# Appendix for Class Adaptive Network Calibration

## A. Penalty functions for ALM

Here, we provide the requirements for a penalty function in Augmented Lagrangian Multiplier (ALM) method.

A function $P : \mathbb{R} \times \mathbb{R}_{++} \times \mathbb{R}_{++} \to \mathbb{R}$ is a Penalty-Lagrangian function such that $P'(z, \rho, \lambda) = \frac{\partial}{\partial y} P(z, \rho, \lambda)$ exists and is continuous for all $z \in \mathbb{R}$, $\rho \in \mathbb{R}_{++}$ and $\lambda \in \mathbb{R}_{++}$. In addition, it should satisfy the following four axioms [1]:

**Axiom 1:** $P'(z, \rho, \lambda) \geq 0 \quad \forall z \in \mathbb{R}, \rho \in \mathbb{R}_{++}, \lambda \in \mathbb{R}_{++}$

**Axiom 2:** $P'(z, \rho, \lambda) = \lambda \quad \forall \rho \in \mathbb{R}_{++}, \lambda \in \mathbb{R}_{++}$

**Axiom 3:** If, for all $j \in \mathbb{N}$, $0 < \lambda_{\text{min}} \leq \lambda^{(j)} \leq \lambda_{\text{max}} < \infty$, then: $\lim\limits_{j \to \infty} \rho^{(j)} = \infty$ and $\lim\limits_{j \to \infty} z^{(j)} > 0$ imply that $\lim\limits_{j \to \infty} P'(z^{(j)}, \rho^{(j)}, \lambda^{(j)}) = \infty$

**Axiom 4:** If, for all $j \in \mathbb{N}$, $0 < \lambda_{\text{min}} \leq \lambda^{(j)} \leq \lambda_{\text{max}} < \infty$, then: $\lim\limits_{j \to \infty} \rho^{(j)} = \infty$ and $\lim\limits_{j \to \infty} z^{(j)} < 0$ imply that $\lim\limits_{j \to \infty} P'(z^{(j)}, \rho^{(j)}, \lambda^{(j)}) = 0$

where the first two axioms guarantee the derivative of the Penalty-Lagrangian function $P$ w.r.t. $z$ is positive and equals to $\lambda$ when $z = 0$, while the last two axioms guarantee that the derivative tends to infinity when the constraint is not satisfied, and zero when the constraint holds.

There are many valid penalty functions [1]. In this paper we adopt the PHR function suggested by [1] and confirmed by our empirical results in Section 5 of the main text. We also empirically compare with another two popular choices, *i.e*. P2 and P3 [1], as shown in Figure 3 of the main text. The formulations of the above three penalty functions are as follows:

$$\mathrm{PHR}(z, \rho, \lambda) = \begin{cases} \lambda z + \frac{1}{2}\rho z^2 & \text{if} \quad \lambda + \rho z \geq 0; \\ -\frac{\lambda^2}{2\rho} & \text{otherwise.} \end{cases} \tag{1}$$

$$P_2(z, \rho, \lambda) = \begin{cases} \lambda z + \lambda \rho z^2 + \frac{1}{6}\rho^2 z^3 & \text{if} \quad z \geq 0 \\ \frac{\lambda z}{1 - \rho z} & \text{if} \quad z \leq 0 \end{cases} \tag{2}$$

$$P_3(z, \rho, \lambda) = \begin{cases} \lambda z + \lambda \rho z^2 & \text{if} \quad z \geq 0 \\ \frac{\lambda z}{1 - \rho z} & \text{if} \quad z \leq 0 \end{cases} \tag{3}$$

## B. Dataset description with implementation details

**Tiny-ImageNet** [4] is a standard benchmark for image classification and commonly used in the calibration literature [11, 18]. It includes $64 \times 64$ dimensional images across 200 classes, with 500 images per class in the train set and 50 per class in the validation set. Following [18], we split out a validation set by randomly choose 50 samples per class from the train set, while the original validation set is used as the test set. We train ResNet-50 [7] model by SGD optimizer with a batch size of 128, and the number of epochs is set to 100 . A multi-step learning rate scheduling strategy is used, *i.e.* learning rate 0.1 for the first 40 epochs, 0.01 for the next 20 epochs and 0.001 for the rest.

