Curriculum by Smoothing

Samarth Sinha, Animesh Garg, Hugo Larochelle

Alex Fedorov

November 6, 2020

Curriculum Learning

Start small, learn easier aspects,
then gradually increase the difficulty level.

Elman, 1993

Maybe first work with NNs

- Elman, Jeffrey L. "Learning and development in neural networks: The importance of starting small." *Cognition* 48.1 (1993): 71-99.
- In humans the greatest learning occurs precisely at that point in time - childhood - when the most dramatic maturational changes also occur
- Possible synergistic interactions between maturational change and the ability to learn a complex domain
- Humans display an exceptional capacity to learn
- Humans are remarkable for the unusually long time it takes to reach maturity.

The importance of starting small

For both “learning device and training input”

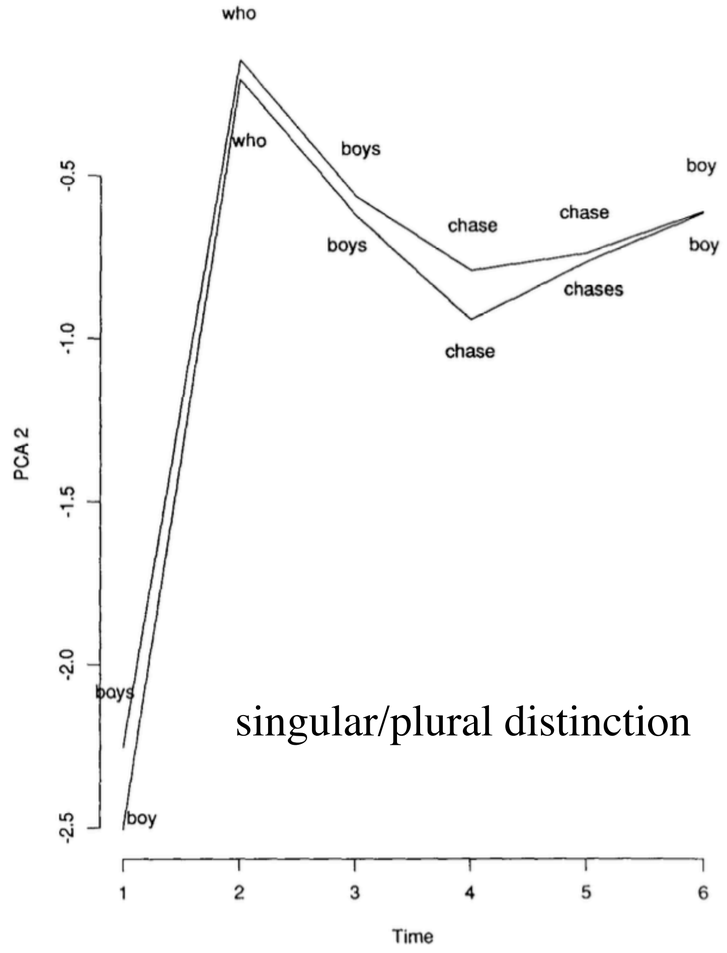
- Insufficiency of the data has been discussed in many contexts
  - (e.g., Baker, 1979; Bowerman, 1987; Pinker, 1989; Wexler & Cullicover, 1980)
- “**Regular** Language” Learning
  - Gold (1967) a language learner is presented inly with positive-only data, only “**regular**” languages can be learned
  - The non-occurrence of an expected form constitutes an *indirect* sort of negative evidence
- Children do learn “**natural**” language
- **Learning and development co-occur**
  - Newport’s (1988, 1990)
    - **“less is more”** proposal, which is very consistent with the results obtained here
  - The typical assumption is that both the learning device and training input are static.
    - Plunkett and Marchman (1990)
      - Better overall learning is achieved when the *training corpus* for a connectionist model is allowed *slowly to grow in size*.

Simulation

- Samples
  - boys who chase dogs see girls.
  - girl who boys who feed cats walk.
  - cats chase dogs
  - mary feeds John
  - dogs see boys who cats who mary feeds chase
- The network was trained to take one word at a time and predict what the next word would be
  - Failed
- Five phases
  - 10000/0 -> 7500/2500 -> 5000/5000 -> 2500/7500 -> 0/10000
  - Generalized to a variety of novel sentences
    - systematically test the capacity to predict grammatically correct forms across a range of different structures

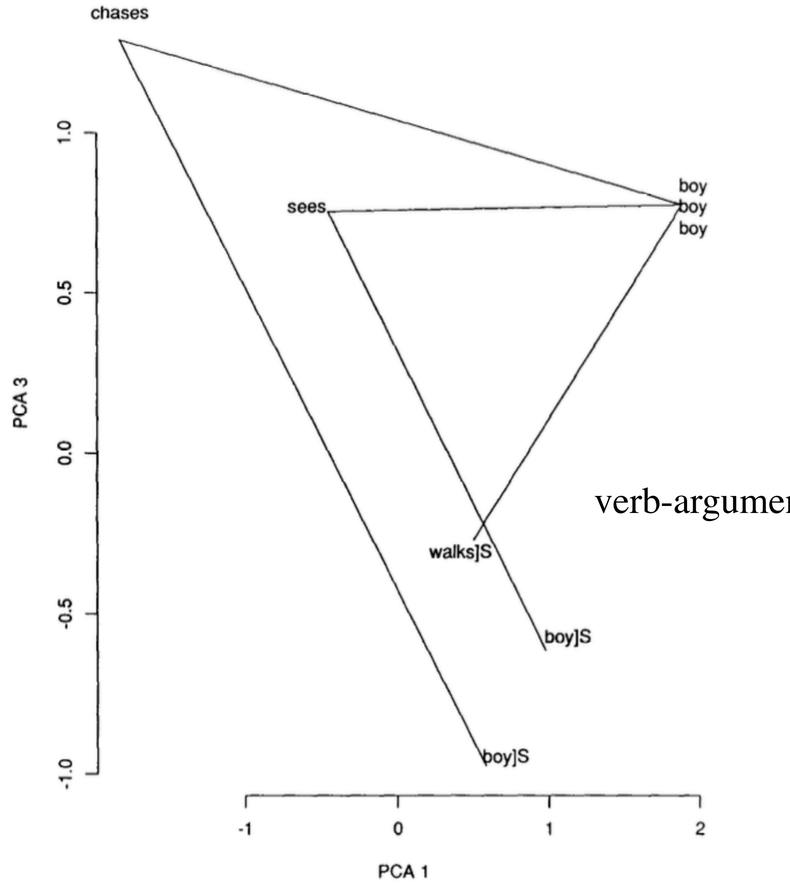
Simulation 2

- Memory window was increased
  - 3-4 -> 4-5 -> 5-6 -> 6-7 -> all words



singular/plural distinction

Figure 2. *Plot of the movement through one dimension of the hidden unit activation space (the second principal component) as the successfully trained network processes the sentences* boy who boys chase chases boy *vs.* boys who boys chase chase boy. *The second principal component encodes the singular/plural distinction in the main clause subject.*



verb-argument structure

Figure 3. *Plot of the movement through two dimensions of hidden unit activation space (first and third principal components) as the network processes the sentences* boy chases boy, boy sees boy, *and* boy walks *(sentence endings are indicated with]S). Nouns occupy the right portion of this plane, and verbs occupy the left side; the axis running from top left to bottom right encodes the verb argument expectations.*

