Designing Network Design Spaces

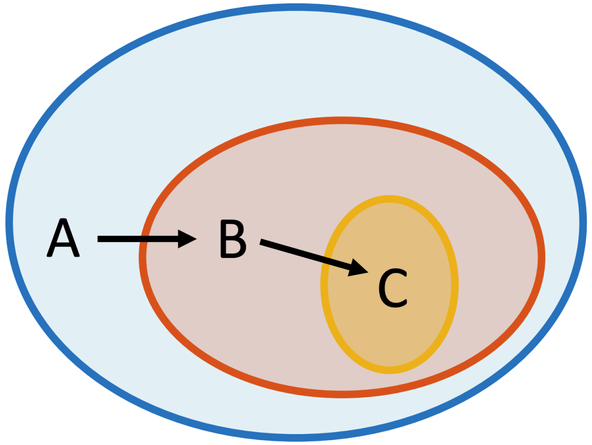
ВШЭ, 2022

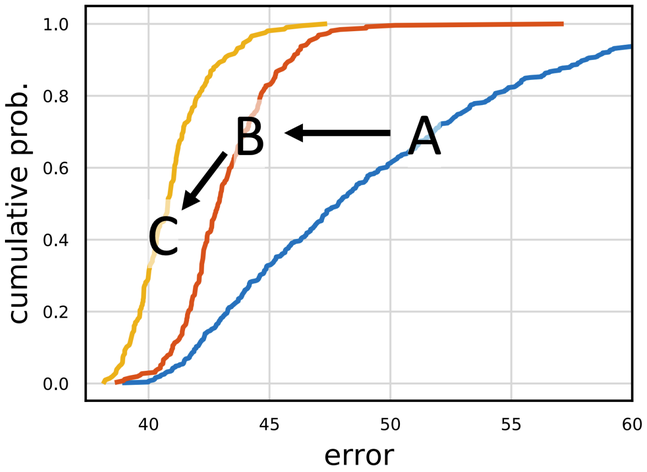
https://arxiv.org/pdf/2003.13678.pdf

TL DR

- **Задача:** поиск оптимальной с точки зрения качества архитектуры нейронной сети

- **Идея:** вместо подгона архитектуры под конкретную задачу, искать пространство “хороших” архитектур сетей, из которого потом можно будет сэмплировать архитектуру под заданные вычислительные ограничения и обучать





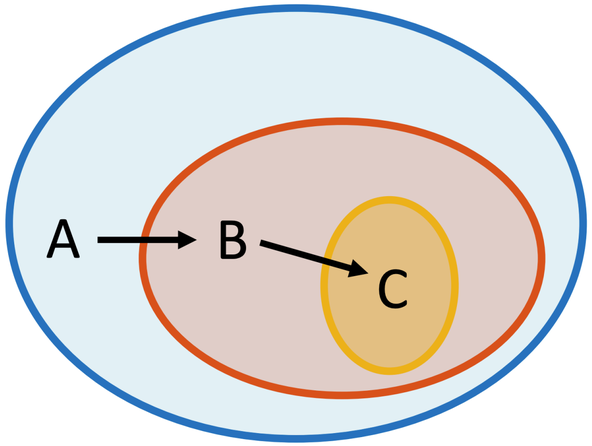
TL DR

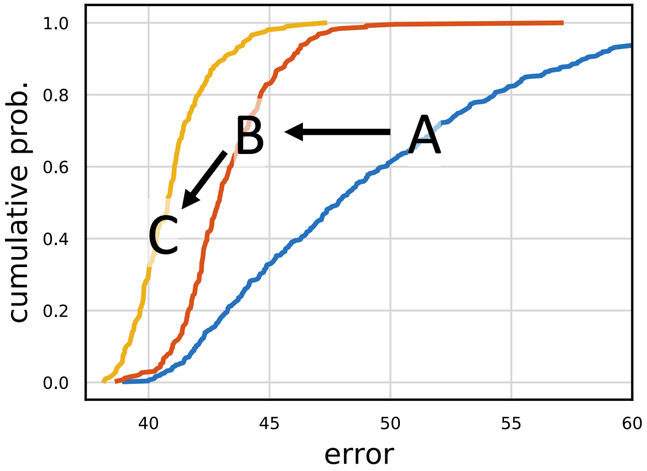
Понятный пайплайн поиска архитектур

1. Сэмплируем n моделей из пространства
2. Обучаем
3. Строим EDF графики; зависимости качества от параметров
4. На основе инсайтов органичиваем пространство поиска
5. Получаем упрощенное пространство поиска с большей концентрацией хороших моделей

TL DR

Понятный пайплайн поиска архитектур





Пайплайн поиска архитектур

Сэмплирование моделей

Основное предположение: пространство моделей можно охарактеризовать сэмплом моделей из этого пространства

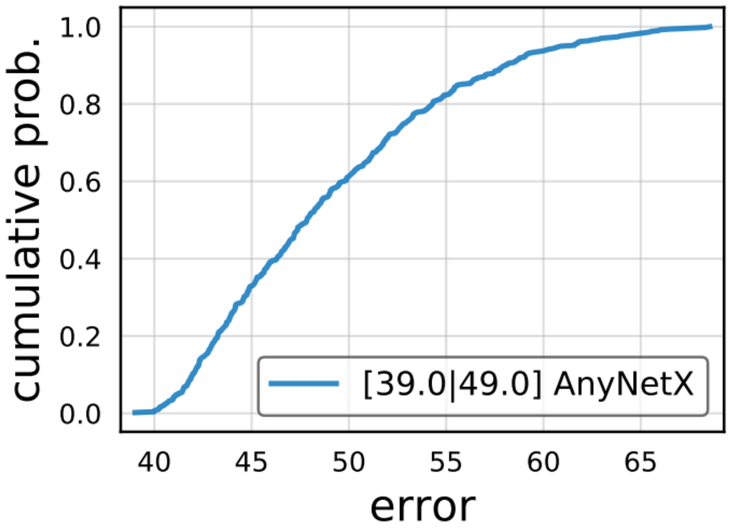
1. Сэмплируем $n = 500$ легких моделей (предлагается брать 400MF) моделей

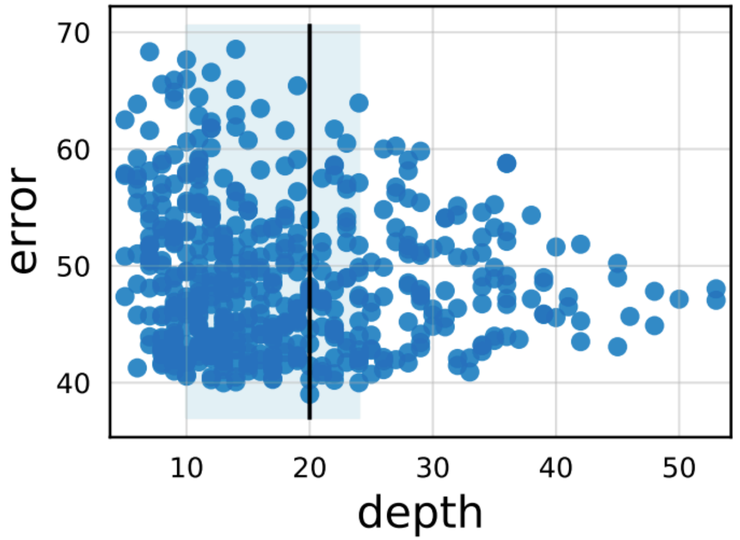
2. Обучаем небольшое число эпох (10)

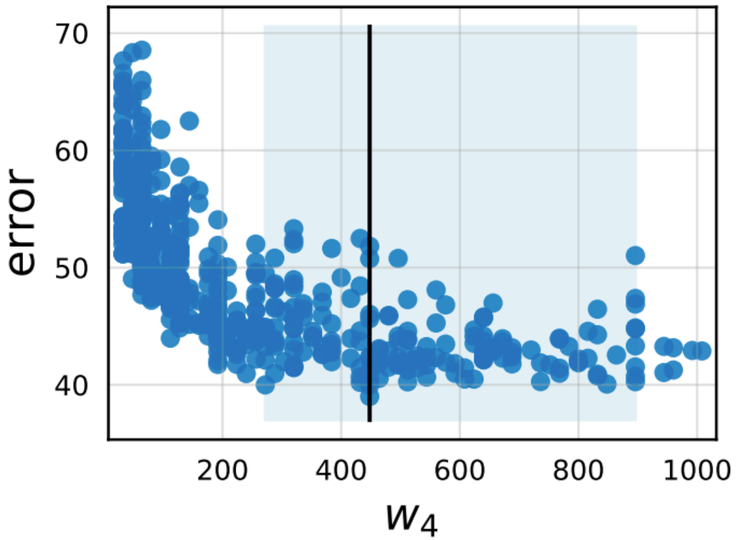
Пайплайн поиска архитектур

Анализ пространства

- Эмпирическая функция распределения ошибки: $F(e) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}[e_i < e]$
- Зависимости ошибки от параметров
- Статистики для выявления наиболее вероятных ограничений (бутстрап)
- На левом графике в легенде записаны минимальная и средняя ошибки



