Ahmed H. Salamah
ahamsala@uwaterloo.ca

Department of Electrical and Computer Engineering
University of Waterloo

UNIVERSITY OF WATERLOO

JPEG Inspired Deep Learning

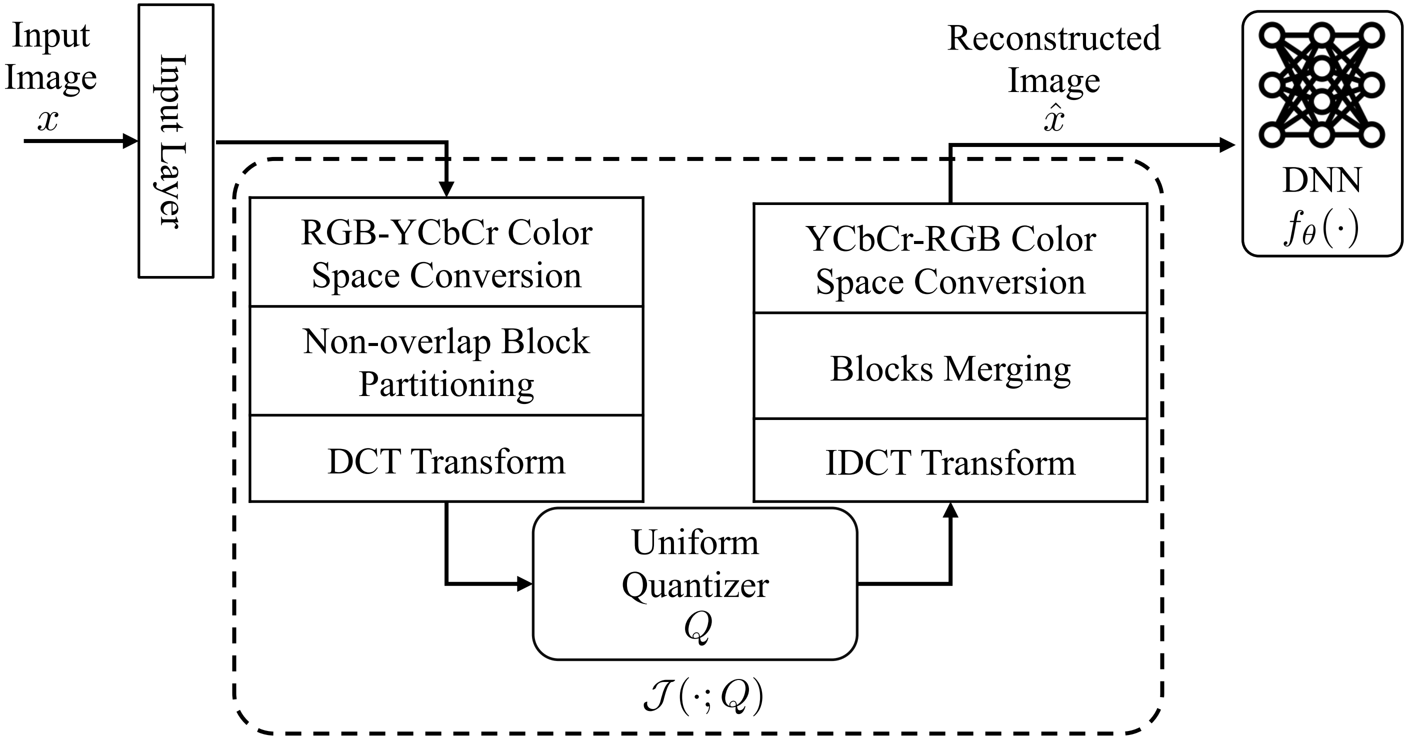
Ahmed H. Salamah[1] , Kaixiang Zheng[1] , Yiwen Liu & En-Hui Yang

ICLR
International Conference On Learning Representations

The Thirteenth International Conference on Learning Representations, Singapore, 2025

[1] Authors contributed equally.

