood of the visual embedding $\mathbf{z}_{\mathcal{V}}$ under the predicted distribution, $\mathcal{G}_{\mathcal{T}}(\mathbf{z})$, for the text embedding. Negative log of Equation 4 leads to a scalable objective function that can be used to learn the optimal parameters for vision and text components of `ProbVLM` ($\mathbf{\Psi}_{\mathcal{V}}(\cdot;\zeta_{\mathcal{V}})$ and $\mathbf{\Psi}_{\mathcal{T}}(\cdot;\zeta_{\mathcal{T}})$),

$$L_{\text{cross}}(\zeta_{\mathcal{V}},\zeta_{\mathcal{T}}) := \underbrace{\left(\frac{|\hat{\mathbf{z}}_{\mathcal{V}}-\mathbf{z}_{\mathcal{T}}|}{\hat{\alpha}_{\mathcal{V}}}\right)^{\hat{\beta}_{\mathcal{V}}} - \log\frac{\hat{\beta}_{\mathcal{V}}}{\hat{\alpha}_{\mathcal{V}}} + \log\Gamma(\frac{1}{\hat{\beta}_{\mathcal{V}}})}_{\text{Cross-modal: vision}\rightarrow\text{text}} +$$
$$\underbrace{\left(\frac{|\hat{\mathbf{z}}_{\mathcal{T}}-\mathbf{z}_{\mathcal{V}}|}{\hat{\alpha}_{\mathcal{T}}}\right)^{\hat{\beta}_{\mathcal{T}}} - \log\frac{\hat{\beta}_{\mathcal{T}}}{\hat{\alpha}_{\mathcal{T}}} + \log\Gamma(\frac{1}{\hat{\beta}_{\mathcal{T}}})}_{\text{Cross-modal: text}\rightarrow\text{vision}} \quad (5)$$

In practice, the exponential of β in the above equation often makes training unstable. To make it more stable, we

		Datasets											
		CUB			**Flowers**			**Flickr**			**COCO**		
M		R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
V-B32	**i2t**	35.3	64.9	79.3	54.5	84.7	94.0	79.0	94.7	97.1	50.6	75.0	83.6
		85.1	89.4	81.9	53.3	55.2	37.2	64.2	61.0	55.1	61.0	62.3	57.2
		92.1	95.0	90.1	69.6	70.6	52.3	77.0	73.6	68.8	75.8	76.5	73.3
	t2i	16.0	34.4	44.6	25.5	47.8	61.8	56.5	82.2	88.3	30.1	55.7	66.8
		63.9	63.0	60.5	37.3	33.5	31.7	36.2	35.5	35.1	35.9	36.9	35.4
		72.8	71.8	70.7	47.4	43.3	43.7	47.5	46.9	46.7	47.2	49.3	47.8
V-B16	**i2t**	34.2	66.2	80.4	52.1	82.8	91.6	82.7	96.2	98.9	53.0	77.1	85.1
		85.1	89.4	81.9	53.3	55.2	37.2	64.2	61.0	55.1	61.0	62.3	57.2
		92.1	95.0	90.1	69.6	70.6	52.3	77.0	73.6	68.8	75.8	76.5	73.3
	t2i	15.0	33.3	44.1	25.4	46.4	57.9	61.0	84.2	89.6	33.3	58.6	68.9
		63.9	63.0	60.5	37.3	33.5	31.7	36.2	35.5	35.1	35.9	36.9	35.4
		72.8	71.8	70.7	47.4	43.3	43.7	47.5	46.9	46.7	47.2	49.3	47.8
RN-50	**i2t**	31.1	61.7	75.9	53.0	87.1	95.0	77.7	95.2	97.3	49.1	72.5	81.8
		85.1	89.4	81.9	53.3	55.2	37.2	64.2	61.0	55.1	61.0	62.3	57.2
		92.1	95.0	90.1	69.6	70.6	52.3	77.0	73.6	68.8	75.8	76.5	73.3
	t2i	15.3	35.0	46.5	31.5	54.3	66.7	55.1	81.2	87.9	28.3	53.1	64.3
		63.9	63.0	60.5	37.3	33.5	31.7	36.2	35.5	35.1	35.9	36.9	35.4
		72.8	71.8	70.7	47.4	43.3	43.7	47.5	46.9	46.7	47.2	49.3	47.8

Table 1: Zero-shot performance on COCO, Flickr, CUB and FLO with for both Image-to-Text (i2t) and Text-to-Image (t2i) Retrieval for CLIP Models (M) with Vision Transformer (V-B32, V-B16) and ResNet (RN-50) backbones.