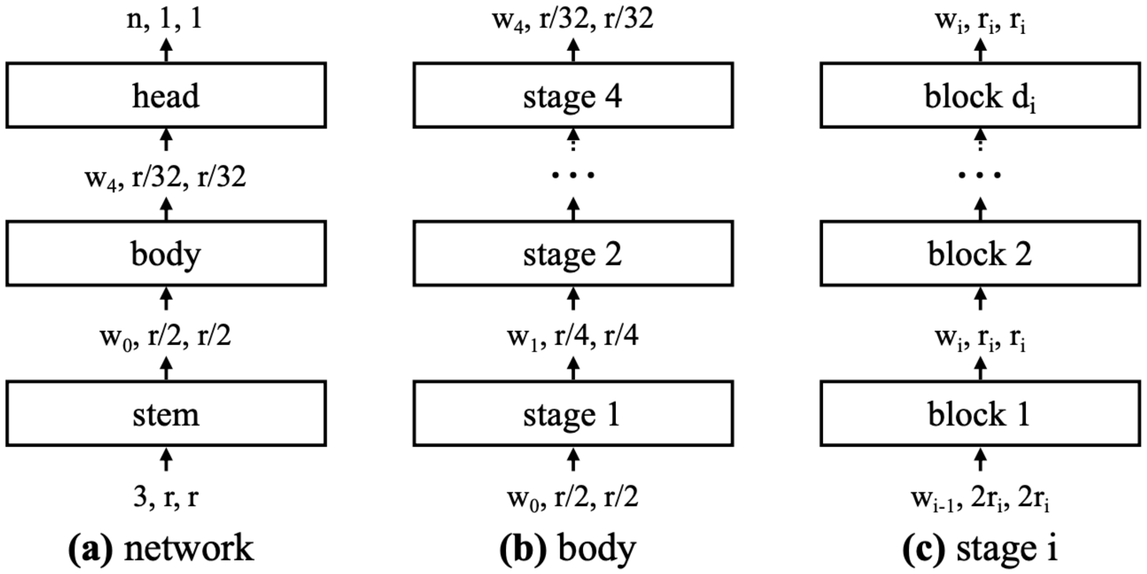


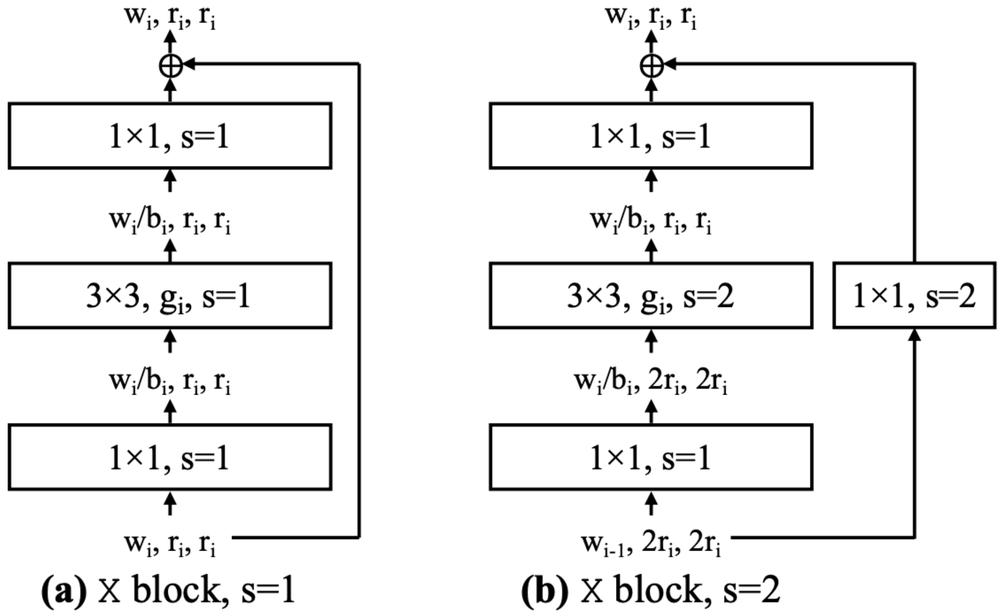


AnyNetX

Исходное пространство поиска

- Модель строится из кирпичиков под названием X block
- X block состоит из двух 1x1 сверток и одной grouped 3x3 свертки
- stem - 3x3 свертка; head - avg pooling + FC

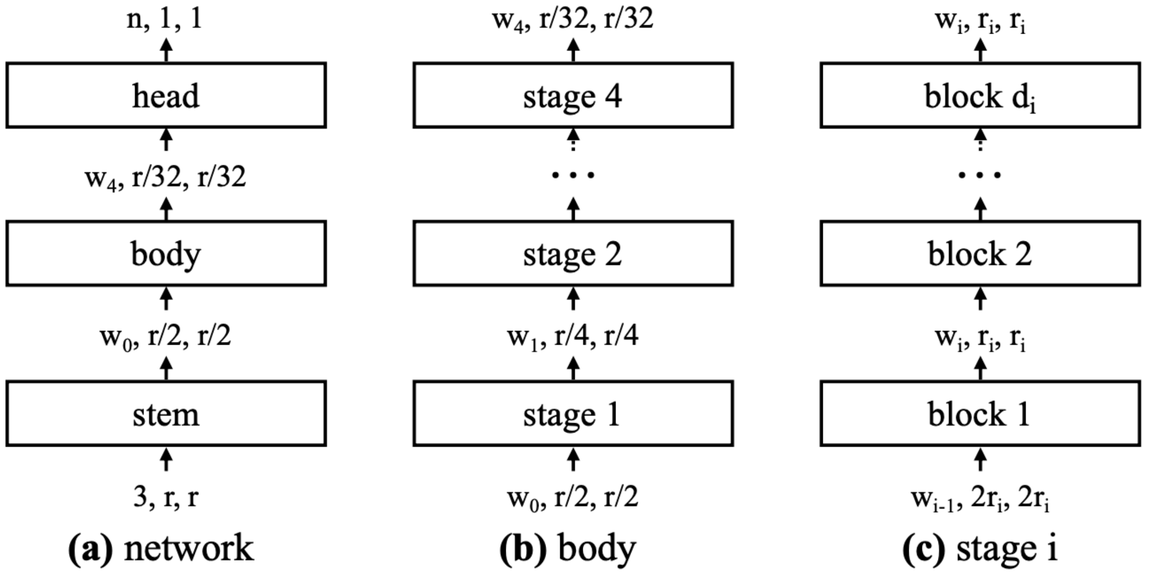




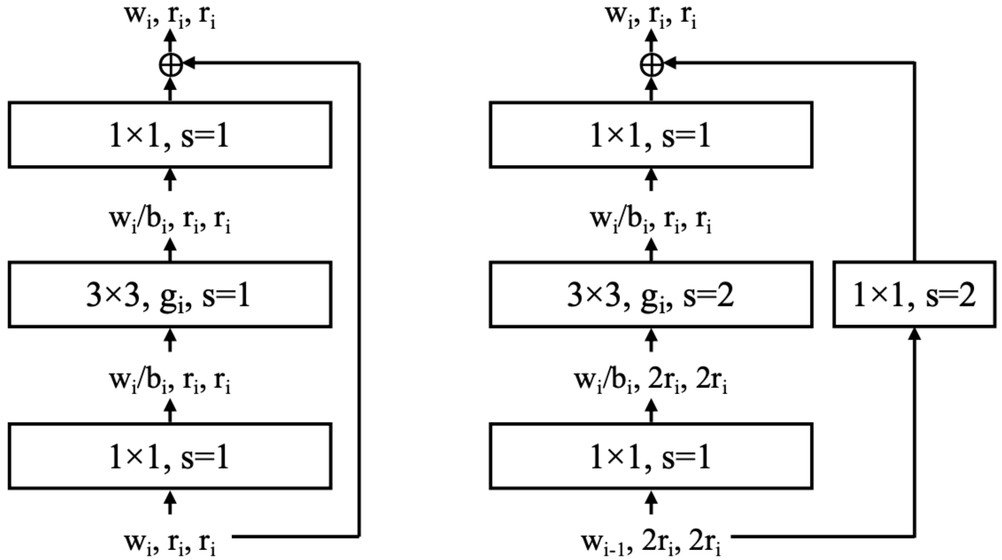
AnyNetX

Исходное пространство поиска. $AnyNetX_A$

- $d_i \leq 16,\ w_i \leq 1024$, кратные 8; сэмплируем из лог нормального распределения
- $b_i \in \{1, 2, 4\},\ g_i \in \{1, 2, \cdots, 32\}$
- получаем $(16 \cdot 128 \cdot 3 \cdot 6)^4 \approx 10^{18}$ моделей



