Matryoshka Representation Learning

Aditya Kusupati*†◇, Gantavya Bhatt*†, Aniket Rege*†,
Matthew Wallingford†, Aditya Sinha◇, Vivek Ramanujan†, William Howard-Snyder†,
Kaifeng Chen◇, Sham Kakade‡, Prateek Jain◇ and Ali Farhadi†
†University of Washington, ◇Google Research, ‡Harvard University
{kusupati,ali}@cs.washington.edu, prajain@google.com

NeurIPS 2022

HUMANE Lab 김태균

2025.04.25

Background

- Limitations of existing fixed dimensional embeddings
  - web-scale bottleneck
  - lack of flexibility
- Previous approaches to add flexibility
  - training multiple models
  - optimizing sub-networks
  - post-hoc compression

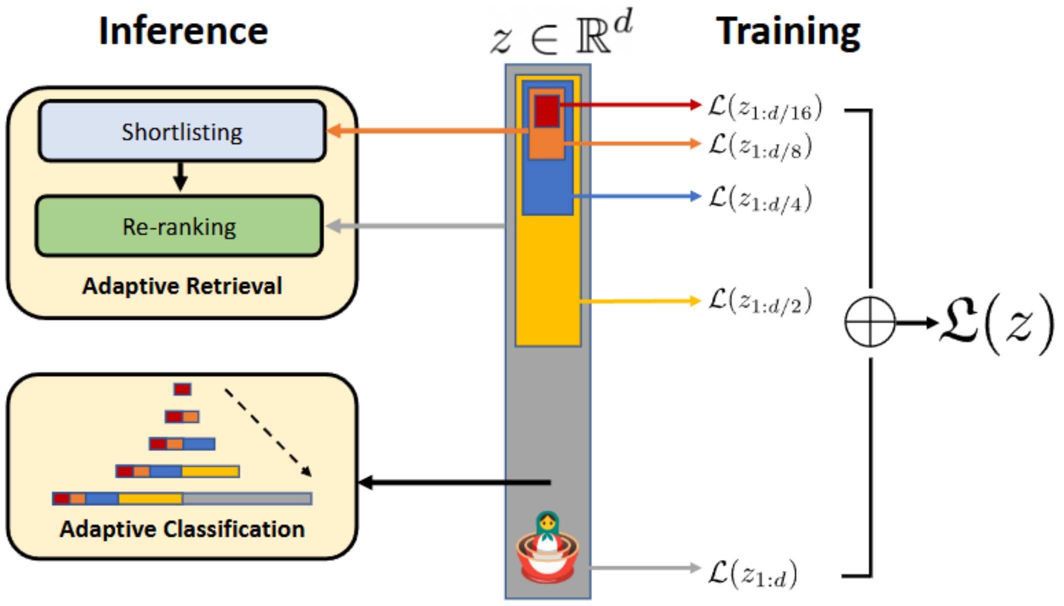
Background

- These approaches fall short for adaptive large-scale deployment
  - high training/maintenance overhead
  - significant accuracy drop

=> Can we design a flexible representation that can adapt to multiple downstream tasks with varying computational resources?

Matryoshka Representation Learning

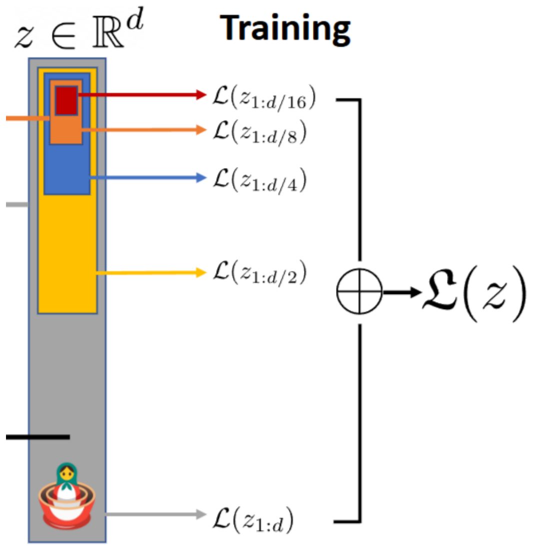
- Learning a single high-dimensional embedding vector such that any prefix of it can serve as a semantically meaningful and effective lower-dimensional embedding on its own



