Appendix

A. Training Details for All Experiments

We provide detailed explanations for experimental settings of all our reported results in the main paper. We have conducted our experiments under two different settings: 1) *Learnable-Dense*, 2) *Frozen-CLS*. The main difference between the two configurations depends on the way of utilizing the visual encoder. Regarding the text decoder, we utilize the same decoder model, which consists of a 6-layer transformer network initialized randomly for both settings.

For the main experiments *i.e.*, Tabs. 1 to 3, we use the *Learnable-Dense* setting where the visual encoder, *i.e.*, pre-trained CLIP ViT-L/14, is tunned during the training phase and both the output [CLS] feature and other spatial features of the CLIP visual encoder are used as visual features. Since the length of output spatial visual features is very long ($16 \times 16 = 256$), we apply 2d-average pooling to the spatial features with a scale factor of 1/4. As a result, the total length of the visual features becomes 17 ($1 + 4 \times 4$). As optimization hyperparameters, we set learning rates to $1e$-5 and $1e$-4 for the visual encoder and the text decoder, respectively, with the same weight decay of $1e$-5.

On the other hand, we adopt the *Frozen-CLS* setting for all other analysis and ablation experiments due to its training efficiency. In the *Frozen-CLS* setting, the visual encoder, *i.e.* pre-trained CLIP ViT-L/14, is frozen and only the output [CLS] feature is used as the visual feature. In this setting, since the sequence length of the visual feature is 1, we use the visual feature as a prefix token of caption tokens like self-attention-based decoding in GIT [47]. We set the learning rate for the caption decoder to 0.0016.

For both settings, we train our model for 10 epochs. The learning rate is warmed up in the first epoch and then follows cosine decay to 0. Also, the network parameters are updated by AdamW [28] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$.

B. Network Architecture Details

We present the network architecture of our alignment-level-controllable captioner in Fig. 10. When a pair of an image and a caption is given, we first calculate the cosine similarity, $s \in \mathbb{R}$, of the pair using pre-trained CLIP ViT-L/14. Then, we convert the similarity score into a discrete alignment level $l \in \{1, ..., K\}$ via the bucketing technique as described in Sec. 3.2. After getting the alignment level, we feed the discrete alignment level as a control signal into a learnable embedding layer to get a control vector that has the same dimension as the image embedding extracted from an image encoder. Finally, we concatenate the control signal and image embedding vectors and feed them to a caption decoder model. During the inference phase, we can simply set the control signal z as the desired alignment level (*e.g.*, $z = 7$) to get a caption describing a given image.

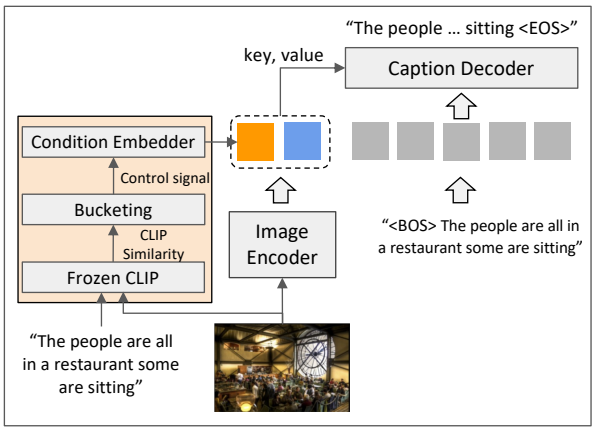


Figure 10: The network architecture of our alignment-level-controllable captioner. During the training phase, the calculated control signal is concatenated to the image embeddings. Then the concatenated vectors are fed into a cross-attention-based caption decoder model as key-value features.

Table 7: Zero-shot results on MSCOCO with different decoder architectures trained on CC3M dataset.

Decoder	Method	B@4	METEOR	SPICE	CIDEr
GiT-like [47]	Vanilla	8.65	14.40	10.32	40.24
	Filtering	11.00	16.44	11.64	47.75
	NoC (z=7)	**12.70**	**18.05**	**12.95**	**51.11**
VirTex-like [12]	Vanilla	9.37	15.02	10.75	42.43
	Filtering	11.24	16.89	12.37	50.43
	NoC (z=7)	**13.42**	**18.73**	**13.49**	**53.18**

C. Backbone Agnostic Property

Our method requires only a minor modification (*i.e.*, adding a control signal) to a conventional image captioning model, which means that our proposed noise-aware learning framework can be easily applied to any captioning model. Despite we have mainly reported the results based on a VirTex-like [12] cross-attention-based transformer network in our main paper, our noise-aware learning framework also can be simply applied to a GIT-like [47] self-attention-based transformer network. For the VirTex-like architecture, the control signal is concatenated to visual features and fed into each cross-attention layer as key-value features like Fig. 10. While in the GIT-like architecture, we can simply use the concatenated features as prefix tokens of a caption.

