Improving CLIP Training

Praroop Chanda
Dept. of Electrical & Computer Engineering
Texas A&M University
College Station, TX
praroop27@tamu.edu

Ishaan Singh Rawal
Dept. of Computer Science
Texas A&M University
College Station, TX
ishaanrawal@tamu.edu

Abstract

This study focuses on systematic hyperparameter optimization for CLIP, SogCLR, and CyCLIP models to enhance the efficiency and performance of contrastive learning methods, particularly under small batch size constraints. Utilizing HyperOpt with ASHA scheduler, we explored optimal configurations for key hyperparameters, including learning rates, temperature, discount factor, in modal and cross modal weights. Training and validation were conducted using a 100k subset of the CC3M dataset, with testing performed on the COCO and ImageNet datasets. Among the models, SogCLR with the AdamW optimizer demonstrated superior performance, achieving the highest zero-shot retrieval mean accuracy (Mean Recall) of 28.45% on COCO dataset and classification accuracy of 55.99% on the top@10 metric for ImageNet dataset, outperforming both CLIP and CyCLIP. Additionally, AdamW consistently outperformed Nadam across all models and configurations. The findings suggest that careful hyperparameter optimization can significantly enhance model performance, even when working with subset datasets and under constrained conditions.

1 Introduction

Self-Supervised Learning (SSL) has emerged as a powerful paradigm for learning generalizable representations from unlabeled data. Within this domain, contrastive learning has gained prominence as an effective approach that learns representations by maximizing similarity between positive pairs while minimizing similarity between negative pairs in the latent space.

One of the most notable advancements in contrastive SSL is CLIP (Contrastive Language-Image Pretraining) [11], which aligns image and text modalities to learn multimodal representations. CLIP uses the InfoNCE loss to align paired image and text embeddings, relying heavily on large batch sizes to ensure a diverse set of negative pairs for effective training. Despite its success, CLIP's dependence on large batches poses significant computational challenges, making it less scalable.

To address this limitation, SogCLR [13] and CyCLIP [6] introduced modifications to the contrastive learning framework. SogCLR introduces a global contrastive loss [13] and approximates it using a stochastic optimization strategy that leverages moving average statistics, enabling contrastive learning with small batch sizes. On the other hand, CyCLIP introduces in-modal and cross-modal consistency components, improving the alignment between modalities without relying exclusively on batch size. While these methods mitigate the batch size dependency, systematic hyperparameter tuning, which is crucial for optimizing their performance, has not been thoroughly explored. For instance, prior works primarily relied on ad hoc or manual tuning approaches, limiting the reproducibility and generalizability of their results.

Motivated by this gap, our project focuses on systematically optimizing the hyperparameters of CLIP, SogCLR, and CyCLIP to improve their performance. While these models have demonstrated strong results, their potential remains underexplored due to the lack of rigorous hyperparameter tuning.

To address this, we employ Ray Tune, a scalable hyperparameter optimization library, using a HyperOpt [1] Estimator for efficient search and the ASHA [9] scheduler for scalable parameter search. We optimize key hyperparameters, such as learning rates, temperature, and consistency weights, for different models. The models were trained and validated on a 100k sample subset of the CCM3 dataset, using a 90-10 train-validation split. For evaluation, we utilized the COCO dataset to test retrieval performance and ImageNet for classification tasks. Metrics such as average retrieval accuracy (mean of top-1, top-5, and top-10 retrieval scores) and classification accuracy (top-1, top-5) are used to evaluate their performance.

Through our systematic approach, we aim to bridge the gap in hyperparameter tuning for contrastive SSL models and provide insights into optimizing their training pipelines. Our most promising results were achieved with the SogCLR model, fine-tuned using the AdamW optimizer.

