WIDE & SPARSE

Neural Networks

Are wider nets better given the same number of parameters?

A. Golubeva, G. Gur-Ari, B. Neyshabur
ICLR 2021, arXiv:2010.14495

- Increasing the number of NN parameters improves performance.

- Increasing the number of NN parameters improves performance.

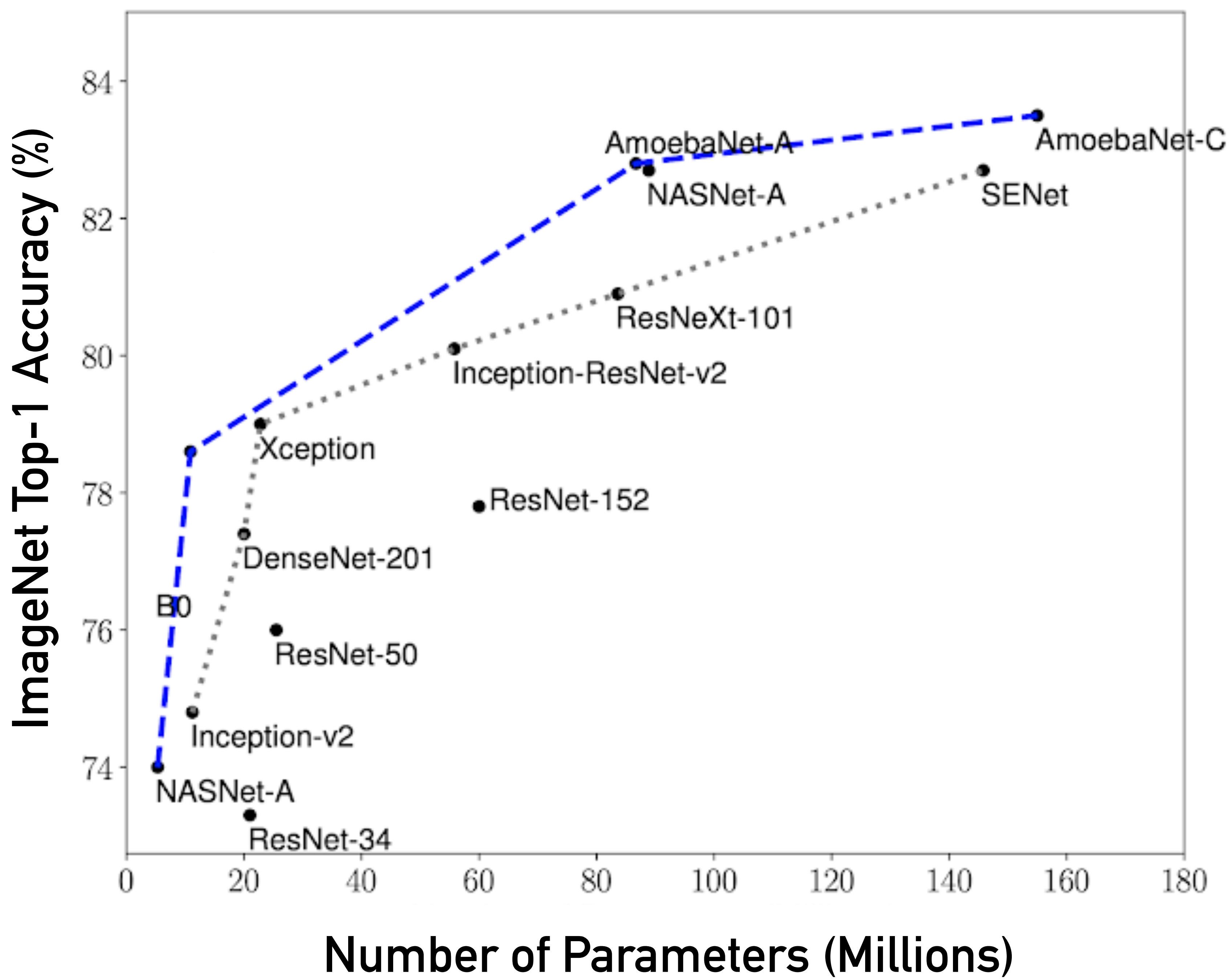


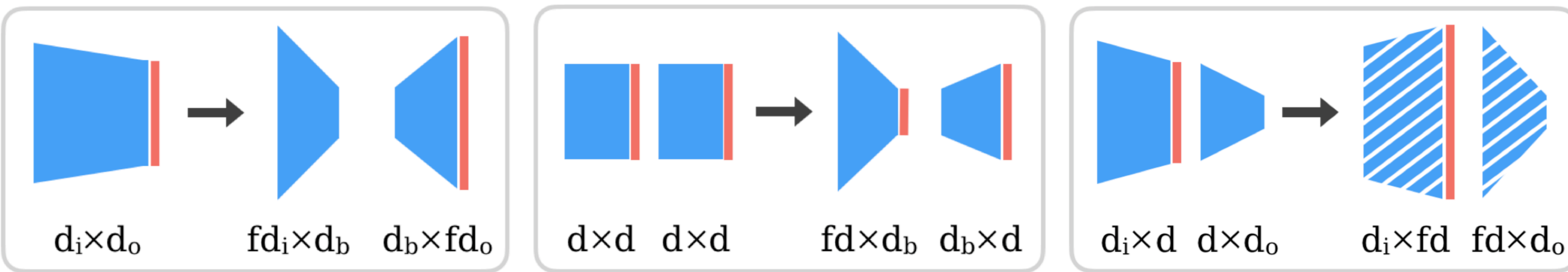
Figure from arXiv:1905.11946

- Increasing the number of NN parameters improves performance.
- The number of parameters is increased along with layer width.