To empirically validate the backbone agnostic property of our algorithm, we present additional comparative experiments in Tab. 7. In this experiment, we freeze the visual encoder and use both [CLS] token and other spatial features, referred to as *Frozen-Dense*, for training efficiency. From the Tab. 7, our proposed model consistently outperforms fairly-controlled comparative baselines by large margins on both decoder architectures.

Models	MSCOCO				nocaps							
					overall		in-domain		near-domain		out-of-domain	
	B@4	M	C	CS	C	CS	C	CS	C	CS	C	CS
Vanilla	10.31	15.48	47.56	62.89	41.58	60.49	38.60	58.64	39.24	59.91	51.22	62.15
Vanilla (Filtering)	12.81	17.30	54.66	64.84	48.96	62.70	46.06	60.74	46.33	62.35	59.50	63.92
Bootstrap	13.51	17.46	55.13	64.31	49.46	62.16	45.23	60.60	47.14	61.62	59.93	63.62
NoC (z=7)	**15.96**	**19.50**	**62.04**	**66.70**	**54.94**	**64.21**	**51.74**	**62.54**	**53.09**	**63.92**	**63.15**	**65.19**

Table 8: Comparison with the Bootstrap baseline. All models are trained on CC3M and evaluated on MSCOCO and nocaps datasets. B@4, M, C, and CS mean BLEU@4, METEOR, CIDEr, and CLIPScore metrics, respectively. Numbers in **bold** indicate the best method.

D. Details for Loss Weighting Baseline

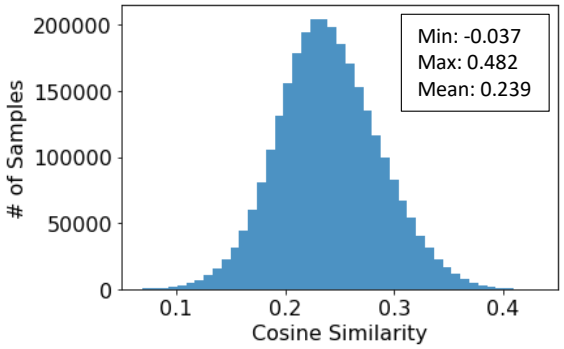
We have defined the Loss weighting baseline in Sec. 4.2 as follows:

$$\mathcal{L}_{\text{weighting}} = -\frac{1}{N}\sum_{i=1}^{N} s_i \log p(c_i|I_i), \tag{5}$$

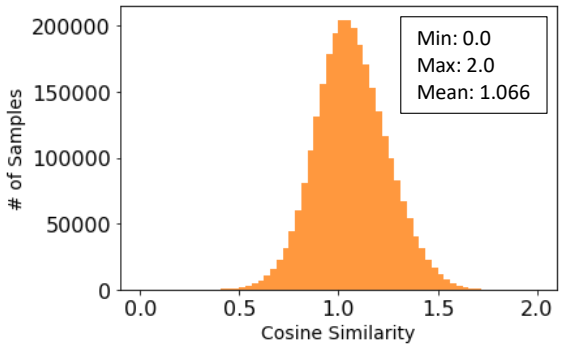
where s_i indicates a cosine similarity computed by CLIP for i^{th} image-text pair in a minibatch of N instances.

Additionally, since the range of cosine similarity from training samples is approximately ranged between 0 and 0.5 as shown in Fig. 11, we apply min-max normalization to s and multiply by 2 for equalizing the averaged loss scale. This loss reweighting strategy makes the model be updated by less for miss-aligned samples and more on well-aligned ones.

For a better understanding, we visualize the distribution of cosine similarities of the CC3M training split in Fig. 11. The range of raw cosine similarities is distributed from -0.037 to 0.482, and the mean of the distribution is 0.239. If we use the cosine similarities directly for loss re-weighting defined in Eq. (5), the scale of the overall loss value becomes low, which could result in slow convergence of the training phase. Thus, to equalize the averaged loss scale, we re-scale the cosine similarities for the loss re-weighting strategy by applying min-max normalization and multiplying by 2. Statistics of the resulting distribution of the re-scaled cosine similarities are 0.0, 2.0, and 1.066 for minimum, maximum, and mean values, respectively.



(a) Distribution of cosine similarities ***before*** re-scaling.



(b) Distribution of cosine similarities ***after*** re-scaling.

Figure 11: Distributions of cosine similarities for all image-text pairs in the CC3M training split. The cosine similarity is calculated by pre-trained CLIP ViT-L/14. Statistics for each distribution are presented in the upper right region in each figure.

E. Comparison with Data Bootstrapping

While BLIP [24] and our method have a similar motivation, *i.e.*, handle the noise issue inherent in the web-crawled dataset, we remark that BLIP requires clean data (*e.g.*, MSCOCO) to learn a filter and captioner for bootstrapping as we discussed in Sec. 2. Despite the necessity of clean data in BLIP, we carefully devised an additional experiment to evaluate the effectiveness of the data bootstrapping technique introduced in BLIP for our zero-shot experimental setting.