2 Background and Related Works

2.1 Clip Architecture and Training

CLIP (Contrastive language-Image Pre training) uses a contrastive loss function to learn joint representation of images and text. Its architecture consists of two encoders:

- An image encoder (ResNet-50[8]) that processes visual information
- A text encoder (DistilBERT [12]) to process textual information

In the original implementation of CLIP, the loss function is formulated as :

$$L = \frac{1}{2}(L_{i2i} + L_{t2i}) \tag{1}$$

- L_{i2t} is the image-to-text loss
- L_{t2i} is the text-to-image loss

Where each directional loss is computed using cross-entropy and we treat temperature as a parameter that is learned during training.

In our project we deviate from the original implementation of CLIP and treat temperature as a hyperparameter.

A random mini-batch of m image-text pairs $\mathcal{B} = \{(x_1, z_1), \dots, (x_B, z_B)\}$ is first sampled. For each image-text pair (x_i, z_i), the mini-batch contrastive loss is defined below:

$$L(w, \tau, x_i, z_i, \mathcal{B}) = \log\left(\frac{1}{|\mathcal{B}_i|}\sum_{z_j \in \mathcal{B}_i} \ell_1(x_i, z_j; \tau)\right) + \log\left(\frac{1}{|\mathcal{B}_i|}\sum_{x_j \in \mathcal{B}_i} \ell_2(z_i, x_j; \tau)\right). \tag{2}$$

This loss contrasts the similarity score between corresponding image-text pairs (x_i, z_i) and non-corresponding image-text pairs (x_i, z_j) and (z_i, x_j) in the same batch. Then the gradient w.r.t. w is computed based on the following local contrastive loss for each image-text pairs.

$$L(w, \tau, \mathcal{B}) := \frac{1}{m}\sum_{x_i \in \mathcal{B}} L(w, \tau, x_i, \mathcal{B}). \tag{3}$$

Here τ becomes a hyperparameter that is not learned during training.

2.2 Global Contrastive objective and SogCLR

The limitation of CLIP is the need of large batch size to perform well, as it replies on the negative samples calculated from the batch to compute its loss. To Address this Yuan et al. [13] introduces a global contrastive loss where instead of sampling image-text pairs from the same batch, the entire dataset is used for sampling. This corresponds to the following **global contrastive objective (GCO)** :

$$L(w,\tau,\mathcal{D}) := \frac{1}{n}\sum_{x_i\in\mathcal{D}} L_1(w,x_i,\mathcal{D}) + \frac{1}{n}\sum_{z_i\in\mathcal{D}} L_2(w,z_i,\mathcal{D}). \tag{4}$$

SogCLR proxies this GCO using a stochastic algorithm by tracking a moving average statistics.

The following sequences are introduced:

$$\begin{aligned} u_{1,i,t} =& (1-\gamma)u_{1,i,t-1} + \gamma g_1(w_t, x_i, \mathcal{B}_{1t}^-), \\ u_{2,i,t} =& (1-\gamma)u_{2,i,t-1} + \gamma g_2(w_t, z_i, \mathcal{B}_{2t}^-). \end{aligned} \tag{5}$$

where $\mathcal{B}_t$ is the mini-batch, $\mathcal{B}_{1t}^-$ and $\mathcal{B}_{2t}^-$ denote the negative texts and images for x_i and z_i in the mini-batch, respectively. Subsequently, we update the parameters with the following gradient estimator

$$G_t = \frac{\tau}{m}\sum_{x_i\in\mathcal{B}_t}\frac{1}{\varepsilon + u_{1,i,t}}\cdot\nabla g_1(w_t,x_i,\mathcal{B}_t) + \frac{\tau}{m}\sum_{z_i\in\mathcal{B}}\frac{1}{\varepsilon + u_{2,i,t}}\cdot\nabla g_2(w_t,z_i,\mathcal{B}_t). \tag{6}$$

Finally, the model parameters are updated using any optimizer of choice.

In our project, τ and γ are the hyperparameters and are not learned during training.

