

Ziplt! Merging Models from Different Tasks without Training

Содержание

- ➡ 1. [Какую проблему решаем.](#)
- ➡ 2. [Zip Operation.](#)
- ➡ 3. [Ziplt! Extensions.](#)
- ➡ 4. [Results. CIFAR-10/CIFAR-100.](#)
- ➡ 5. [Results. ImageNet-1k \(200+200\).](#)
- ➡ 6. [Results. Multiple datasets.](#)
- ➡ 7. [Hyperparameter analysis.](#)
- ➡ 8. [Ziplt! for SinGANs.](#)
- ➡ 9. [Итоги доклада.](#)

Какую проблему решаем



Хотим объединять несколько моделей в одну.



Умеем переобучать модель с одного домена данных на другой, но при этом наблюдается эффект забывания первоначального домена.



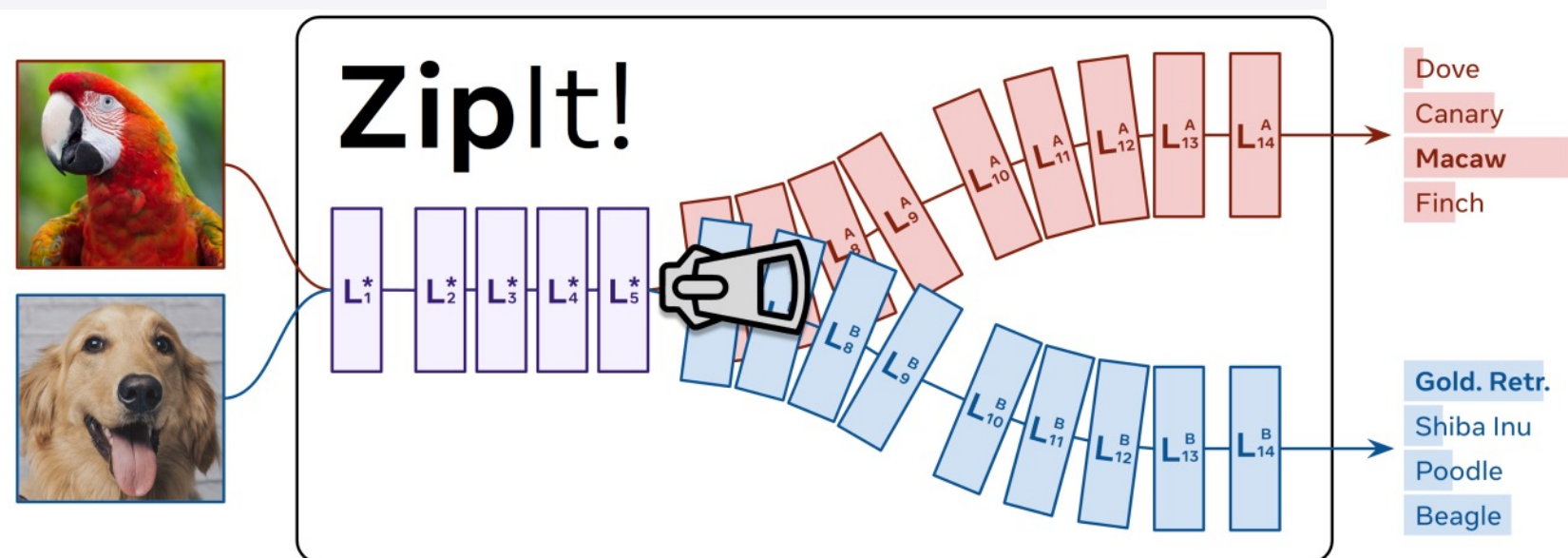
Умеем с помощью Git Re-Basin и REPAIR объединять модели, обученные на одном датасете, но с разными инициализациями.



Умеем объединять модели finetuned с разных инициализаций на одном датасете.



Не умеем объединять модели, обученные под разные задачи. Хотим делать это без дополнительного обучения.



Ziplt! Zip operation

$$f_i^* = M_i (f_i^A \parallel f_i^B)$$

$$U_i = 2M_i^T$$

$$M_{i[p,j]} = M_{i[p,k]} = 1/2$$

Merge matrix

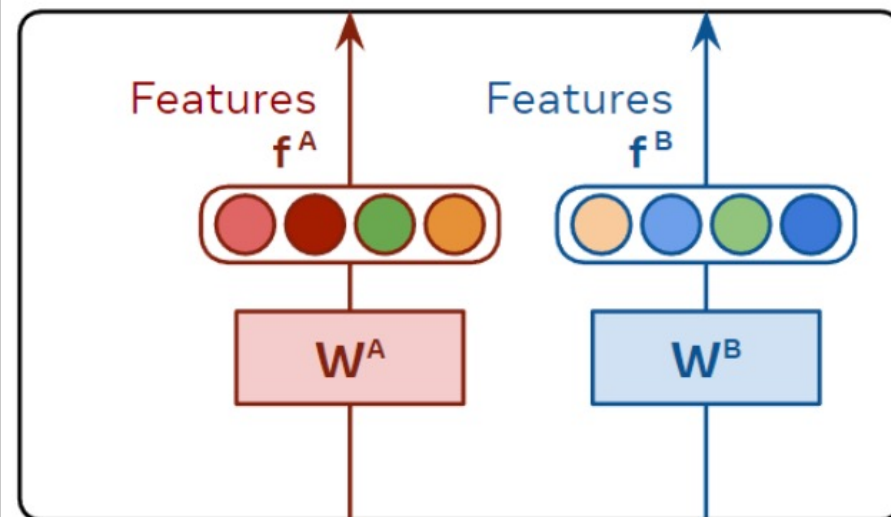
$$f_{i+1}^A = L_{i+1}^A (U_i^A f_i^*) \quad f_{i+1}^B = L_{i+1}^B (U_i^B f_i^*)$$