Firstly, as the captioner and filter models of BLIP are fine-tuned using the clean MSCOCO dataset, we implement a replacement for the purpose of zero-shot evaluation on MSCOCO. Specifically, we employ our Vanilla (Filtering) model, which has been trained on the filtered CC3M dataset, to replace the captioner, and we utilize the pre-trained CLIP ViT-B/32 as a substitute for the filter. Subsequently, we partitioned the original CC3M dataset into two groups. The first group consists of image-text pairs with a CLIP similarity greater than 0.3, which are considered well-aligned annotations. While the second group comprises image-text pairs with a CLIP similarity lower than 0.3, which are

Models	MSCOCO				nocaps							
					overall		in-domain		near-domain		out-of-domain	
	B@4	M	C	CS	C	CS	C	CS	C	CS	C	CS
Vanilla	10.31	15.48	47.56	62.89	41.58	60.49	38.60	58.64	39.24	59.91	51.22	62.15
Vanilla (Filtering)	12.81	17.30	54.66	64.84	48.96	62.70	46.06	60.74	46.33	62.35	59.50	63.92
Loss weighting	11.16	16.15	50.86	63.87	43.89	61.18	39.30	59.23	41.80	60.50	53.84	63.04
NoC (z=1)	12.65	16.44	52.78	63.23	43.95	60.55	41.39	58.50	42.11	60.30	51.70	61.62
NoC (z=2)	3.77	8.21	12.83	45.24	10.26	43.43	11.96	46.29	10.76	44.46	7.45	40.60
NoC (z=3)	3.70	9.19	17.33	49.67	15.12	48.05	16.45	49.56	15.68	48.66	12.40	46.48
NoC (z=4)	8.19	13.52	39.61	59.86	32.25	57.23	30.04	56.01	31.76	57.23	35.39	57.62
NoC (z=5)	11.88	16.28	52.11	64.01	45.68	61.72	41.25	59.57	43.74	61.18	55.05	63.23
NoC (z=6)	14.38	18.27	58.76	65.82	51.50	63.52	48.02	61.82	49.60	63.13	60.06	63.52
NoC (z=7)	**15.96**	**19.50**	**62.04**	**66.70**	**54.94**	**64.21**	**51.74**	**62.54**	**53.09**	**63.92**	**63.15**	**65.19**
NoC (z=8)	12.82	17.46	53.50	64.94	48.05	62.64	42.44	60.16	45.86	62.20	59.08	64.21

Table 9: Zero-shot captioning results for all bin indices from models trained on CC3M. Numbers in **bold** and underlined indicate the best and second-best ones, respectively.

considered less-aligned annotations. After separating the dataset, we use the trained Vanilla (Filtering) model to generate pseudo-captions for the images in the second group, as suggested by BLIP. Finally, we train a Vanilla model with the bootstrapped dataset under the *Learnable-Dense* setting.

From the Tab. 8, we observed that the Bootstrap baseline shows marginally higher captioning scores than the Vanilla (Filtering). In contrast, the Bootstrap baseline shows significantly lower scores than our proposed model. We carefully hypothesize that the bootstrapping technique would yield optimal results when a clean dataset of the target domain is available for training the captioner and filter. On the other hand, our proposed model can effectively mitigate the negative impact of noisy pairs utilizing only the web-crawled data. Consequently, it shows superior zero-shot generalization performance in comparison to the data bootstrapping technique.

Models	MSCOCO			Flickr30k		
	R@1	R@5	R@10	R@1	R@5	R@10
GT Caption	34.57	59.30	69.91	63.08	86.50	92.00
Vanilla	25.44	50.38	61.66	47.10	76.60	85.90
Vanilla (Filtering)	31.64	58.90	70.36	56.50	85.50	92.50
Loss weighting	28.78	54.44	65.44	48.00	78.90	87.50
NoC (z=1)	25.92	49.92	61.98	44.60	77.20	86.60
NoC (z=2)	2.44	7.86	11.46	8.30	21.50	29.20
NoC (z=3)	5.02	13.12	18.98	12.70	30.90	40.00
NoC (z=4)	17.34	38.86	50.30	32.30	66.70	75.30
NoC (z=5)	29.02	54.32	66.26	50.30	84.10	91.10
NoC (z=6)	35.26	62.78	73.88	60.00	89.90	95.40
NoC (z=7)	**40.00**	**66.78**	**77.53**	**65.10**	**92.00**	**96.20**
NoC (z=8)	32.30	57.60	69.72	54.90	85.80	93.50

Table 10: Self-retrieval capability for all bin indices on MSCOCO and Flickr30k datasets. Numbers in **bold** and underlined indicate the best and second-best ones, respectively.