**ImageNet** [4] is a large-scaled image classification benchmark. We use the version of ILSVRC-2012 (or ImageNet-1K) in our experiments (referred as ImageNet in this paper). It consists of 1K object classes with 1.2M images for training and 5K for validation. The average resolution of an image is $469 \times 387$. We follow [17] for evaluating calibration performance on ImageNet, *i.e.* reserving $20\%$ for validation and the remaining $80\%$ for testing. Besides ResetNet-50 [7], we also train state-of-the-art transformer based network, *i.e.* SwinV2-T [12], on this dataset. AdamW [15] optimizer is applied, and a cosine learning rate scheduler [14] with an initial learning rate of 0.001 is used. The number of training epochs is set to 200 and 300 for ResNet-50 and SwinV2-T respectively. The input size is $224 \times 224$ for ResNet-50 and $256 \times 256$ for SwinV2-T, while the batch size is 1024 for training both networks. Regular data augmentation techniques like random resized crop, random horizontal flips, random color jitter, and random pixel erasing are applied on the training samples.

**ImageNet-LT** [13] is truncated from ImageNet by sampling a subset so that the labels of the training set follow a long-tailed distribution. Overall, it has 115.8K images belonging to 1K classes, and the number of samples per class ranges from 5 to 1280. Both the validation and test sets are balanced, where the validation set includes 20 images per class and the original validation set in ImageNet is employed as the test set. Regarding the networks and training details, we use the same settings as those on ImageNet.

**PASCAL VOC2012** [5] is a natural semantic segmentation

| | TinyImageNet | | ImageNet | | ImageNet-LT | | 20 News | |
|---|---|---|---|---|---|---|---|---|
| Method | Pre-TS | Post-TS | Pre-TS | Post-TS | Pre-TS | Post-TS | Pre-TS | Post-TS |
| CE | 3.73 | $1.86_{1.1}$ | 9.19 | $3.88_{1.6}$ | 28.12 | $3.72_{1.7}$ | 22.75 | $3.01_{3.1}$ |
| LS | 3.17 | $1.79_{0.9}$ | 2.57 | $2.57_{1.0}$ | 10.46 | $3.32_{1.3}$ | 8.07 | $3.69_{1.2}$ |
| FL | 2.96 | $1.74_{0.9}$ | 1.60 | $1.60_{1.0}$ | 18.37 | $2.52_{1.5}$ | 10.80 | $3.33_{1.4}$ |
| FLSD | 2.91 | $1.74_{0.9}$ | 2.08 | $2.08_{1.0}$ | 17.77 | $3.40_{1.4}$ | 10.87 | $4.10_{1.4}$ |
| CPC | 4.88 | $2.66_{1.5}$ | 3.66 | $2.00_{1.1}$ | 16.00 | $3.22_{1.2}$ | 9.46 | $4.35_{1.4}$ |
| MbLS | 1.64 | $1.64_{1.0}$ | 4.44 | $2.07_{1.1}$ | 6.16 | $2.60_{1.1}$ | 5.40 | $2.09_{1.1}$ |
| CALS-HR | 2.50 | $1.82_{0.9}$ | 5.63 | $1.68_{1.4}$ | 2.83 | $2.83_{1.0}$ | 6.99 | $3.14_{1.1}$ |
| **CALS-ALM** | **1.54** | $\mathbf{1.54}_{1.0}$ | **1.46** | $\mathbf{1.28}_{1.1}$ | **2.15** | $\mathbf{1.81}_{0.9}$ | **2.04** | $\mathbf{1.86}_{1.1}$ |

Table 1. Calibration performance (ECE in %) when adding post temperature scaling (best T value for each method in subscript). The architecture is fixed to ResNet-50 for the vision datasets and GPCN for 20 News dataset.