Introduction



JPEG Compression Pipeline given an underlying model f_θ

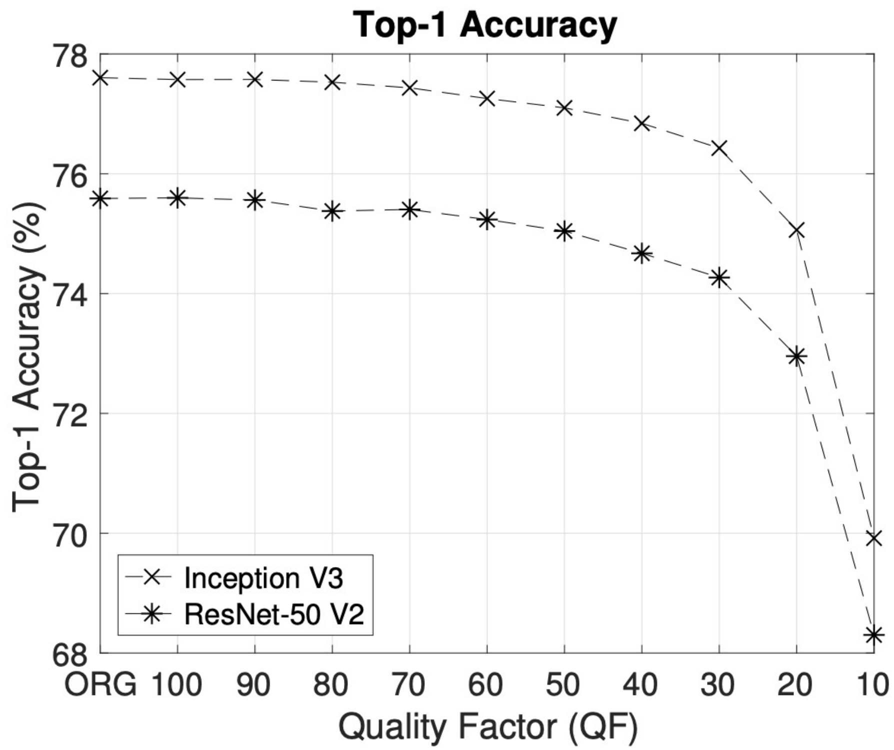
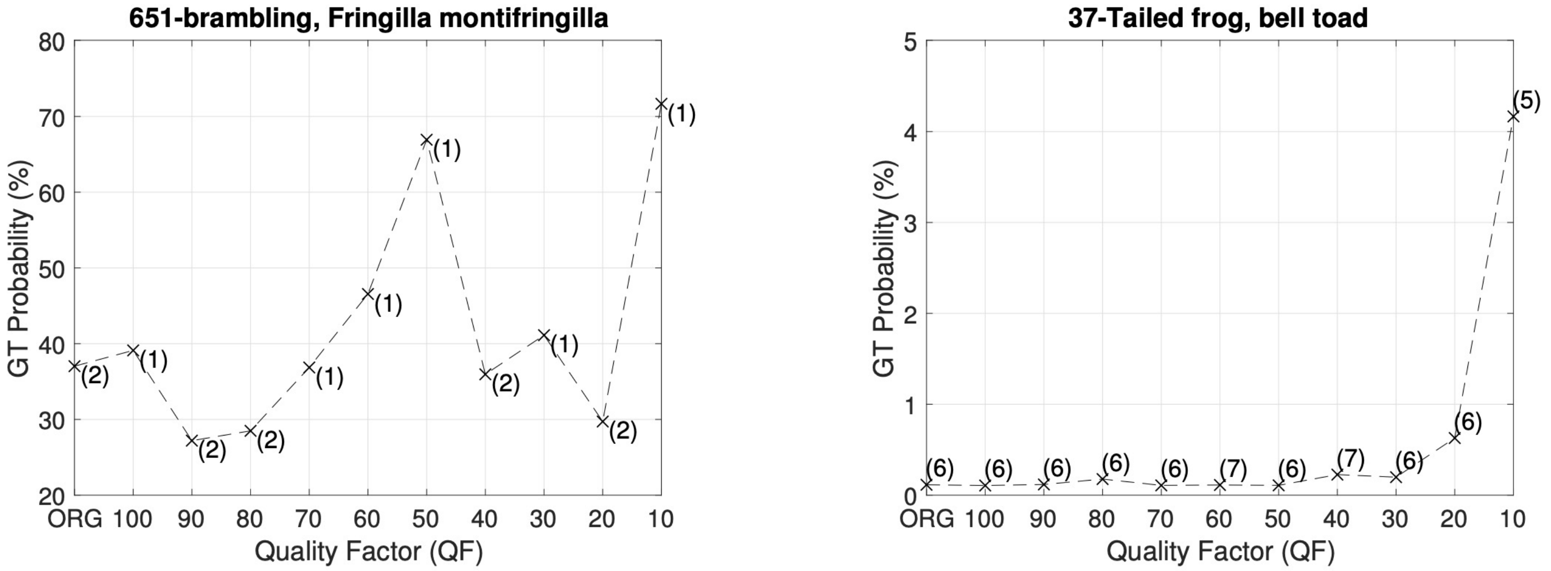


Fig. Top 1 accuracy accuracy degradation phenomenon for Inception V3 and ResNet-50 V2 in the case of the "one QF vs. all images" approach.

1. Samuel Dodge and Lina Karam. Understanding how image quality affects deep neural networks. In 2016 eighth international conference on QoMEX, pp. 1–6. IEEE, 2016.
2. Zihao Liu, Tao Liu, Wujie Wen, Lei Jiang, Jie Xu, Yanzhi Wang, and Gang Quan. DeepN-JPEG: A deep neural network favorable JPEG-based image compression framework. In Proceedings of the 55th Annual Design Automation Conference, pp. 18. ACM, 2018.
3. En-Hui Yang, Hossam Amer, and Yanbing Jiang. Compression helps deep learning in image classification. Entropy, 23(7):881, 2021.

Motivation



(**a**) Image # 651, GT label: Brambling

(**b**) Image # 37, GT label: Tailed Frog

Fig. The perspective of one image vs. all QFs—the ranks and probabilities of the GT label of an image across different QFs: (a) Image # 651; and (b) Image # 37.

1. En-Hui Yang, Hossam Amer, and Yanbing Jiang. Compression helps deep learning in image classification. Entropy, 23(7):881, 2021.
2. Kaixiang Zheng, Ahmed H. Salamah, Linfeng Ye, and En-Hui Yang. Jpeg compliant compression for dnn vision. In 2023 IEEE International Conference on Image Processing (ICIP), pp. 1875–1879, 2023.
3. Ahmed H Salamah, Kaixiang Zheng, Linfeng Ye, and En-Hui Yang. Jpeg compliant compression for dnn vision. IEEE Journal on Selected Areas in Information Theory, 2024b.

