A Appendix

A.1 Gaussian as the Posterior

Here we explain why setting the reward distribution as a Gaussian distribution is feasible. Recall that we consider a non-stationary environment. We design the reward distribution to model the rewards observed in recent history. Within a small number of steps, the model is unlikely to change drastically. Therefore, the observed rewards, which are computed as the ratios of decrement in loss, are unlikely to be skewed. As a result, Gaussian is a feasible choice.

A.2 Data

For the VQA task, we conduct downstream fine-tuning and testing on the VQA 2.0 dataset [14], which consists of 83k images and 444k questions for training, 41k images, and 214k questions for validation. For the image captioning task on COCO, we use [6] for training and testing. It contains 11k images for training and 5k images for validation and 5k images for testing.

A.3 Teacher Model Implementation Details

We fine-tune a pre-trained CoCa-Large model [48] as the teacher. It contains 672M parameters in Transformer layers and contains a total of 787M parameters including the embedding size. It contains 24 layers in the image encoder, and 12 layers in the text encoder, and 12 layers in the multimodal decoder. We use a vocabulary size of 64k. We use 576 as the image resolution and 18 as the patch size for image inputs. We use 64 as the max sequence length for text inputs. We follow [48] for fine-tuning hyper-parameters for all tasks, as listed in Table 2.

Table 2: Hyper-parameters for fine-tuning CoCa-Large teacher models.

Hyper-parameters	VQA	SNLI-VE	NLVR2	COCO Caption
Optimizer	Adafactor with Decoupled Weight Decay			
Adam βs	$(0.9, 0.999)$			
Gradient Clipping	1.0			
Learning Rate Schedule	Linear Schedule Decaying to Zero			
Warm-up Steps	1k			
Weight Decay Rate	0.1			
Pooler Learning Rate	5×10^{-4}	1×10^{-3}	5×10^{-3}	N/A
Encoder Learning Rate	2×10^{-5}	5×10^{-5}	2×10^{-5}	1×10^{-5}
RandAugment	$1, 10$	$1, 10$	None	None
Training Steps	100k	50k	50k	50k
Batch Size	64	128	64	128
Dropout of Task Layer	0.5	0.5	0.5	N/A

For multimodal understanding tasks, we follow [48] to apply an attentional pooler with a single query to extract embedding from the decoder output, and train a linear classifier on top of the pooled embedding. For NLVR2, we construct two input sequences, each containing the concatenation of the description and one image. The two output representations are further concatenated as the input to the classifier. For image captioning, we apply the captioning loss proposed in [48]. We do not use the CIDEr metric-specific optimization [29]. We use a greedy strategy for decoding.

A.4 Distillation Implementation Details

For each task, we distill a CoCa-Tiny$_{12}$ student and a CoCa-Tiny$_6$ student from a fine-tuned CoCa-Large teacher on that task. CoCa-Tiny$_{12}$ contains 102M parameters in the Transformer layers and contains a total of 152M parameters including the embedding size. CoCa-Tiny$_6$ contains 55M parameters in the Transformer layers and contains a total of 105M parameters including the embedding size. We use 576 as the image resolution and 18 as the patch size for image inputs. To tokenize text input, we use a sentence-piece model [33, 22] with a vocabulary size of 64k trained on the sampled pre-training dataset. We use 64 as the max sequence length for text inputs. We follow [48] for fine-tuning hyper-parameters for all tasks, as listed in Table 3.

We conduct a two-stage distillation for CoCa-Tiny$_6$. We first distill CoCa-Tiny$_{12}$ from CoCa-Large, then use the distilled CoCa-Tiny$_{12}$ as the teacher to teach CoCa-Tiny$_6$. Existing works have shown that introducing an intermediate-sized teacher reduces the gap between the teacher and the student model, which allows the distillation to be more effective [26].

We distill the model for a total of T' steps, i.e., a total of $T = T'/P$ rounds. Among the T rounds, the first $T_0 \cdot K$ rounds are used for initialization of the parameters of the reward distribution.

Table 3: Hyper-parameters for distilling CoCa-Tiny student models.