benchmark including 20 foreground object classes and an additional background class. As the original test set is not publicly released and it is unable to evaluate the calibration performance via the official evaluation server, we split out a validation set by randomly selecting 20% images from the training set and treat the original validation set as our test set. Overall, the training/validation/test split contains $1171/293/1449$ images. For segmentation model training, we employ DeepLabV3 [2] implemented by the popular public library[1], where we use ResNet-34 as encoder initialized with pre-trained weights on ImageNet, and the decoder is trained from scratch. The batch size is set to 8 and AdamW optimizer is used with an initial learning rate of 0.001 alongside a cosine learning rate scheduler. Finally, the maximum training epoch is set to 100.

**20 Newsgroups** [9]. To evaluate the generalization of the proposed method, we include a non-vision dataset, *i.e*. 20 Newsgroups, which is a text classification benchmark and also used in previous calibration papers [11, 18]. It contains $20K$ news articles from 20 different groups according to the content, *e.g*. rec.motorcycles, rec.autos, sci.space, etc. We use the standard data split setting : $15,098$ documents for training, 900 for validation and $3,999$ for testing. The Glove word embedding [19] is used to encode the text and then a Global Pooling Convolutional Network (GPCN) [10] is trained. During training, we use Adam optimizer with an initial learning rate of 0.001. We train the model for 100 epochs, where the learning rate is decayed by a factor of 0.1 after the first 50 epochs.

## C. Additional results

Table 1 reports the results of post-training temperature scaling (post-TS) on the outputs of the trained models [6]. Since this post-process technique is orthogonal to training based methods, we also present the results of applying it to our method, as well as the related works. We can see that our method without temperature scaling (pre-TS) outperforms previous methods, even post-TS, across all the benchmarks. Additionally, the ECE of our model is further reduced with post-TS in some cases, for instance on ImageNet (1.46% → 1.28%) and ImageNet-LT (2.04% → 1.81%).

| Method | ImageNet | ImageNet-LT |
|---|---|---|
| CE | 0.036 | 0.090 |
| LS | 0.029 | 0.072 |
| FL | 0.030 | 0.087 |
| FLSD | 0.029 | 0.087 |
| CPC | 0.049 | 0.078 |
| MbLS | 0.030 | 0.072 |
| CALS-HR | 0.029 | 0.071 |
| **CALS-ALM** | **0.027** | **0.069** |

Table 2. Class-wise Calibration Error (CWCE in %) computed for different approaches on ImageNet and ImageNet-LT. The architecture is fixed to ResNet-5. Best method is highlighted in bold.

Table 2 reports the performance on the two natural image datasets, *i.e*. ImageNet and ImageNet-LT, in terms of Class-wise Calibration Errors (CWCE) [16], which is a class-wise extension of ECE. Our method consistently achieve the best scores, with relative improvements of 25.0% on ImageNet and 23.3% on ImageNet-LT.

In Table 3, we present results on the out-of-distribution (OOD) scenario [17]. It is shown nn both settings, our method achieves the lowest ECE on the target domain. These

| | CE | LS | FL | MbLS | **Ours** |
|---|---|---|---|---|---|
| ImageNet → ImageNet-C | 26.25 | 24.00 | 23.73 | 26.55 | **22.52** |
| ImageNet-LT → ImageNet-C | 19.99 | 27.51 | 15.80 | 15.40 | **12.91** |

Table 3. ECE (%) on the out-of-distribution dataset, *i.e*. ImageNet-C (Gaussian noise corruption with severity level 5), for models trained on in-distribution datasets, *i.e*. ImageNet and ImageNetLT.

[1] https://github.com/qubvel/segmentation_models.pytorch

| Hyper-parameter | Value |
|---|---|
| Margin $m$ (all vision tasks) | 10 |
| Margin $m$ (text classification) | 6 |
| Initial multiplier $\boldsymbol{\lambda}^{(0)}$ | $10^{-6} \cdot \mathbf{1}_K$ |
| Initial Penalty parameter $\boldsymbol{\rho}^{(0)}$ | $\mathbf{1}_K$ |
| Penalty increasing factor $\gamma$ | 1.2 |
| Constraint improvement factor $\tau$ | 0.9 |
| Period of penalty parameter update | 10 |

Table 4. Hyper-parameters for our method, *i.e*. CALS-ALM.

results confirm the effectiveness of our method in the OOD scenario.

