

APPENDIX

A COMPARISON TO PMI^k

By assuming $P_{test}(\mathbf{t})$ to be a “flatten” version of $P_{train}(\mathbf{t})$, our Equation 7 can interpolate between scenario 1 (same train and test priors) and 2 (balanced test priors):

$$P_{test}(\mathbf{t}) \propto P_{train}(\mathbf{t})^{1-\alpha} \Rightarrow \text{Optimal score is } \frac{P_{train}(\mathbf{t}|\mathbf{i})}{P_{train}(\mathbf{t})^\alpha} \quad (11)$$

In fact, the above equation can be rewritten using the language of PMI^k (Role & Nadif, 2011; Daille, 1994), a well-known variant of PMI that controls the amount of debiasing (Li et al., 2016; Li & Jurafsky, 2016; Wang et al., 2020) in information retrieval:

$$\frac{P_{train}(\mathbf{t}|\mathbf{i})}{P_{train}(\mathbf{t})^\alpha} = \frac{P_{train}(\mathbf{t}, \mathbf{i})}{P_{train}(\mathbf{i})P_{train}(\mathbf{t})^\alpha} \quad (12)$$

$$\propto \frac{P_{train}(\mathbf{t}, \mathbf{i})^{\frac{1}{\alpha}}}{P_{train}(\mathbf{i})P_{train}(\mathbf{t})} \quad , \text{ as } P_{train}(\mathbf{i}) \text{ is constant in I-to-T} \quad (13)$$

$$= \text{pmi}_{P_{train}}^k(\mathbf{t}, \mathbf{i}), \text{ where } k = \frac{1}{\alpha} \geq 1 \quad (14)$$

where

$$\text{pmi}_P(\mathbf{t}, \mathbf{i}) = \frac{P(\mathbf{t}, \mathbf{i})}{P(\mathbf{t})P(\mathbf{i})} = \frac{P(\mathbf{t}|\mathbf{i})}{P(\mathbf{t})} = \frac{P(\mathbf{i}|\mathbf{t})}{P(\mathbf{i})} \quad (15)$$

PMI is an information-theoretic measure that quantifies the *association* between two variables (Yao et al., 2010; Henning & Ewerth, 2017; Shrivastava et al., 2021). In the context of image-text retrieval, it measures how much more (or less) likely the image-text pair co-occurs than if the two were independent. Eq. 15 has found applications in diverse sequence-to-sequence modelling tasks (Wang et al., 2020; Li & Jurafsky, 2016; Li et al., 2016) as a retrieval (reranking) objective. Compared to the conditional likelihood $P(\mathbf{t}|\mathbf{i})$, PMI reduces the learned bias for preferring “common” texts with high marginal probabilities $P(\mathbf{t})$ (Li et al., 2016; Li & Jurafsky, 2016; Wang et al., 2020). This can be an alternative explanation for the effectiveness of our debiasing solutions.

B ABLATION STUDIES ON α -TUNING

Estimating $P_{train}(\mathbf{t})$ via null (Gaussian noise) images is more sample-efficient. We use Winoground to show that sampling Gaussian noise images to calculate $P_{train}(\mathbf{t})$ can be more efficient than sampling trainset images. As demonstrated in Table 4, a limited number of Gaussian noise images (e.g., 3 or 10) can surpass the results obtained with 1000 LAION images. Moreover, using null images produces less variance in the results.

Sample Size	Gaussian Noise Images		Trainset Images	
	$\alpha=\alpha_{test}^*$	α_{test}^*	$\alpha=\alpha_{test}^*$	α_{test}^*
3	35.95 _(0.5)	0.821 _(0.012)	32.20 _(1.6)	0.706 _(0.150)
10	36.25 _(0.4)	0.827 _(0.016)	33.60 _(0.9)	0.910 _(0.104)
100	36.35 _(0.1)	0.840 _(0.010)	34.70 _(0.6)	0.910 _(0.039)
1000	36.25 _(0.0)	0.850 _(0.000)	35.15 _(0.3)	0.960 _(0.033)

Table 4: **Comparing sampling of Gaussian noise images and trainset images for estimating $P_{train}(\mathbf{t})$.** We report text scores of α -tuning on Winoground I-to-T retrieval task. We ablate 3/10/100/1000 Gaussian noise and LAION samples and report both mean and std using 5 sampling seeds. The optimal $\alpha^* \in [0, 1]$ is searched on testset via a step size of 0.001. The Gaussian noise images are sampled with a mean calculated from the LAION subset and a fixed std of 0.25.