F. Detailed Results for All Alignment Levels

We provide comprehensive results for zero-shot captioning and self-retrieval tasks originally reported in Tabs. 1 and 3, with the aim of using these expended results to help verify that our noise-aware model is trained according to our intentions. Tabs. 9 and 11 present zero-shot captioning results. Each model is trained on either CC3M or various scales of COYO and evaluated on MSCOCO and nocaps datasets. From the tables, our model consistently outperforms other baselines when ($z \geq 6$). On the other hand, when ($z < 6$), our model shows lower scores than the comparative baselines. This trend is reasonable considering that image-text pairs with bin indices of $z < 6$ are relatively less-aligned pairs. Consequently, our model, when conditioned on these indices, is expected to generate less-aligned captions.

From the Tabs. 10 and 12, we observe a similar trend in retrieval performance as in captioning results, *i.e.*, increasing the bin index leads to higher scores. Notably, as illustrated in Tab. 12, self-retrieval performance consistently improves as models are trained on larger-scale datasets. This suggests that a model can generate more distinctive captions as it learns a broader range of visual concepts from larger datasets.

We note that a model trained in CC3M shows irregular performance at the lowest bin index ($z = 1$). We hypothesize that this is due to the extremely small number of samples for $z = 1$ in CC3M (only 21 samples included), which prevents the model from being fully trained for the noisiest samples. In contrast, for COYO, which is much noisier than CC3M, the number of noisiest samples is sufficiently large, resulting in the model showing the worst performance at $z = 1$.

Ours(z=1) Person, a photographer, spent travelling to capture the breath - taking photographs in the book
(z=2) Person, click for a larger version
(z=3) Person and I had a lot of fun.
(z=4) Cows grazing in a meadow
(z=5) Cows grazing in a field
(z=6) Cattle graze in a pasture
(z=7) Cattle graze in a pasture with storm cloud overhead.
(z=8) Cattle graze in a pasture with storm cloud overhead.

Ours(z=1) A city is a restaurant
(z=2) Person, a place of worship
(z=3) The view from the dining room
(z=4) The restaurant is a popular place to eat
(z=5) The restaurant is a popular place for locals and tourists alike to eat and drink
(z=6) The restaurant is a popular place for locals and tourists alike to enjoy a meal
(z=7) The restaurant is a nod to the late 19th century, with a large glass - walled dining room
(z=8) The restaurant features a large, rustic wooden table and chairs with a large glass - topped bowl of wine

Figure 12: Examples of generated captions from all alignment levels of our model.

G. Discussion on Similarity Computation Module

We examine how robust our model is to the variations on the alignment level computation module used for computing cosine similarities between given image-text pairs. We change the alignment level computation module from pre-trained CLIP ViT-L/14 to CLIP ViT-B/32 which has fewer parameters resulting in faster computations of alignment levels but shows lower performance than the ViT-L/14-based model in other downstream tasks [38]. The zero-shot captioning and self-retrieval results are presented in Tab. 13 and Tab. 14, respectively. From the tables, we observe that our model shows almost consistent performance on both the captioning and retrieval tasks, even if the alignment level computation module is changed to a model having relatively lower representation power. Also, we expect that if we can use a more powerful model trained to capture fine-grained alignments between images and paired texts, such as Florence [55] or FILIP [53][3], for measuring the alignment levels, our proposed noise-aware learning framework can perform better than current results as the powerful alignment computation module measures the alignment level more accurately.

H. More Qualitative Results

H.1. Zero-shot Captioning Results

We present the captioning results for all alignment levels of our model to show that our model can effectively control the quality of generated captions in Fig. 12. From the Fig. 12, our quality controllable model generates miss aligned or less descriptive captions with the control signals corresponding to low alignment levels (*i.e.*, $z \leq 3$). While, with the control signals meaning to middle alignment levels (*i.e.*, $4 \leq z \leq 6$), our model describes the images using common words as other baselines do. Finally, our model with the control signals corresponding to high alignment levels (*i.e.*, $z \geq 7$) can generate captions satisfying both *descriptiveness* and *distinctiveness*.

In addition, we provide more zero-shot captioning examples in Fig. 13. As discussed in Sec. 4.7, baseline models tend to describe given images using common words (or phrases), while our model with a higher alignment level (*e.g.*, $z = 7$) describes the images with more various words (or phrases) based on learned rich visual knowledge.

H.2. Self-retrieval Results

We present more examples of self-retrieval results on the MSCOCO dataset for Vanilla (Filtering) baseline and our method in Fig. 14.

[3]Checkpoints of the models have not been released publicly yet.