Simulation 2

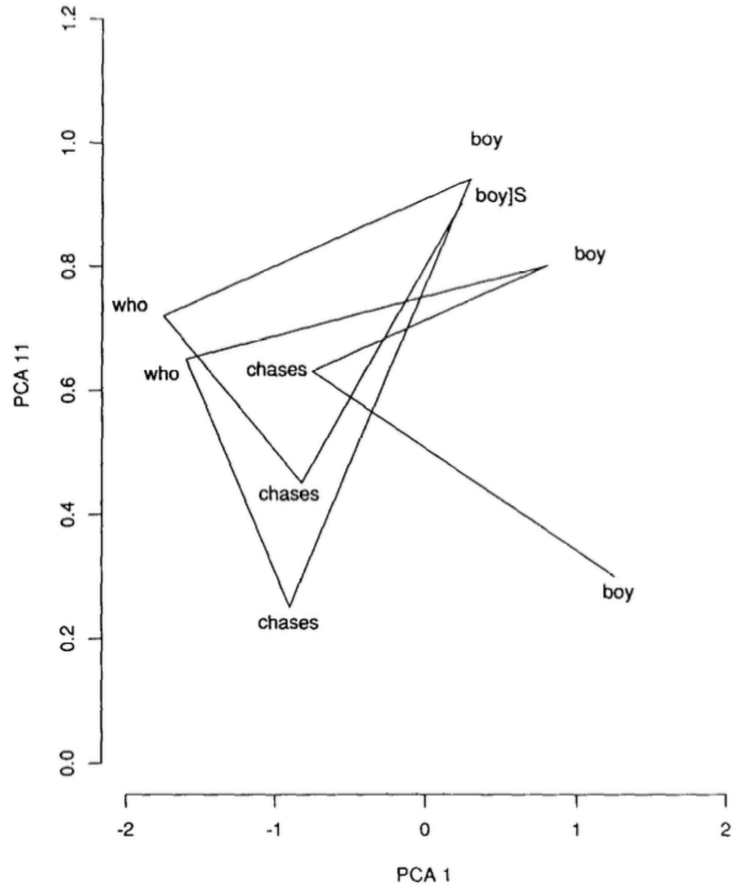


Figure 4. *Plot of the movement through two dimensions of hidden unit activation space (first and eleventh principal components) as the network processes the sentences* boy chases boy *and* boy who chases boy who chases boy. *The depth of embedding is encoded by a shift to the left in the trajectory for the canonical simple sentences.*

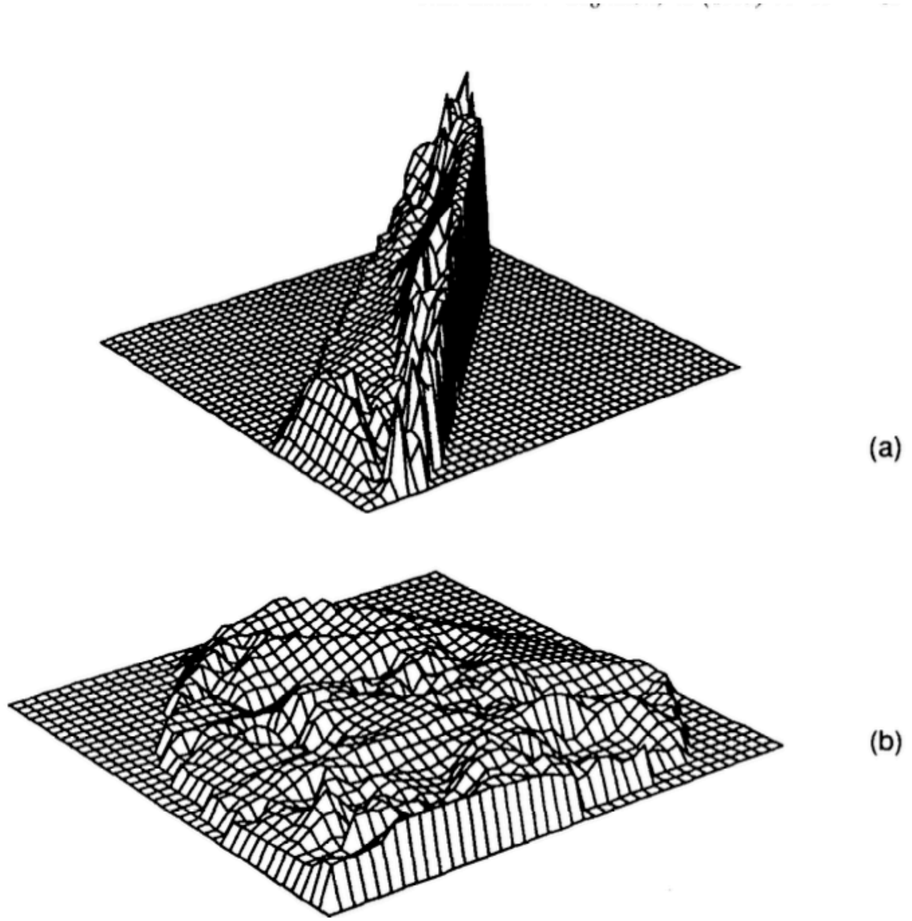


Figure 5. *(a) Sample of points visited in the hidden unit state space of a successfully trained network as it processes 1,000 randomly chosen sentences. (b) Sample of points visited in the hidden unit state space of a network which has failed to learn the task, as it processes 1,000 randomly chosen sentences.*

How networks learn

Property 1: statistics as the basis for learning; the problem of sample size

- In NN learning algorithms, the driving force for inference is statistics
- A lookup table of co-occurrence facts, then the approach is indeed doomed
  - Neural networks are function approximators, not compilers of lookup tables
    - Rule system (Miller and Chomsky) (extrapolation)
      - This allow generalization
- Minimal constraints are imposed due limited data

Pattern	*Classification*
1 0 1 1 0 1	1
0 0 0 0 0 0	1
0 0 1 1 0 0	1
0 1 0 1 1 0	0
1 1 1 0 1 1	0
0 0 0 1 1 1	0

0 1 1 1 0 1

How networks learn

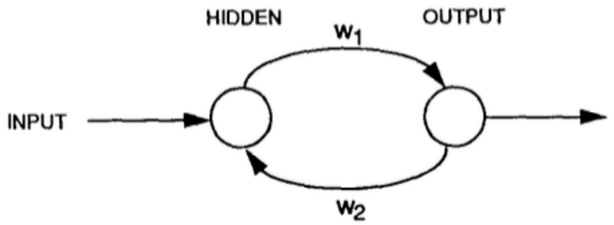
Property 2: the representation of experience

- Lifetime of the data
- Ideas
  - Data is accumulated
  - Preserved as long needed (Exemplar-based models)
    - Experiences are stored as individually retrievable retrievable exemplars
  - Connectionist
    - Effect is immediate
    - Data are as an effect
    - However, Catastrophic forgetting problem

How networks learn

Property 3: constraints on new hypotheses; the continuity of search

- Gradient Descent
  - Impose constrains to prevent generating widely different hypothesis
  - Smooth and small changes
  - But can stack in local minima



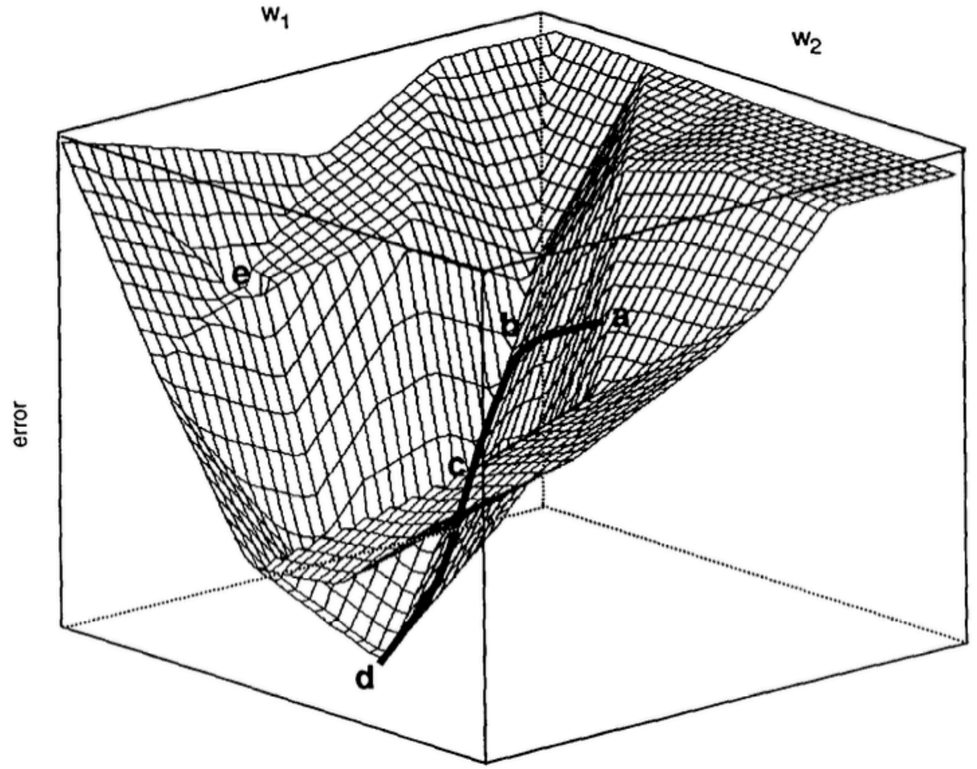


Figure 6. *Hypothetical error surface (bottom) associated with a network with two trainable weights (top). The z coordinate indicates the error that is produced if the network has weights corresponding to the values at the x and y coordinates; low regions in this surface correspond to low error.*