Details of Gaussian noise samples. Unless otherwise specified, the Gaussian noise images are sampled with a mean of 1.0 and a standard deviation of 0.25. By default, we use 100 images for Winoground, 30 images for EqBen, and 3 images for the rest of the benchmarks. We also fix the

sampling seed in our code to ensure reproducibility. We leave more advanced techniques of generating null images to future works.

Alternative approach on COCO/Flickr30k: estimating $P_{train}(t)$ using testset images. For large-scale retrieval benchmarks like COCO (Lin et al., 2014) and Flickr30k (Young et al., 2014), we can directly average scores of all candidate images (in the order of thousands) to efficiently approximate $P_{train}(t)$ without the need to sample additional images. This approach incurs zero computation cost as we have already pre-computed scores between each candidate image and text. We show in Table 5 that using testset images indeed results in better performance than sampling 3 Gaussian noise images.

Metric	Benchmark	$P_{train}(t i)$	Sampling Method	$\frac{P_{train}(t i)}{P_{train}(t)^\alpha}$		
				$\alpha=1$	$\alpha=\alpha_{val}^*$	α_{val}^*
R@1 / R@5	COCO	19.7 / 40.6	Testset Images	46.2 / 73.1	48.0 / 74.2	0.819
			Null Images	24.4 / 52.6	40.4 / 66.6	0.600
	Flickr30k	34.6 / 59.0	Testset Images	58.7 / 88.0	63.6 / 89.2	0.719
			Null Images	27.8 / 62.2	48.5 / 79.0	0.427

Table 5: **I-to-T retrieval on COCO/Flickr30k using different sampling methods.** Estimating $P_{train}(t)$ by averaging the scores of testset images (with zero computational cost) demonstrates superior performance compared to sampling additional Gaussian noise images.

Tuning α with a validation set. In Table 6 similar performance trends are observed across validation and test splits of COCO and Flickr30k I-to-T retrieval benchmarks using the same $\alpha \in [0, 1]$. Furthermore, α_{test}^* and α_{val}^* are empirically close. As such, our method can function as a reliable training-free debiasing method. Future studies may explore fine-tuning methods to further improve the debiasing performance.

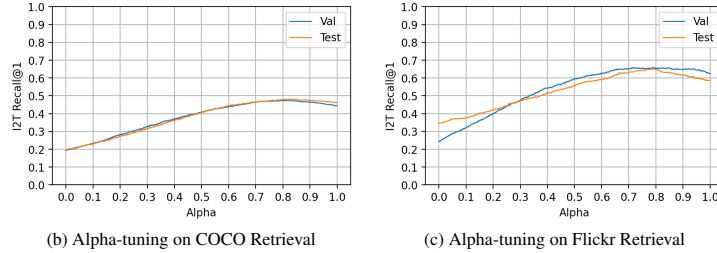


Table 6: **α -tuning results on both val set and test set for COCO/Flickr30k I-to-T retrieval.** We observe that validation and test performance are strongly correlated while we interpolate $\alpha \in [0, 1]$.

C IS VISUALGPTSCORE A BIASED ESTIMATOR OF $P_{train}(t|i)$?

Retrieval performance on trainset (LAION). This paper is built on the assumption that VisualGPTScore is a reliable estimator of $P_{train}(t|i)$. However, this simplifying assumption does not completely hold for the BLIP model we examine. We speculate that such OTS generative scores are biased towards more common texts. We witness this same phenomenon in Table 7 where we perform image-text retrieval on random subsets from training distribution LAION-114M (Li et al., 2022).