Models	MSCOCO				nocaps							
					overall		in-domain		near-domain		out-of-domain	
	B@4	M	C	CS	C	CS	C	CS	C	CS	C	CS
Vanilla	4.41	10.49	22.54	58.25	21.30	57.62	17.76	55.97	18.98	56.74	31.26	59.79
Vanilla (Filtering)	4.65	11.52	22.48	59.67	21.29	59.04	18.15	56.22	19.25	58.68	30.07	60.56
NoC (z=1)	0.00	2.40	0.09	30.86	0.08	30.85	0.05	30.72	0.09	30.29	0.07	31.95
NoC (z=2)	0.00	2.38	0.13	31.20	0.15	31.36	0.24	30.17	0.15	30.89	0.10	32.60
NoC (z=3)	0.31	4.28	3.15	41.82	2.31	40.49	2.74	40.33	2.32	40.52	1.97	40.47
NoC (z=4)	2.40	8.30	16.21	54.88	17.05	54.33	13.20	52.25	15.45	53.53	24.93	56.44
NoC (z=5)	5.40	11.37	25.80	60.35	24.21	59.36	20.17	56.96	21.84	58.81	34.70	61.10
NoC (z=6)	**6.60**	13.39	**28.90**	62.50	26.92	61.22	**23.19**	58.33	24.84	61.00	36.24	62.48
NoC (z=7)	6.59	14.19	28.10	**63.33**	**27.24**	**62.09**	22.17	**59.32**	**25.29**	**61.86**	**37.09**	**63.34**
NoC (z=8)	6.06	**14.39**	24.72	63.13	24.85	61.88	19.57	58.67	23.14	61.63	34.11	63.31

(a) Zero-shot captioning results from models trained on COYO3M

Models	MSCOCO				nocaps							
					overall		in-domain		near-domain		out-of-domain	
	B@4	M	C	CS	C	CS	C	CS	C	CS	C	CS
Vanilla	5.34	11.31	27.52	60.69	24.00	59.61	18.32	56.52	21.95	58.96	34.59	61.77
Vanilla (Filtering)	6.51	12.71	29.25	64.11	26.88	62.83	21.39	60.87	24.84	62.52	37.30	63.98
NoC (z=1)	0.10	2.37	0.13	28.44	0.12	27.39	0.17	26.64	0.11	27.17	0.11	28.03
NoC (z=2)	0.00	2.44	0.28	32.28	0.23	31.75	0.29	30.62	0.26	31.63	0.12	32.30
NoC (z=3)	0.52	4.33	3.56	41.89	2.46	40.45	2.85	40.32	2.49	40.39	2.08	40.60
NoC (z=4)	2.90	8.94	19.62	56.20	18.78	55.61	15.70	52.46	16.80	54.92	27.33	57.84
NoC (z=5)	6.22	12.09	29.86	62.60	27.27	61.44	21.82	58.53	25.45	61.04	36.97	63.08
NoC (z=6)	**7.32**	14.05	**32.63**	65.33	30.57	63.91	25.33	60.63	**28.28**	63.77	41.65	65.12
NoC (z=7)	7.05	15.03	30.59	66.26	**30.66**	65.08	**26.11**	61.95	27.91	64.94	**42.72**	66.27
NoC (z=8)	6.92	**15.59**	27.75	**66.65**	29.07	**65.35**	24.75	**63.06**	26.02	**65.20**	41.93	**66.32**

(b) Zero-shot captioning results from models trained on COYO10M

Models	MSCOCO				nocaps							
					overall		in-domain		near-domain		out-of-domain	
	B@4	M	C	CS	C	CS	C	CS	C	CS	C	CS
Vanilla	5.17	11.32	27.07	62.06	25.32	60.93	19.29	58.43	23.05	60.26	36.87	62.95
Vanilla (Filtering)	7.18	13.33	32.52	65.87	29.85	64.65	25.33	62.48	27.64	64.25	40.18	66.05
NoC (z=1)	0.00	2.96	0.08	33.45	0.08	31.54	0.11	30.90	0.08	31.34	0.07	32.12
NoC (z=2)	0.00	3.14	0.18	35.89	0.18	34.65	0.26	33.81	0.19	34.39	0.08	35.38
NoC (z=3)	0.36	4.33	2.57	41.26	1.68	39.40	1.88	39.34	1.78	39.70	1.20	38.86
NoC (z=4)	2.81	9.03	19.97	56.93	18.84	56.11	14.33	53.29	17.19	55.30	27.34	58.46
NoC (z=5)	6.11	12.05	30.25	63.67	27.29	62.53	22.40	59.81	25.11	62.06	37.75	64.21
NoC (z=6)	**7.59**	14.19	**33.43**	66.65	**30.60**	65.17	**26.59**	62.64	28.11	65.00	**41.45**	66.24
NoC (z=7)	7.57	15.54	31.72	68.12	30.55	66.59	25.25	64.00	**28.52**	66.46	40.85	67.60
NoC (z=8)	7.01	**16.37**	25.45	**68.80**	27.11	**67.39**	21.02	**64.92**	24.34	**67.26**	40.33	**68.36**