2.3 CyCLIP: In-Modal and Cross-Modal Consistency

Cyclip introduces additional components to improve representation learning:

- **In-modal Consistency:** Enforces alignment within each modality (e.g., images with images, texts with texts).
- **Cross-modal Consistency:** Ensures alignment between the modalities (e.g., images with their corresponding text).

The CyCLIP loss is defined as:

$$\begin{aligned} \mathcal{L}_{\text{C-Cyclic}} &= \frac{1}{N}\sum_{j=1}^{N}\sum_{k=1}^{N}(\langle I_j^e, T_k^e\rangle - \langle I_k^e, T_j^e\rangle)^2 \\ \mathcal{L}_{\text{I-Cyclic}} &= \frac{1}{N}\sum_{j=1}^{N}\sum_{k=1}^{N}(\langle I_j^e, T_k^e\rangle - \langle T_k^e, T_j^e\rangle)^2 \\ \mathcal{L}_{\text{CYCLIP}} &= \mathcal{L}_{\text{CLIP}} + \lambda_1\mathcal{L}_{\text{I-Cyclic}} + \lambda_2\mathcal{L}_{\text{C-Cyclic}} \end{aligned} \tag{7}$$

In our project τ ,$\lambda 1$ and $\lambda 2$ are the hyperparameters and are not learned during training.

Looking at some similar works, SimCLR [3] introduced a simple framework for contrastive learning by leveraging strong data augmentations to create positive and negative pairs within a batch. It minimizes a InfoNCE loss to align embeddings of positive pairs while separating them from negatives. However, SimCLR too relies on large batch sizes for its accuracy.

MoCo[7] addresses the large batch size limitation of contrastive learning by introducing a memory bank to store negative samples. This enables contrastive learning with smaller batch sizes, as the memory bank acts as a dynamic dictionary of embeddings. MoCo achieves this by using a momentum-updated key encoder for consistency across training steps, making it computationally efficient and conceptually similar to SogCLR's optimization approach.

3 Experiments

3.1 Setup

For all CLIP variants, we use ResNet-50 [8], pretrained on ImageNet [4], as the image encoder and DistilBERT [12], pretrained on BookCorpus [14] and English Wikipedia, as the text encoder. To

Table 1: List of hyperprameters tunes for all the models. Note that we tune the learning rates (lr) for both AdamW and Nadam optimizer configurations (τ is the softmax tempeature, γ is the discount factor for SogCLR, and λ_1, λ_2 are the in-modal and cross-modal loss weights for CyCLIP). #trials represents the total number of unique hyperparameter combinations sampled by HyperOpt from the search space.

Model	Tuned Hyperparameters	#trials
CLIP	lr, τ	5
CyCLIP	lr, τ, λ_1, λ_2	10
SogCLR	lr, τ, γ	10

Table 2: Hyperparameter search space and distributions for HyperOpt search.

Hyperparameter	Distribution	Range
lr	loguniform	[1e-5, 1e-2]
τ	loguniform	[1e-3, 1e-1]
γ	loguniform	[0.5, 0.9]
λ_1	uniform	[0.25, 0.75]
λ_2	uniform	[0.25, 0.75]

ensure fair evaluation, we fix the batch size to 128 and the maximum number of epochs to 30. A 100K subset of the CC3M dataset, consisting of image-caption pairs provided in the problem statement, is used. The dataset is split into training (90K samples) and validation (10K samples) sets in a 9:1 ratio.

For validation, we define **average recall** as the mean of Recall@1, Recall@5, and Recall@10, averaged across both image-to-text and text-to-image retrieval tasks. For testing, we use two different subsets from MSCOCO and ImageNet for retrieval and zero-shot classification respectively. We used the codebase provided with the problem statement, and performed the experiments on NVIDIA A100 GPUs, supported by TAMU HPRC.