Unmerge matrix

$$W_i^* = M_i^A W_i^A U_{i-1}^A + M_i^B W_i^B U_{i-1}^B$$

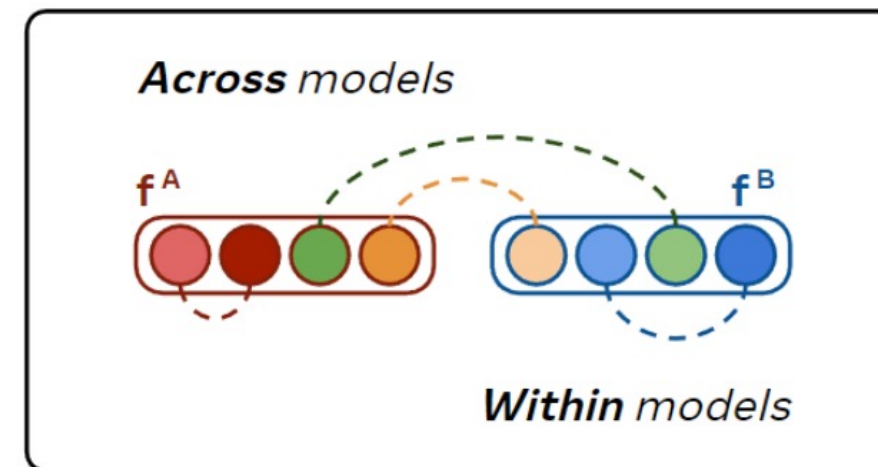
Merged weights

The Zip Operation

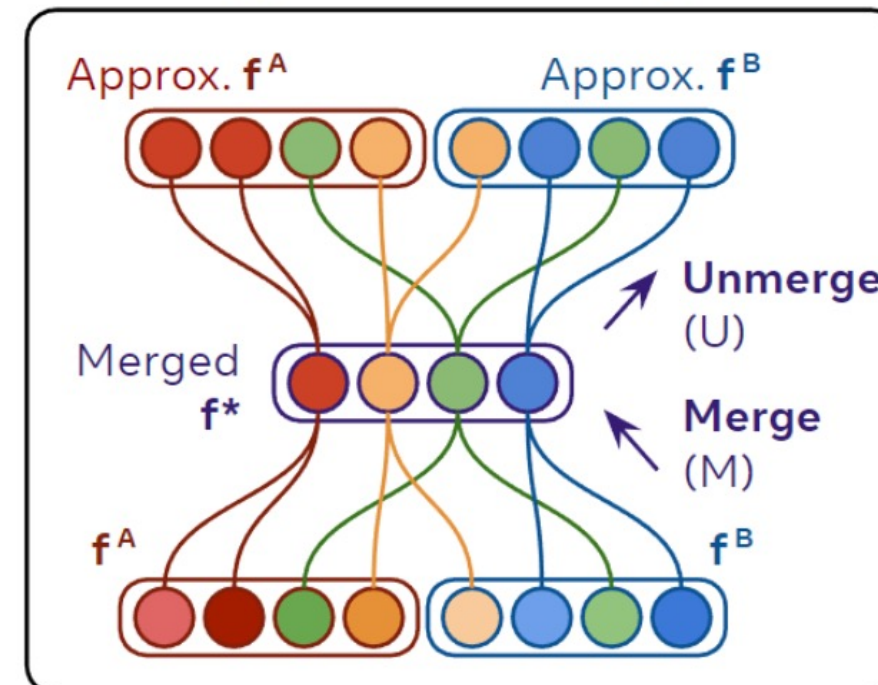


(a) Two **Disjoint** Layers

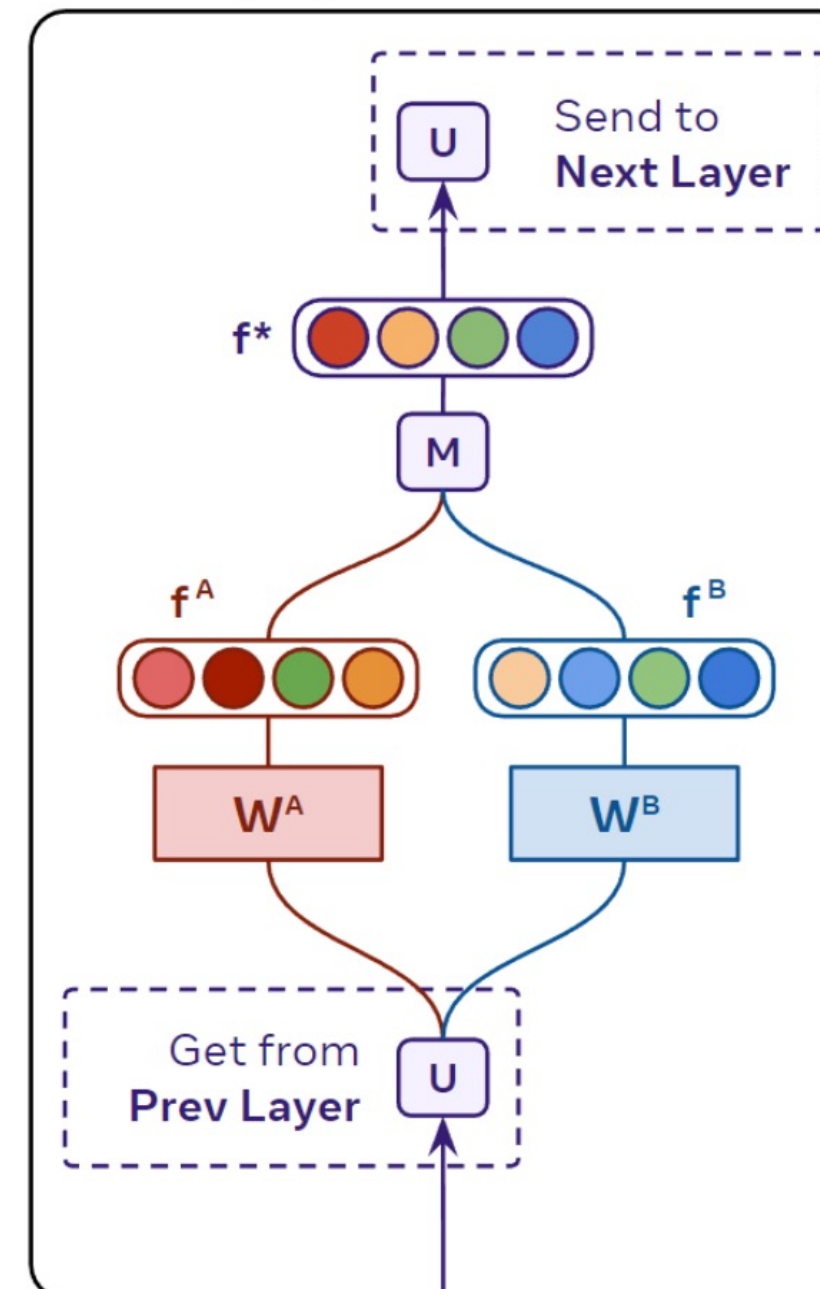
(b) Find **Redundant** Features



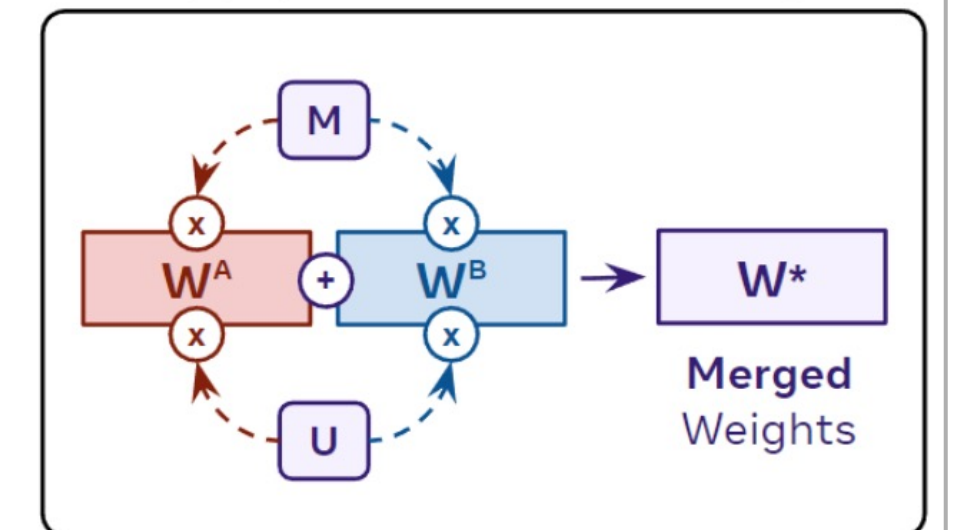