## D. Visualization of learned classwise multipliers

Figure 1 shows the evolution of learned multipliers $\lambda_k$ on ImageNet for the three classes with the highest average and the three classes with the lowest average. This highlights the advantages of our method: 1) assigning distinct penalty weights for different classes; 2) adaptively updating the weight for each class throughout the training process.

## E. Reliability diagram

Figure 2 presents the reliability diagrams for different models trained on ImageNet and ImageNet-LT, which is a standard way of visualizing calibration performance. The curve of a perfectly calibrated model in the reliability diagram should match the dashed red line, where the prediction confidence perfectly reflects the accuracy of the model. It is shown that the models trained with CE (*left-most plots*) are over-confident, with accuracy mostly lower than confidence. Our method, CALS, is the most effective one to pull the curves closer to the expected lines, showing nearly perfect calibration performances. In particular, the improvement on ImageNet-LT is substantial compared to the other methods like LS and FL, which further demonstrates that the proposed class adaptive learning method could address the class imbalance issue in the long-tailed dataset. On ImageNet, LS and FL also present strong calibration performance, but decrease the final accuracy as shown in Table 1 of the main text. Overall, our method achieves the best compromise between calibration and accuracy. It is noted that the observation from Figure 2 is supported by the quantitative scores reported in Table 1 of the main text.

## F. Hyper-parameter setting

Table 4 gives details of the hyper-parameter settings in our method, *i.e*. CALS-ALM. Note, the margin values are set by following [11], *i.e*. 10 for all the vision tasks including classification and segmentation, and 6 for the text classification on 20 Newsgroups.

Regarding the other methods in Table 1 of the main text, we set their hyper-parameters by following previous works, except that the values for MMCE [8] and CPC [3] are empirically set according to our implementation. Detailed hyper-parameter settings for each method are as follows:

- MMCE [8]: balancing weight $\lambda = 0.1$.
- ECP [20]: balancing weight $\lambda = 0.1$.
- LS [21]: smoothing factor $\alpha = 0.05$.
- FL [18]: scaling factor $\gamma = 3$
- FLSD [18]: scaling factor $\gamma$ is set to $5$ for $s_k \in [0, 0.2)$ and $3$ for $s_k \in [0.2, 1)$, where $k$ is the right class for the sample.
- CPC [3]: balancing weights for the binary discrimination penalty and binary exclusion penalty are set to $10$ and $1$ respectively. It is noted that we re-implement CPC since the official code is not publicly available.
- MbLS [11]: balancing weight $\lambda = 0.1$, margin $m = 10$ for all the vision tasks, and $m = 6$ for the text classification task.