(a) network **(b)** body **(c)** stage i

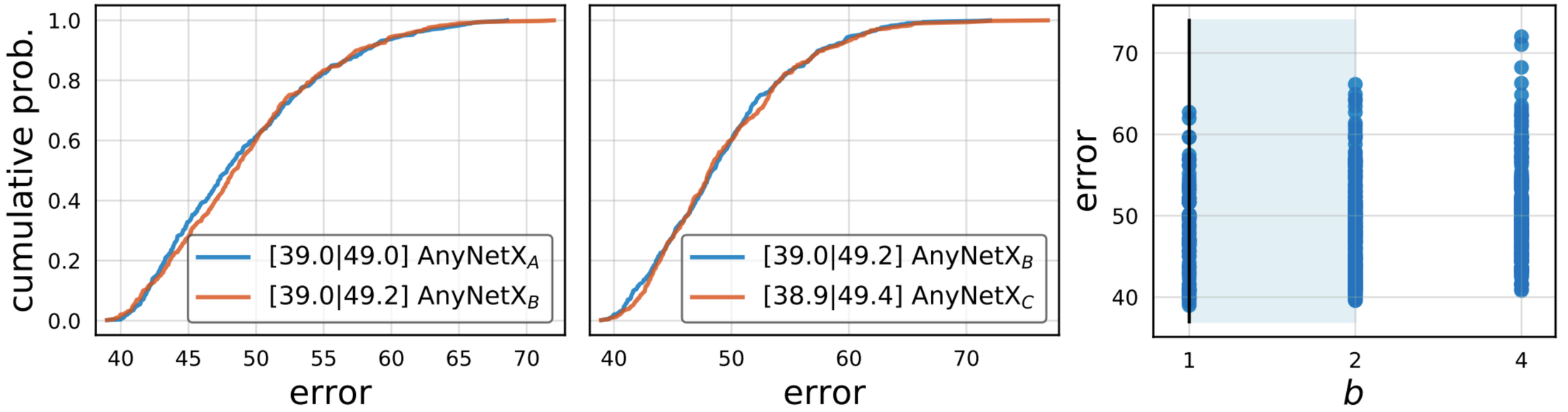


(a) X block, s=1 **(b)** X block, s=2

AnyNetX

Уменьшение пространства поиска

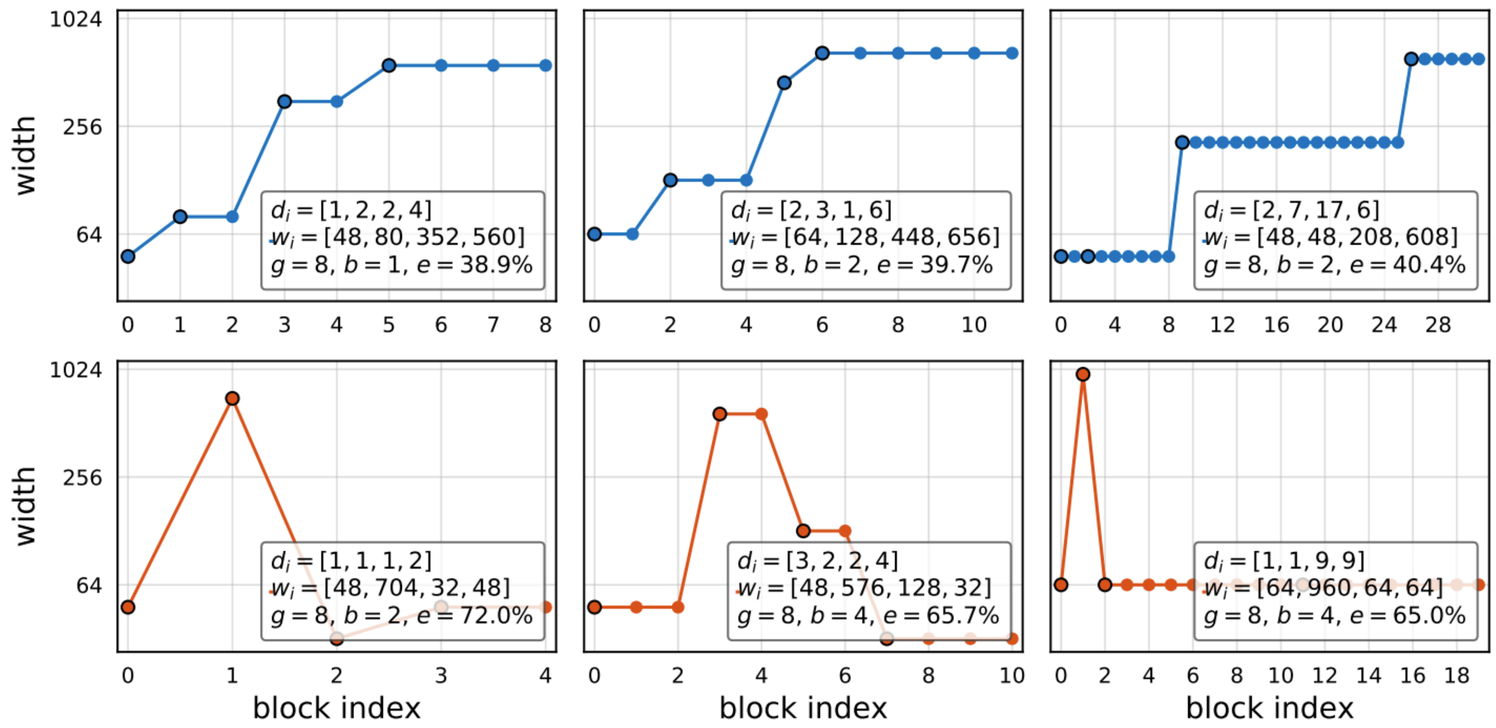
- $\boldsymbol{AnyNetX_B}$: общие веса b_i для всех блоков
- $\boldsymbol{AnyNetX_C}$: общие веса g_i для всех блоков; что интересно, получилось, что лучше брать $g > 1$