(c) **Merge** and **Unmerge**



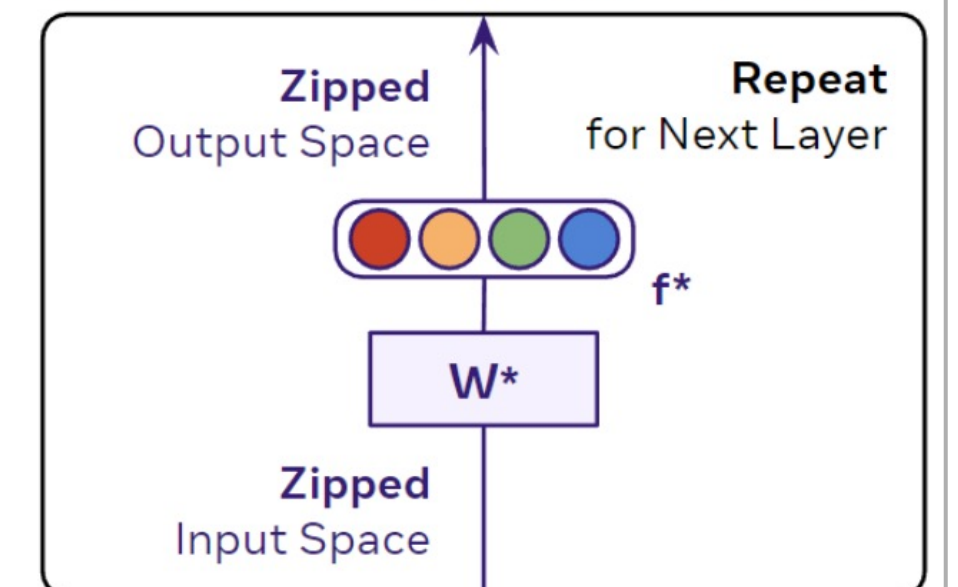
(d) **Propagate M** and **U**



(e) Apply **M** and **U** to **Weights**



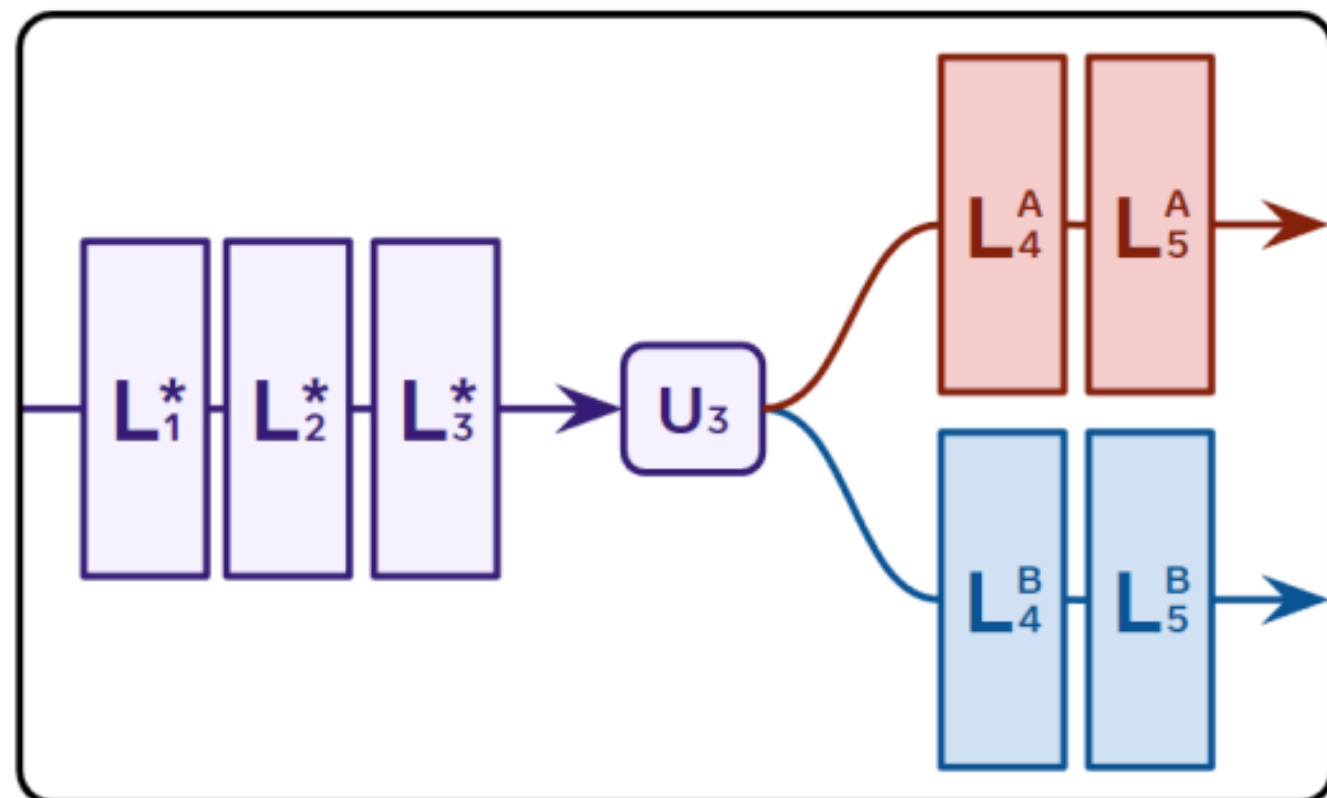
(f) **Result: One Zipped Layer**



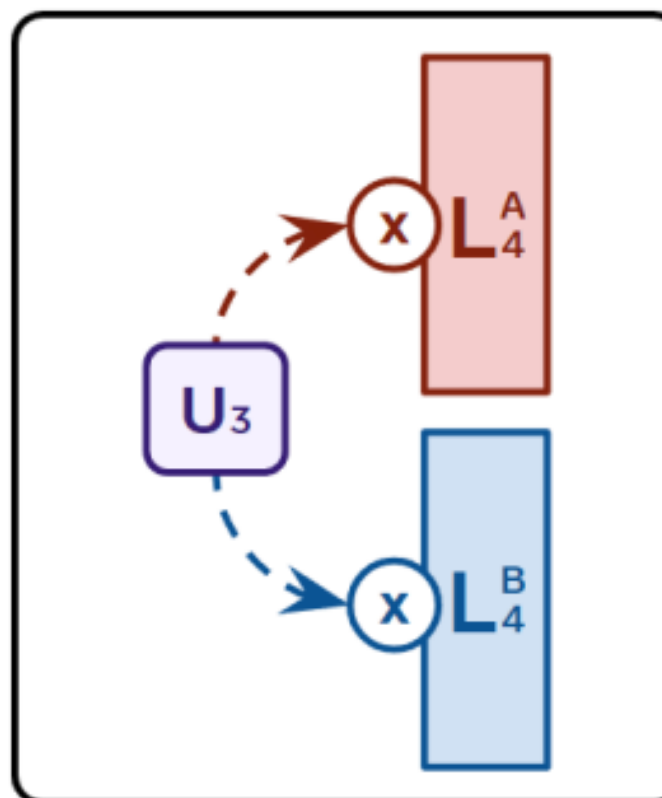
Extensions

- Операцию Zip можно применять не ко всем слоям модели, а только к первым нескольким.
- Если моделей больше 2, при merge заменяем каждую пару признаков на их комбинацию и определяем корреляцию комбинации пары признаков с другими признаками как $\alpha \cdot \min(\text{корреляция каждого признака из пары})$.
- Коэффициент β регулирует долю признаков, которые допускается комбинировать внутри одной модели.