- Increasing the number of NN parameters improves performance.
- The number of parameters is increased along with layer width.

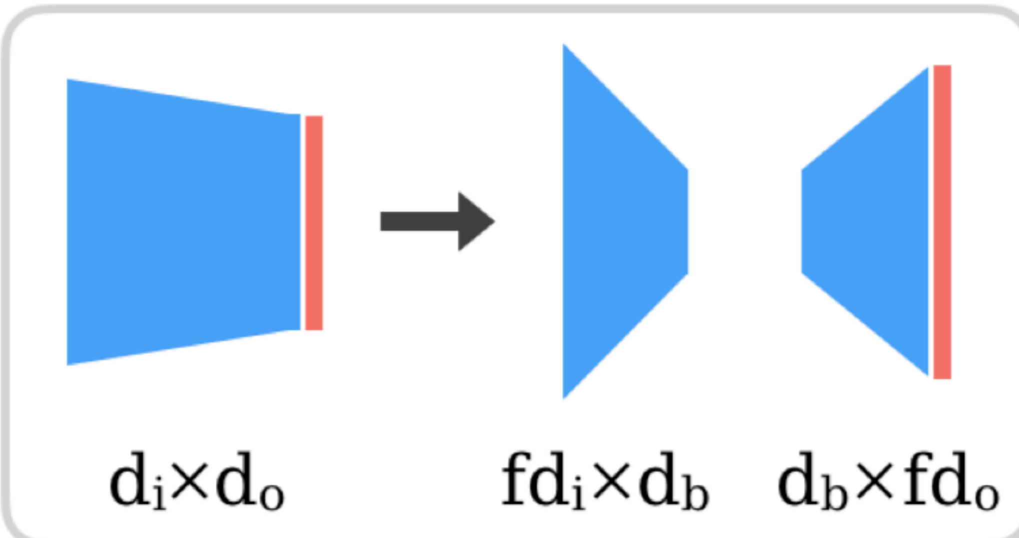
► Is the performance gain due to
more params or larger width?

How to increase width independently of the number of params?



(a) Linear Bottleneck (b) Non-linear Bottleneck (c) Static Sparsity

Linear Bottleneck



- split each layer in two:

$$W \rightarrow W_1 \, W_2$$

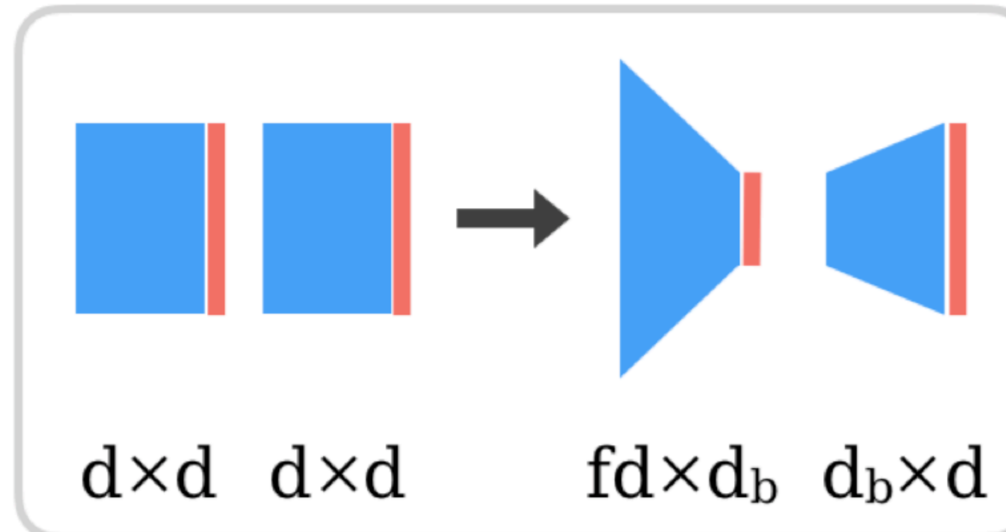
$$\mathbb{R}^{d_i \times d_o} \qquad \mathbb{R}^{d_i \times d_b} \quad \mathbb{R}^{d_b \times d_o}$$