AnyNetX

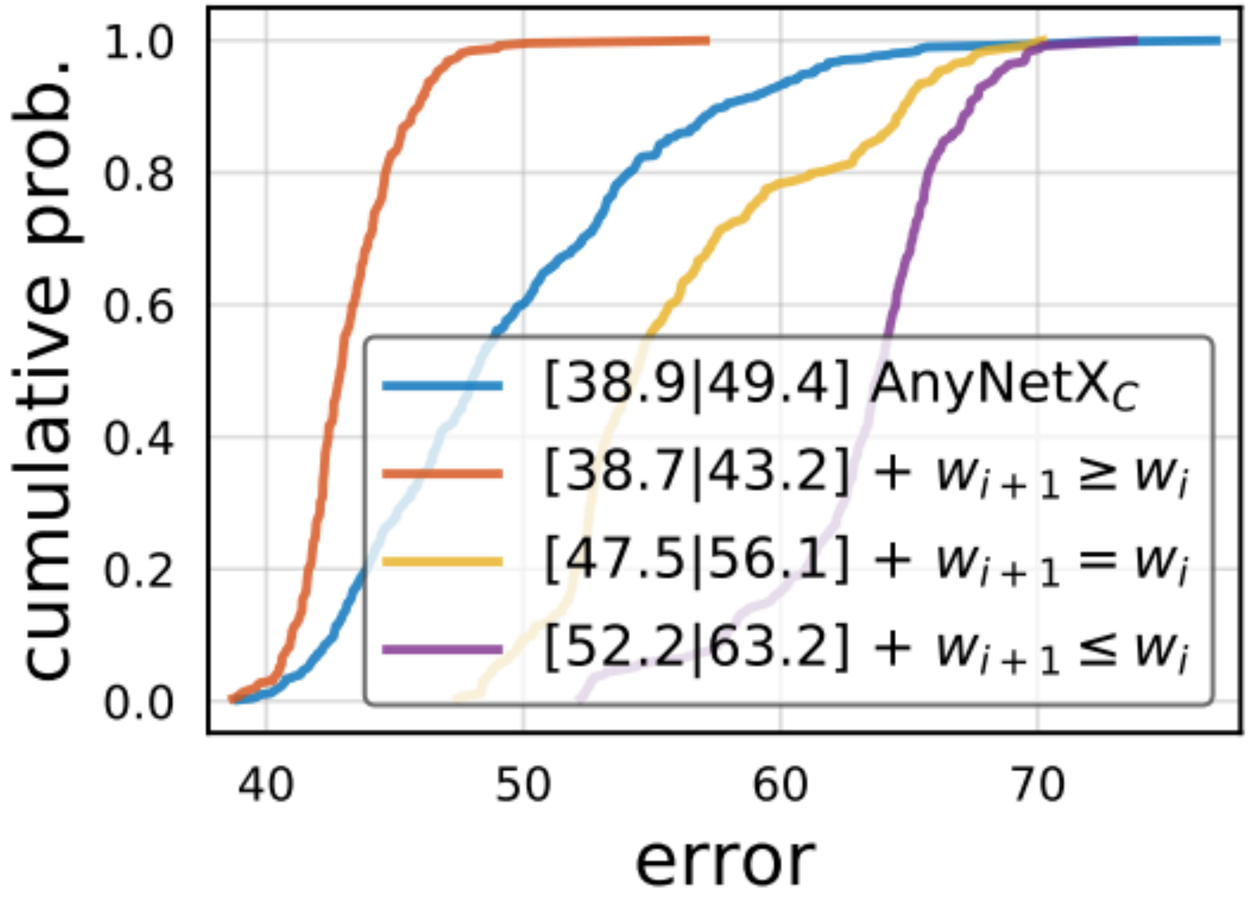
Уменьшение пространства поиска

- $\boldsymbol{AnyNetX_D}$: лучше брать модели с возрастающим числом каналов $w_{i+1} \geq w_i$



AnyNetX

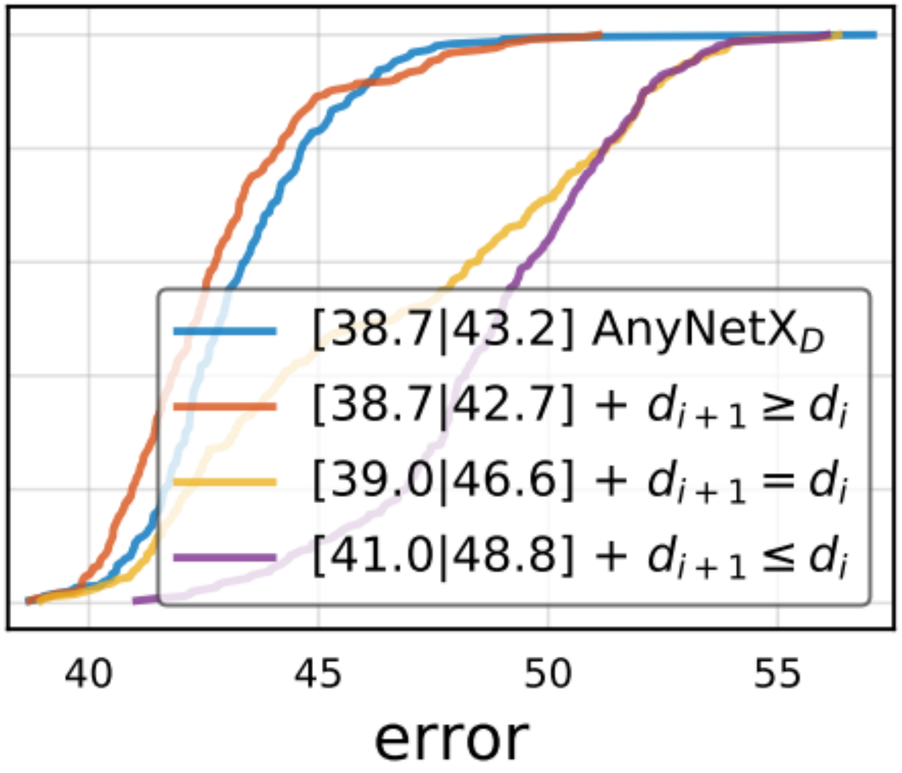
Уменьшение пространства поиска



AnyNetX

Уменьшение пространства поиска

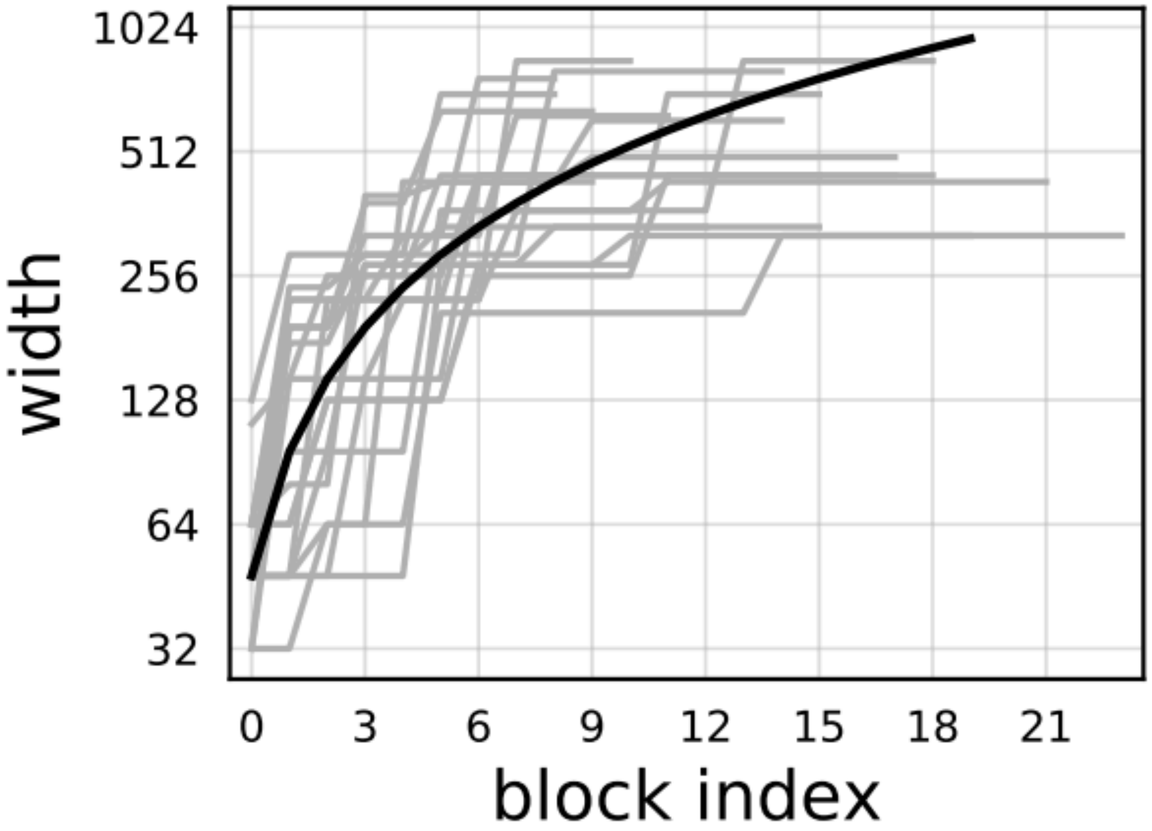
- $\boldsymbol{AnyNetX_E}$: глубину стоит увеличивать в увеличением уровня, т.е. $d_{i+1} \geq d_i$
- В итоге перешли к пространству, у которого в $\approx 10^7$ меньше степеней свободы



RegNet

Параметризация числа каналов w_i

- Нарисовали график для $\boldsymbol{AnyNetX_E}$ и заметили, что w_j очень хорошо аппроксимируется линейной функцией $w_j = 48 \cdot (j + 1)$
- Параметризуем $u_j = w_0 + w_a \cdot j$, где $w_0, w_a > 0$
- Для квантизации добавим параметр $w_m > 0$, что $u_j = w_0 \cdot w_m^{s_j}$, где $1.5 \leq w_m \leq 3$
- Находим s_j, тогда $\hat{w}_j = w_0 \cdot w_m^{\lfloor s_j \rceil}$
- $d_i = \sum_j \left[\lfloor s_j \rceil = i \right]$