Problem Formulation

In supervised learning, each $x \in \mathcal{X}$ corresponds to a ground truth label $y \in \mathcal{Y}$. Let f_θ represent a DNN model with trainable weights θ, and let $\mathcal{L}$ denote the loss function used to train this DNN. In standard DL, the primary objective is to solve the following minimization problem:

$$\min_{\theta} \ \mathbb{E}[\mathcal{L}(f_\theta(x), y)]. \tag{1}$$

In contrast, JPEG-DL tries to improve the performance of DNN by jointly training it with the JPEG operation. As a result, the formulation should be instead:

$$\min_{\theta, Q} \ \mathbb{E}[\mathcal{L}(f_\theta(\mathcal{J}(x; Q)), y)]. \tag{2}$$

However, in order to solve (2) with gradient descent, the key challenge is caused by the non-differentiable quantization operation, which makes the gradients w.r.t. Q almost zero everywhere. To address this issue, we will introduce a *differentiable soft quantizer* (Q_d) in the next slides, replacing the uniform quantizer (Q_u) used in $\mathcal{J}$.

Differentiable Soft Quantizer (1/2)

Denote the index set of uniform quantization as

$$\mathcal{A} = \{-L, -L+1, \ldots, 0, \ldots, L-1, L\}.$$

For convenience, $\mathcal{A}$ is also regarded as a vector of length $2L+1$. Multiplying $\mathcal{A}$ with a quantization step size q, we get the corresponding reconstruction space

$$\hat{\mathcal{A}} = q \times [-L, -L+1, \ldots, 0, \ldots, L-1, L].$$

Again, we will regard $\hat{\mathcal{A}}$ as both a vector and a set.
To randomly quantize a DCT coefficient z to an element in $\hat{\mathcal{A}}$, we invoke from Yang *et al.* a trainable conditional probability mass function (CPMF) $P_\alpha(\cdot|z)$ over the reconstruction space $\hat{\mathcal{A}}$ or equivalently the index set $\mathcal{A}$ given z, where $\alpha > 0$ is a trainable parameter:

$$P_\alpha(iq|z) = \frac{e^{-\alpha(z-iq)^2}}{\sum_{j\in\mathcal{A}} e^{-\alpha(z-jq)^2}}, \ \forall i \in \mathcal{A}. \tag{3}$$

Extend z to a vector of length $2L+1$, i.e., $[z]_{2L+1} = [\ \overbrace{z, \ldots, z}^{2L+1 \text{ times}}\ ]$. Then, the CPMF $P_\alpha(\cdot|z)$, regarded as a vector of length $2L+1$, can be easily computed via the softmax operation $\sigma(\cdot)$:

$$\big[P_\alpha(\cdot|z)\big]_{2L+1} = \sigma\Big(-\alpha \times \Big(\big[z\big]_{2L+1} - \hat{\mathcal{A}}\Big)^2\Big). \tag{4}$$

Differentiable Soft Quantizer (2/2)

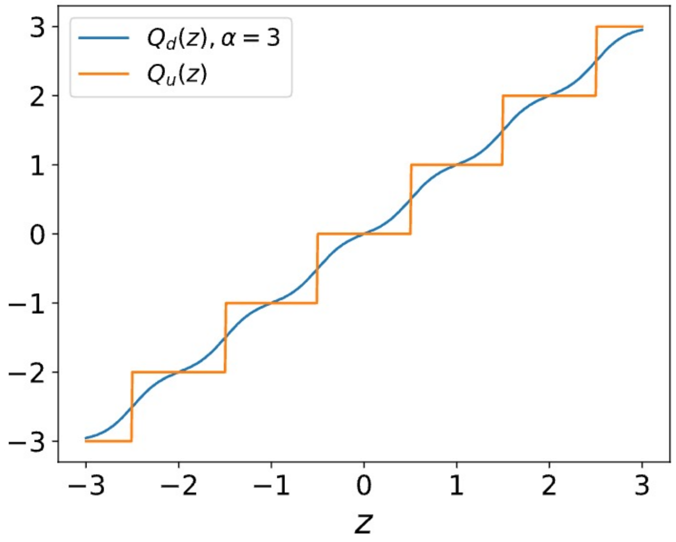
$$P_\alpha(iq|z) = \frac{e^{-\alpha(z-iq)^2}}{\sum_{j\in\mathcal{A}} e^{-\alpha(z-jq)^2}}, \ \forall i \in \mathcal{A}. \tag{3}$$

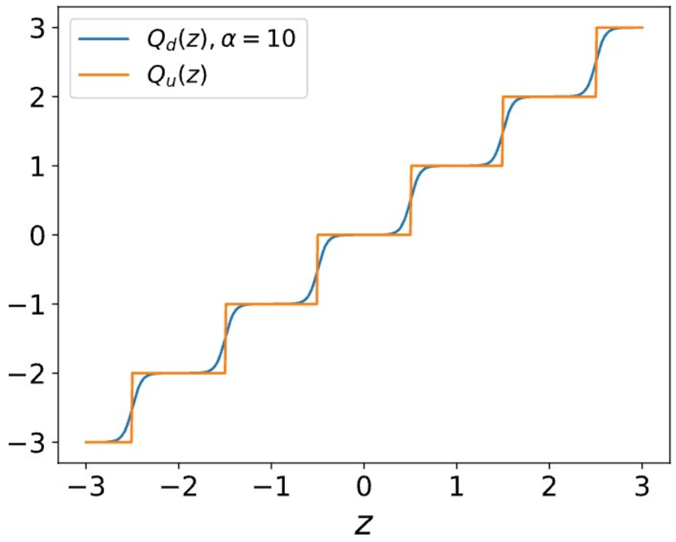
$$\big[P_\alpha(\cdot|z)\big]_{2L+1} = \sigma\Big(-\alpha\times\Big(\big[z\big]_{2L+1}-\hat{\mathcal{A}}\Big)^2\Big) \tag{4}$$