increase d_i , d_o , reduce d_b

no activation function added

- ▶ changes depth
- ▶ strongly affects trainability

Non-Linear Bottleneck

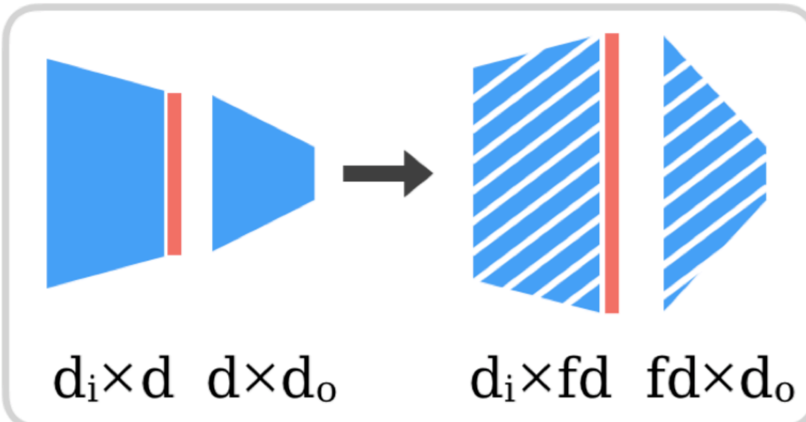


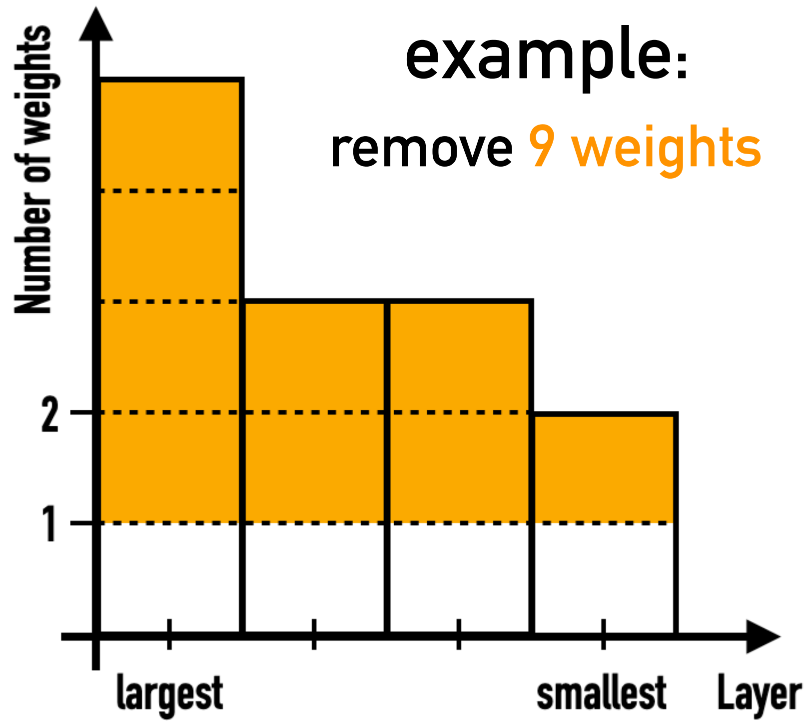
- modify layers in pairs:

increase the outer dimensions,
reduce the inner dimension

- ▶ leads to worse performance

Static Sparsity





- sparsity pattern: random, applied at initialization, static
- per-layer distribution according to layer size
- in-layer distribution uniform across all layer dimensions
- method advantage: it does not alter the NN structure

❗ we are not aiming for performance gains through sparsity

Our approach in summary:

- select model type and architecture

  baseline: dense model (full connectivity)

e.g. ResNet-18,
with 32 output channels
in the first conv layer

- fix the number of weights

- build a family of models having different widths and sparsity, but same number of weights

  wide & sparse: increase the width and
  remove excess weights

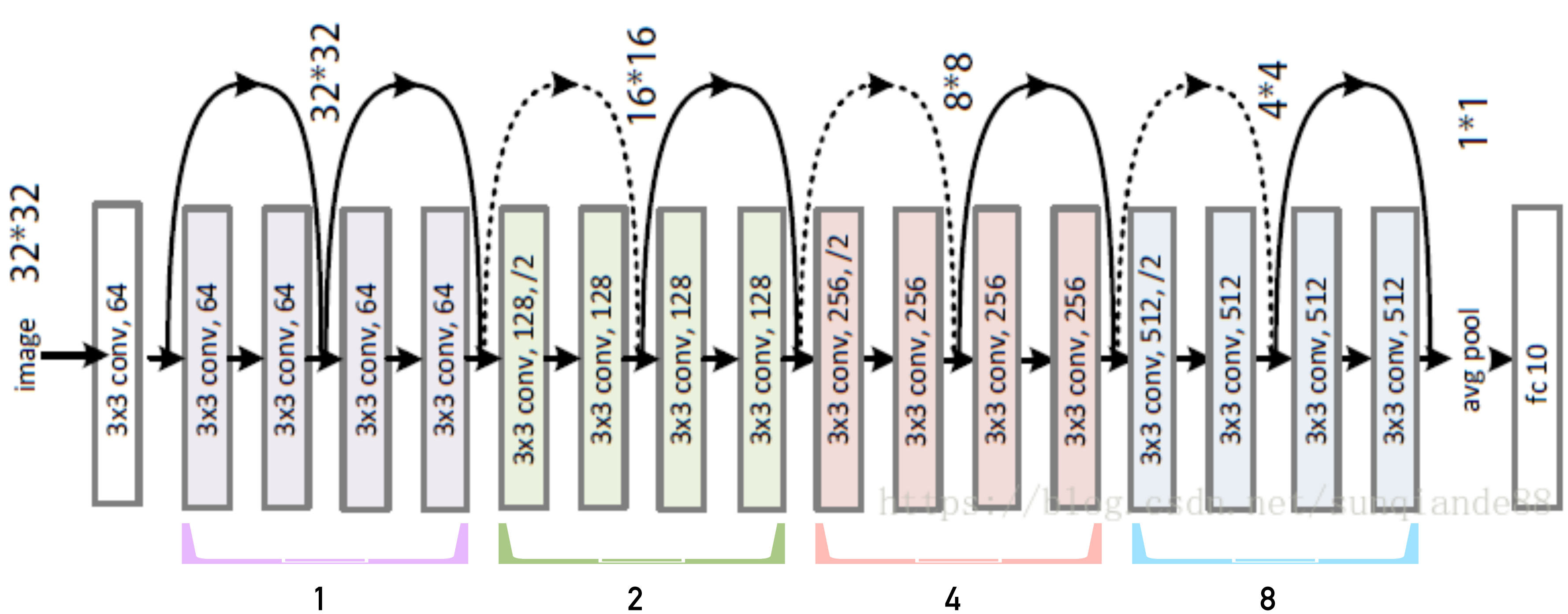
- train and compare performance

  (task: image classification)

Our approach in summary:

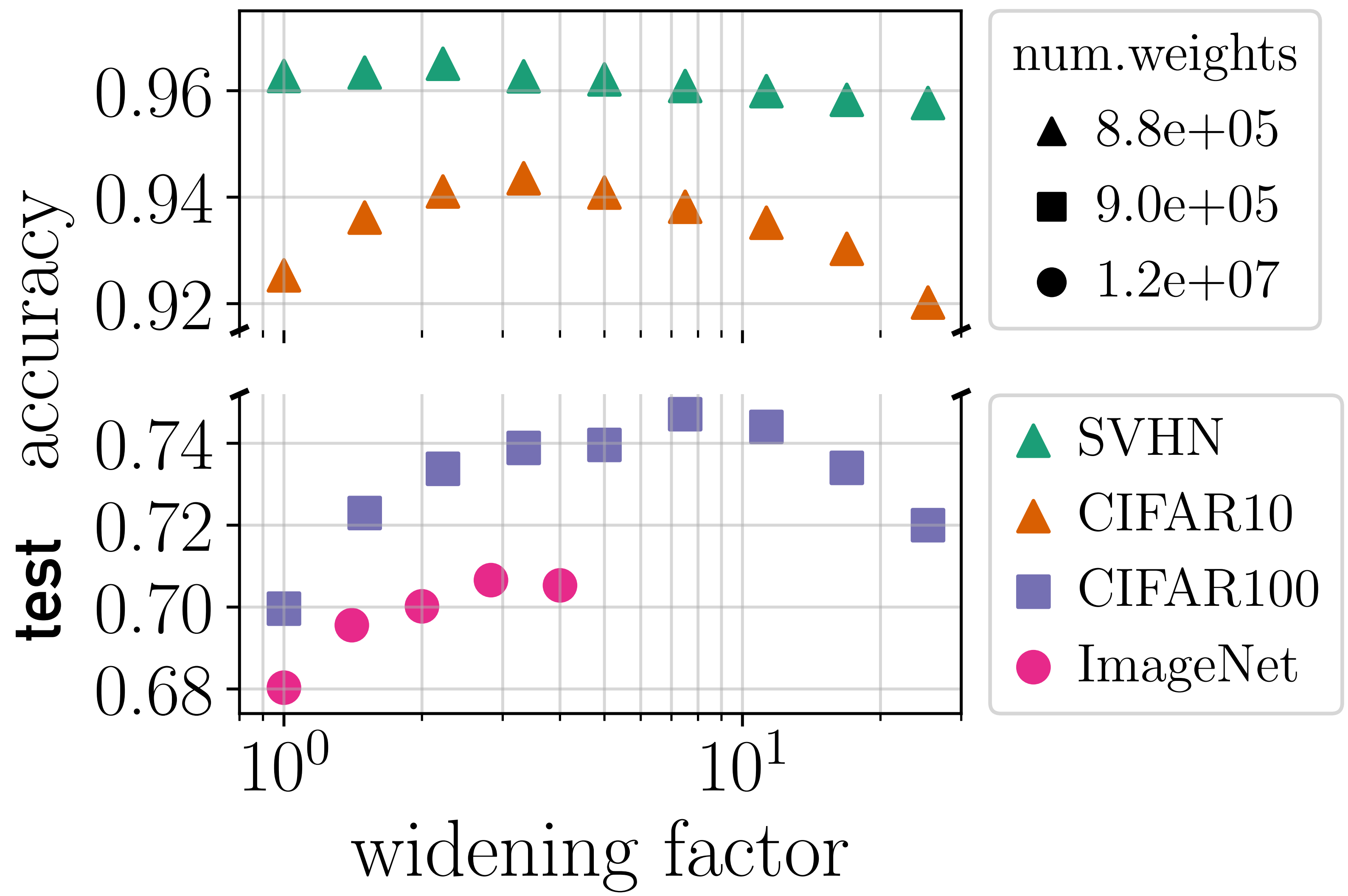
- select model type and architecture

  e.g. ResNet-18:



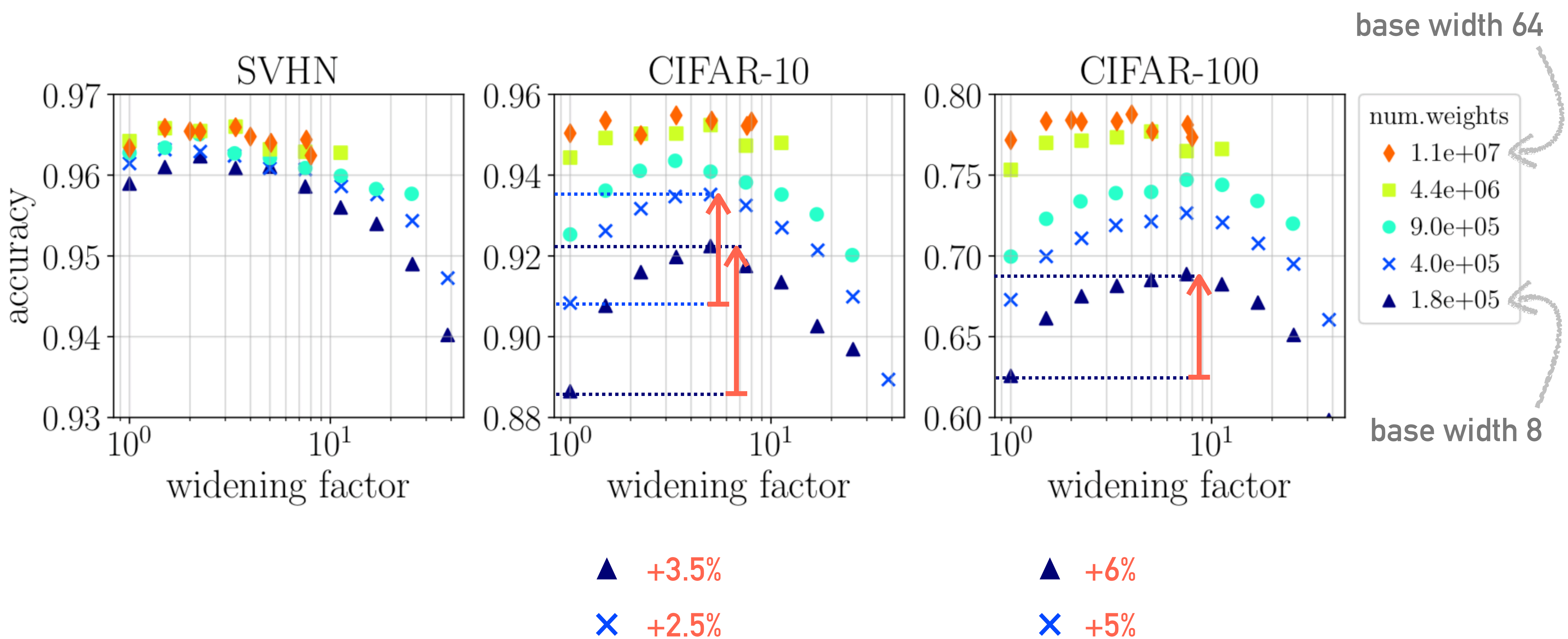
- ▸ model width = number of output channels in the first convolutional layer
- ▸ the widths of all subsequent layers are set according to 1:2:4:8