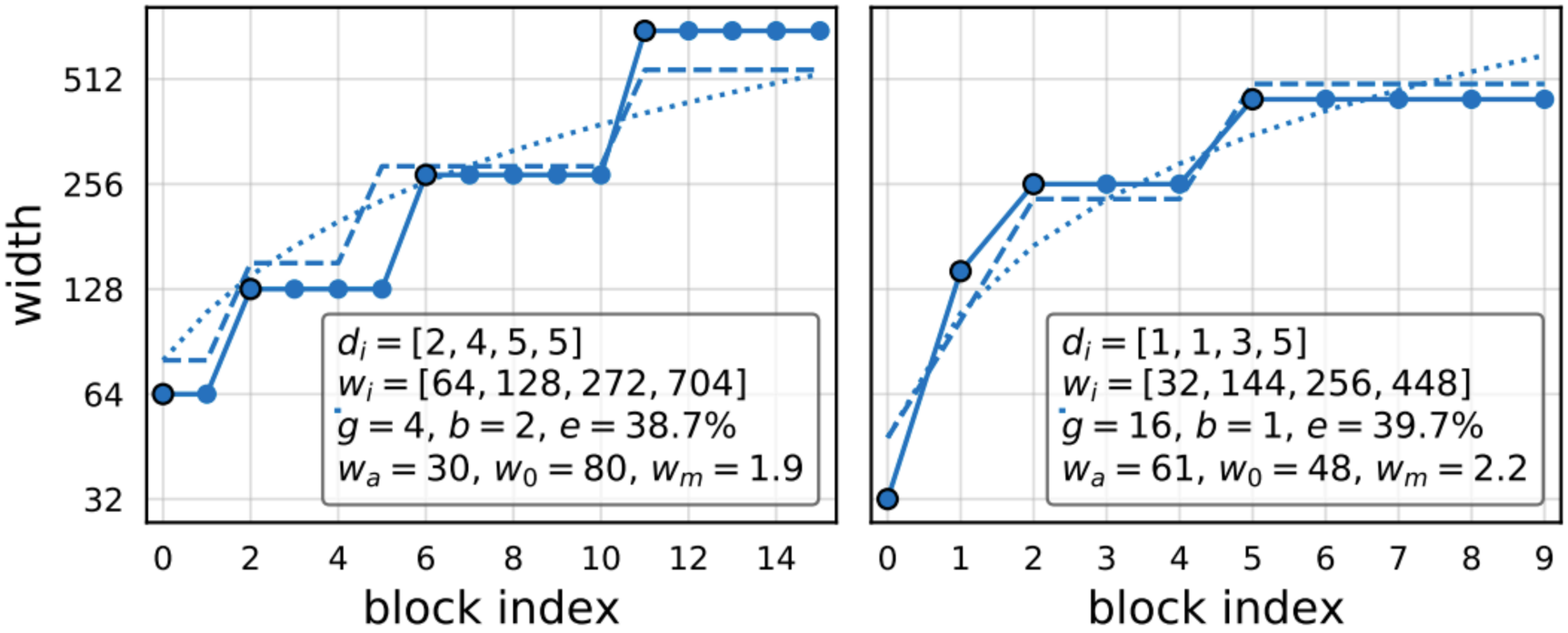


RegNet

Как проверяли?

- Брали конкретную модель
- Запускали гридсерч по w_0, w_a, w_m минимизируя среднюю log-ratio ошибку $\log \frac{\hat{w}_i}{w_i}$
- Предложенная параметризация, как оказалось, неплохо описывает w_i у хороших моделей

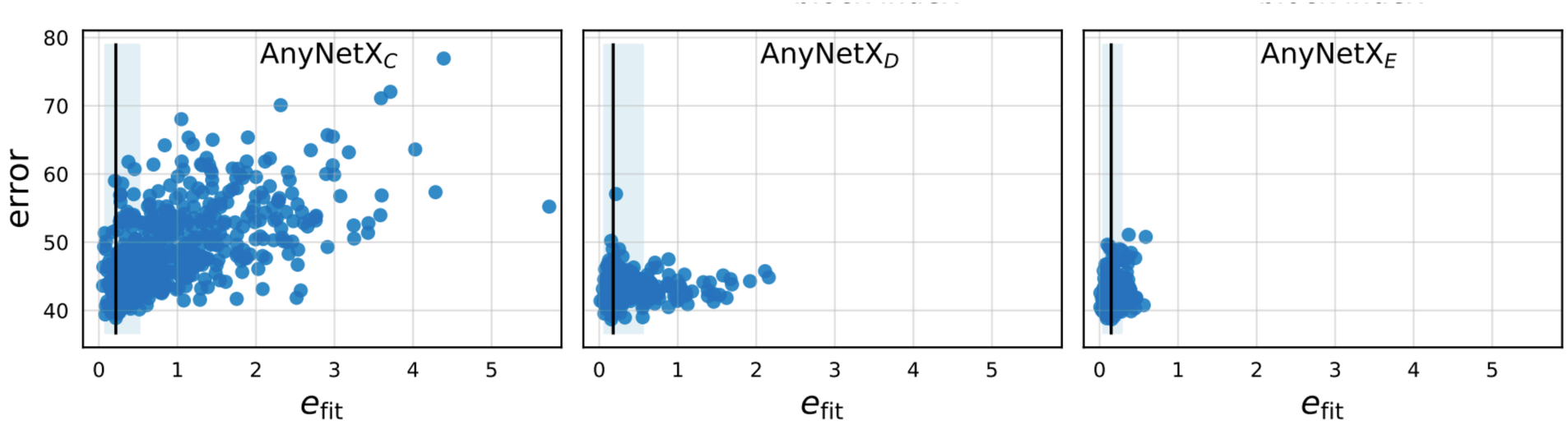
RegNet
Как проверяли?



RegNet

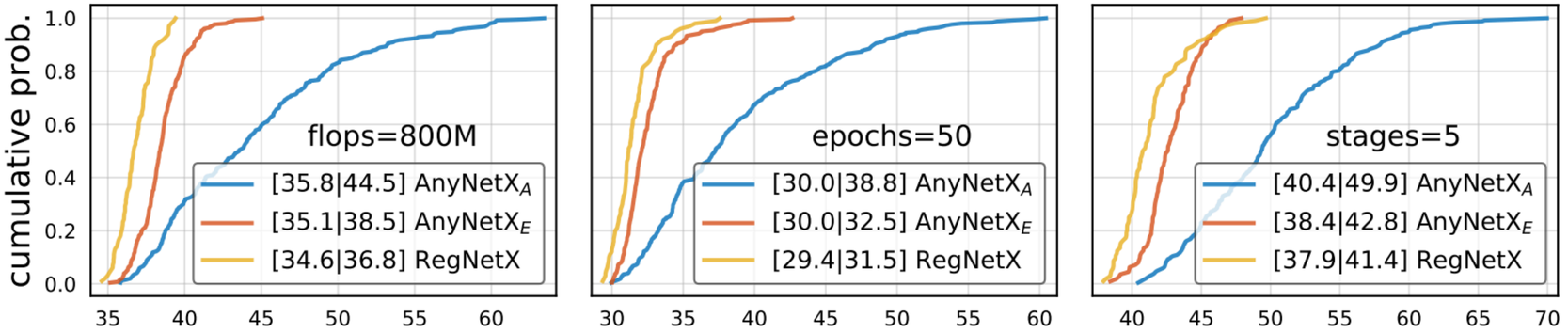
Как проверяли?