With the CPMF $P_\alpha(\cdot|z)$, z is now quantized to each $iq \in \hat{\mathcal{A}}$ with probability $P_\alpha(iq|z)$. Note that as $\alpha \to \infty$, $P_\alpha(\cdot|z)$ approaches an one-hot vector with probability 1 at the nearest point to z in $\hat{\mathcal{A}}$ and 0 elsewhere. Therefore, the resulting random quantizer effectively functions as the deterministic uniform quantizer $Q_u(z) = \lfloor z/q \rceil \cdot q$.
Based on the CPMF $P_\alpha(\cdot|z)$, we can now define a differentiable soft quantizer Q_d as the conditional expectation of iq given z, i.e.,

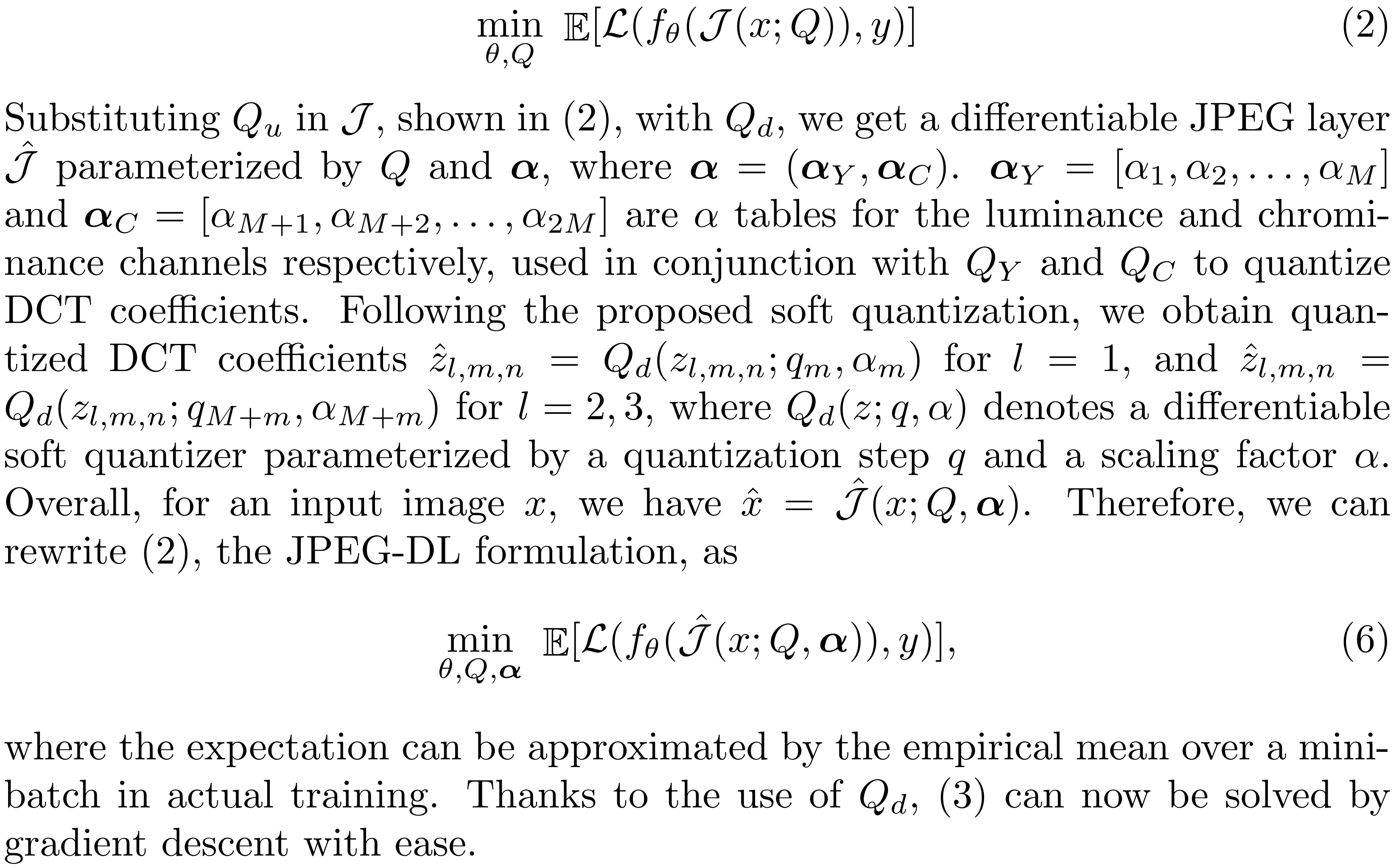
$$Q_d(z) = \mathbb{E}[iq|z] = \sum_{i\in\mathcal{A}} P_\alpha(iq|z)\cdot iq. \tag{5}$$

Similarly, as $\alpha \to \infty$, Q_d also goes to Q_u. On the left, the figures show how the shape of Q_d varies w.r.t α, given a fixed q.





Overall Framework of JPEG-DL



$$\min_{\theta,Q} \ \mathbb{E}[\mathcal{L}(f_\theta(\mathcal{J}(x;Q)),y)] \tag{2}$$