Results: ResNet-18



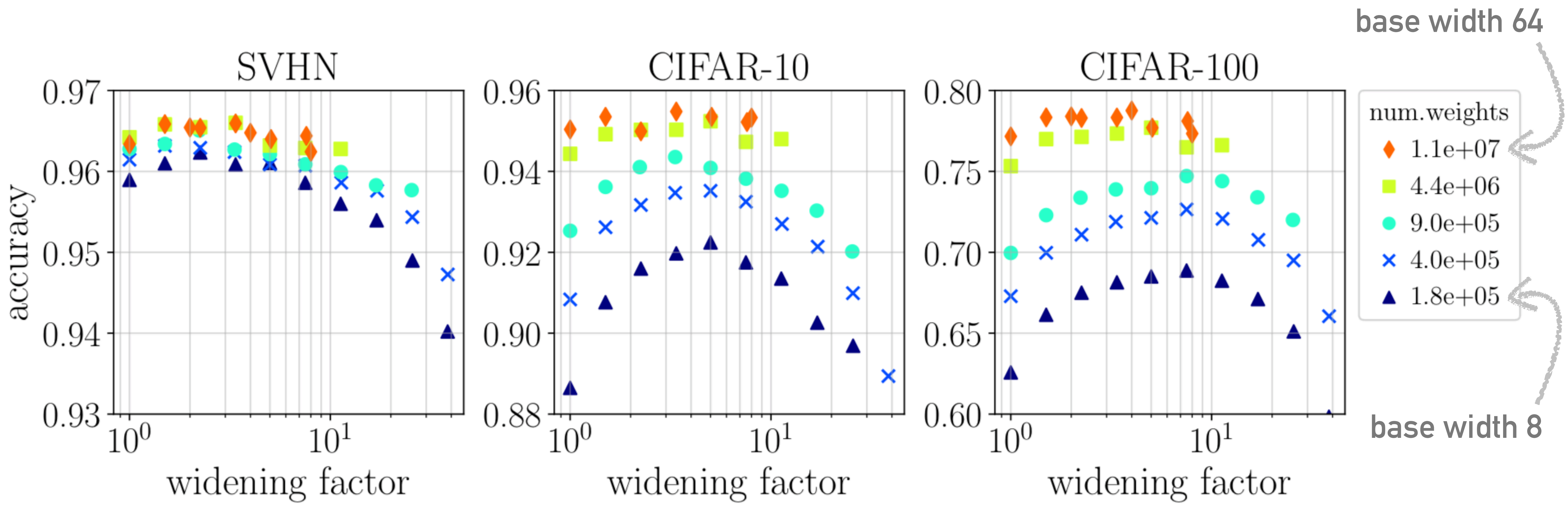
- performance improves as width is increased, even though the number of weights is fixed

ResNet-18 on CIFAR and SVHN



- the improvement is strongest for smaller models & harder tasks

ResNet-18 on CIFAR and SVHN



▲ +3.5%
✕ +2.5%

▲ +6%
✕ +5%

- the improvement is strongest for smaller models & harder tasks

ResNet-18 on ImageNet

width	64	90	128	181	256
dense	68.03 (11.7)	69.11 (22.8)	70.22 (45.7)	70.91 (90.7)	71.89 (180.6)
sparse	–	69.56 (11.7)	70.02 (11.7)	70.66 (11.7)	70.53 (11.7)