How networks learn

Property 4: how the ability to learn changes over time; early flexibility versus late rigidity

- Backprogation
  - credit/blame assignment
- The more system knows the harder it is to learn something new

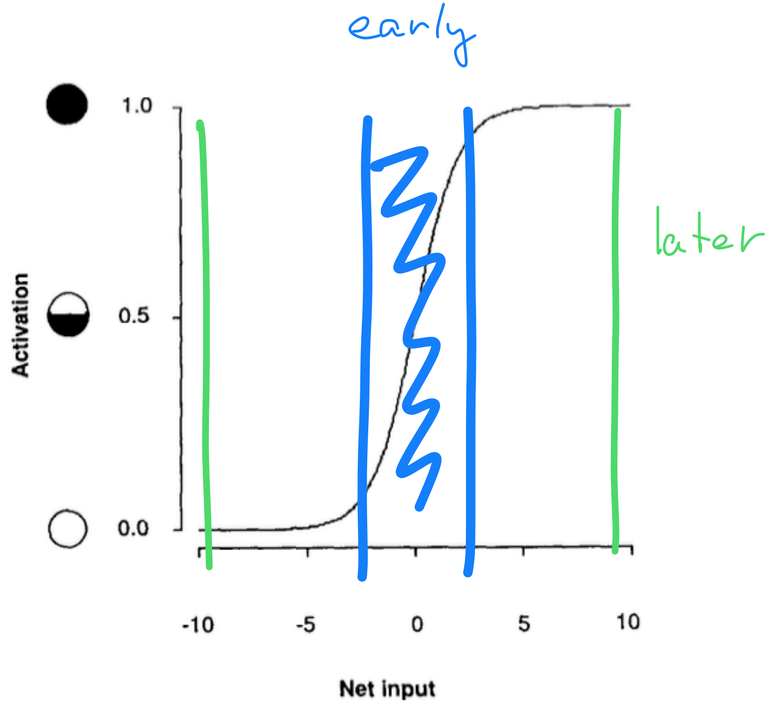
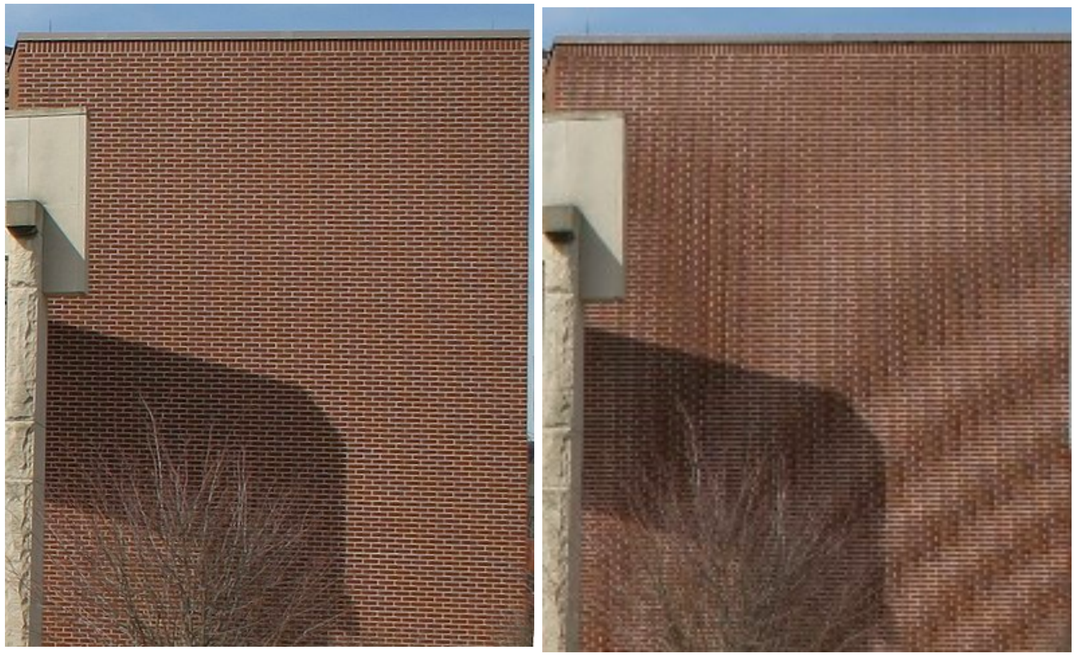


Figure 7. *The logistic function used to determine node activations. One term in the weight change equation is the slope of this function; when node activations saturate at either 1.0 or 0.0, this slope asymptotically approaches 0.*

Curriculum by Smoothing

Aliasing

aliasing is an effect that causes different signals to become indistinguishable (or *aliases* of one another) when sampled.