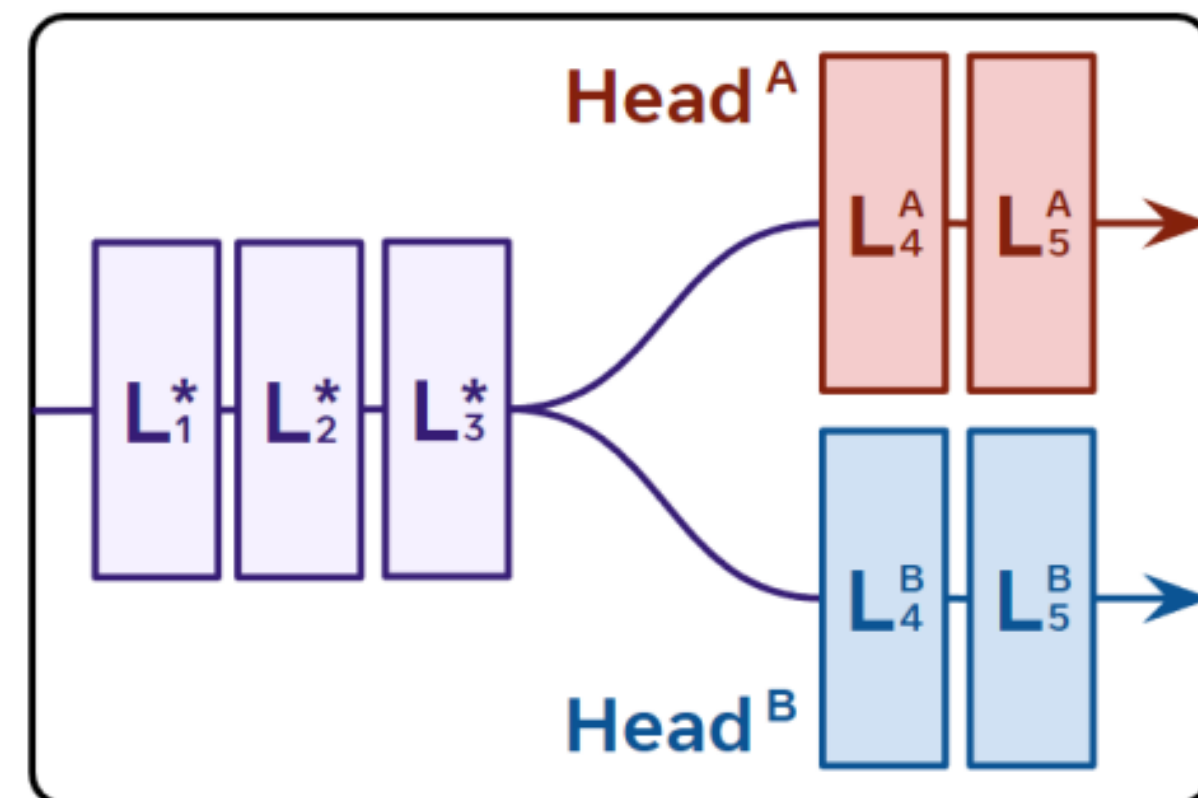
(a) **Stop zipping**



(b) **Unmerge**



(c) Result is **Multi-Head**



Results. CIFAR-10/CIFAR-100

- ResNet-20x4 для CIFAR-10 / ResNet-20x8 для CIFAR-100.
- Обучали модели в каждом датасете на двух непересекающихся наборах из 5 (50 для CIFAR-100). После чего делали Zip.

| Method | FLOPs (G) | Joint Acc (%) | Per-Task (%) | | |
|-------------------------------|--------------|--------------------------------|--------------------------------|--------------------------------|--------------------------------|
| | | | Task A | Task B | Avg |
| Model A | 0.68 | 48.2 \pm 1.0 | 97.0 \pm 0.6 | 45.1 \pm 8.6 | 71.0 \pm 4.4 |
| Model B | 0.68 | 48.4 \pm 3.8 | 49.1 \pm 9.3 | 96.1 \pm 1.1 | 72.6 \pm 4.9 |
| W. Avg (Eq. 1) | 0.68 | 43.0 \pm 1.6 | 54.1 \pm 1.4 | 67.5 \pm 1.2 | 60.8 \pm 4.5 |
| Git Re-Basin [2] | 0.68 | 46.2 \pm 0.8 | 76.8 \pm 8.9 | 82.7 \pm 5.1 | 79.8 \pm 6.5 |
| Permute (Eq. 2) | 0.68 | 58.4 \pm 6.8 | 86.6 \pm 2.1 | 87.4 \pm 1.1 | 87.4 \pm 1.4 |
| ZipIt!_{20/20} | 0.68 | 79.1\pm1.1 | 92.9\pm1.1 | 91.2\pm1.4 | 92.1\pm1.0 |
| Ensemble | 1.37 | 87.4 \pm 2.6 | 97.0 \pm 0.6 | 96.1 \pm 1.1 | 96.6 \pm 0.4 |
| ZipIt!_{13/20} | 0.91 | 83.8\pm3.1 | 95.1\pm0.7 | 94.1\pm1.5 | 94.6\pm0.6 |

(a) **CIFAR-10 (5+5)**. Using ResNet-20 (4 \times width).

| Method | FLOPs (G) | Joint Acc (%) | Per-Task (%) | | |
|-------------------------------|--------------|--------------------------------|--------------------------------|--------------------------------|--------------------------------|
| | | | Task A | Task B | Avg |
| Model A | 2.72 | 41.6 \pm 0.3 | 82.9 \pm 0.7 | 24.8 \pm 0.4 | 53.9 \pm 0.5 |
| Model B | 2.72 | 41.6 \pm 0.2 | 25.1 \pm 1.2 | 82.8 \pm 0.2 | 54.0 \pm 0.6 |
| W. Avg (Eq. 1) | 2.72 | 17.0 \pm 1.7 | 23.8 \pm 6.9 | 24.8 \pm 5.9 | 24.3 \pm 1.9 |
| Git Re-Basin [2] | 2.72 | 40.9 \pm 0.2 | 57.3 \pm 1.5 | 56.7 \pm 0.7 | 57.0 \pm 0.8 |
| Permute (Eq. 2) | 2.72 | 42.8 \pm 0.7 | 61.6 \pm 1.4 | 60.5 \pm 0.5 | 61.0 \pm 0.8 |
| ZipIt!_{20/20} | 2.72 | 54.9\pm0.8 | 68.2\pm0.8 | 67.9\pm0.6 | 68.0\pm0.4 |
| Ensemble | 5.45 | 73.5 \pm 0.4 | 82.9 \pm 0.7 | 82.8 \pm 0.2 | 82.8 \pm 0.4 |
| ZipIt!_{13/20} | 3.63 | 70.2\pm0.4 | 80.3\pm0.8 | 80.1\pm0.7 | 80.2\pm0.6 |

(b) **CIFAR-100 (50+50)**. Using ResNet-20 (8 \times width).

Результаты (accuracy) сравнения метода ZipIt! с другими методами объединения моделей. ZipIt! 13 из 20 слоев модели показывает лучшее качество среди аналогов, немного отстает от качества метода ансамблирования.