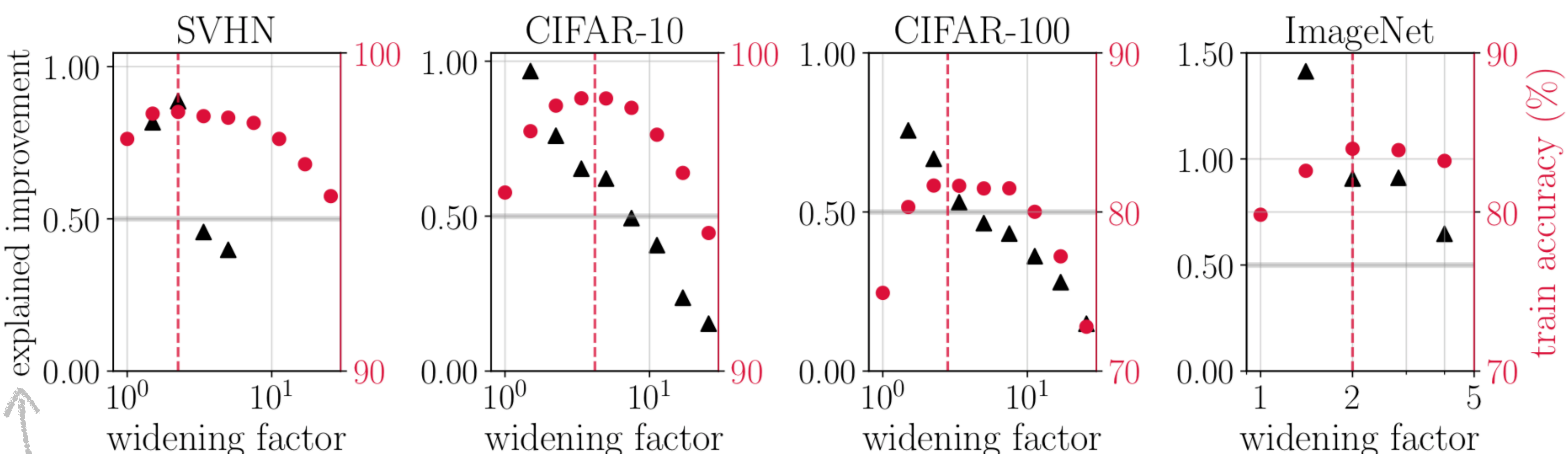
top-1 test accuracy in %

num. weights in 10^6

- the improvement with increasing width obtained by the sparse models is on par with the dense models (up to a ceretain max. width)

How much improvement is due to width only?

- compare perf increase for wide & sparse to wide & dense models



$$= \frac{\Delta \text{ test acc. (sparse)}}{\Delta \text{ test acc. (dense)}}$$

improvement due to width

improvement due to width + weights

- as long as the model can achieve high training accuracy, most of the improvement in performance can be attributed to the width

Theory: ∞-width limit and GP kernels

GP = Gaussian Process

- hypothesis: performance improvement is correlated with having a GP kernel that is closer to the ∞-width kernel
- hypothesis: the distance to the ∞-width kernel can be reduced by increasing network width without adding weights

- ▸ compute GP kernel of a sparse ReLU net with 1 hidden layer
- ▸ find strong correlation between the model performance and the distance to the ∞-width kernel

Theory: ∞-width limit and GP kernels

- network

$f : \mathbb{R}^d \to \mathbb{R}^D \qquad f(x) = \alpha v[ux]_+$

ReLU

const.

$x \in \mathbb{R}^d \qquad \underbrace{u \in \mathbb{R}^{n\times d} \qquad v \in \mathbb{R}^{D\times n}}_{\text{parameters } \theta} \qquad [z]_+ = zH(z) \qquad \alpha \in \mathbb{R}$

Heaviside step function

$$\mathbf{Pr}(\theta) = p\, \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\theta^2}{2\sigma^2}\right) + (1-p)\ \delta(\theta)$$

probability of param being non-zero
= connectivity

- GP kernel: $\Theta(x,y) = \nabla_v f(x) \cdot \nabla_v f(y)$
  - distance from the ∞-width kernel