Hyper-parameters	VQA	SNLI-VE	NLVR2	COCO Caption
T_0	10			
P	100			
T	1000			
γ	0.98			
αs	$(0.0, 1.0, 1 \times 10^{-2})$			
Optimizer	Adafactor with Decoupled Weight Decay			
Adam βs	$(0.9, 0.999)$			
Gradient Clipping	1.0			
Learning Rate Schedule	Linear Schedule Decaying to Zero			
Learning Rate	1×10^{-3}			
Warm-up Steps	1k			
Weight Decay Rate	0.1			
RandAugment	$1, 10$	$1, 10$	None	None
Training Steps (T')	125k	100k	100k	100k
Batch Size	128	384	256	256
Dropout of Task Layer	0.5	0.5	0.5	N/A

A.5 Statistics of Experimental Results

We report the median of five random seeds for experiment results on CoCa-Tiny$_{12}$ and CoCa-Tiny$_6$. Table 4 show the standard deviations of the experimental results in Table 1.

Table 4: The standard deviation of the experimental results in Table 1.

Method	VQA Acc	SNLI-VE Acc	NLVR2 Acc	COCO Caption CIDEr
CoCa-Tiny$_6$	0.20	0.25	0.11	0.15
CoCa-Tiny$_6$ (OPTIMA)	0.22	0.13	0.35	0.15
CoCa-Tiny$_{12}$	0.17	0.23	0.15	0.88
CoCa-Tiny$_{12}$ (OPTIMA)	0.08	0.15	0.30	0.32

A.6 Design of Reward

Recall that we design the reward (Eq. 6) as the averaged ratio of loss decrements over three types of distillation losses: $\mathcal{D}_{\mathrm{KL}}$ (Eq. 1), $\mathcal{L}_{\mathrm{hidn}}$ (Eq. 3) and $\mathcal{L}_{\mathrm{attn}}$ (Eq. 4). Figure 7 compares ours with two variants: 1) r_{KD}: the ratio of loss decrement of $\mathcal{D}_{\mathrm{KL}}$; 2) r_{LWD}: the averaged ratio of loss decrements over $\mathcal{L}_{\mathrm{hidn}}$ and $\mathcal{L}_{\mathrm{attn}}$. We can observe that r_{KD} performs better than r_{LWD} in NLVR2 but reversely in COCO. Since captioning tasks often rely more on contextual knowledge in the layerwise representations than classification tasks, the layerwise representation distance may better characterize the distillation performance in COCO. By taking both distance metrics into consideration, r_{OPTIMA} performs well on both tasks.

A.7 Design of the Reward Distribution

Recall that we design the mean of the reward distribution (Eq. 7) as the exponential moving average (EMA) of the past rewards. Figure 8 shows a hyper-parameter study on the halflife of the EMA, computed as $-\frac{1}{\log_2 \gamma}$. Halflife is the number of rounds the EMA decays by one-half. We can observe that a too-large or too-small halflife, meaning that counting too many old rewards or counting only instantaneous rewards can both be harmful to the student's performance. This corroborates that the actual contribution of each module is non-stationary in the long term and stationary in the short term, and using EMA with an appropriate γ can correctly track the changing contribution.

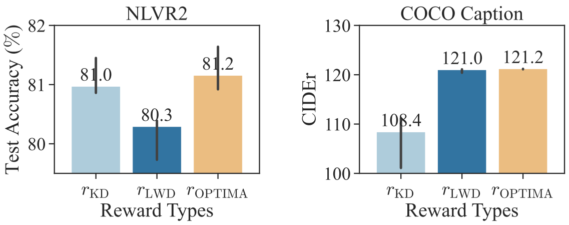

Figure 7: A comparison of three variants of reward: 1) r_{KD}: the ratio of decrement of $\mathcal{D}_{\mathrm{KL}}$; 2) r_{LWD}: the averaged ratio of decrement over $\mathcal{L}_{\mathrm{hidn}}$ and $\mathcal{L}_{\mathrm{attn}}$; 3) r_{OPTIMA}.

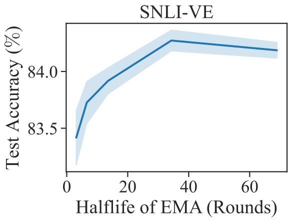

Figure 8: A hyper-parameter study on the halflife of the EMA, computed as $-\frac{1}{\log_2 \gamma}$.