Modelling the language bias in VisualGPTScore. As evidenced in Table 7 we believe VisualGPTScore is biased towards more common texts due to modelling error. To consider this error in our analysis, we rewrite the VisualGPTScore as:

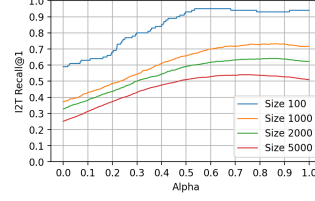
$$\text{VisualGPTScore}(t, i) := \hat{P}_{train}(t|i) = P_{train}(t|i) \cdot P_{train}(t)^\beta, \quad (16)$$

where \hat{P} represents the (biased) model estimate and P represents the true distribution. The model bias towards common texts is encoded by an unknown parameter β .

Monte Carlo estimation using \hat{P} . Because our Monte Carlo sampling method relies on $\hat{P}_{train}(t|i)$, it is also a biased estimator of $P_{train}(t)$:

Dataset Size	I-to-T Retrieval					T-to-I Retrieval	
	ITM	$\frac{P_{train}(\mathbf{t} \mathbf{i})}{P_{train}(\mathbf{t})^\alpha}$				ITM	$P_{train}(\mathbf{t} \mathbf{i})$
		$\alpha=0$	$\alpha=1$	$\alpha=\alpha^*$	α^*		
100	96.0	59.0	94.0	95.0	0.535	95.0	97.0
1000	90.9	37.1	71.7	85.7	0.733	92.0	93.1
2000	87.2	32.8	62.3	64.3	0.840	87.8	89.8
5000	79.8	25.1	50.9	54.1	0.727	81.9	84.4

(a) Performance on LAION trainset retrieval



(b) Alpha-tuning on LAION

Table 7: **Retrieval performance on randomly sampled LAION114M subsets with varied sizes.** Table (a) shows that while OTS generative scores are robust for T-to-I retrieval, its performance degrades on I-to-T retrieval tasks when the number of candidate texts increases. This implies that OTS generative scores suffer from language biases towards certain texts even in the training set. Nonetheless, we show that our debiasing solution using either $\alpha = 1$ or optimal $\alpha^* \in [0, 1]$ with a step size of 0.001, can consistently boost the performance. Figure (b) visualizes α -tuning results on LAION subsets, where each curve represents a different sample size.

$$\hat{P}_{train}(\mathbf{t}) := \frac{1}{n} \sum_{k=1}^n \hat{P}_{train}(\mathbf{t}|\mathbf{i}_k) = P_{train}(\mathbf{t})^{1+\beta}. \quad (17)$$

Rewriting optimal I-to-T objective with \hat{P} . We can rewrite Equation 4 as:

$$P_{test}(\mathbf{t}|\mathbf{i}) \propto P_{train}(\mathbf{t}|\mathbf{i}) \frac{P_{test}(\mathbf{t})}{P_{train}(\mathbf{t})} \quad (18)$$

$$= \hat{P}_{train}(\mathbf{t}|\mathbf{i}) \frac{P_{test}(\mathbf{t})}{P_{train}(\mathbf{t})^{1+\beta}} \quad (19)$$

$$= \hat{P}_{train}(\mathbf{t}|\mathbf{i}) \frac{P_{test}(\mathbf{t})}{\hat{P}_{train}(\mathbf{t})} \quad (20)$$

α -tuning with \hat{P} . Using Equation 20, we can reformulate α -tuning (Equation 7) as follows:

$$P_{test}(\mathbf{t}) \propto P_{train}(\mathbf{t})^{1-\alpha} \Rightarrow \text{Optimal score is } \frac{\hat{P}_{train}(\mathbf{t}|\mathbf{i})}{\hat{P}_{train}(\mathbf{t})^\alpha} \quad (21)$$

where $\alpha = \frac{\hat{\alpha}+\beta}{1+\beta}$. Notably, the above equation has the same structure as before (Equation 7). This implies that even if $P_{train}(\mathbf{t}) = P_{test}(\mathbf{t})$, we still anticipate $\alpha = \frac{\beta}{1+\beta} \neq 0$. This accounts for why the optimal α is not 0 when we perform I-to-T retrieval on trainset in Table 7.