		CLIP backbones fine-tuned on											
		CUB			**Flowers**			**Flickr**			**COCO**		
D		V-B32	V-B16	RN-50	V-B32	V-B16	RN-50	V-B32	V-B16	RN-50	V-B32	V-B16	RN-50
CUB	**i2t**	**58.8**	**66.1**	**53.9**	25.2	23.8	13.4	32.4	31.1	26.2	31.5	32.5	26.8
	t2i	**41.3**	**42.3**	**37.4**	18.4	16.8	13.1	16.6	17.1	16.1	16.6	16.9	14.3
Flowers	**i2t**	54.5	51.1	44.3	**80.7**	**82.0**	**73.8**	49.5	55.2	49.7	47.9	47.2	43.6
	t2i	25.5	31.2	29.6	**57.8**	**59.0**	**53.3**	31.3	29.3	30.8	28.7	29.2	31.7
Flickr	**i2t**	68.9	73.5	48.2	51.4	62.4	24.4	**90.0**	**92.7**	**87.1**	86.7	90.2	87.7
	t2i	48.6	54.7	31.4	32.3	40.5	17.0	**73.4**	**77.5**	**68.3**	69.9	74.5	68.7
COCO	**i2t**	32.6	42.6	22.0	24.8	31.8	8.9	56.9	61.5	52.0	**73.4**	**69.5**	**64.3**
	t2i	19.5	27.1	12.5	32.3	19.7	6.8	38.7	43.9	33.0	**49.8**	**52.3**	**45.3**

Table 2: Result for fine-tuning CLIP on different Datasets (D) for Image-to-Text (i2t) and Text-to-Image (t2i) retrieval.

make use of the Taylor-series expansion and note that

$$\left(\frac{|\hat{\mathbf{z}}-\mathbf{z}|}{\hat{\alpha}}\right)^{\hat{\beta}} = \left(1 + (\frac{|\hat{\mathbf{z}}-\mathbf{z}|}{\hat{\alpha}} - 1)\right)^{\hat{\beta}} \approx 1 - \hat{\beta} + \hat{\beta}(\frac{|\hat{\mathbf{z}}-\mathbf{z}|}{\hat{\alpha}}) \qquad (6)$$

This way, the variable $\hat{\beta}$ no longer in exponent and as a result loss becomes more stable during optimization.

B. Additional Quantitative Experiments

We provide the zero-shot results for the CLIP model trained with different visual backbones in Table 1, while the results after fine-tuning are presented in Table 2. While Zero-Shot CLIP achieves promising results on all 4 datasets, these are much worse when compared to the results obtained when fine-tuning on the desired target dataset (42.3 vs. 15.0 for a ViT B/16 on CUB t2i R@1). However, this comes at the cost of worse performance on the remaining datasets due to catastrophic forgetting and has to be mitigated via several strategies.

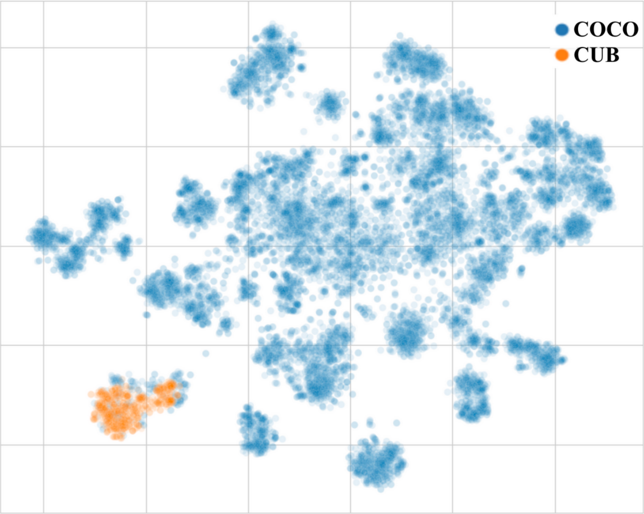

Figure 2: tSNE plot for MS-COCO and CUB image embeddings illustrating the diversity of MS-COCO.

Figure 2 shows the tSNE plots for the CLIP embeddings obtained from a relatively diverse dataset (e.g., COCO) compared to a niche dataset (e.g., CUB consisting of only birds). As indicated in the plot, a niche dataset will likely not be able to capture all the representations spread in the embedding space, leading to poor generalization, as shown in Table 2. This is because CUB has images that only contain birds, whereas COCO is a much larger datasets containing 80 different object categories (including birds). Therefore fine-tuning either the VLM or `ProbVLM` on a larger, more diverse dataset such as COCO would lead to better generalization, and trasnferrability across datasets.

C. Implementation Details and Code

Stable Diffusion interpretation. The U-Net based decoder used by Stable Diffusion takes the pre-final layer of the CLIP text encoder as input, which expects the input to be in a shape *tokens* x *features* However, the usual training of `ProbVLM` takes the pooled output from the text encoder to enable cross-modal alignment with the vision encoder. For this experiment, we re-train `ProbVLM` to operate on the pre-final layer without the cross-modal alignment.

References

[1] Yichun Shi and Anil K Jain. Probabilistic face embeddings. In *ICCV*, 2019. 1

[2] Hamza Soury and Mohamed-Slim Alouini. New results on the sum of two generalized gaussian random variables. In *GlobalSIP*, 2015. 1