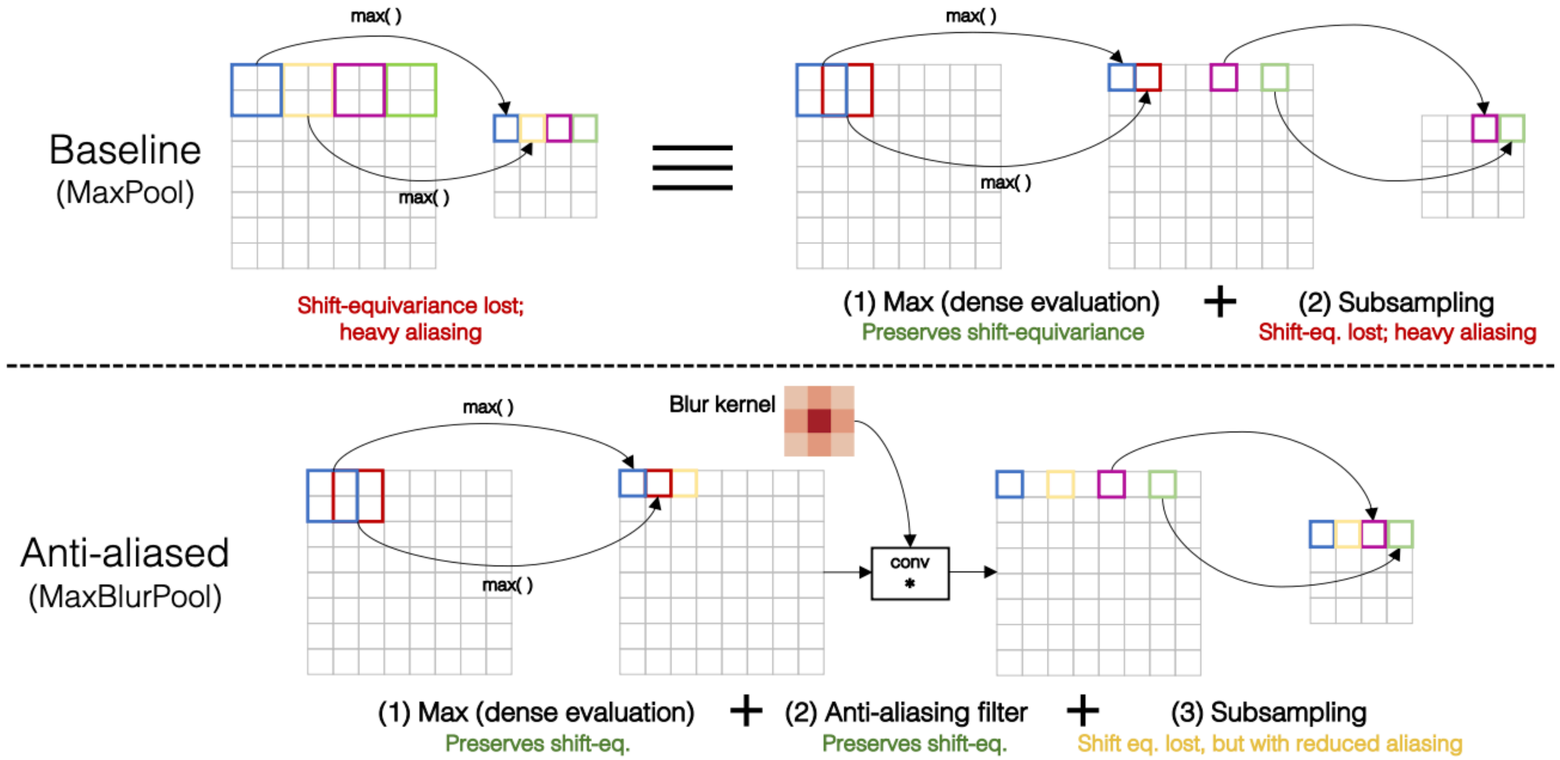
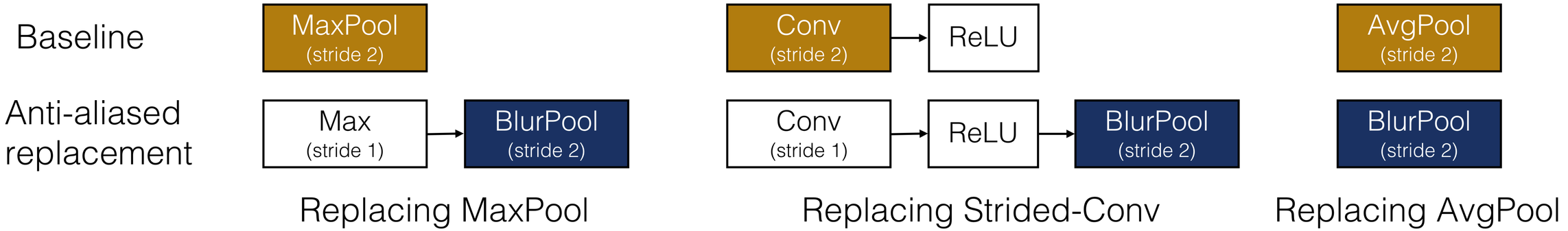


Figure 3. **Anti-aliased max-pooling. (Top)** Pooling does not preserve shift-equivariance. It is functionally equivalent to densely-evaluated pooling, followed by subsampling. The latter ignores the Nyquist sampling theorem and loses shift-equivariance. **(Bottom)** We low-pass filter between the operations. This keeps the first operation, while anti-aliasing the appropriate signal. Anti-aliasing and subsampling can be combined into one operation, which we refer to as **BlurPool**.



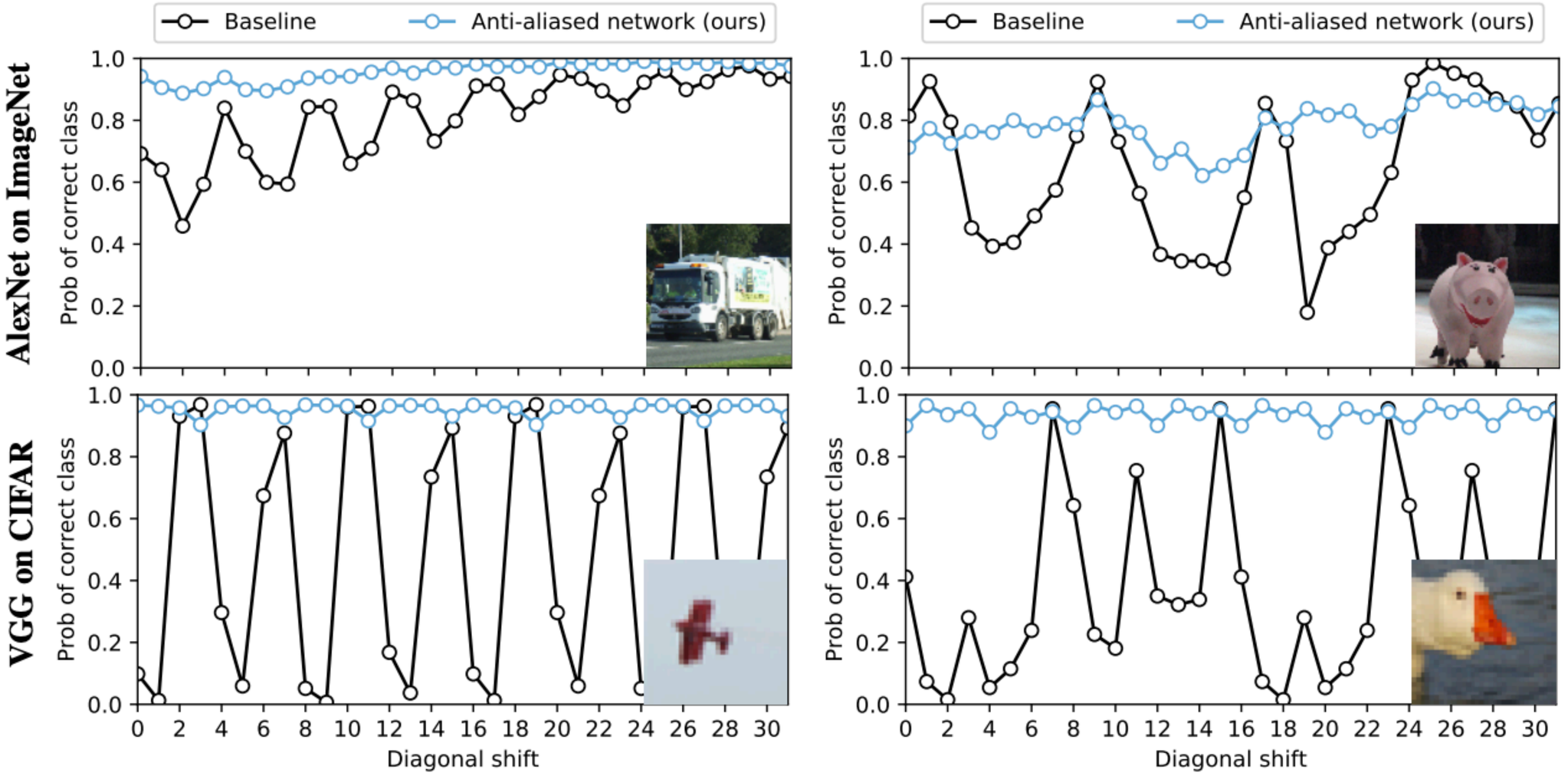
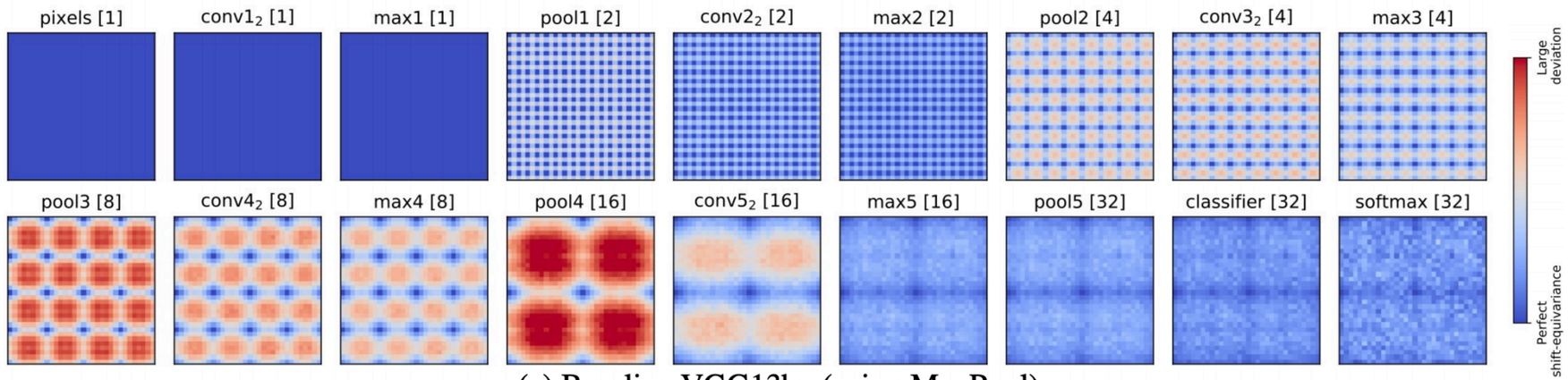
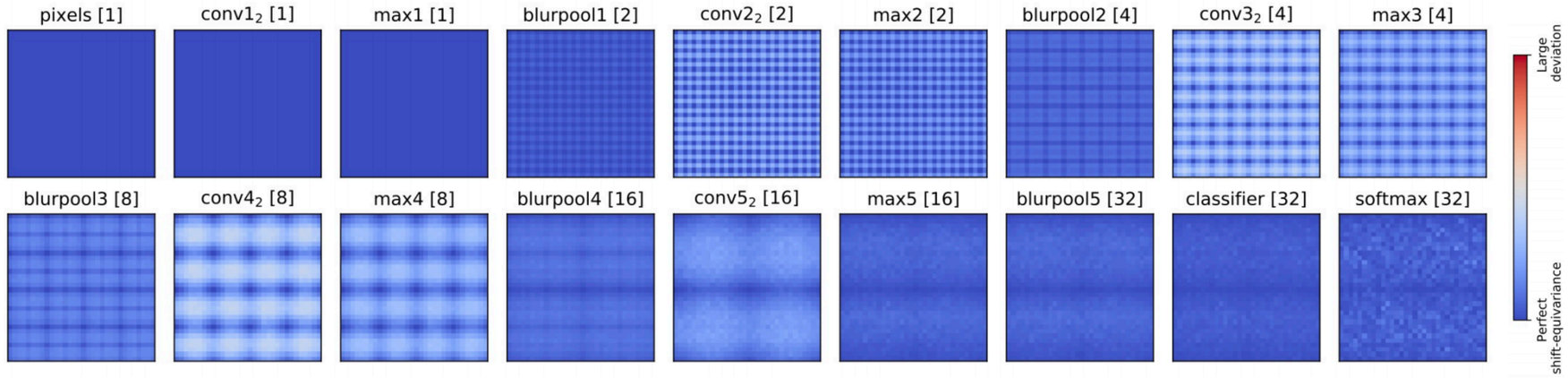