We report hyperparameter tuning, followed by best-model evaluation for both – AdamW [10] and Nadam [5] optimizers. Note that we only search for optimal learning rates of the optimizers and set other optimizer-specific hyperparameters to their default values. We use HyperOpt [2] with the Tree-structured Parzen Estimator, a Bayesian optimization algorithm, to identify the optimal hyperparameter combination that maximizes average recall on the validation set. We use ASHA Scheduler [9] with 8 maximum epochs and a grace period of 3 to efficiently scale our experiments. The grace period and maximum epochs refer to the minimum and maximum number of epochs allocated to a trial before it is either stopped or evaluated for promotion based on performance. Table 1 enlists the hyperparameters tuned for each model, and Table 2 contains their corresponding search space.

4 Results

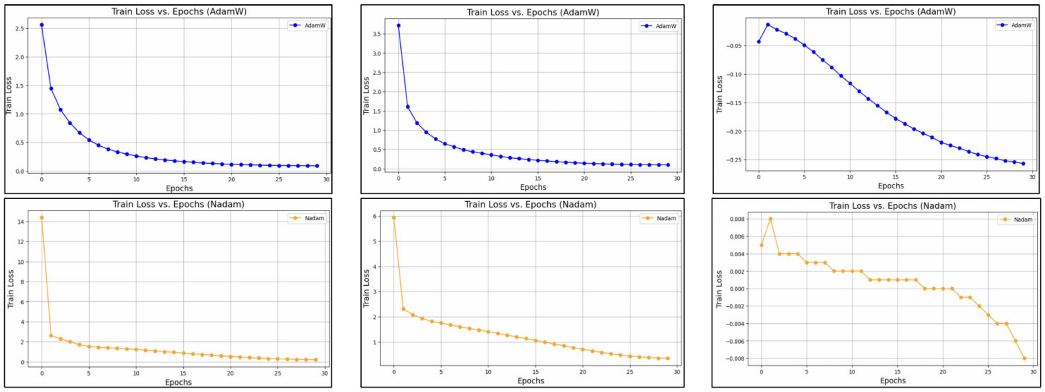
We identify the optimal hyperparameters for all model-optimizer configurations and present them in Tables 6, 7, and 8. Using these optimal hyperparameter combinations, we train the best-performing models and provide the corresponding training and validation curves in Fig. 1 and 2, and the best validation accuracies in Table 5.

1. Models trained with Nadam consistently perform significantly worse than those trained with AdamW in terms of validation accuracy.

2. Although the training loss decreases for both Nadam and AdamW, the loss curves are smoother and more stable for AdamW.

3. On the validation set, CyCLIP achieves the best performance, followed by CLIP and SogCLR.

We report the image retrieval and zero-shot classification results on COCO and ImageNet for the model achieving the highest mean recall on the validation set in Tables 3 and 4. We find the following:



(a) CLIP (b) CyCLIP (c) SogCLR

Figure 1: Training curves for (a) CLIP, (b) CyCLIP and (c) SogCLR for the best hyperparameters for both AdamW (top row) and Nadam (bottom row) optimizers.

1. SogCLR with AdamW consistently outperforms all other models across all evaluation metrics, followed by CyCLIP and SogCLR models trained with AdamW.

2. Consistent with the validation results, models trained with Nadam perform significantly worse than those trained with AdamW across both benchmarks.

3. Interestingly, although SogCLR shows the lowest validation performance, it achieves the highest test performance. This suggests a distribution shift between the validation set (curated from CC3M) and the test sets (ImageNet, COCO). This finding supports the authors' claim that iSogCLR can learn generalizable multimodal representations, even with a small batch size of 128.

4. For validation results on COCO, recall on the text-to-image retrieval task is consistently higher than on the image-to-text retrieval task, indicating potential modality bias in the dataset and/or the trained models.

Table 3: Zeroshot test performance of models with the best validation performance on COCO dataset. T2I recall and I2T recall are the average text-to-image and image-to-text recalls (averaged across top@1, top@5 and top@10). Mean recall is the average of T2I and I2T recalls.