Substituting Q_u in $\mathcal{J}$, shown in (2), with Q_d, we get a differentiable JPEG layer $\hat{\mathcal{J}}$ parameterized by Q and $\boldsymbol{\alpha}$, where $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_Y, \boldsymbol{\alpha}_C)$. $\boldsymbol{\alpha}_Y = [\alpha_1, \alpha_2, \ldots, \alpha_M]$ and $\boldsymbol{\alpha}_C = [\alpha_{M+1}, \alpha_{M+2}, \ldots, \alpha_{2M}]$ are α tables for the luminance and chrominance channels respectively, used in conjunction with Q_Y and Q_C to quantize DCT coefficients. Following the proposed soft quantization, we obtain quantized DCT coefficients $\hat{z}_{l,m,n} = Q_d(z_{l,m,n}; q_m, \alpha_m)$ for $l = 1$, and $\hat{z}_{l,m,n} = Q_d(z_{l,m,n}; q_{M+m}, \alpha_{M+m})$ for $l = 2, 3$, where $Q_d(z; q, \alpha)$ denotes a differentiable soft quantizer parameterized by a quantization step q and a scaling factor α. Overall, for an input image x, we have $\hat{x} = \hat{\mathcal{J}}(x; Q, \boldsymbol{\alpha})$. Therefore, we can rewrite (2), the JPEG-DL formulation, as

$$\min_{\theta,Q,\boldsymbol{\alpha}} \ \mathbb{E}[\mathcal{L}(f_\theta(\hat{\mathcal{J}}(x;Q,\boldsymbol{\alpha})),y)], \tag{6}$$

where the expectation can be approximated by the empirical mean over a minibatch in actual training. Thanks to the use of Q_d, (3) can now be solved by gradient descent with ease.

JPEG-DL on CIFAR100 and ImageNet

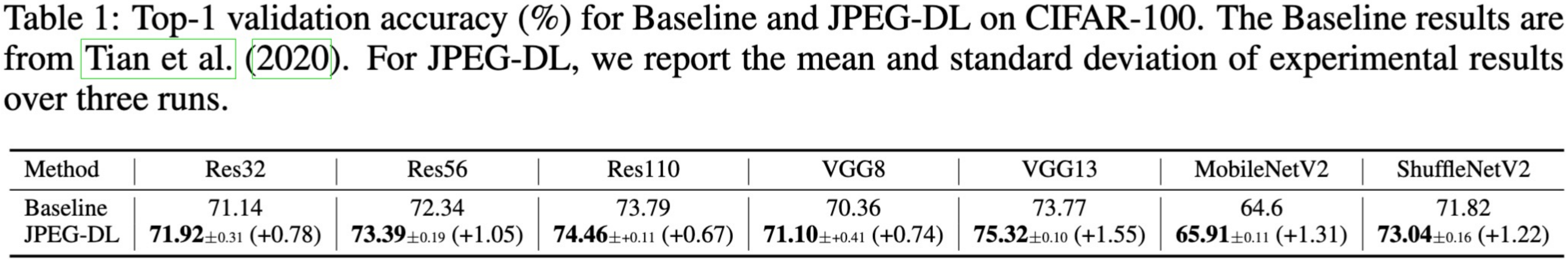


Table 1: Top-1 validation accuracy (%) for Baseline and JPEG-DL on CIFAR-100. The Baseline results are from Tian et al. (2020). For JPEG-DL, we report the mean and standard deviation of experimental results over three runs.

Method	Res32	Res56	Res110	VGG8	VGG13	MobileNetV2	ShuffleNetV2
Baseline	71.14	72.34	73.79	70.36	73.77	64.6	71.82
JPEG-DL	**71.92**$_{\pm 0.31}$ (+0.78)	**73.39**$_{\pm 0.19}$ (+1.05)	**74.46**$_{\pm +0.11}$ (+0.67)	**71.10**$_{\pm +0.41}$ (+0.74)	**75.32**$_{\pm 0.10}$ (+1.55)	**65.91**$_{\pm 0.11}$ (+1.31)	**73.04**$_{\pm 0.16}$ (+1.22)

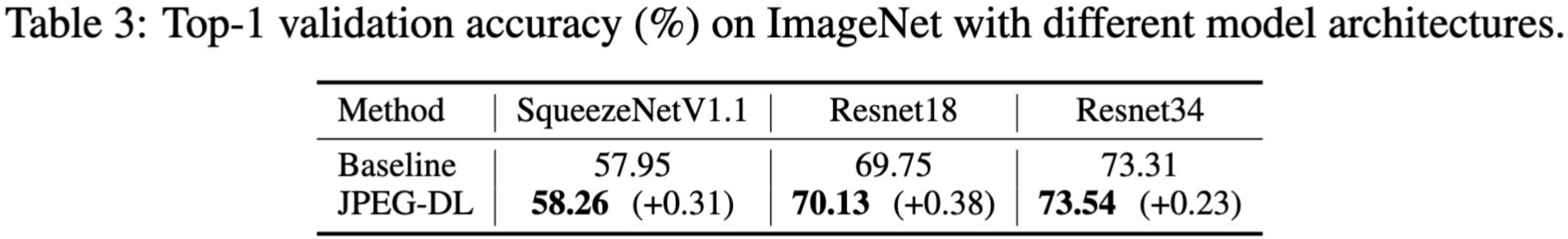
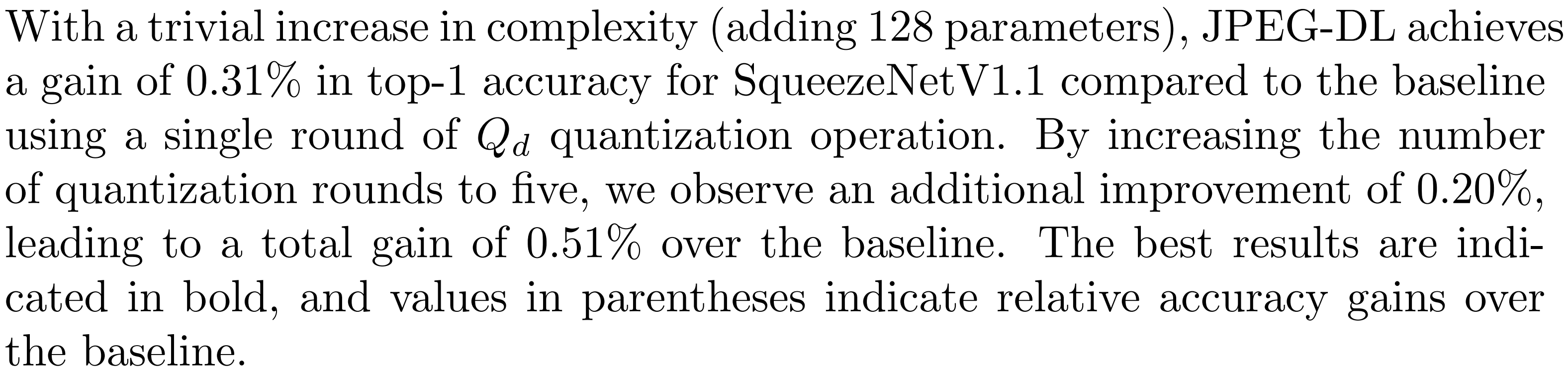