- У хороших моделей низкая log-ratio ошибка



RegNet

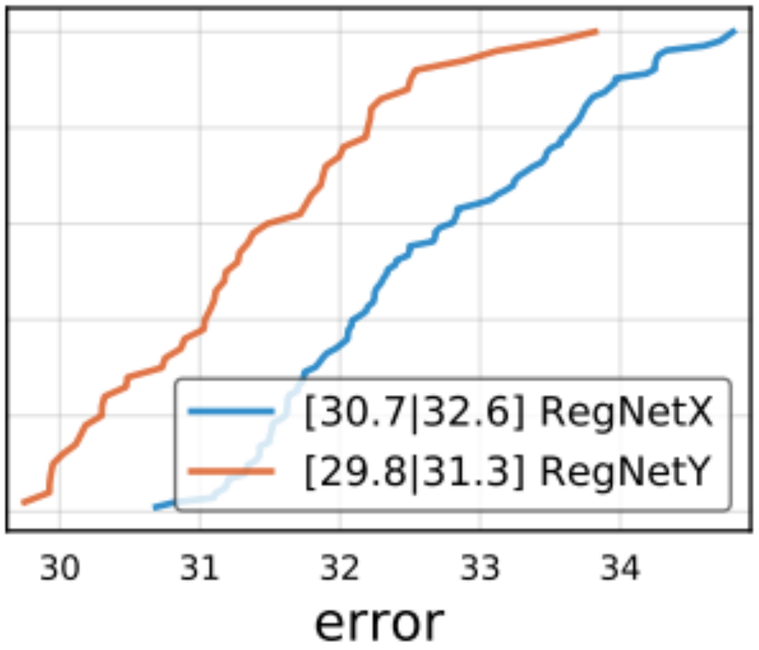
- Улучшили и упростили пространство, но при этом сохранили разнообразие моделей

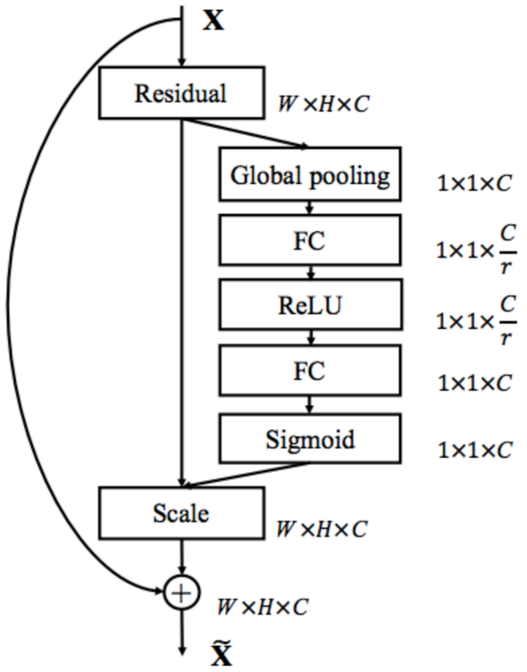


	restriction	dim.	combinations	total
AnyNetX$_A$	none	16	$(16 \cdot 128 \cdot 3 \cdot 6)^4$	$\sim 1.8 \cdot 10^{18}$
AnyNetX$_B$	$+ b_{i+1} = b_i$	13	$(16 \cdot 128 \cdot 6)^4 \cdot 3$	$\sim 6.8 \cdot 10^{16}$
AnyNetX$_C$	$+ g_{i+1} = g_i$	10	$(16 \cdot 128)^4 \cdot 3 \cdot 6$	$\sim 3.2 \cdot 10^{14}$
AnyNetX$_D$	$+ w_{i+1} \geq w_i$	10	$(16 \cdot 128)^4 \cdot 3 \cdot 6 / (4!)$	$\sim 1.3 \cdot 10^{13}$
AnyNetX$_E$	$+ d_{i+1} \geq d_i$	10	$(16 \cdot 128)^4 \cdot 3 \cdot 6 / (4!)^2$	$\sim 5.5 \cdot 10^{11}$
RegNet	quantized linear	6	$\sim 64^4 \cdot 6 \cdot 3$	$\sim 3.0 \cdot 10^8$

RegNet
RegNetY

- Добавим Squeeze-and-Excitation блок
- Получим лучшие результаты

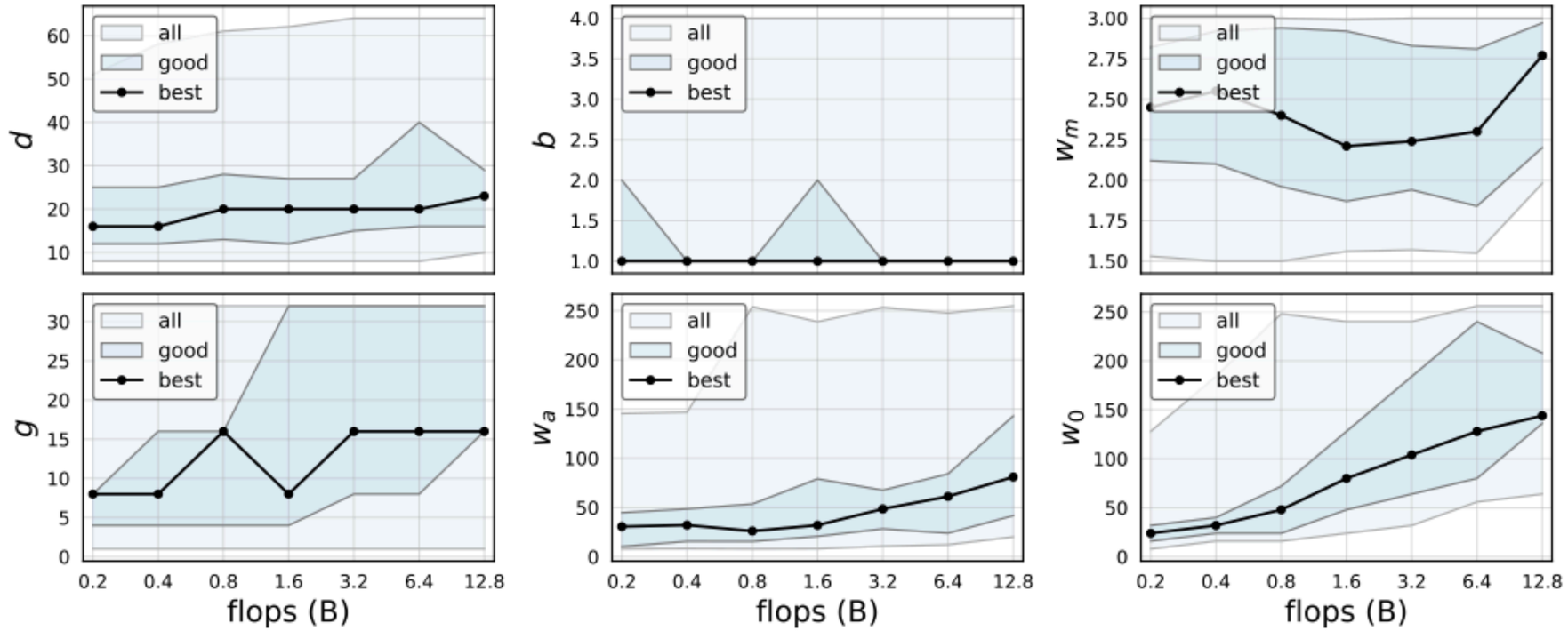




SE-ResNet Module

RegNet
Анализ пространства

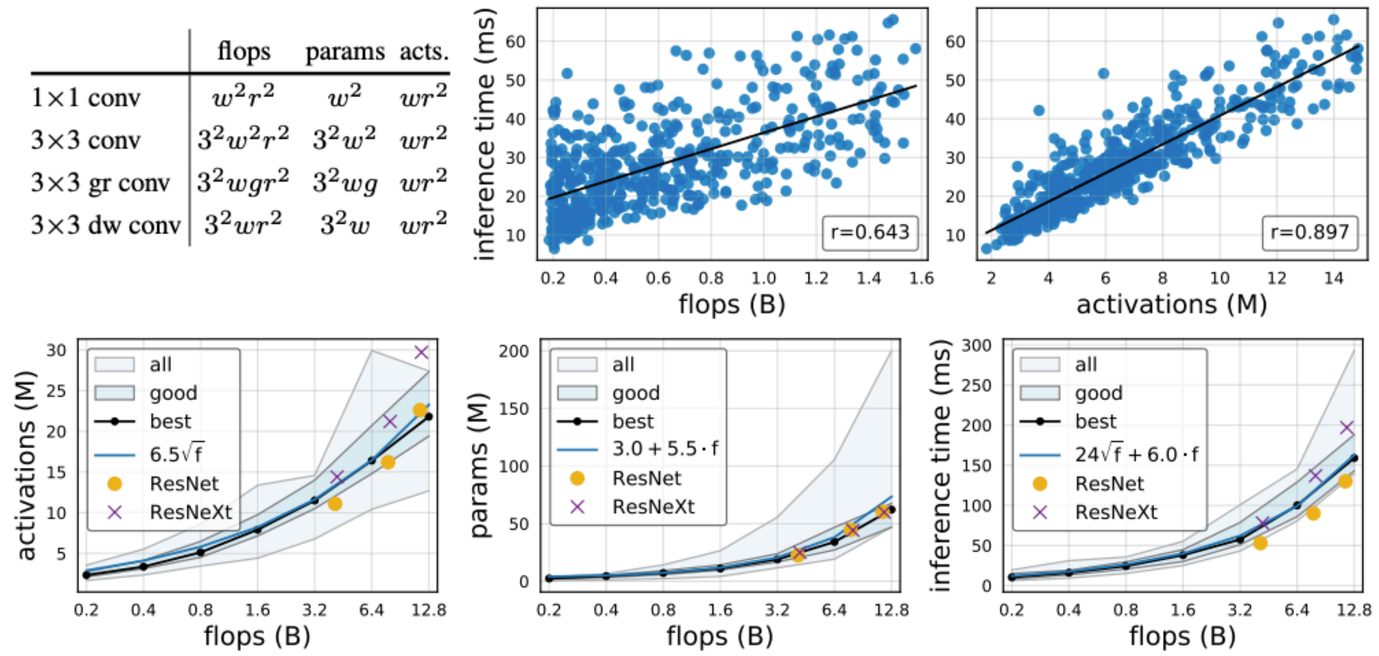
- Оптимальная глубина статична на уровне 20
- Оптимальный bottle-neck ration $b = 1$