Training Process in MRL

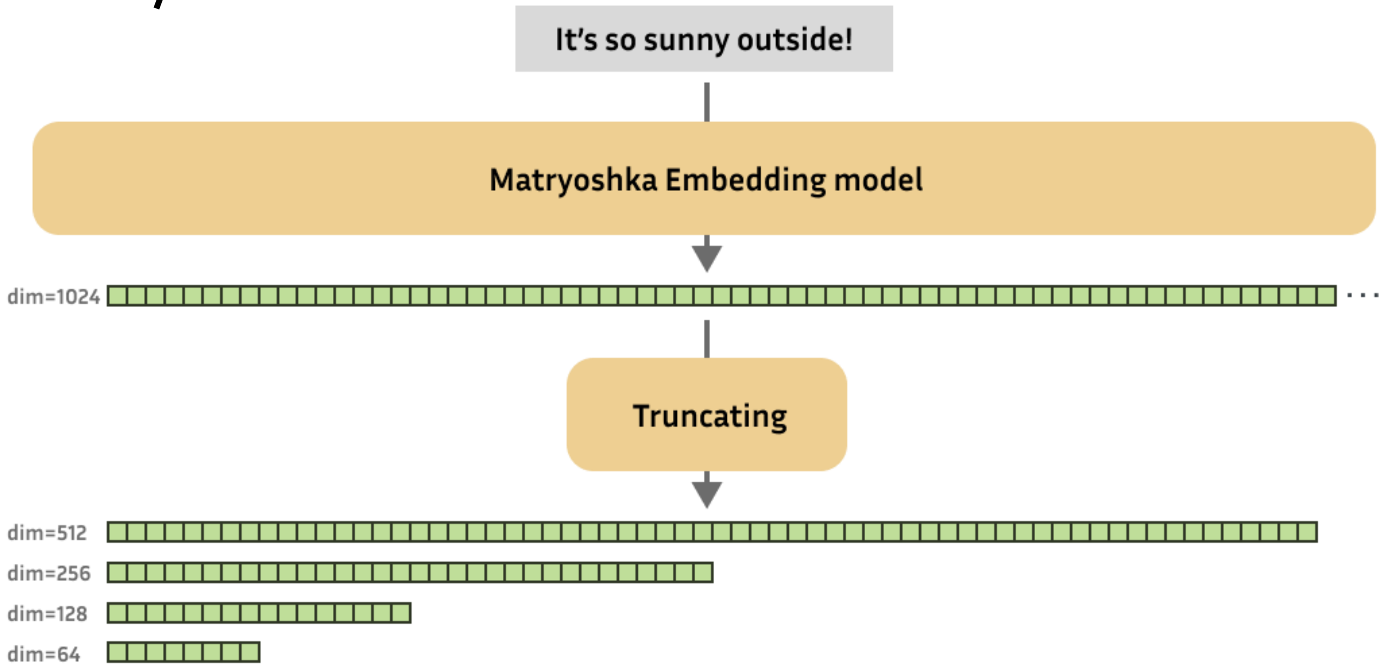
- Extract the first m dimensions $z_{1:m}$ from the full vector $z \in \mathbb{R}^d$
  - e.g. for each $\mathrm{m} \in M = \{8, 16, \dots, 2048\}$ where $d = 2048$
- Use a corresponding classifier $W^{(m)}$ to make predictions from $z_{1:m}$, and compute the loss against the ground truth label y
- Sum all losses across different m and optimize the total loss

$$\min_{\{\mathbf{W}^{(m)}\}_{m\in\mathcal{M}},\ \theta_F} \frac{1}{N} \sum_{i\in[N]} \sum_{m\in\mathcal{M}} c_m \cdot \mathcal{L}\left(\overset{\mathbb{R}^{L\times m}}{\mathbf{W}^{(m)}} \cdot \underbrace{\overset{\mathbb{R}^{m}}{F(x_i;\theta_F)_{1:m}}}_{:= z_{1:m}} ;\ \overset{\mathbb{R}^{L}}{y_i}\right)$$



Training Process in MRL

- After training, F generates a d-dimensional vector z
- For any $\mathrm{m} \in M$, using only the first m dimensions $z_{1:m}$ is sufficient to perform the task effectively

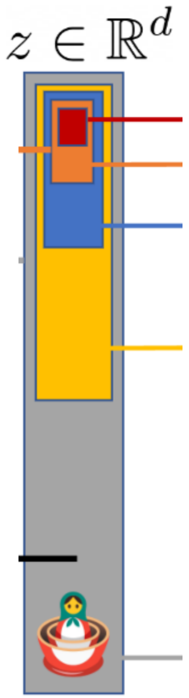


MRL-E

- Using a separate classifier $W^{(m)}$ for each m increases the number of parameters
- To reduce this, MRL-E adopts weight-tying
  - using a single large classifier W, and setting $W^{(m)} = W_{1:m}$
  - i.e. the first m columns of W