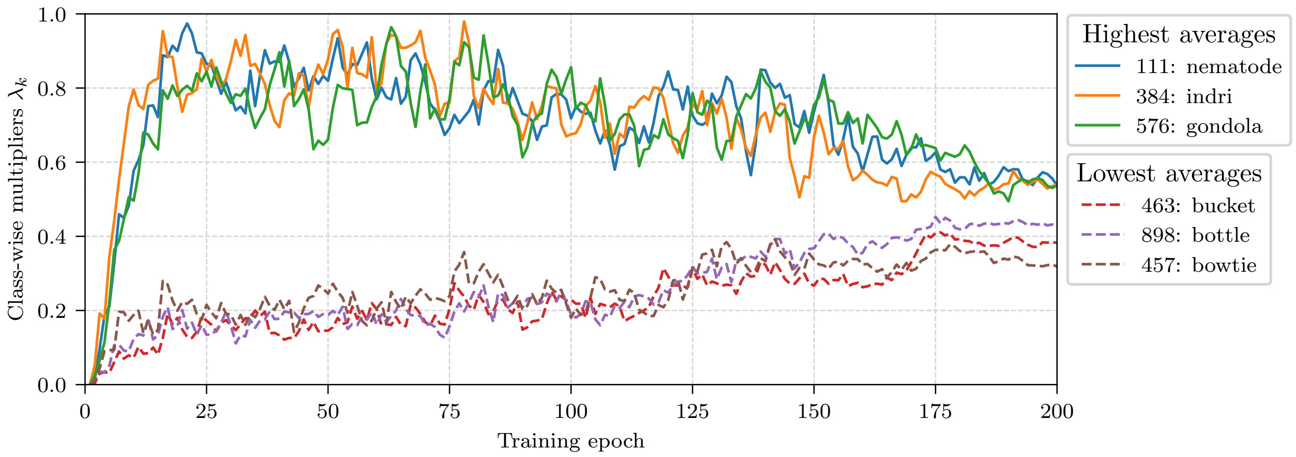

Figure 1. Visualization of learned multipliers $\lambda_k$ during the training of the ResNet-50 model on ImageNet. We show classes with the highest average (*Solid lines*) and the lowest average (*dashed lines*).