Table 3: Top-1 validation accuracy (%) on ImageNet with different model architectures.

Method	SqueezeNetV1.1	Resnet18	Resnet34
Baseline	57.95	69.75	73.31
JPEG-DL	**58.26** (+0.31)	**70.13** (+0.38)	**73.54** (+0.23)



With a trivial increase in complexity (adding 128 parameters), JPEG-DL achieves a gain of 0.31% in top-1 accuracy for SqueezeNetV1.1 compared to the baseline using a single round of Q_d quantization operation. By increasing the number of quantization rounds to five, we observe an additional improvement of 0.20%, leading to a total gain of 0.51% over the baseline. The best results are indicated in bold, and values in parentheses indicate relative accuracy gains over the baseline.

Comparison with more Baselines

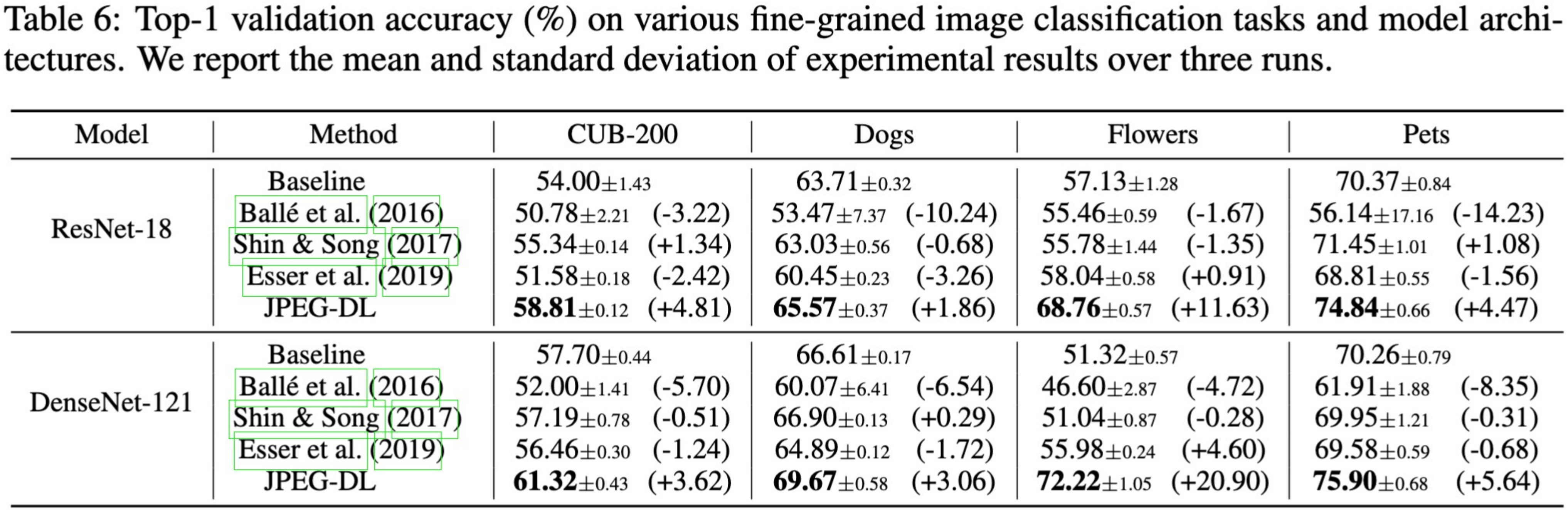


Table 6: Top-1 validation accuracy (%) on various fine-grained image classification tasks and model architectures. We report the mean and standard deviation of experimental results over three runs.

Model	Method	CUB-200	Dogs	Flowers	Pets
ResNet-18	Baseline	54.00±1.43	63.71±0.32	57.13±1.28	70.37±0.84
	Ballé et al. (2016)	50.78±2.21 (-3.22)	53.47±7.37 (-10.24)	55.46±0.59 (-1.67)	56.14±17.16 (-14.23)
	Shin & Song (2017)	55.34±0.14 (+1.34)	63.03±0.56 (-0.68)	55.78±1.44 (-1.35)	71.45±1.01 (+1.08)
	Esser et al. (2019)	51.58±0.18 (-2.42)	60.45±0.23 (-3.26)	58.04±0.58 (+0.91)	68.81±0.55 (-1.56)
	JPEG-DL	**58.81**±0.12 (+4.81)	**65.57**±0.37 (+1.86)	**68.76**±0.57 (+11.63)	**74.84**±0.66 (+4.47)
DenseNet-121	Baseline	57.70±0.44	66.61±0.17	51.32±0.57	70.26±0.79
	Ballé et al. (2016)	52.00±1.41 (-5.70)	60.07±6.41 (-6.54)	46.60±2.87 (-4.72)	61.91±1.88 (-8.35)
	Shin & Song (2017)	57.19±0.78 (-0.51)	66.90±0.13 (+0.29)	51.04±0.87 (-0.28)	69.95±1.21 (-0.31)
	Esser et al. (2019)	56.46±0.30 (-1.24)	64.89±0.12 (-1.72)	55.98±0.24 (+4.60)	69.58±0.59 (-0.68)
	JPEG-DL	**61.32**±0.43 (+3.62)	**69.67**±0.58 (+3.06)	**72.22**±1.05 (+20.90)	**75.90**±0.68 (+5.64)