Results. ImageNet-1k (200+200)

- Разбили датасет на 5 непересекающихся подмножеств классов (по 200 классов).
- Обучали 5 ResNet-50: по одной на каждом подмножестве.
- Для каждой пары подмножеств делали Zip соответствующих моделей и измеряли качество на 400 классах.

| Method | FLOPs (G) | Joint Acc (%) | Per-Task (%) | | |
|-------------------------|--------------|------------------|----------------|-----------------|----------------|
| | | | Task A | Task B | Avg |
| Model A | 4.11 | 37.2 \pm 2.0 | 74.3 \pm 4.0 | 0.5 \pm 0.1 | 37.4 \pm 2.0 |
| Model B | 4.11 | 35.3 \pm 1.6 | 0.5 \pm 0.1 | 70.5 \pm 3.2 | 35.5 \pm 1.6 |
| W. Avg (Eq. 1) | 4.11 | 0.3 \pm 0.1 | 0.6 \pm 0.1 | 0.7 \pm 0.1 | 0.6 \pm 0.1 |
| Git Re-Basin [2] | 4.11 | 3.1 \pm 1.2 | 5.3 \pm 2.6 | 5.7 \pm 2.4 | 5.5 \pm 1.7 |
| Permute (Eq. 2) | 4.11 | 8.6 \pm 5.8 | 10.1 \pm 4.4 | 15.3 \pm 11.1 | 12.7 \pm 7.7 |
| ZipIt! _{50/50} | 4.11 | 8.6 \pm 4.7 | 12.4 \pm 5.9 | 14.7 \pm 7.8 | 13.5 \pm 6.6 |
| Ensemble | 8.22 | 63.3 \pm 4.9 | 74.3 \pm 4.0 | 70.5 \pm 3.2 | 72.4 \pm 2.5 |
| ZipIt! _{22/50} | 6.39 | 55.8 \pm 4.1 | 65.9 \pm 2.5 | 64.1 \pm 3.0 | 65.0 \pm 2.3 |
| ZipIt! _{10/50} | 7.43 | 60.9 \pm 4.1 | 70.7 \pm 3.0 | 69.0 \pm 2.9 | 69.9 \pm 1.9 |

Результаты (accuracy) сравнения метода ZipIt! с другими методами объединения моделей. ZipIt! 10 из 50 слоев показывает лучшее качество среди аналогов, немного отстает от качества метода ансамблирования.

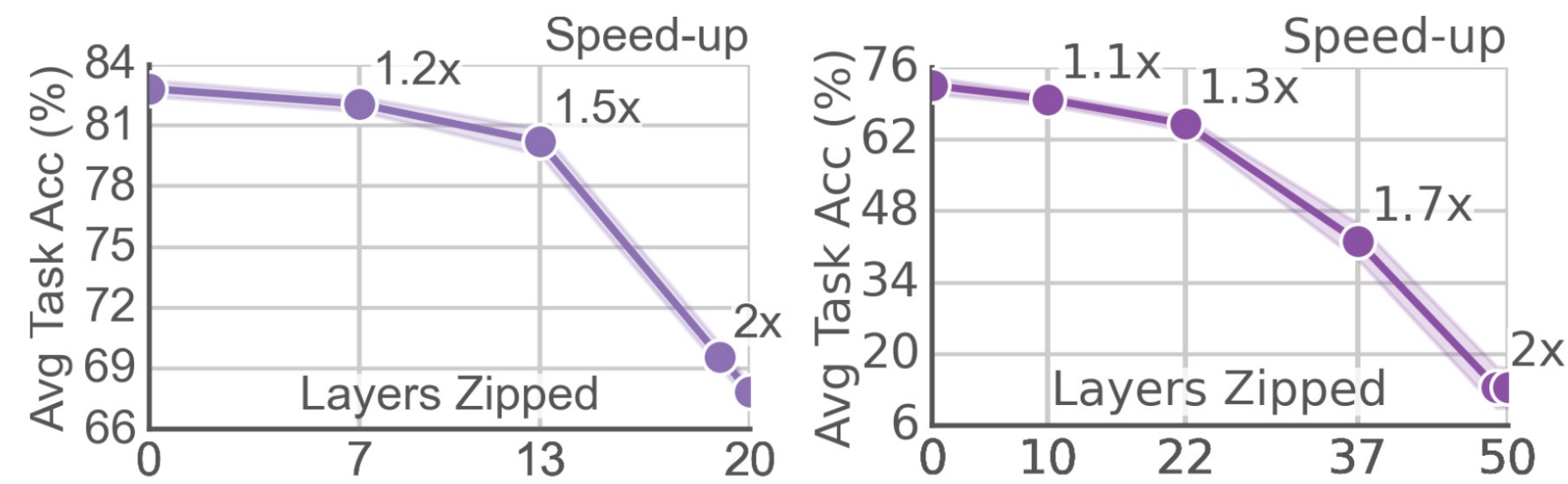
Results. Multiple datasets

- 4 датасета: Stanford Dogs (SD), Oxford Pets (OP), CUB200 (CUB), and NABirds (NAB).
- Обучали ResNet-50 на каждом.
- Делали Zip всех 4 моделей в одну и 6 попарных Zip операций.

| Method | FLOPs | Per-Task (%) | | | | |
|--------------------------------|-------|--------------|-------------|-------------|-------------|-------------|
| | (G) | SD | OP | CUB | NAB | Avg |
| Merging Pairs | | | | | | |
| W. Avg (Eq. 1) | 4.11 | 15.1 | 23.8 | 11.8 | 2.1 | 13.2 |
| Permute (Eq. 2) | 4.11 | 51.3 | 64.7 | 36.7 | 15.5 | 42.1 |
| ZipIt! _{49/50} | 4.11 | 51.2 | 67.7 | 40.6 | 15.6 | 43.8 |
| Ensemble | 8.22 | 72.7 | 83.2 | 71.0 | 77.2 | 76.0 |
| ZipIt! _{37/50} | 4.92 | 56.8 | 73.8 | 54.6 | 37.9 | 55.8 |
| ZipIt! _{22/50} | 6.39 | 65.3 | 79.7 | 64.8 | 61.2 | 67.7 |
| Merging All 4 | | | | | | |
| W. Avg (Eq. 1) | 4.12 | 0.7 | 3.4 | 0.4 | 0.2 | 1.2 |
| Permute (Eq. 2) | 4.12 | 34.2 | 55.4 | 13.4 | 5.7 | 27.2 |
| ZipIt! _{49/50} | 4.12 | 32.1 | 55.3 | 14.7 | 6.9 | 27.3 |
| Ensemble | 16.44 | 72.7 | 83.2 | 71.0 | 77.2 | 76.0 |
| ZipIt! _{37/50} | 6.5 | 39.9 | 66.4 | 44.3 | 24.6 | 43.8 |
| ZipIt! _{22/50} | 11.0 | 58.2 | 78.5 | 58.6 | 55.1 | 62.6 |