Model	**Optimizer**	**T2I recall**	**I2T recall**	**Mean Recall**
CLIP	AdamW	28.47	23.75	26.11
	Nadam	11.97	9.54	10.76
CyCLIP	AdamW	28.91	23.70	26.31
	Nadam	13.08	10.56	11.82
SogCLR	AdamW	31.81	25.21	28.51
	Nadam	4.42	3.82	4.12

Table 4: Zeroshot test performance of models with the best validation performance on Imagenet dataset.

Model	**Optimizer**	**top@1**	**top@5**	**top@10**
CLIP	AdamW	23.24	44.14	52.90
	Nadam	5.72	15.58	21.92
CyCLIP	AdamW	23.99	44.20	52.57
	Nadam	4.51	13.81	20.15
SogCLR	AdamW	25.05	47.19	55.99
	Nadam	1.73	6.19	9.88

Table 5: Best validation recall and corresponding epoch for each model-optimizer combination trained with optimal hyperparameters. Best model highlighted in yellow.

Model	**Optimizer**	**Mean Validation Recall**	**Epoch**
CLIP	AdamW	24.66	26
	Nadam	12.30	30
CyCLIP	AdamW	25.53	30
	Nadam	12.89	30
SogCLR	AdamW	24.47	25
	Nadam	4.27	30

Table 6: Results of hyperparameter tuning for CLIP, where the selected configurations are highlighted in yellow. Columns represent learning rate (**lr**) and softmax temperature ($\boldsymbol{\tau}$).

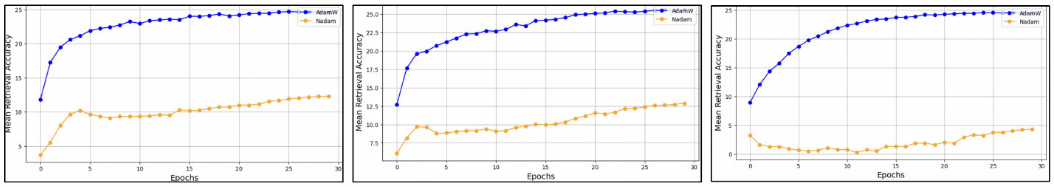
Optimizer	**lr**	$\boldsymbol{\tau}$
AdamW	3.09e-3	0.0012
	6.21e-5	0.0028
	6.73e-5	0.0657
	2.36e-3	0.0025
	7.97e-3	0.0354
Nadam	1.87e-3	0.0290
	4.16e-4	0.0134
	1.28e-5	0.0016
	2.73e-5	0.0023
	1.00e-4	0.0016

Table 7: Results of hyperparameter tuning for CyCLIP. The selected configurations for AdamW and Nadam are highlighted in yellow. Columns represent learning rate (**lr**), softmax temperature ($\boldsymbol{\tau}$), in-modal loss weight ($\boldsymbol{\lambda_1}$) and cross-model loss weight ($\boldsymbol{\lambda_2}$).

Optimizer	**lr**	$\boldsymbol{\tau}$	$\boldsymbol{\lambda_1}$	$\boldsymbol{\lambda_2}$
AdamW	1.77e-3	0.0281	0.3309	0.6436
	3.89e-5	0.0736	0.4546	0.3126
	2.96e-5	0.0179	0.2972	0.5124
	2.82e-4	0.0162	0.5940	0.4977
	6.06e-3	0.0018	0.5666	0.3165
	4.54e-4	0.0366	0.6963	0.3390
	3.31e-3	0.0900	0.6236	0.5086
	1.29e-5	0.0039	0.5289	0.3416
	6.54e-4	0.0011	0.3592	0.7432
	4.99e-4	0.0034	0.3952	0.5219
Nadam	1.99e-3	0.0223	0.4334	0.5958
	5.48e-5	0.0196	0.3317	0.4003
	1.25e-4	0.0052	0.4538	0.6589
	2.30e-4	0.0065	0.6890	0.7441
	1.40e-3	0.0087	0.4327	0.6608
	2.90e-5	0.0121	0.3738	0.4913
	1.92e-5	0.0568	0.6302	0.4824
	5.04e-4	0.0035	0.4103	0.6807
	2.47e-4	0.0417	0.6116	0.4036
	9.87e-3	0.0106	0.3707	0.6316

Table 8: Results of hyperparameter tuning for SogCLR, where the selected configurations are highlighted in yellow. Columns represent learning rate (**lr**), SogCLR gamma (γ), and temperature (τ).