Implication for vision-language modelling. Our analysis indicates that similar to generative LLMs (Li et al., 2016; Li & Jurafsky, 2016), contemporary image-conditioned language models also experience issues related to imbalanced learning (Kang et al., 2019). Potential solutions could be: (a) refined sampling techniques for Monte Carlo estimation of $P(\mathbf{t})$ such as through dataset distillation (Wu et al., 2023), and (b) less biased modelling of $P(\mathbf{t}|\mathbf{i})$ such as through controllable generation (Keskar et al., 2019).

D EXPERIMENTS WITH BLIP-2

We provide BLIP-2 results for completeness.

BLIP-2 (Li et al., 2023) overview. BLIP-2 leverages frozen pre-trained image encoders (Fang et al., 2022) and large language models (Chung et al., 2022; Zhang et al., 2022) to bootstrap vision-language pre-training. It proposes a lightweight Querying Transformer (Q-Former) that is trained in two stages. Similar to BLIP (Li et al., 2022), Q-Former is a mixture-of-expert model that can calculate ITC, ITM, and captioning loss given an image-text pair. Additionally, it introduces a set of trainable query tokens, whose outputs serve as *visual soft prompts* prepended as inputs to LLMs. In its first training stage, Q-Former is fine-tuned on the same LAION dataset using the same objectives

(ITC+ITM+captioning) as BLIP. In the second stage, the output query tokens from Q-Former are fed into a frozen language model, such as FLAN-T5 (Chung et al., 2022) or OPT (Chung et al., 2022), after a linear projection trained only with captioning loss. BLIP-2 achieves state-of-the-art performance on various vision-language tasks with significantly fewer trainable parameters.

BLIP-2 results. We present retrieval performance of the BLIP-2 model that uses ViT-L as the frozen image encoder. We report results for both the first-stage model (denoted as Q-Former) and the second-stage model which employs FLAN-T5 (Chung et al., 2022) as the frozen LLM.

Benchmark	Dataset	Random	w. Q-Former			w. Flan-T5
			ITC	ITM	$P_{train}(t i)$	$P_{train}(t i)$
ARO	VG-Relation	50.0	46.4	67.2	90.7	89.1
	VG-Attribution	50.0	76.0	88.1	94.3	90.9
	COCO-Order	20.0	28.5	25.2	96.8	99.3
	Flickr30K-Order	20.0	25.3	28.6	97.5	99.7
Crepe	Atom-Foils	16.7	20.8	20.9	74.7	69.7
	Negate	16.7	13.4	14.2	79.1	90.0
	Swap	16.7	13.4	18.0	79.5	79.1
VL-CheckList	Object	50.0	89.7	89.2	90.1	84.1
VL-CheckList	Attribute	50.0	76.6	79.3	73.9	70.6
VL-CheckList	Relation	50.0	70.5	72.3	89.9	56.7
SugarCrepe	Replace	50.0	86.7	88.5	93.0	82.4
SugarCrepe	Swap	50.0	69.8	80.9	91.2	80.8
SugarCrepe	Add	50.0	86.5	88.0	92.7	76.2

Table 8: BLIP-2 on ARO/Crepe/VL-CheckList/SugarCrepe.

Benchmark	Model	I-To-T (Text Score)						T-To-I (Image Score)		
		ITC	ITM	$\frac{P_{train}(t i)}{P_{train}(t)^{\alpha}}$				ITC	ITM	$P_{train}(t i)$
				$\alpha=0$	$\alpha=1$	$\alpha=\alpha^*$	α^*			
Winoground	BLIP	28.0	35.8	27.0	33.0	36.5	0.836	9.0	15.8	21.5
	BLIP2-QFormer	30.0	42.5	24.3	29.3	33.0	0.882	10.5	19.0	20.0
	BLIP2-FlanT5	-	-	25.3	31.5	34.3	0.764	-	-	19.5
EqBen (Val)	BLIP	20.9	26.0	9.6	19.8	19.8	0.982	20.3	20.3	26.1
	BLIP2-QFormer	32.1	36.2	12.2	21.9	22.2	0.969	23.4	28.4	26.6
	BLIP2-FlanT5	-	-	8.5	22.0	22.0	1.000	-	-	20.9

Table 9: BLIP-2 on Winoground/EqBen.

E ADDITIONAL REPORTS

Computational resources. All experiments use a single NVIDIA GeForce 3090s GPU.