(c) Zero-shot captioning results from models trained on COYO23M

Models	MSCOCO				nocaps							
					overall		in-domain		near-domain		out-of-domain	
	B@4	M	C	CS	C	CS	C	CS	C	CS	C	CS
Vanilla	4.92	11.18	28.03	62.40	25.29	61.73	20.25	59.14	22.56	60.98	37.63	63.90
Vanilla (Filtering)	7.80	13.48	34.55	66.75	30.93	65.56	26.90	63.47	28.75	65.35	40.77	66.58
NoC (z=1)	0.00	2.04	0.09	33.84	0.09	31.25	0.16	30.30	0.08	30.71	0.10	32.56
NoC (z=2)	0.00	2.04	0.20	35.42	0.18	33.37	0.38	32.51	0.15	32.94	0.17	34.44
NoC (z=3)	0.45	4.38	2.98	40.84	1.96	38.61	2.00	38.88	2.15	39.09	1.30	37.62
NoC (z=4)	3.11	9.18	21.26	57.13	19.15	56.31	15.32	53.77	17.37	55.53	27.59	58.54
NoC (z=5)	6.68	12.37	31.54	64.60	27.62	63.35	22.70	60.74	25.68	62.98	37.34	64.82
NoC (z=6)	**8.44**	14.53	**35.82**	67.72	**31.62**	66.32	**27.12**	64.30	28.98	66.18	43.33	67.19
NoC (z=7)	8.08	15.87	32.86	69.24	31.55	67.74	24.05	65.75	**29.05**	67.66	**44.95**	68.46
NoC (z=8)	7.12	**16.73**	23.85	**69.87**	25.35	**68.51**	18.08	**66.73**	22.44	**68.41**	39.85	**69.24**

(d) Zero-shot captioning results from models trained on COYO125M

Table 11: Zero-shot caption generation performance on MSCOCO and nocaps when scaling up the training dataset sizes using COYO. Models of each dataset are trained for the same number of steps. Numbers in **bold** and underlined indicate the best and second-best ones, respectively.

Models	MSCOCO			Flickr30k		
	R@1	R@5	R@10	R@1	R@5	R@10
GT Caption	34.57	59.30	69.91	63.08	86.50	92.00
Vanilla	24.92	48.96	59.90	47.90	73.90	82.10
Vanilla (Filtering)	31.78	57.96	69.08	52.20	82.00	90.80
NoC (z=1)	0.08	0.24	0.44	0.20	1.10	1.90
NoC (z=2)	0.08	0.32	0.78	0.10	0.90	1.70
NoC (z=3)	1.32	4.20	6.50	3.10	10.10	15.00
NoC (z=4)	14.16	34.10	44.30	28.40	54.90	66.60
NoC (z=5)	30.16	57.14	67.86	54.30	81.10	88.90
NoC (z=6)	39.28	68.30	77.86	62.80	88.30	92.90
NoC (z=7)	44.96	72.56	81.42	66.50	89.40	94.70
NoC (z=8)	**45.76**	**73.02**	**81.90**	**69.70**	**91.00**	**94.80**

(a) Self-retrieval results from models trained on COYO3M

Models	MSCOCO			Flickr30k		
	R@1	R@5	R@10	R@1	R@5	R@10
GT Caption	34.57	59.30	69.91	63.08	86.50	92.00
Vanilla	31.44	55.48	66.42	56.30	79.90	86.20
Vanilla (Filtering)	43.56	70.60	80.42	67.80	90.30	95.00
NoC (z=1)	0.06	0.22	0.44	0.20	1.00	1.80
NoC (z=2)	0.18	0.44	0.94	0.40	1.40	2.20
NoC (z=3)	1.48	4.72	7.08	3.10	9.70	14.90
NoC (z=4)	16.24	36.94	48.14	33.80	60.40	71.70
NoC (z=5)	35.98	63.52	75.08	59.70	86.60	92.70
NoC (z=6)	48.24	76.44	84.82	71.20	92.60	96.80
NoC (z=7)	54.70	80.68	87.94	75.50	94.30	**97.40**
NoC (z=8)	**56.60**	**81.92**	**88.76**	**78.30**	**95.40**	**97.40**

(b) Self-retrieval results from models trained on COYO10M

Models	MSCOCO			Flickr30k		
	R@1	R@5	R@10	R@1	R@5	R@10
GT Caption	34.57	59.30	69.91	63.08	86.50	92.00
Vanilla	34.70	60.42	71.02	58.70	83.20	88.30
Vanilla (Filtering)	48.70	75.62	84.28	74.10	93.30	96.30
NoC (z=1)	0.02	0.18	0.30	0.00	0.70	0.90
NoC (z=2)	0.06	0.34	0.68	0.40	1.20	1.90
NoC (z=3)	1.30	3.82	6.42	2.90	8.20	13.90
NoC (z=4)	17.44	38.68	50.14	33.50	61.00	72.00
NoC (z=5)	39.04	66.58	77.42	65.90	89.50	94.60
NoC (z=6)	52.80	79.42	87.62	76.60	93.50	96.90
NoC (z=7)	61.40	85.32	91.38	81.30	96.60	98.00
NoC (z=8)	**64.92**	**86.46**	**92.60**	**84.10**	**96.80**	**98.40**