Figure 1. **Classification stability for selected images.** Predicted probability of the correct class changes when shifting the image. The baseline (black) exhibits chaotic behavior, which is stabilized by our method (blue). We find this behavior across networks and datasets. Here, we show selected examples using AlexNet on ImageNet **(top)** and VGG on CIFAR10 **(bottom)**. Code and anti-aliased versions of popular networks are available at https://richzhang.github.io/antialiased-cnns/.



(a) Baseline VGG13bn (using MaxPool)



(b) Anti-aliased VGG13bn (using MaxBlurPool, *Bin-5*)

Figure 5. **Deviation from perfect shift-equivariance, throughout VGG.** Feature distance between left & right-hand sides of the shift-equivariance condition (Eqn 1). Each pixel in each heatmap is a shift $(\Delta h, \Delta w)$. Blue indicates perfect shift-equivariance; red indicates large deviation. Note that the dynamic ranges of distances are different per layer. For visualization, we calibrate by calculating the mean distance between two different images, and mapping red to half the value. Accumulated downsampling factor is in [brackets]; in layers `pool5`, `classifier`, and `softmax`, shift-equivariance and shift-invariance are equivalent, as features have no spatial extent. Layers up to `max1` have perfect equivariance, as no downsampling yet occurs. **(a)** On the **baseline network**, shift-equivariance is reduced each time downsampling takes place. Periodic-N shift-equivariance holds, with N doubling with each downsampling. **(b)** With our **antialiased network**, shift-equivariance is better maintained, and the resulting output is more shift-invariant.

Zhang, Richard. "Making Convolutional Networks Shift-Invariant Again." *International Conference on Machine Learning*. 2019.

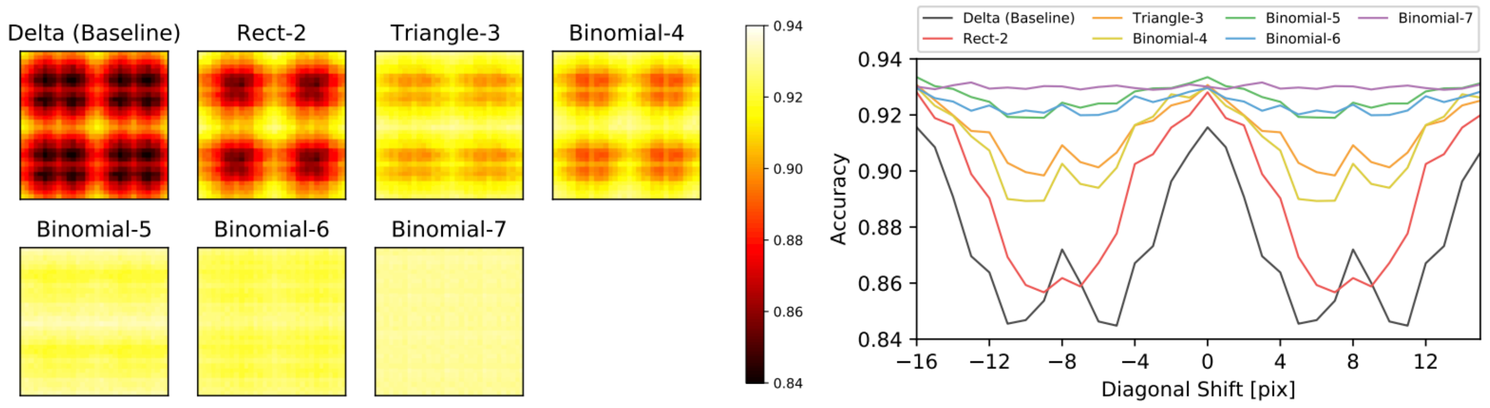


Figure 9. **Average accuracy as a function of shift. (Left)** We show classification accuracy across the test set as a function of shift, given different filters. **(Right)** We plot accuracy vs diagonal shift in the input image, across different filters. Note that accuracy degrades quickly with the baseline, but as increased filtering is added, classifications become consistent across spatial positions.

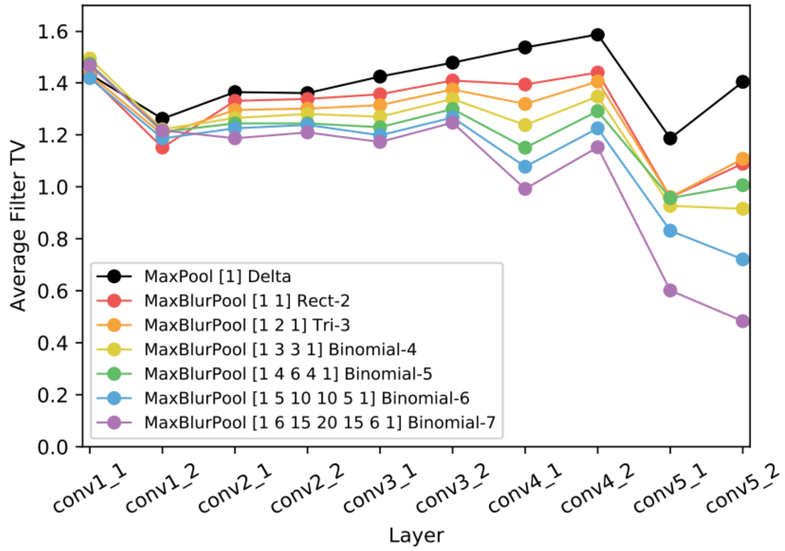


Figure 10. **Total Variation (TV) by layer.** We compute average smoothness of learned conv filters per layer (lower is smoother). Baseline MaxPool is in black, and adding additional blurring is shown in colors. Note that the *learned* convolutional layers become smoother, indicating that a smoother feature extractor is induced.