Matryoshka

- Set of wooden dolls of decreasing size placed one inside another



Applications

- Downstream applications of MRL for flexible large-scale deployment
  - Adaptive Classification (AC)
  - Adaptive Retrieval (AR)

Representation Learning Setups

- Supervised learning for vision
  - ResNet50 on ImageNet-1K
  - ViT-B/16 on JFT-300M
- Contrastive learning for vision + language
  - ALIGN on ALIGN data
- Masked language modelling
  - BERT on Wikipedia and BooksCorpus

Representation Learning Setups

- Baselines
    - FF (Fixed Feature)
    - SVD
    - Slimmable networks
    - Randomly selected features

Classification

- At least as accurate as each FF model

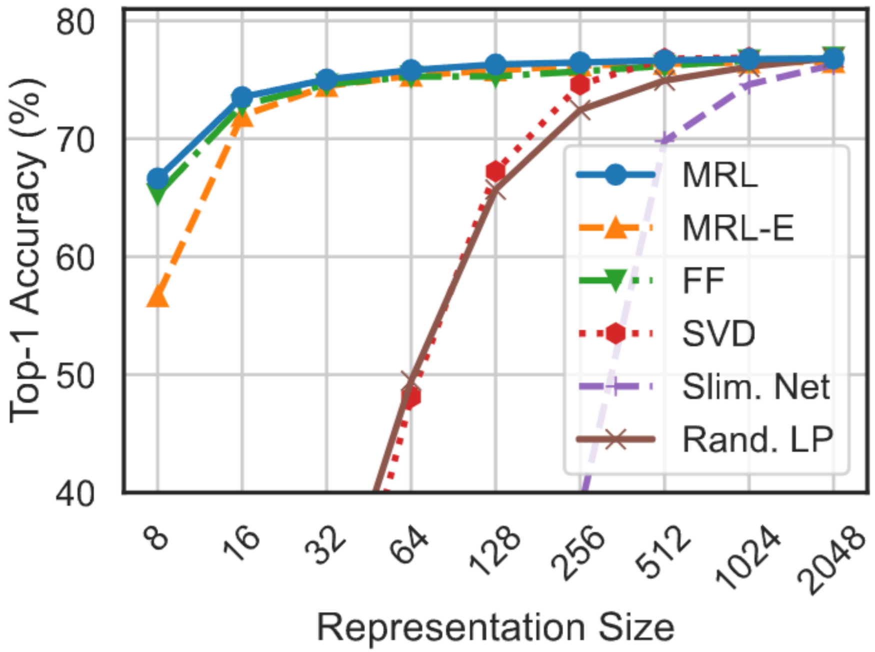


Figure 2: ImageNet-1K linear classification accuracy of ResNet50 models. MRL is as accurate as the independently trained FF models for every representation size.

Classification

- High accuracy on large-scale models and datasets (scalability)
- Intermediate dimensional representations also perform well (flexibility)

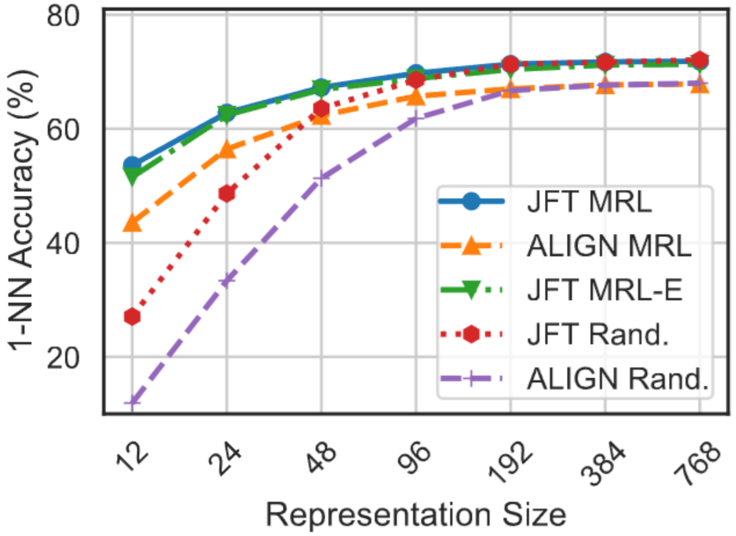


Figure 4: ImageNet-1K 1-NN accuracy for ViT-B/16 models trained on JFT-300M & as part of ALIGN. MRL scales seamlessly to web-scale with minimal training overhead.