Optimizer	lr	γ	τ
AdamW	3.38e-3	0.7337	0.0071
	1.42e-3	0.6090	0.0020
	2.11e-5	0.5390	0.0559
	2.60e-3	0.7195	0.0612
	8.69e-3	0.8842	0.0818
	3.28e-5	0.5265	0.0060
	1.94e-3	0.6618	0.0306
	3.43e-5	0.7359	0.0030
	5.84e-3	0.8044	0.0423
	1.03e-5	0.5016	0.0277
Nadam	3.59e-3	0.8462	0.0017
	4.18e-4	0.6235	0.0535
	3.10e-3	0.7649	0.0150
	7.56e-4	0.6635	0.0200
	1.71e-5	0.6914	0.0050
	3.55e-5	0.7729	0.0030
	3.89e-5	0.5591	0.0084
	1.41e-3	0.8738	0.0028
	5.61e-4	0.6152	0.0045
	3.87e-3	0.8128	0.0010



(a) CLIP

(b) CyCLIP

(c) SogCLR

Figure 2: Validation curves for (a) CLIP, (b) CyCLIP and (c) SogCLR for the best hyperparameters for both AdamW (blue) and Nadam (yellow) optimizers.

5 Conclusion

In this project, we conducted a systematic hyperparameter search for the CLIP model and its two popular variants – CyCLIP and SogCLR. Our findings confirm that SogCLR excels at learning more generalizable representations compared to others under restricted data and batch size settings. While our study was limited by time and resources, it highlights the valuable insights that can be uncovered through such experiments. We aim to extend this work by conducting large-scale experiments with an expanded search space, larger datasets, and additional training epochs.

6 Contributions

Praroop and Ishaan ideated and conducted all the experiments together. Both created the graphs and analyzed the results. Praroop wrote the Abstract, Introduction and Related Work sections, and Ishaan wrote the Experiment and Results sections.

References

[1] James Bergstra, Daniel Yamins, and David Cox. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In Sanjoy Dasgupta

and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 115–123, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR.

[2] James Bergstra, Daniel Yamins, and David Cox. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In *International conference on machine learning*, pages 115–123. PMLR, 2013.

[3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, pages 1597–1607, 2020.

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[5] Timothy Dozat. Incorporating nesterov momentum into adam. 2016.

[6] Shashank Goel, Hritik Bansal, Sumit Bhatia, Ryan Rossi, Vishwa Vinay, and Aditya Grover. Cyclip: Cyclic contrastive language-image pretraining. *Advances in Neural Information Processing Systems*, 35:6704–6719, 2022.

[7] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the 33rd IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[9] Liam Li, Kevin Jamieson, Afshin Rostamizadeh, Ekaterina Gonina, Jonathan Ben-Tzur, Moritz Hardt, Benjamin Recht, and Ameet Talwalkar. A system for massively parallel hyperparameter tuning. *Proceedings of Machine Learning and Systems*, 2:230–246, 2020.

[10] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

[11] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, pages 8748–8763, 2021.

[12] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[13] Zhuoning Yuan, Yuexin Wu, Zi-Hao Qiu, Xianzhi Du, Lijun Zhang, Denny Zhou, and Tianbao Yang. Provable stochastic optimization for global contrastive learning: Small batch does not harm performance. In *International Conference on Machine Learning*, pages 25760–25782. PMLR, 2022.

[14] Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.