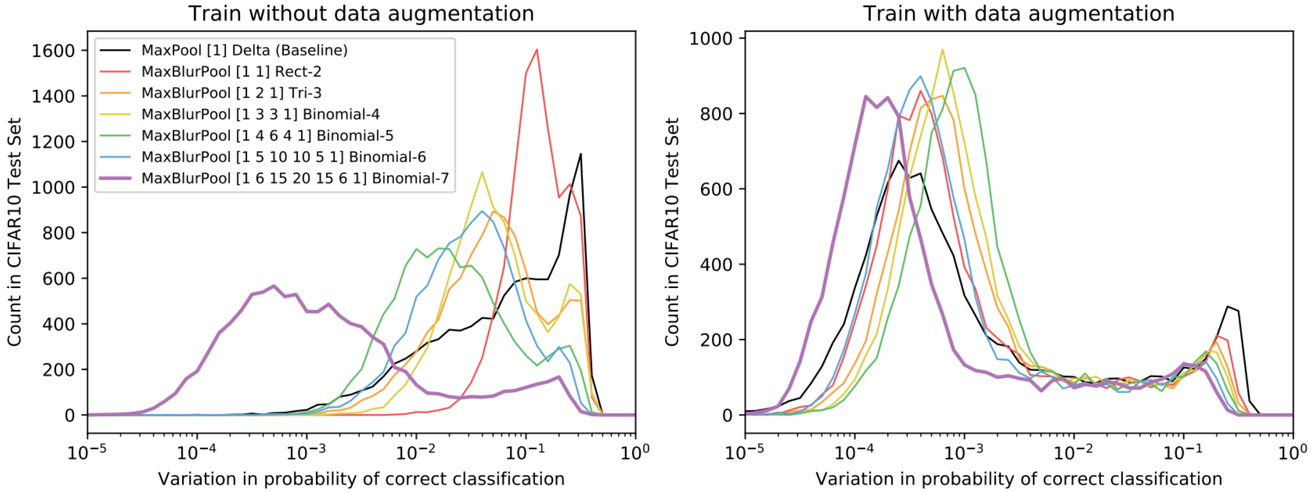


Figure 11. **Distribution of per-image classification variation.** We show the distribution of classification variation in the test set, **(left)** without and **(right)** with data augmentation at training. Lower variation means more consistent classifications (and increased shift-invariance). Training with data augmentation drastically reduces variation in classification. Adding filtering further decreases variation.

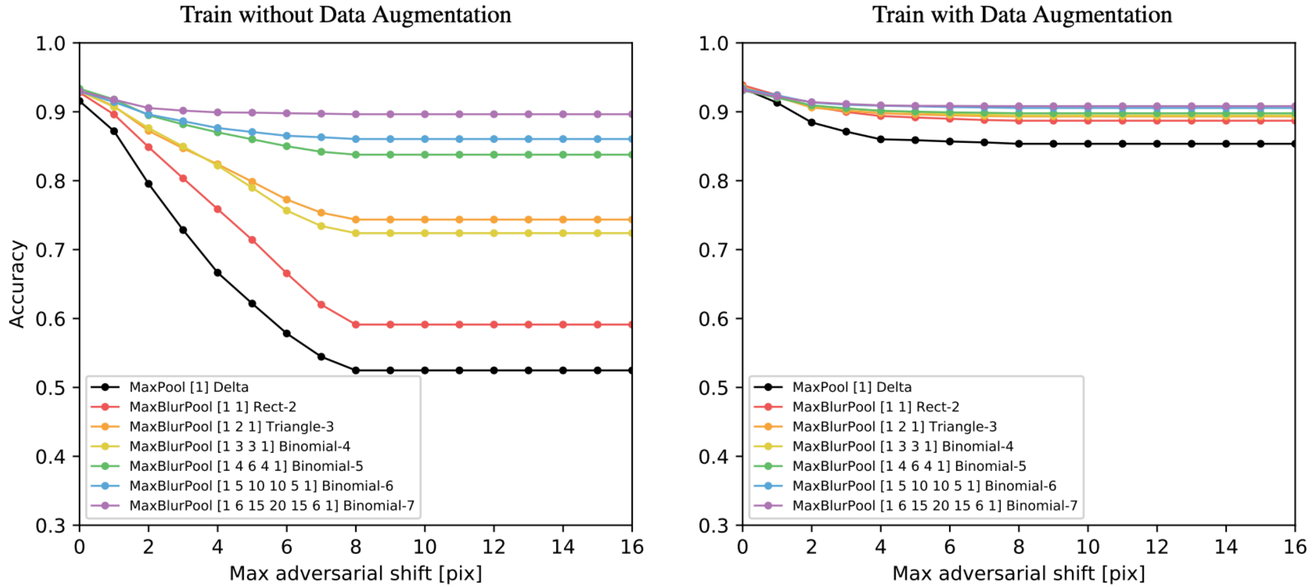


Figure 12. **Robustness to shift-based adversarial attack.** Classification accuracy as a function of the number of pixels an adversary is allowed to shift the image. Applying our proposed filtering increases robustness, both without **(left)** and with **right** data augmentation.

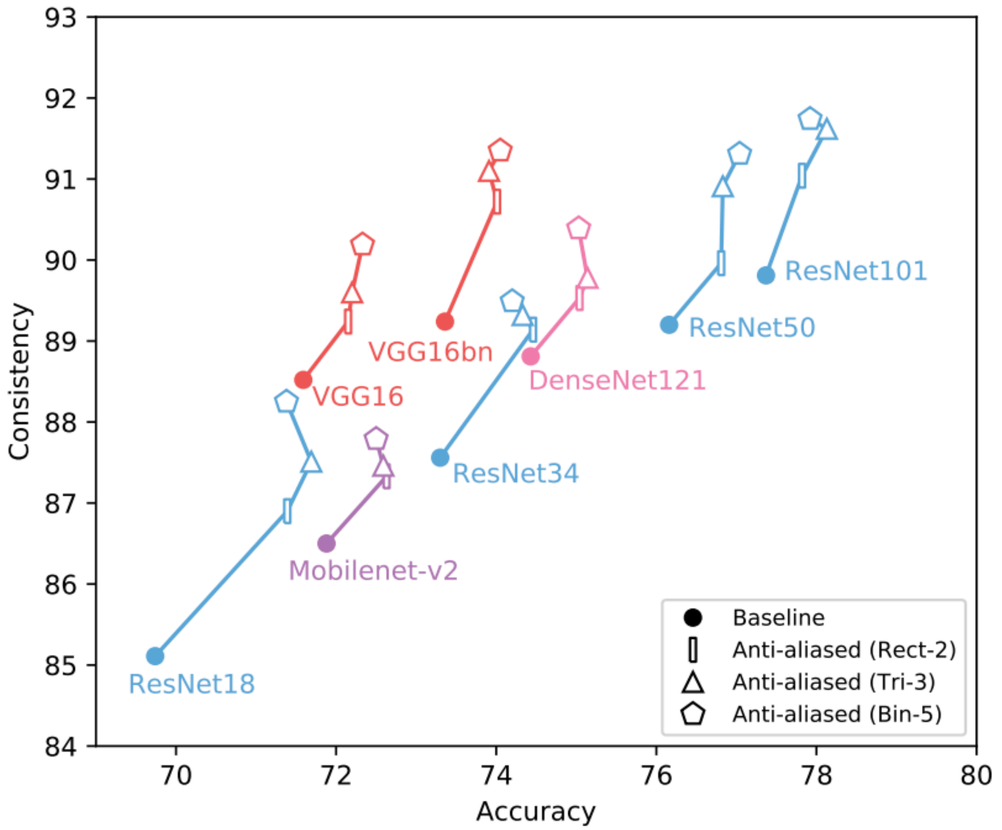


Figure 6. **ImageNet Classification consistency vs. accuracy.** Up (more consistent to shifts) and to the right (more accurate) is better. Different shapes correspond to the baseline (circle) or variants of our anti-aliased networks (bar, triangle, pentagon for length 2, 3, 5 filters, respectively). We test across network architectures. As expected, low-pass filtering helps shift-invariance. Surprisingly, classification accuracy is also improved.