Details of Table 1. For CLIP, LAION2B-CLIP, and LAION5B-CLIP, we report the results from Hsieh et al. (2023) using the ViT-B-32, ViT-bigG-14, and xlm-roberta-large-ViT-H-14 models respectively. The results of NegCLIP, Structure-CLIP, SVLC, and xgVL, DAC-LLM, and DAC-SAM are directly copied from their original papers. We run BLIP-ITC and BLIP-ITM using our own codebase, which will be released to the public.

Group scores on Winoground/EqBen using BLIP (Table 10).

Method	Winoground			EqBen		
	Text Score	Image Score	Group Score	Text Score	Image Score	Group Score
ITCScore	28.0	9.0	6.5	20.9	20.3	10.6
ITMScore	35.8	15.8	13.3	26.0	20.3	12.6
VisualGPTScore α^*	36.5	21.5	16.8	20.4	26.1	11.7

Table 10: Performance comparison of BLIP’s ITCScore, ITMScore, and α -tuned VisualGPTScore α^* on Winoground (all) and EqBen (val).

Fine-grained tags on Winoground (Table 11).

Performance on SugarCrepe (Table 12).

Dataset	Size	Method	Text Score	Image Score	Group Score
NoTag	171	ITCScore	32.6	11.6	8.1
		ITMScore	41.9	21.5	19.2
		VisualGPTScore α^*	43.0	28.5	23.8
NonCompositional	30	ITCScore	43.3	16.7	16.7
		ITMScore	50.0	23.3	16.7
		VisualGPTScore α^*	43.3	33.3	26.7
AmbiguouslyCorrect	46	ITCScore	32.6	8.7	6.5
		ITMScore	28.3	6.5	2.2
		VisualGPTScore α^*	26.1	19.6	8.7
VisuallyDifficult	38	ITCScore	29.0	7.9	7.9
		ITMScore	26.3	10.5	7.9
		VisualGPTScore α^*	31.6	13.2	7.9
UnusualImage	56	ITCScore	32.5	8.9	8.9
		ITMScore	21.4	10.7	7.1
		VisualGPTScore α^*	30.4	10.7	8.9
UnusualText	50	ITCScore	20.0	8.0	6.0
		ITMScore	38.0	12.0	12.0
		VisualGPTScore α^*	30.0	18.0	12.0
ComplexReasoning	78	ITCScore	16.7	2.6	1.3
		ITMScore	21.8	5.1	2.6
		VisualGPTScore α^*	21.8	10.3	6.4

Table 11: BLIP performance on Winoground subtags (Diwan et al., 2022). We report the number of test instances for each subtag and their respective text score, image score, group score.

Method	Model	SugarCrepe			
		Replace	Swap	Add	AVG
Human Performance	-	98.67	99.50	99.00	99.06
Random Chance	-	50.00	50.00	50.00	50.00
Text-Only Baseline	Vera	49.46	49.30	49.50	49.42
	Grammar	50.00	50.00	50.00	50.00
$P_{LLM}(t)$	Bart	48.41	51.93	61.16	53.83
	Flan-T5	51.41	57.59	40.94	49.98
	OPT	58.53	66.58	45.78	56.96
$P_{train}(t)$	BLIP	75.90	77.14	70.89	74.64
ITCScore	CLIP-LAION2B	86.50	68.56	88.37	81.14
	CLIP-LAION5B	84.98	67.95	89.62	80.85
	BLIP	85.76	73.79	85.66	81.74
	BLIP-2	86.66	69.77	86.50	80.98
	NegCLIP-SugarCrepe	88.27	74.89	90.16	84.44
ITMScore	BLIP	88.68	81.29	87.57	85.85
	BLIP2-Qformer	88.45	80.87	87.96	85.76
$P_{train}(t i)$	BLIP	93.33	91.00	90.98	91.77
	BLIP2-Qformer	93.00	91.24	92.69	92.31
	BLIP2-FlanT5	82.44	76.57	76.24	78.42
$\frac{P_{train}(t i)}{P_{train}(t)}\alpha^*$	BLIP	95.09	92.39	97.36	94.95
	BLIP2-Qformer	94.62	92.27	97.58	94.82
	BLIP2-FlanT5	85.69	78.80	91.76	85.42