RegNet
Анализ пространства

- Число активаций в общем смысле лучше коррелирует со временем на инференс модели

	flops	params	acts.
1×1 conv	w^2r^2	w^2	wr^2
3×3 conv	$3^2w^2r^2$	3^2w^2	wr^2
3×3 gr conv	3^2wgr^2	3^2wg	wr^2
3×3 dw conv	3^2wr^2	3^2w	wr^2



Эксперименты
Mobile regime

- Mobile regime – легковесные модели по 600MF
- Без специальных усилий удалось получить сравнимую с SOTA модель (меньше - лучше)

	flops (B)	params (M)	top-1 error
MobileNet [9]	0.57	4.2	29.4
MobileNet-V2 [25]	0.59	6.9	25.3
ShuffleNet [33]	0.52	-	26.3
ShuffleNet-V2 [19]	0.59	-	25.1
NASNet-A [35]	0.56	5.3	26.0
AmoebaNet-C [23]	0.57	6.4	24.3
PNASNet-5 [17]	0.59	5.1	25.8
DARTS [18]	0.57	4.7	26.7
RegNetX-600MF	0.60	6.2	$25.9_{\pm 0.03}$
RegNetY-600MF	0.60	6.1	$24.5_{\pm 0.07}$

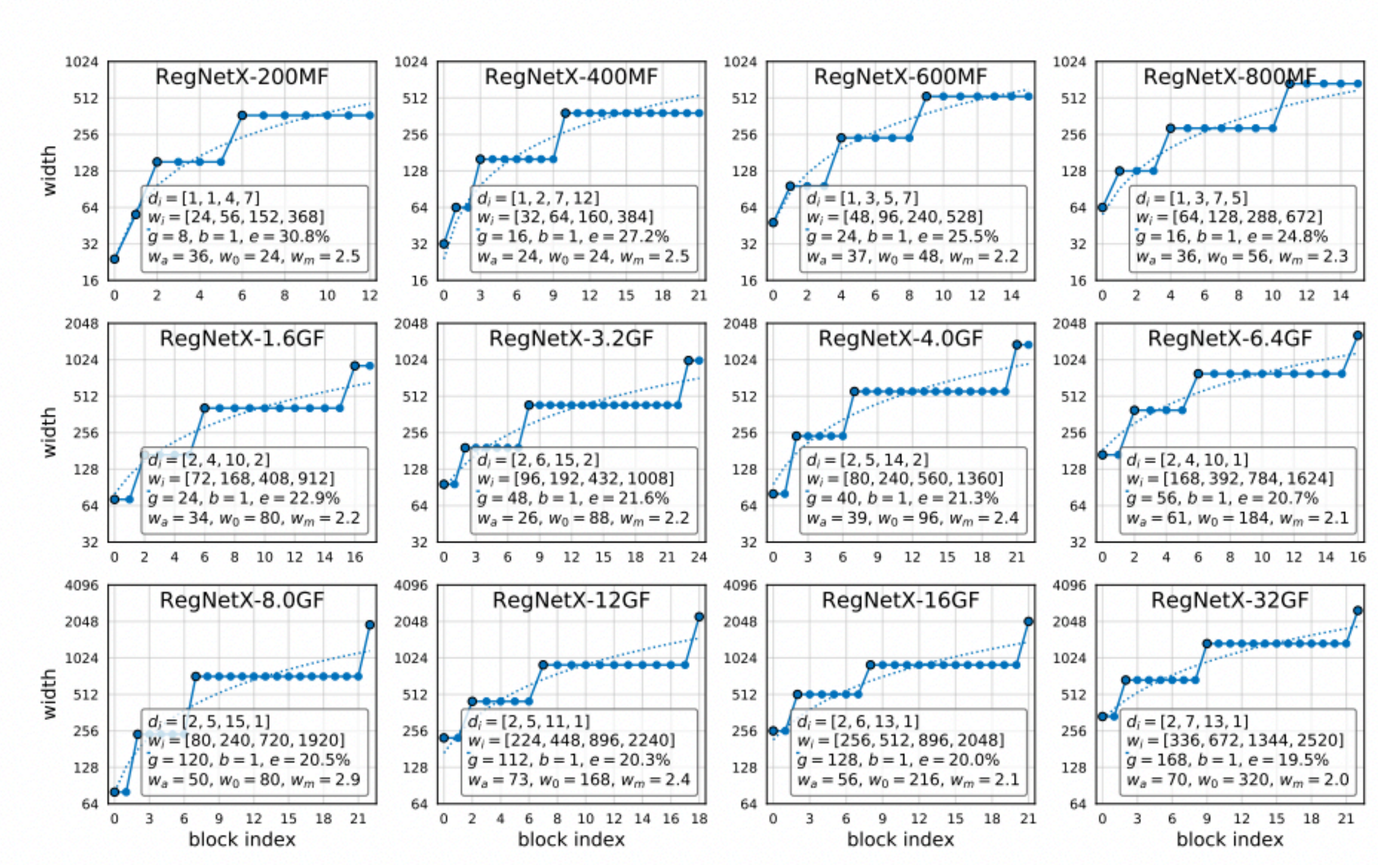
Эксперименты

- Для каждого режима брали самую лучшую модель из 25 случайных
- Учили лучшую модель 5 раз по 100 эпох

	flops (B)	params (M)	acts (M)	batch size	infer (ms)	train (hr)	error (top-1)
RegNetX-200MF	0.2	2.7	2.2	1024	10	2.8	31.1±0.09
RegNetX-400MF	0.4	5.2	3.1	1024	15	3.9	27.3±0.15
RegNetX-600MF	0.6	6.2	4.0	1024	17	4.4	25.9±0.03
RegNetX-800MF	0.8	7.3	5.1	1024	21	5.7	24.8±0.09
RegNetX-1.6GF	1.6	9.2	7.9	1024	33	8.7	23.0±0.13
RegNetX-3.2GF	3.2	15.3	11.4	512	57	14.3	21.7±0.08
RegNetX-4.0GF	4.0	22.1	12.2	512	69	17.1	21.4±0.19
RegNetX-6.4GF	6.5	26.2	16.4	512	92	23.5	20.8±0.07
RegNetX-8.0GF	8.0	39.6	14.1	512	94	22.6	20.7±0.07
RegNetX-12GF	12.1	46.1	21.4	512	137	32.9	20.3±0.04
RegNetX-16GF	15.9	54.3	25.5	512	168	39.7	20.0±0.11
RegNetX-32GF	31.7	107.8	36.3	256	318	76.9	19.5±0.12

Эксперименты

- Получились интересные результаты, что с утяжелением модели, число блоков на последнем уровне падает



Эксперименты
Mobile regime

- Mobile regime – легковесные модели по 600MF
- Без специальных усилий удалось получить сравнимую с SOTA модель (меньше - лучше)

	flops (B)	params (M)	top-1 error
MobileNet [9]	0.57	4.2	29.4
MobileNet-V2 [25]	0.59	6.9	25.3
ShuffleNet [33]	0.52	-	26.3
ShuffleNet-V2 [19]	0.59	-	25.1
NASNet-A [35]	0.56	5.3	26.0
AmoebaNet-C [23]	0.57	6.4	24.3
PNASNet-5 [17]	0.59	5.1	25.8
DARTS [18]	0.57	4.7	26.7
RegNetX-600MF	0.60	6.2	$25.9_{\pm 0.03}$
RegNetY-600MF	0.60	6.1	$24.5_{\pm 0.07}$