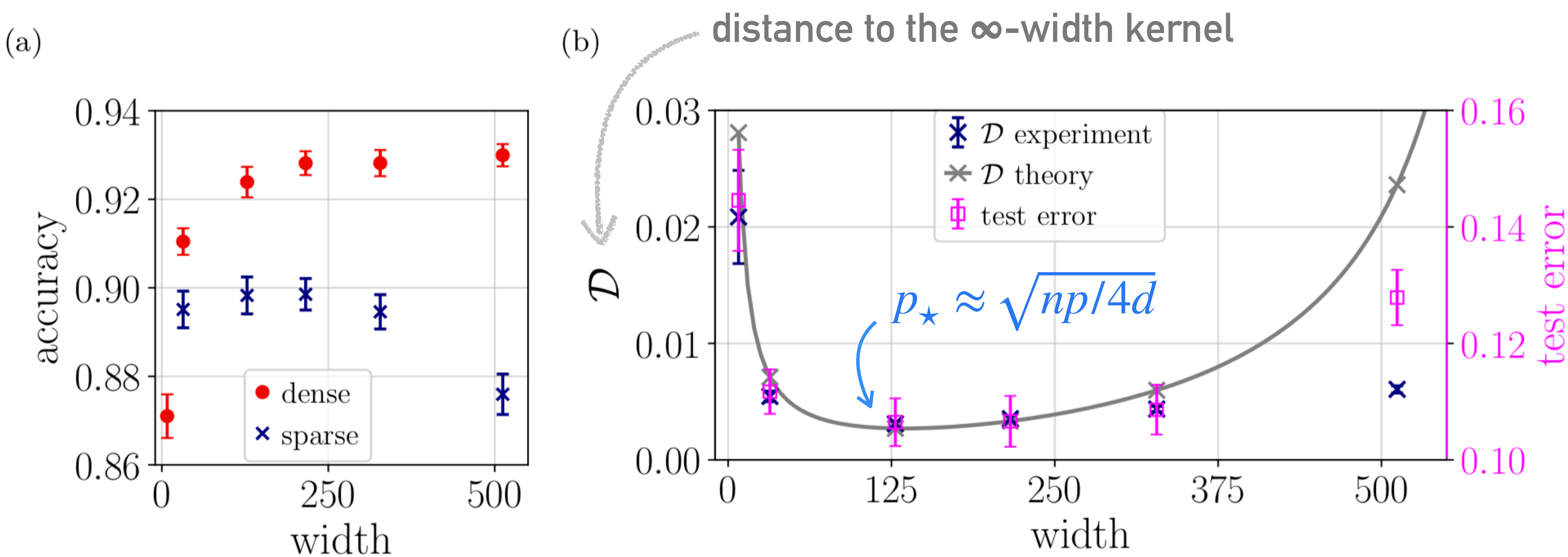
$$\mathbb{E}_\theta \left[(\Theta_\infty(x,y) - \Theta_{n,p}(x,y))^2 \right]$$

infinite-width kernel, dense

finite width n, connectivity p

Sparse GP kernel and its distance to the ∞-width kernel

MLP on MNIST, 1 hidden layer, ReLU



theory predicts optimal connectivity $p_\star$ with $np =$ const.

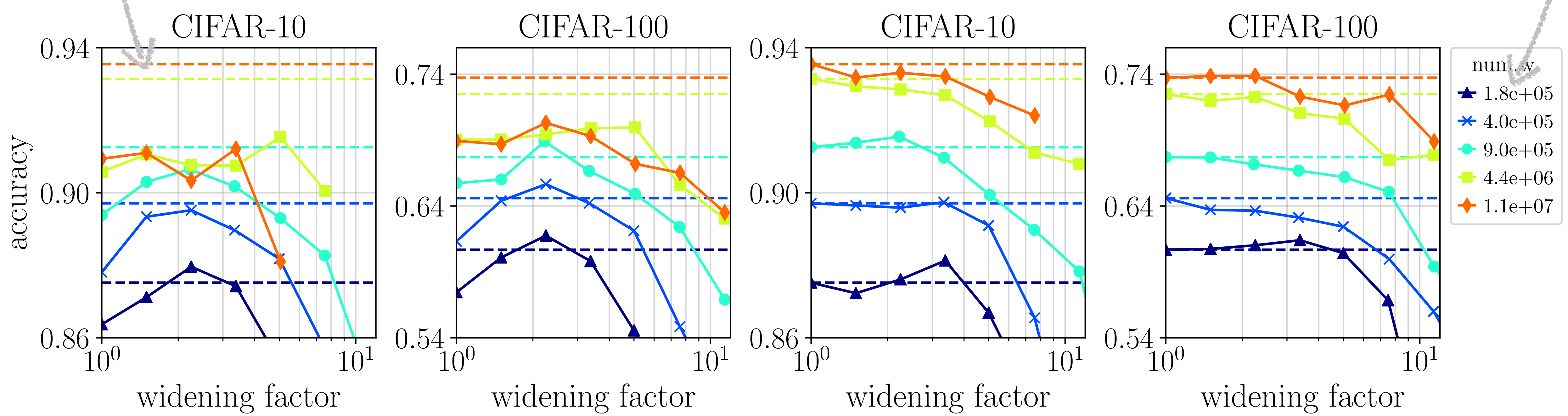
thanks!

appx

Bottleneck Results

dashed lines:
performance of the baseline net

num. weights
in the baseline net



Linear Bottleneck

Non-Linear Bottleneck

MLP on MNIST
1 hidden layer