Related recent work

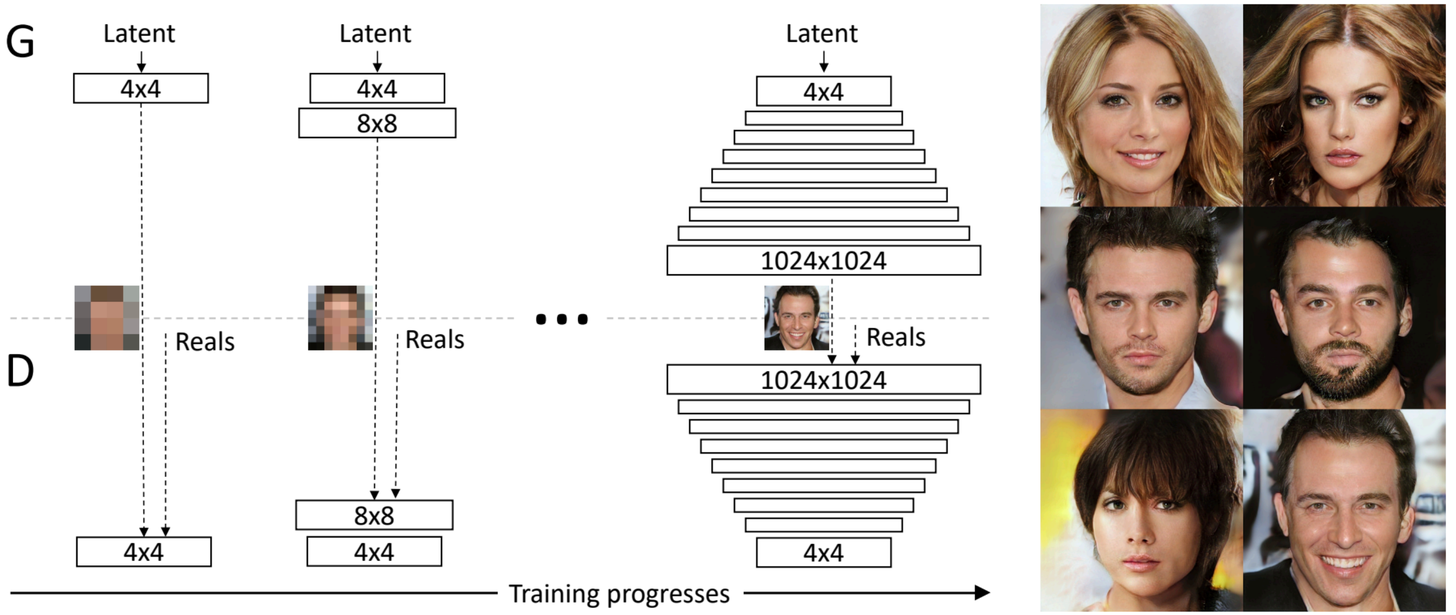


Figure 1: Our training starts with both the generator (G) and discriminator (D) having a low spatial resolution of 4×4 pixels. As the training advances, we incrementally add layers to G and D, thus increasing the spatial resolution of the generated images. All existing layers remain trainable throughout the process. Here $\boxed{N \times N}$ refers to convolutional layers operating on $N \times N$ spatial resolution. This allows stable synthesis in high resolutions and also speeds up training considerably. One the right we show six example images generated using progressive growing at 1024×1024.

Karras, Tero, et al. "Progressive growing of gans for improved quality, stability, and variation." *arXiv preprint arXiv:1710.10196* (2017).

Related recent work

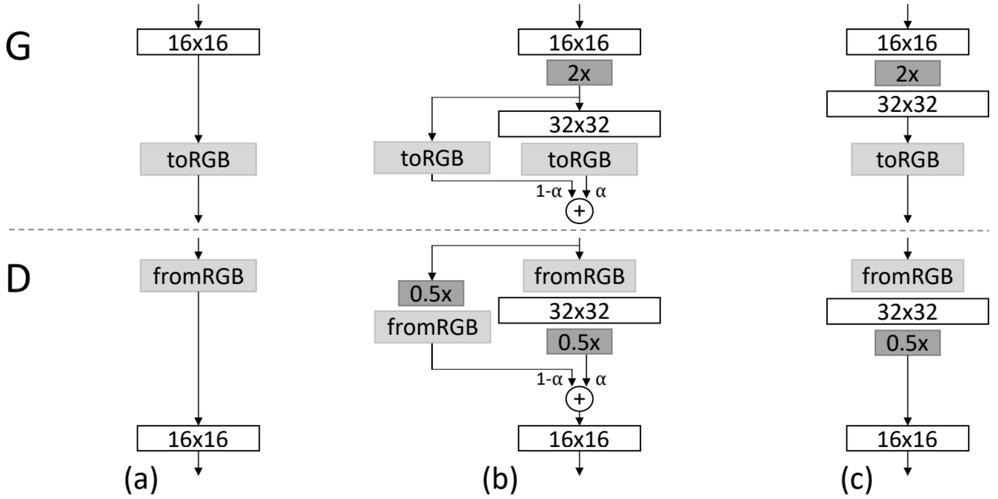


Figure 2: When doubling the resolution of the generator (G) and discriminator (D) we fade in the new layers smoothly. This example illustrates the transition from 16×16 images (a) to 32×32 images (c). During the transition (b) we treat the layers that operate on the higher resolution like a residual block, whose weight α increases linearly from 0 to 1. Here $\boxed{2\times}$ and $\boxed{0.5\times}$ refer to doubling and halving the image resolution using nearest neighbor filtering and average pooling, respectively. The $\boxed{\text{toRGB}}$ represents a layer that projects feature vectors to RGB colors and $\boxed{\text{fromRGB}}$ does the reverse; both use 1×1 convolutions. When training the discriminator, we feed in real images that are downscaled to match the current resolution of the network. During a resolution transition, we interpolate between two resolutions of the real images, similarly to how the generator output combines two resolutions.

Karras, Tero, et al. "Progressive growing of gans for improved quality, stability, and variation." *arXiv preprint arXiv:1710.10196* (2017).

Related recent work

Mao et al. (2016b) (128 × 128) Gulrajani et al. (2017) (128 × 128) Our (256 × 256)

Figure 6: Visual quality comparison in LSUN BEDROOM; pictures copied from the cited articles.

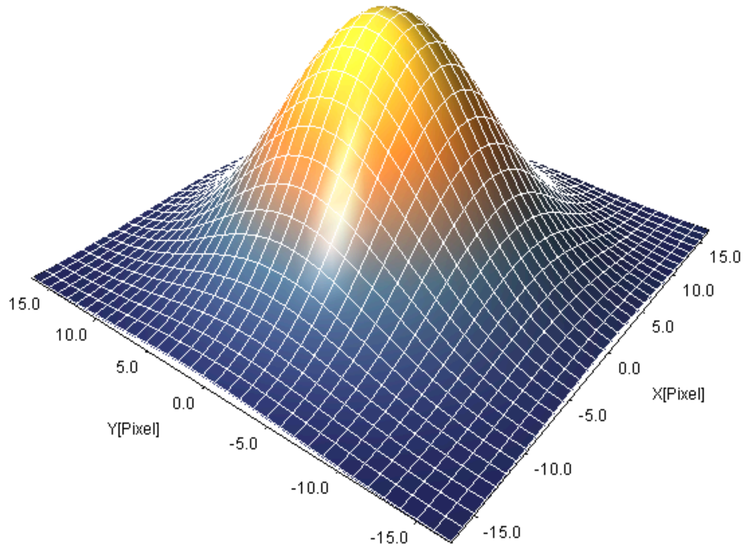
Karras, Tero, et al. "Progressive growing of gans for improved quality, stability, and variation." *arXiv preprint arXiv:1710.10196* (2017).