Table 12: **Performance on SugarCrepe (Hsieh et al., 2023)**. SugarCrepe is the most recent visio-linguistic compositionality benchmark which improves upon previous Crepe (Ma et al., 2022) by using state-of-the-art large language models (including ChatGPT), instead of rule-based templates, to generate more natural negative text captions. We show that text-only baselines and LLM-based methods indeed fail to succeed on SugarCrepe. However, our OTS generative approaches still achieve competitive results compared against SOTA discriminative approaches. The results of human performance, text-only baseline, and SOTA CLIP and NegCLIP-SugarCrepe are directly taken from the Hsieh et al. (2023). For other approaches, we evaluate their performance following the same procedure as described in main texts.

F BENCHMARK VISUALIZATION

We include random samples from each benchmark in [Table 13](#).






Dataset	Image	Positive Caption	Negative Caption(s)
VG-Relation		the bus is to the right of the trees	the trees is to the right of the bus
VG-Attribution		the striped zebra and the large tree	the large zebra and the striped tree
COCO-Order		two dogs sharing a frisby in their mouth in the snow	two frisby sharing a mouth in their snow in the dogs in dogs the in frisby sharing two mouth their a snow two dogs sharing in a frisby their mouth in snow the a frisby in the snow two dogs sharing their mouth in
Flickr30K-Order		a white duck spreads its wings while in the water	a white wings spreads its water while in the duck a white duck the its wings while in water spreads white a duck spreads its wings in while the water while in the spreads its wings water a white duck
SugarCrepe Add-Attribute		They are going to serve pizza for lunch today.	They are going to serve pizza topped with pineapple for lunch today.
SugarCrepe Add-Object		A man kisses the top of a woman's head.	A man kisses the top of a woman's head with a flower in his hand.
SugarCrepe Replace-Attribute		A kid standing with a small suitcase on a street.	A kid standing with a big suitcase on a street.
SugarCrepe Replace-Object		A duck floating in the water near a bunch of grass and rocks	A swan floating in the water near a bunch of grass and rocks.
SugarCrepe Replace-Relation		A clock tower stands in front of a large mirrored sky scraper.	A clock tower stands behind a large mirrored sky scraper.
SugarCrepe Swap-Attribute		A tennis player is taking a swing on a red court.	A red player is taking a swing on a tennis court.
SugarCrepe Swap-Object		A woman holding a game controller with a man looking on.	A man holding a game controller with a woman looking on.
Crepe-AtomFolds		microwave in a kitchen, and sink in a kitchen.	microwave in a cupboard, and sink in a kitchen microwave in a bar, and sink in a kitchen line in a kitchen, and sink in a kitchen microwave in a kitchen, and shower in a kitchen microwave in a kitchen, and tap in a kitchen
Crepe-Negate		a chair next to a table, with the back of the chair visible.	A chair is not next to a table, with the back of the chair visible A chair next to a table, with the back not of the chair visible A chair next to a table, with the back of the chair visible A chair next to a table, with something of the chair visible. There is no back. There is no chair next to a table, with the back of the chair visible
Crepe-Swap		a car driving on a road with a line next to a tree.	a car driving on a bright green leaves with a line next to a tree a bright green leaves driving on a road with a line next to a tree a car driving on a tree with a line next to a road a car driving on a road with a line next to a white car a car driving on a road with a line next to a street
VL-CheckList Relation (spatial)		person read book	person carry book
VL-CheckList Relation (action)		sign near boy	sign far from book
Winoground		a person on top of the world	the world on top of a person
		the world on top of a person	a person on top of the world
EqBen		The person is touching the dish which is in front of him/her.	The person is holding the dish which is in front of him/her.
		The person is holding the dish which is in front of him/her.	The person is touching the dish which is in front of him/her.

Table 13: **Visualization of benchmarks.** ARO (VG-Relation/VG-Attribution/COCO-Order/Flickr30K-Order), Crepe (AtomFolds/Negate/Swap), VL-CheckList (Object/Attribute/Relation), SugarCrepe (Replace/Swap/Add) are constructed by generating hard negative captions for an image-text pair. On the other hand, each sample of Winoground and EqBen has two image-text pairs.