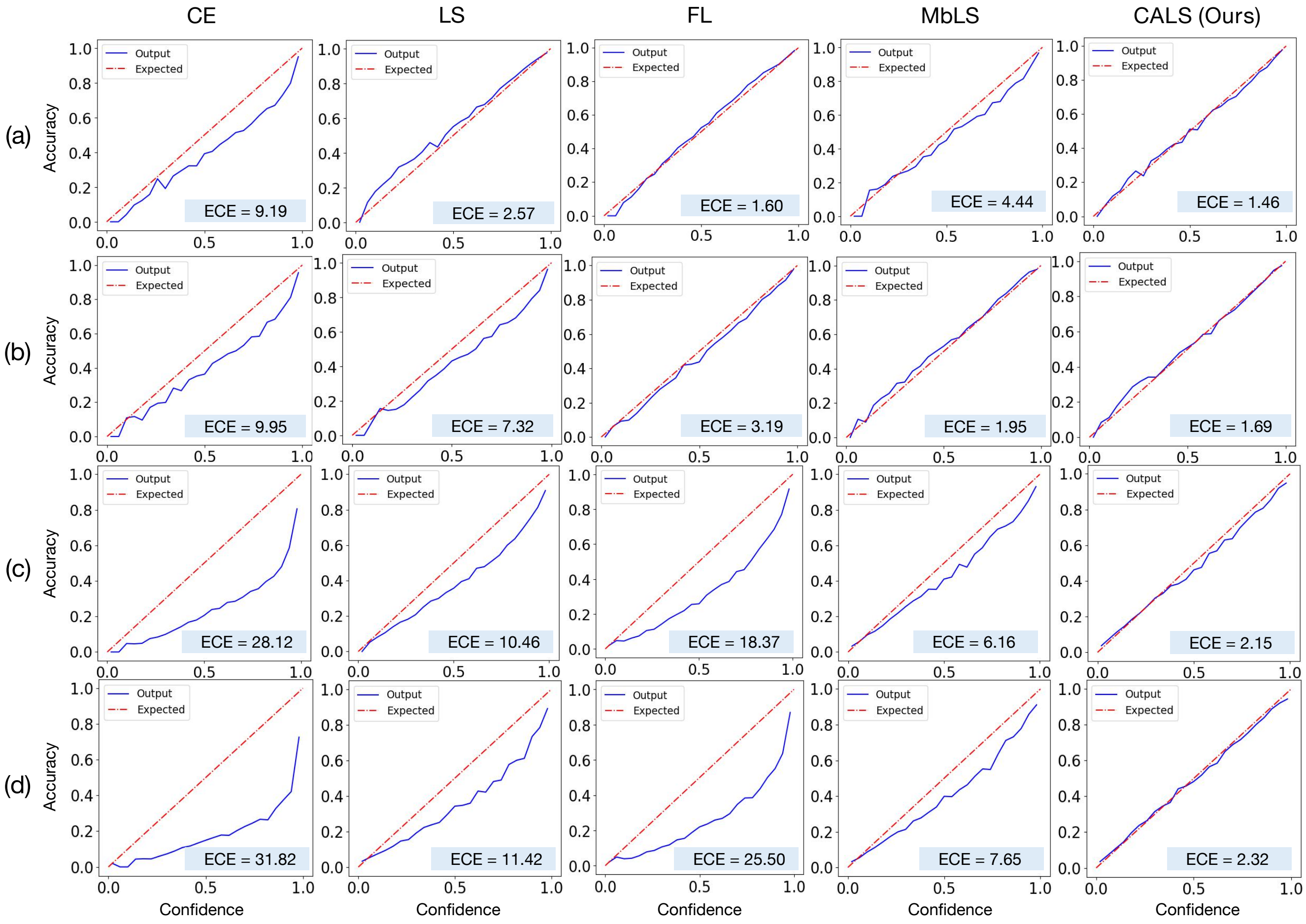

Figure 2. **Calibration visualizations: (a) ImageNet (ResNet-50), (b) ImageNet (SwinV2-T), (c) ImageNet-LT (ResNet-50), and (d) ImageNet-LT (SwinV2-T) .** We present the reliability diagrams of our method (CALS), compared with those of baselines and closely related works. The number of bins to plot reliability diagrams is set to 25.

# References

[1] Ernesto G Birgin, Romulo A Castillo, and José Mario Martínez. Numerical comparison of augmented lagrangian algorithms for nonconvex problems. *Computational Optimization and Applications*, 31(1), 2005. 1

[2] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. In *CVPR*, 2017. 2

[3] Jiacheng Cheng and Nuno Vasconcelos. Calibrating deep neural networks by pairwise constraints. In *CVPR*, 2022. 3

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 1

[5] Mark Everingham, S. M. Eslami, Luc Gool, Christopher K. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *IJCV*, 111(1):98–136, 2015. 1

[6] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *ICML*, 2017. 2

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1

[8] Aviral Kumar, Sunita Sarawagi, and Ujjwal Jain. Trainable calibration measures for neural networks from kernel mean embeddings. In *ICML*, 2018. 3

[9] Ken Lang. Newsweeder: Learning to filter netnews. In *ICML*, 1995. 2

[10] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. In *ICLR*, 2014. 2

[11] Bingyuan Liu, Ismail Ben Ayed, Adrian Galdran, and Jose Dolz. The devil is in the margin: Margin-based label smoothing for network calibration. In *CVPR*, 2022. 1, 2, 3

[12] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, Furu Wei, and Baining Guo. Swin transformer v2: Scaling up capacity and resolution. In *CVPR*, 2022. 1

[13] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X. Yu. Large-scale long-tailed recognition in an open world. In *CVPR*, 2019. 1

[14] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *ICLR*, 2017. 1

[15] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 1

[16] Lena Maier-Hein et al. Metrics reloaded: Pitfalls and recommendations for image analysis validation. *arXiv preprint arXiv:2206.01653*, 2022. 2

[17] Matthias Minderer, Josip Djolonga, Rob Romijnders, Frances Hubis, Xiaohua Zhai, Neil Houlsby, Dustin Tran, and Mario Lucic. Revisiting the calibration of modern neural networks. In *NeurIPS*, 2021. 1, 2

[18] Jishnu Mukhoti, Viveka Kulharia, Amartya Sanyal, Stuart Golodetz, Philip HS Torr, and Puneet K Dokania. Calibrating deep neural networks using focal loss. In *NeurIPS*, 2020. 1, 2, 3

[19] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *EMNLP*, 2014. 2

[20] Gabriel Pereyra, George Tucker, Jan Chorowski, Łukasz Kaiser, and Geoffrey Hinton. Regularizing neural networks by penalizing confident output distributions. In *ICLR*, 2017. 3

[21] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016. 3