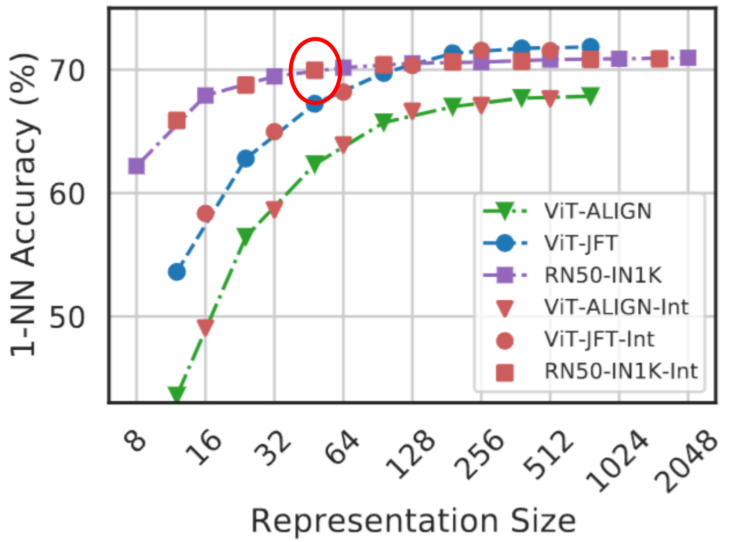


Figure 5: Despite optimizing MRL only for $O(\log(d))$ dimensions for ResNet50 and ViT-B/16 models; the accuracy in the intermediate dimensions shows interpolating behaviour.

Retrieval

- At least as accurate as each FF model

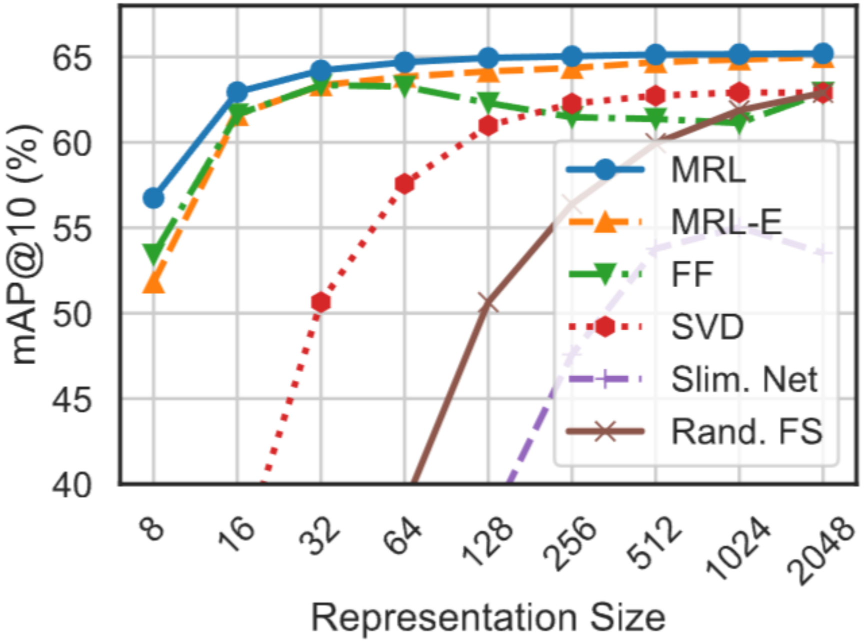


Figure 7: mAP@10 for Image Retrieval on ImageNet-1K with ResNet50. MRL consistently produces better retrieval performance over the baselines across all the representation sizes.

MLM Accuracy

- Although it shows slightly lower accuracy than independently trained FF models at each dimension, the difference is minimal

Rep. Size	BERT-FF	BERT-MRL
12	60.12	59.92
24	62.49	62.05
48	63.85	63.40
96	64.32	64.15
192	64.70	64.58
384	65.03	64.81
768	65.54	65.00

MRL in Practice

- Recent embedding models based on MRL
  - OpenAI's text-embedding-3 series
  - Alibaba's gte-multilingual-base
  - ...

Conclusions

- MRL provides flexible representations for adaptive deployment
  - enabling efficient and scalable performance across varying resource constraints

Open Questions

- What about using the postfix ($z_{-m:}$) in MRL?