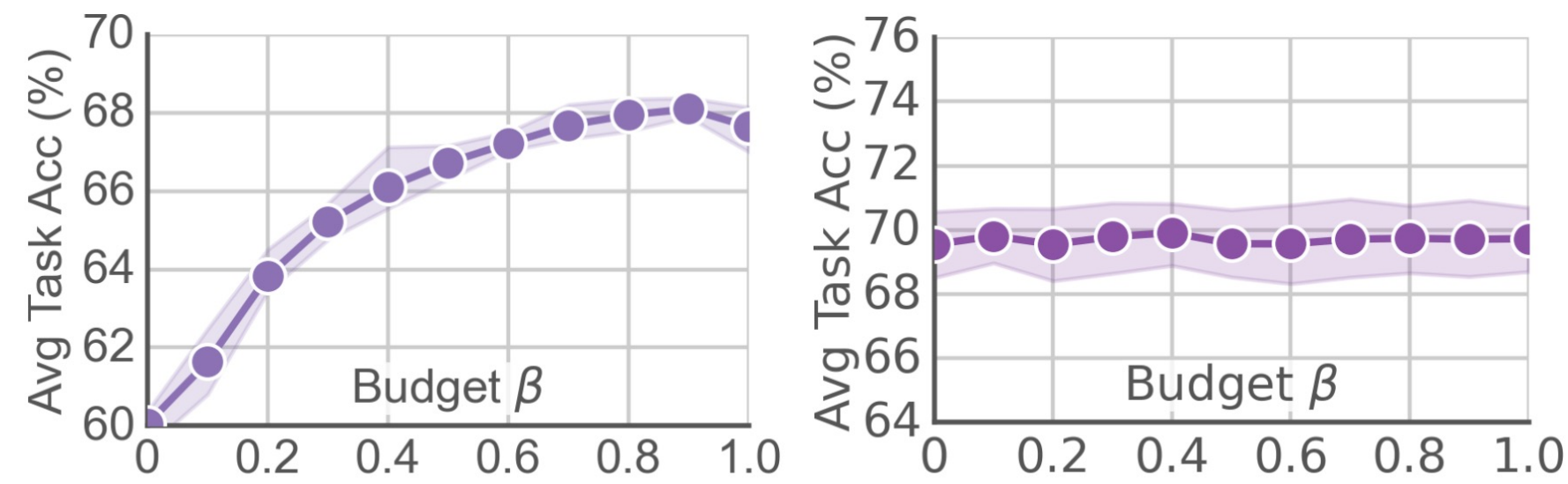
Результаты (ассурасу) сравнения метода ZipIt! с другими методами объединения моделей. ZipIt! всех моделей в одну показывает результат хуже, чем попарный ZipIt!, но все еще лучше остальных методов (кроме ансамблирования).

Hyperparameter analysis



(a) **CIFAR-100 50+50.** (b) **ImageNet-1k 200+200.**

Зависимость качества работы моделей от количества слоев, к которым применили Zip операцию.

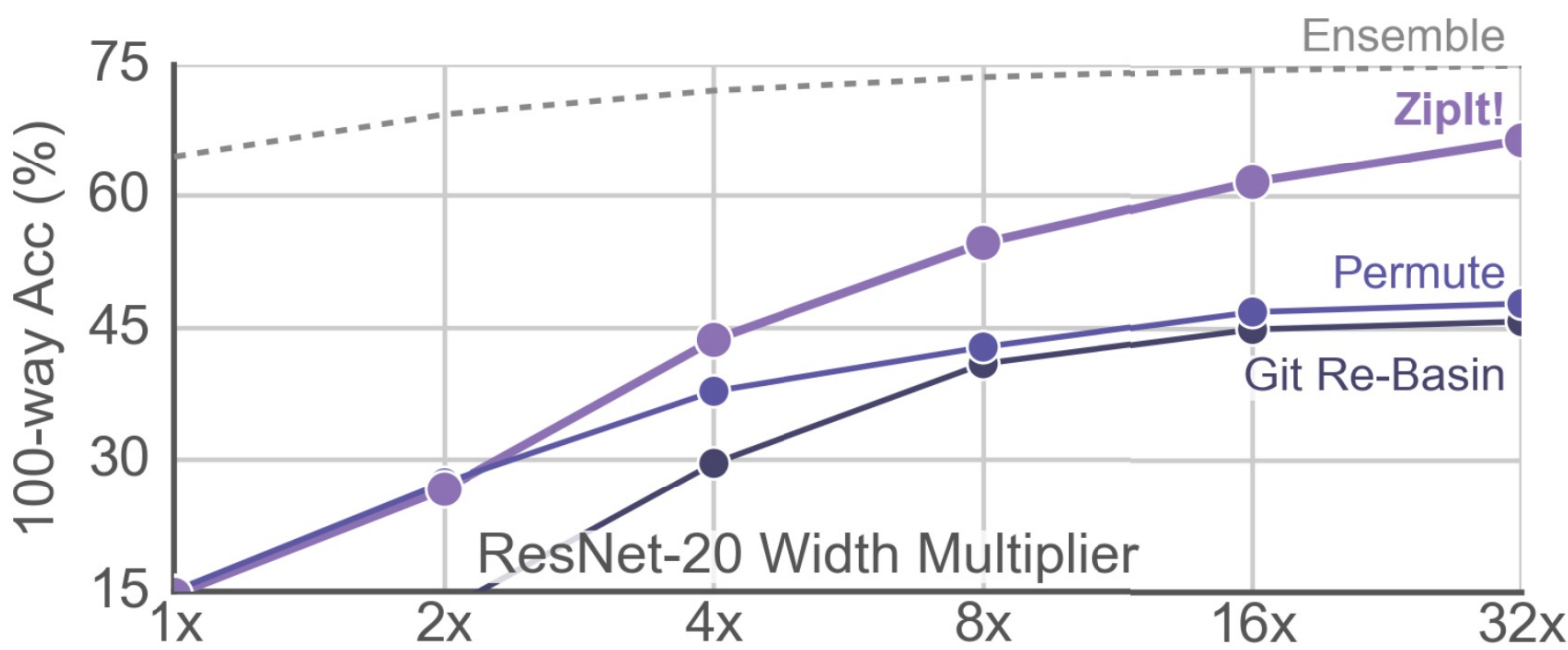


(a) **CIFAR-100 50+50.** (b) **ImageNet-1k 200+200.**

Зависимость качества работы моделей от значения гиперпараметра β .