Layer Replacement

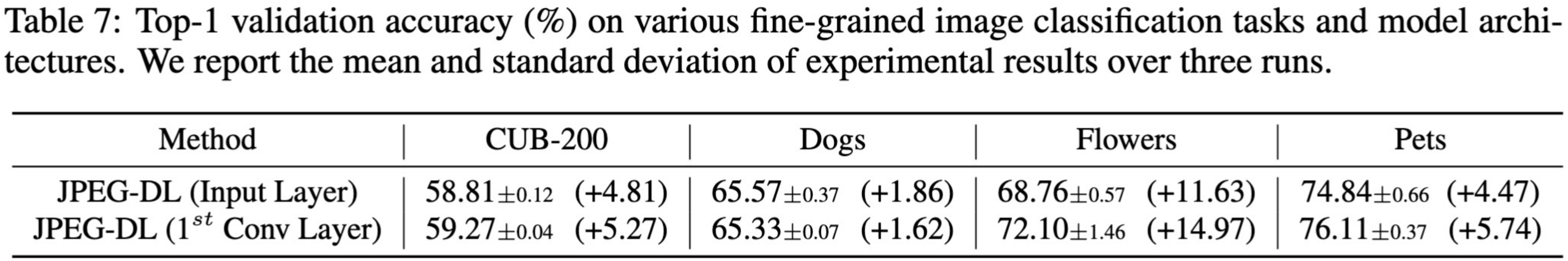
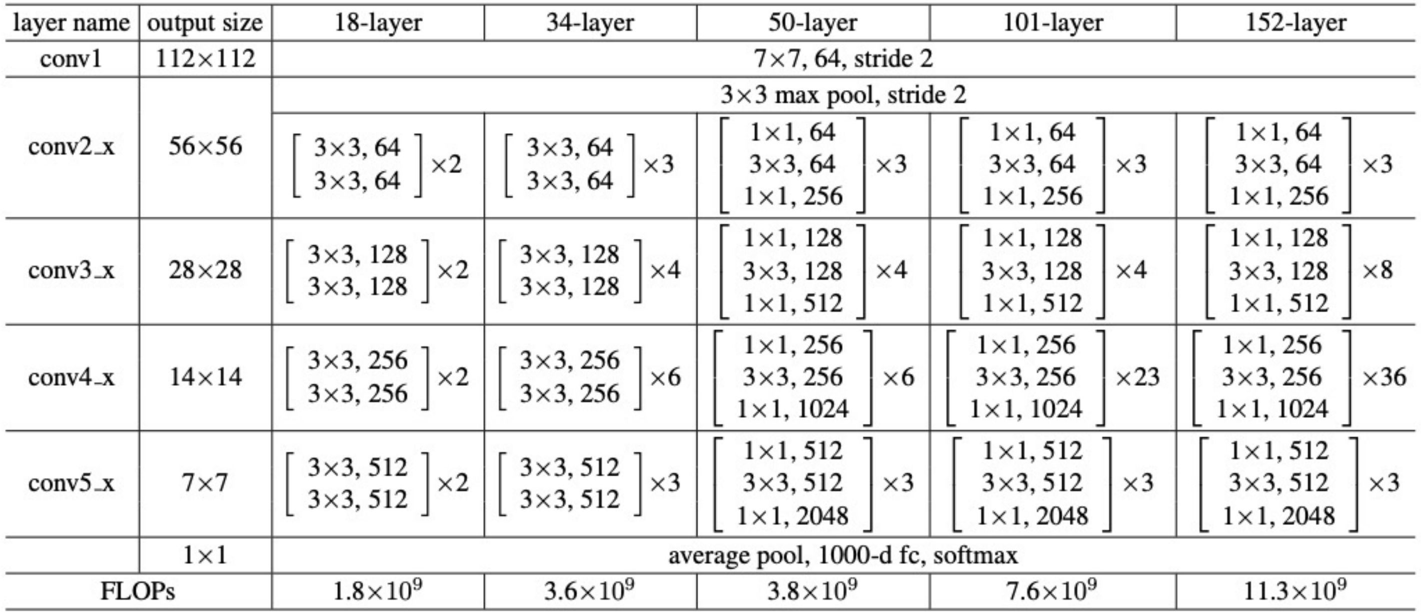


Table 7: Top-1 validation accuracy (%) on various fine-grained image classification tasks and model architectures. We report the mean and standard deviation of experimental results over three runs.