Related recent work

Figure 5: 1024 × 1024 images generated using the CELEBA-HQ dataset. See Appendix F for a larger set of results, and the accompanying video for latent space interpolations.

Karras, Tero, et al. "Progressive growing of gans for improved quality, stability, and variation." *arXiv preprint arXiv:1710.10196* (2017).

$$k(x,y) = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$



https://docs.pyvista.org/examples/01-filter/gaussian-smoothing.html

Gaussian Kernel Layer

- By annealing the value of σ as training progresses

```
# Use the Gaussian kernel after the convolution operation
h = gaussian_kernel(conv1(x))
# Add non-linearity and pooling
h = activation(pool(h))

# Same operation after each conv. layer
h = gaussian_kernel(conv2(h))
h = activation(pool(h))
```

Gaussian Kernel Layer

Table 8: Ablation study by applying a Gaussian kernel on only specific layers of a simple 3 layer CNN with Max-Pooling and ReLU activations. **Bolded** value corresponds to the proposed model which uses the kernel layer after each convolutional layer.

None	First	Second	Third	All
57.04	60.03	59.87	59.88	**61.41**

Table 1: **Image Classification**. Top-1 classification accuracy on CIFAR10, CIFAR100 and SVHN for CNNs trained normally and CNNs trained using CBS. We show significant improvements over three standard datasets using four different network architectures: VGG-16 [51], ResNet18 [19], Wide ResNet-50 [65], and ResNext-50 [63].

	SVHN	CIFAR10	CIFAR100
VGG-16	96.6 ± 0.2	85.8 ± 0.2	57.0 ± 0.2
VGG-16 + CBS	**97.0** ± 0.2	**88.9** ± 0.3	**61.4** ± 0.3
ResNet-18	97.2 ± 0.2	87.1 ± 0.3	62.4 ± 0.3
ResNet-18 + CBS	**98.7** ± 0.2	**90.2** ± 0.3	**65.4** ± 0.2
Wide-ResNet-50	97.7 ± 0.1	91.8 ± 0.1	73.3 ± 0.1
Wide-ResNet-50 + CBS	**98.3** ± 0.3	**93.9** ± 0.1	**75.9** ± 0.2
ResNeXt-50	97.7 ± 0.2	93.1 ± 0.1	74.1 ± 0.3
ResNeXt-50 + CBS	**99.0** ± 0.2	**95.1** ± 0.2	**77.0** ± 0.1

Table 2: **Image Classification**. Top-1 and Top-5 classification accuracy on ImageNet for CNNs trained normally and trained using CBS. We see significant improvements on for VGG-16 and a ResNet-18 networks when trained with CBS.

	ImageNet (Top-1)	ImageNet (Top-5)
VGG-16	63.45 ± 0.4	83.81 ± 0.3
VGG-16 + CBS	**66.02** ± 0.5	**86.26** $\pm$ **0.3**
ResNet-18	67.90 ± 0.7	85.86 ± 0.5
ResNet-18 + CBS	**71.02** ± 0.8	**89.55** ± 0.6

Table 3: **Feature Extraction for Classification**. Top-1 classification accuracy on CIFAR10, CIFAR100 and SVHN when the CNNs trained on ImageNet normally and CNNs trained using Curriculum By Smoothing (CBS) are used as feature extractors on a different dataset. The CNN weights are then frozen, and the features from the images are used to train a 3-layer Multi-Layer Perception with ReLU activation. We show considerable improvement for all three datasets, as well as both network architectures.

	SVHN	CIFAR10	CIFAR100
VGG-16	69.31 ± 0.2	71.94 ± 0.2	46.10 ± 0.1
VGG-16 + CBS	**72.04** ± 0.2	**73.82** ± 0.3	**48.79** ± 0.1
ResNet-18	72.12 ± 0.5	72.98 ± 0.5	51.30 ± 0.3
ResNet-18 + CBS	**76.30** ± 0.5	**75.92** ± 0.6	**54.99** ± 0.3

Table 4: **Transfer Learning**. Results for transfer learning on a different task on the Pascal VOC Dataset. For all semantic segmentation experiments we use Fully Convolutional Network with VGG-16 network, trained on ImageNet from Section 4. For all Object Detection experiments we use Fast-RCNN with the same VGG-16 backbone.

	Semantic Segmentation (% mIoU)	Object Detection (% mAP)
CNN	55.7 ± 0.2	67.9 ± 0.4
CBS	**57.9**± 0.3	**70.0** ± 0.2

Table 5: **Zero-Shot Domain Adaptpation**. Comparison of different architectures trained with and without curricula for ZSDA for the digit recognition task. We present mean and standard deviation over 5 runs. Adding CBS during training improves each networks performance by learning better and more robust representations, which then improves zero-shot performance.

Source → Target	Backbone	MNIST → USPS	USPS → MNIST	SVHN → MNIST
Source Only	WideResnet-50	79.37 ± 0.24	46.66 ± 0.64	72.70 ± 0.18
Source Only + CBS	WideResnet-50	**81.69** ± 0.24	**49.89** ± 0.21	**74.32** ± 0.45
Source Only	ResNext-50	70.70 ± 0.31	40.74 ± 0.24	64.45 ± 0.23
Source Only + CBS	ResNext-50	**71.23** ± 0.12	**43.35** ± 0.20	**67.89** ± 0.84
Target Only	WideResnet-50	96.29 ± 0.21	99.85 ± 0.10	99.85 ± 0.10

Table 6: **Unsupervised Representation Learning with Generative Models**. We evaluate the ability of a model to learn unsupervised representations from data using a VAE [28] and a β-VAE [20], with $\beta = 10$ on two benchmark representation learning datasets: MNIST [32] and CelebA [37]. We show that using CBS, we are able to learn significantly better reconstructions, as shown by NLL, and richer latent spaces, as shown by Mutual Information and # of Active Units. We report the mean and standard deviation over 3 random seeds.

MNIST [32]	NLL	Mutual Information	# of Active Units
VAE	83.9 ± 0.2	125.0 ± 0.8	**36** ± 0.5
VAE + CBS	**82.0** ± 0.3	**127.3** ± 0.4	**36** ± 0.3
β-VAE	126.1 ± 0.5	6.3 ± 0.3	8 ± 1.1
β-VAE + CBS	**125.0** ± 0.2	**7.2** ± 0.4	**11** ± 0.9
CelebA [37]	NLL	Mutual Information	# of Active Units
VAE	66.1 ± 0.4	108.5 ± 0.8	44 ± 0.9
VAE + CBS	**64.9** ± 0.3	**108.7** ± 0.6	**48** ± 0
β-VAE	92.6 ± 0.3	3.6 ± 0.1	**34** ± 0.9
β-VAE + CBS	**91.3** ± 0.3	**3.9** ± 0.3	**34** ± 0.5

Table 7: **Ablation study**. Applying smoothing to different components of the network. We report the mean and standard deviation over 5 random seeds using a ResNet-18. We see that applying Gaussian smoothing on the images or without decaying the value of σ, the network is unable to learn effective representations.

	Image Only	Image + Features	Constant $\sigma = 1$	Network	CBS
CIFAR-10	80.0 ± 0.3	84.1 ± 0.2	85.3 ± 0.4	87.1 ± 0.3	**90.2** ± 0.3
CIFAR-100	45.7 ± 0.3	49.6 ± 0.3	54.0 ± 0.2	62.4 ± 0.3	**65.4** ± 0.2

Dream **big**.
Start **small**.
But most of all, **start**.

- Simon Sinek

Goalcast