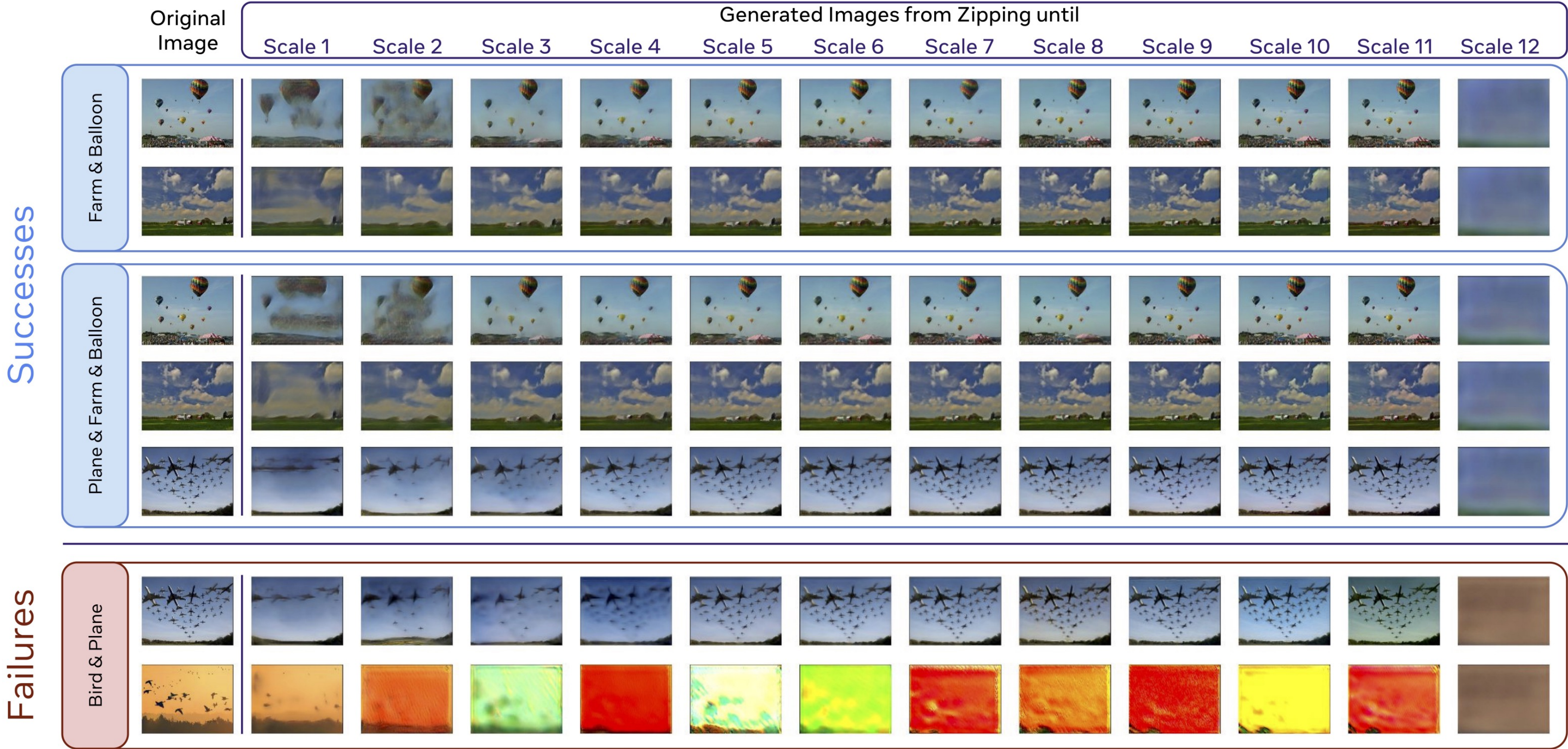
| Algorithm | $A \leftrightarrow A/B \leftrightarrow B?$ | Acc | Time |
|----------------------|--|----------------------------------|---------|
| Identity (Eq. 1) | \times | 43.0 ± 3.1 | 1.8 ms |
| Permute (Eq. 2) | \times | 58.4 ± 1.3 | 28 ms |
| K-Means | \checkmark | 29.1 ± 5.5 | 19 sec |
| Zip (Eq. 7) | | | |
| Optimal Match | \checkmark | 79.6 ± 1.7 | 11 min |
| Greedy Match | \checkmark | 79.0 ± 1.8 | 1.1 sec |
| Greedy, $\alpha=0.1$ | \checkmark | 79.1 ± 2.1 | 1.2 sec |

Сравнение качества и времени работы моделей, сжатых с использованием разных алгоритмов сжатия признаков.



Зависимость качества работы моделей, сжатых с помощью разных алгоритмов, от размеров этих моделей.

Ziplt! for SinGANs



Результат работы модели SinGAN, сжатых до определенного слоя (верхняя строка), обученных на картинках (левый столбец).

ИТОГИ

- С помощью Ziplt! научились объединять модели одной архитектуры, обученные под разные задачи.
- Объединять модели с помощью Ziplt! можно как полностью, так и частично, оставляя отдельные головы.
- Ziplt! не требует дополнительного обучения.
- По результатам экспериментов, описанных в статье, Ziplt! показывает более высокое качество, чем аналогичные методы (Git Re-Basin, Permute).
- Ziplt! может быть использован с любыми типами задач, однако большинство экспериментов в статье проводилось на задачах классификации изображений.

Stoica G. et al. Ziplt! Merging Models from Different Tasks without Training
//arXiv preprint arXiv:2305.03053. – 2023.