(c) Self-retrieval results from models trained on COYO23M

Models	MSCOCO			Flickr30k		
	R@1	R@5	R@10	R@1	R@5	R@10
GT Caption	34.57	59.30	69.91	63.08	86.50	92.00
Vanilla	36.20	60.84	71.44	60.90	83.20	89.20
Vanilla (Filtering)	52.10	77.34	86.10	77.20	95.20	97.70
NoC (z=1)	0.02	0.12	0.28	0.00	0.40	0.80
NoC (z=2)	0.04	0.34	0.58	0.10	0.90	1.50
NoC (z=3)	0.94	3.56	5.66	2.70	9.10	13.50
NoC (z=4)	17.02	37.60	49.36	34.40	64.10	75.10
NoC (z=5)	42.00	69.54	79.56	68.70	91.40	95.70
NoC (z=6)	57.24	82.06	89.24	82.60	96.30	98.20
NoC (z=7)	65.32	87.68	92.84	85.40	97.40	98.50
NoC (z=8)	**69.66**	**89.28**	**94.12**	**88.20**	**98.00**	**99.30**

(d) Self-retrieval results from models trained on COYO125M

Table 12: Self-retrieval performance on MSCOCO and Flickr30k when scaling up the training dataset sizes using COYO. Models of each dataset are trained for the same number of steps. Numbers in **bold** and underlined indicate the best and second-best ones, respectively.

Models	MSCOCO				nocaps							
					overall		in-domain		near-domain		out-of-domain	
	B@4	M	C	CS	C	CS	C	CS	C	CS	C	CS
Alignment Level Computation using ViT-L/14												
NoC (z=1)	3.59	10.36	18.33	50.68	13.46	48.28	13.69	48.97	12.90	48.41	15.08	47.84
NoC (z=2)	1.53	6.80	6.45	41.06	4.38	38.85	4.95	36.54	4.42	39.68	3.81	36.54
NoC (z=3)	1.99	8.13	11.42	47.31	8.86	45.80	10.36	46.92	8.72	46.18	8.26	44.76
NoC (z=4)	5.36	11.78	27.88	57.27	24.06	55.41	23.72	53.38	23.65	55.31	25.61	55.91
NoC (z=5)	9.19	15.01	42.27	62.89	38.59	61.07	35.13	59.54	35.57	60.37	50.73	62.86
NoC (z=6)	11.79	17.18	49.28	65.58	44.00	63.33	39.53	62.06	41.25	62.84	56.00	64.61
NoC (z=7)	12.11	18.34	49.18	**66.65**	45.09	**64.40**	39.25	**63.02**	42.41	**64.18**	57.87	**65.21**
NoC (z=8)	12.23	18.43	48.81	66.31	46.21	64.10	40.23	62.84	43.50	63.77	59.15	64.11
Alignment Level Computation using ViT-B/32												
NoC (z=1)	9.42	13.79	33.68	54.00	29.52	50.99	33.43	51.32	28.52	50.82	29.98	51.23
NoC (z=2)	1.83	7.26	7.68	43.43	5.63	41.52	6.37	43.09	5.36	42.00	5.98	40.16
NoC (z=3)	2.31	8.73	13.24	49.10	10.54	47.36	12.15	48.49	10.09	47.62	10.88	46.51
NoC (z=4)	6.01	12.56	30.81	59.03	25.34	56.72	25.06	56.10	23.90	56.44	30.19	57.41
NoC (z=5)	9.34	15.29	43.10	63.38	39.19	61.33	34.74	59.24	36.22	60.70	51.89	63.14
NoC (z=6)	11.70	17.18	49.01	65.23	45.29	63.22	41.19	61.09	42.29	62.86	57.81	64.53
NoC (z=7)	**12.32**	**18.67**	**49.68**	66.26	**47.19**	63.86	**42.07**	62.14	**44.23**	63.51	**60.31**	65.01
NoC (z=8)	11.31	16.51	47.77	63.67	43.71	61.43	39.70	60.45	41.11	60.87	54.91	62.79

Table 13: Zero-shot caption generation performances on MSCOCO and nocaps trained on CC3M when replacing the pre-trained CLIP ViT-L/14 with CLIP ViT-B/32 for alignment level computation. We observe our model seems robust to the pre-trained model. Numbers in **bold** and underlined indicate the best and second-best ones for each model, respectively.