Эксперименты
Standard Baselines: ResNe(X)t

- Учились с одними и теми же настройками
- Тоже получили сравнительно лучшие результаты

	flops (B)	params (M)	acts (M)	infer (ms)	train (hr)	top-1 error ours$\pm$std [orig]
ResNet-50	4.1	22.6	11.1	53	12.2	23.2$\pm$0.09 [23.9]
RegNetX-3.2GF	3.2	15.3	11.4	57	14.3	**21.7**$\pm$0.08
ResNeXt-50	4.2	25.0	14.4	78	18.0	21.9$\pm$0.10 [22.2]
ResNet-101	7.8	44.6	16.2	90	20.4	21.4$\pm$0.11 [22.0]
RegNetX-6.4GF	6.5	26.2	16.4	92	23.5	**20.8**$\pm$0.07
ResNeXt-101	8.0	44.2	21.2	137	31.8	20.7$\pm$0.08 [21.2]
ResNet-152	11.5	60.2	22.6	130	29.2	20.9$\pm$0.12 [21.6]
RegNetX-12GF	12.1	46.1	21.4	137	32.9	**20.3**$\pm$0.04

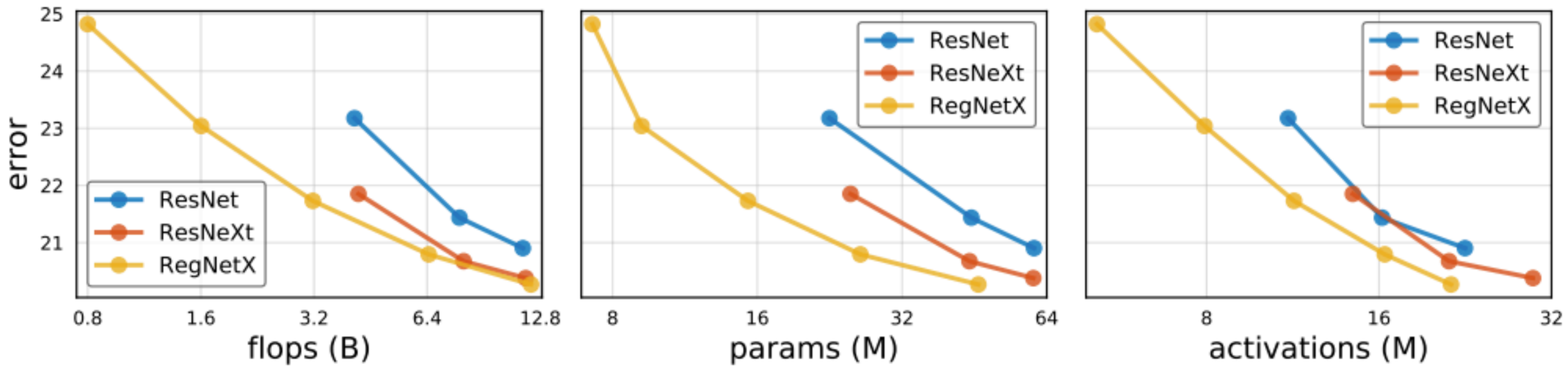
(a) Comparisons grouped by **activations**.

ResNet-50	4.1	22.6	11.1	53	12.2	23.2$\pm$0.09 [23.9]
ResNeXt-50	4.2	25.0	14.4	78	18.0	21.9$\pm$0.10 [22.2]
RegNetX-4.0GF	4.0	22.1	12.2	69	17.1	**21.4**$\pm$0.19
ResNet-101	7.8	44.6	16.2	90	20.4	21.4$\pm$0.11 [22.0]
ResNeXt-101	8.0	44.2	21.2	137	31.8	**20.7**$\pm$0.08 [21.2]
RegNetX-8.0GF	8.0	39.6	14.1	94	22.6	**20.7**$\pm$0.07
ResNet-152	11.5	60.2	22.6	130	29.2	20.9$\pm$0.12 [21.6]
ResNeXt-152	11.7	60.0	29.7	197	45.7	20.4$\pm$0.06 [21.1]
RegNetX-12GF	12.1	46.1	21.4	137	32.9	**20.3**$\pm$0.04

(b) Comparisons grouped by **flops**.

Эксперименты
Standard Baselines: ResNe(X)t

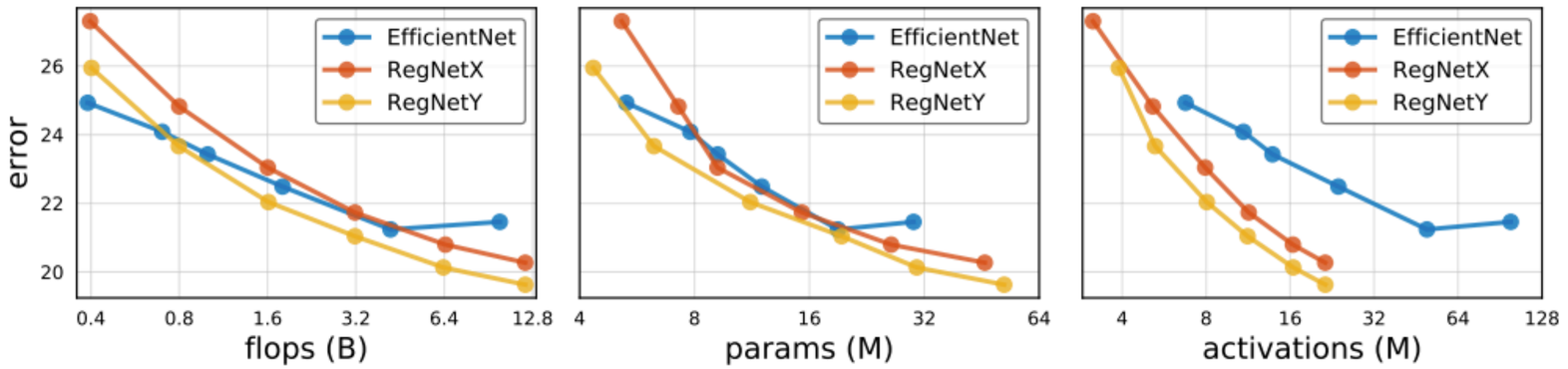
- Хорошие модели доступны на всем спектре режимов



Эксперименты

Сравнение с SOTA: EfficientNet

- Учили в одних и тех же настройках: 100 эпох; с weight decay и адаптивным lr
- На маленьких моделях хуже; на больших – лучше



Эксперименты
Сравнение с SOTA: EfficientNet

- В моделях сильно меньше активаций, что приводит к почти в 5 раз меньшему времени на инференс и обучение

	flops (B)	params (M)	acts (M)	batch size	infer (ms)	train (hr)	top-1 error ours$\pm$std [orig]
EfficientNet-B0	0.4	5.3	6.7	256	34	11.7	**24.9**$\pm$0.03 [23.7]
RegNetY-400MF	0.4	4.3	**3.9**	1024	19	5.1	25.9$\pm$0.16
EfficientNet-B1	0.7	7.8	10.9	256	52	15.6	**24.1**$\pm$0.16 [21.2]
RegNetY-600MF	0.6	6.1	**4.3**	1024	19	5.2	24.5$\pm$0.07
EfficientNet-B2	1.0	9.2	13.8	256	68	18.4	**23.4**$\pm$0.06 [20.2]
RegNetY-800MF	0.8	6.3	**5.2**	1024	22	6.0	23.7$\pm$0.03
EfficientNet-B3	1.8	12.0	23.8	256	114	32.1	22.5$\pm$0.05 [18.9]
RegNetY-1.6GF	1.6	11.2	**8.0**	1024	39	10.1	**22.0**$\pm$0.08
EfficientNet-B4	4.2	19.0	48.5	128	240	65.1	21.2$\pm$0.06 [17.4]
RegNetY-4.0GF	4.0	20.6	**12.3**	512	68	16.8	**20.6**$\pm$0.08
EfficientNet-B5	9.9	30.0	98.9	64	504	135.1	21.5$\pm$0.11 [16.7]
RegNetY-8.0GF	8.0	39.2	**18.0**	512	113	28.1	**20.1**$\pm$0.09

Рецензия
Publication & Authors

- Работа опубликована в 2020 году на конференции IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), это главное ежегодное мероприятие по компьютерному зрению.

- Авторы статьи - команда Facebook AI Research (FAIR)

Рецензия
Competitors

- On Network Design Spaces for Visual Recognition вышедшая в 2020 году на конференции ICCV

Рецензия
Strengths and weaknesses

- Сильные стороны: Проектирование пространств сетевого дизайна является перспективным направлением для будущих исследований

- Слабые стороны: необходимо соблюдений многих условий, которые, на практике выполнить практически невозможно

.