Method	CUB-200	Dogs	Flowers	Pets
JPEG-DL (Input Layer)	58.81±0.12 (+4.81)	65.57±0.37 (+1.86)	68.76±0.57 (+11.63)	74.84±0.66 (+4.47)
JPEG-DL (1^{st} Conv Layer)	59.27±0.04 (+5.27)	65.33±0.07 (+1.62)	72.10±1.46 (+14.97)	76.11±0.37 (+5.74)

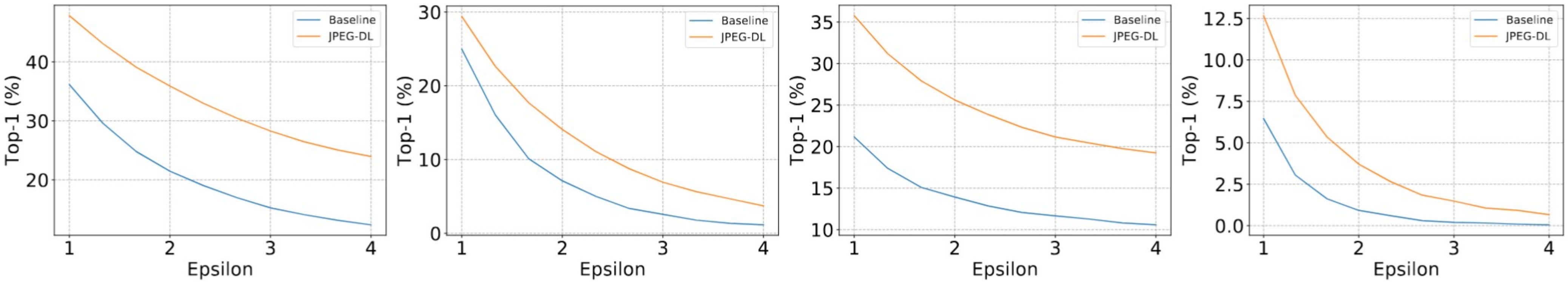
JPEG layer →



layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times23$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

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, pp. 770–778, 2016.

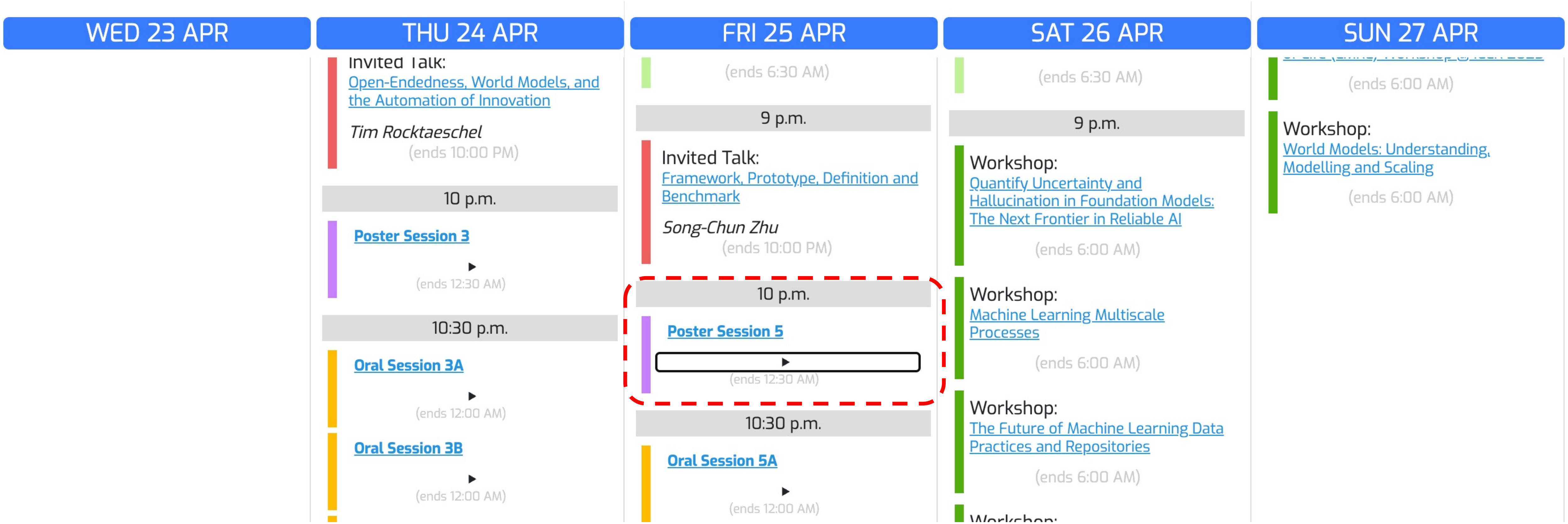
Robustness



(a) FGSM for VGG13 (b) PGD for VGG13 (c) FGSM for Res56 (d) PGD for Res56

Figure 3: Evaluate the adversarial robustness of JPEG-DL models in comparison to standard DNN on VGG13 and Res56 for CIFAR-100 against FGSM and PGD attacks.

Meet us …



WED 23 APR

THU 24 APR

Invited Talk:
Open-Endedness, World Models, and the Automation of Innovation

Tim Rocktaeschel
(ends 10:00 PM)

10 p.m.

Poster Session 3

▶
(ends 12:30 AM)

10:30 p.m.

Oral Session 3A

▶
(ends 12:00 AM)

Oral Session 3B

▶
(ends 12:00 AM)

FRI 25 APR

(ends 6:30 AM)

9 p.m.

Invited Talk:
Framework, Prototype, Definition and Benchmark

Song-Chun Zhu
(ends 10:00 PM)

10 p.m.

Poster Session 5

▶
(ends 12:30 AM)

10:30 p.m.

Oral Session 5A

▶
(ends 12:00 AM)

SAT 26 APR

(ends 6:30 AM)

9 p.m.

Workshop:
Quantify Uncertainty and Hallucination in Foundation Models: The Next Frontier in Reliable AI

(ends 6:00 AM)

Workshop:
Machine Learning Multiscale Processes

(ends 6:00 AM)

Workshop:
The Future of Machine Learning Data Practices and Repositories

(ends 6:00 AM)

Workshop:

SUN 27 APR

(ends 6:00 AM)

Workshop:
World Models: Understanding, Modelling and Scaling

(ends 6:00 AM)