Models	MSCOCO			Flickr30k		
	R@1	R@5	R@10	R@1	R@5	R@10
Alignment Level Computation using ViT-L/14						
NoC (z=1)	5.12	14.90	21.68	14.30	33.70	43.60
NoC (z=2)	1.08	3.22	4.98	2.90	9.30	14.40
NoC (z=3)	2.80	8.10	12.12	7.50	21.90	29.90
NoC (z=4)	11.72	28.84	39.04	27.40	57.00	66.20
NoC (z=5)	25.24	52.10	63.78	45.80	79.20	89.30
NoC (z=6)	38.18	65.60	76.72	62.60	91.00	95.70
NoC (z=7)	**44.16**	**71.18**	**81.16**	**69.20**	**93.90**	97.20
NoC (z=8)	43.02	70.74	80.34	67.90	92.90	**97.30**
Alignment Level Computation using ViT-B/32						
NoC (z=1)	9.26	23.08	31.12	20.40	45.80	56.70
NoC (z=2)	1.38	5.06	7.82	4.60	14.10	19.40
NoC (z=3)	3.92	11.66	17.70	11.50	29.00	39.90
NoC (z=4)	16.22	36.72	47.88	33.00	64.40	74.50
NoC (z=5)	28.66	55.76	66.82	49.70	82.30	90.40
NoC (z=6)	36.40	64.76	75.92	60.10	91.00	96.00
NoC (z=7)	**43.20**	**70.38**	**80.50**	**67.30**	**91.20**	**96.30**
NoC (z=8)	29.50	56.42	68.64	56.50	84.50	91.10

Table 14: Self-retrieval performance on MSCOCO and Flickr30k trained on CC3M when replacing the pre-trained CLIP ViT-L/14 with CLIP ViT-B/32 for alignment level computation. Numbers in **bold** and underlined indicate the best and second-best ones for each model, respectively.

Vanilla Person in action in the race

Vanilla (Filtering) A horse being ridden by a driver

Ours A horse and jockey ride around the track during a race

Vanilla The boat is a popular attraction for visitors to the city

Vanilla (Filtering) The team of canoeing down river

Ours A group of men and women paddle a canoe through the water during the regatta

Vanilla Little boy sitting on a pier

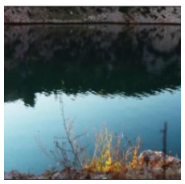
Vanilla (Filtering) Little boy sitting on a wooden dock

Ours Little boy sitting on a wooden pier near the lake and looking at the water

Vanilla Tired traveler sitting on the airport

Vanilla (Filtering) Businessman waiting for departure at the airport

Ours Sad and depressed businessman sitting on the airport with a suitcase



Vanilla Lake in the spring at sunset

Vanilla (Filtering) Lake with a mountain range in the background

Ours A beautiful lake with a reflection of the sky and a rock in the water

Vanilla Dogs playing in the park

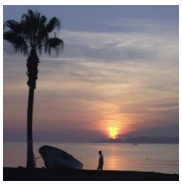
Vanilla (Filtering) Dogs playing in the grass

Ours Dogs wrestling on a green grass

Vanilla Original fine art by person

Vanilla (Filtering) Flowers in a vase by painting artist

Ours A still life painting of a bunch of red and yellow flowers in a vase



Vanilla Silhouette of a man walking on the beach at sunset

Vanilla (Filtering) Silhouette of a man walking on a pier at sunset

Ours A man walks along the beach at sunset with a sailboat in the background

Vanilla Person in a hat and suit

Vanilla (Filtering) A man in a suit with a bow tie

Ours A man in a brown fedora and a blue shirt with a bow tie

Vanilla A base with a painting of a vase

Vanilla (Filtering) Vase with a flower on the table

Ours A vintage vase with a beautiful floral pattern

Vanilla This is what my dog looks like

Vanilla (Filtering) A dog with glasses reading a book

Ours A dog dressed as a nerd reads a book

Vanilla Aerial view of the island

Vanilla (Filtering) Aerial view of a tropical island

Ours Aerial view of a tropical island with blue water and white sand beaches

Vanilla A bench in the shade of a tree

Vanilla (Filtering) A view of the back of the barn

Ours A white bench sits in a field of tall grass with a large tree in the background

Vanilla Pink flowers on a tree

Vanilla (Filtering) Pink flowers on a tree

Ours Beautiful purple orchids blooming in a park with green trees and blue sky background

Vanilla Ice skating on the frozen lake

Vanilla (Filtering) Ice skating on the frozen lake

Ours Ice skating on frozen lake in the park on a sunny winter day

Vanilla Sunset - admire the beauty of creations

Vanilla (Filtering) Sunset over the trees in the forest

Ours a red and orange sunset over a forest of pine trees

Figure 13: Examples of generated captions sampled from MSCOCO and CC3M validation splits. The captions of our model are generated with the control signal $z = 7$. Expressions capturing fine details from images in ours are highlighted in red.

Figure 14: Examples of self-retrieval in MSCOCO. For each example (of three rows), the first column indicates the input image and the generated captions by the specified model, while 2-6th columns show the top-5 retrieved images using the generated captions—by our method and Vanilla (Filtering) baseline—or ground-truth caption.