# Decoupled Training: Return of Frustratingly Easy Multi-Domain Learning

Ximei Wang[1], **Junwei Pan**[1], Xingzhuo Guo[2], Dapeng Liu[1], Jie Jiang[1]

[1] Tencent Inc, China [2]School of Software, Tsinghua University, China

## Summary

We explicitly uncover the main challenges of dataset bias and domain domination in multi-domain Learning (MDL).

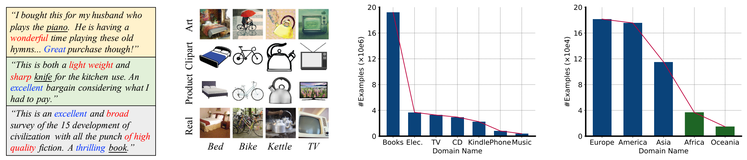
We propose a frustratingly easy and hyperparameter-free general-to-specific Decoupled Train (D-Train) method.

D-Train highlights the domain-independent fine-tuning to alleviate the obstacles of *seesaw effect* across domains.

## Challenges of Multi-Domain Learning (MDL)

**Dataset Bias**: different domains in MDL have various appearances, backgrounds, styles, and keywords

**Domain Domination**: the distributions of sample amount across domains are usually imbalanced, or even long-tailed.

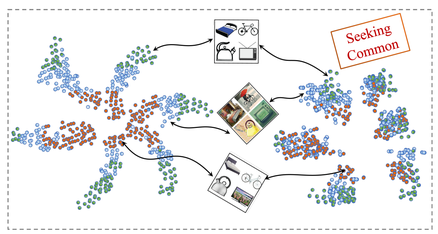



(a) Review Examples (b) Visual Examples (c) #Samples in Amazon (d) #Samples in FMoW

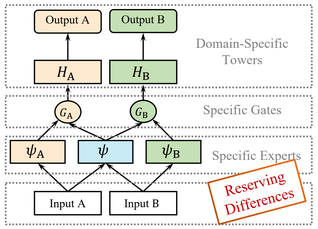
## Existing Approaches to MDL

**Seeking Commonalities** by aligning distributions across domains to reduce domain gap

**Reserving Differences** by implementing domain-specific towers, gates, and even experts




(e) Seeking Commonalities




(f) Reserving Differences

**Limitations**: existing MDL approaches are becoming sophisticated, consisting of multifarious network architectures or complex loss functions

## Method: D-Train

**Pre-train**: Warm Up a Root Model

$$\psi_0, h_0 = \arg\min_{\psi,h} \frac{1}{T}\sum_{t=1}^{T}\frac{1}{n_t}\sum_{i=1}^{n_t}\mathcal{L}\left[(h\circ\psi)(\mathbf{x}_i^t), \mathbf{y}_i^t\right], \quad (1)$$

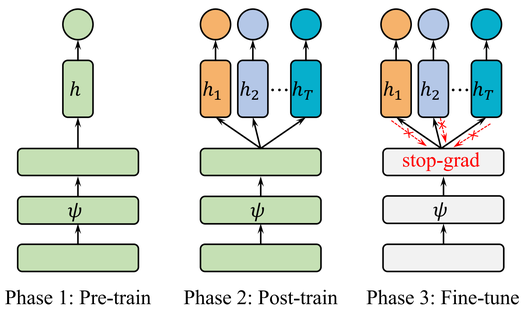
**Post-Train**: Split Into Multi-Heads

$$\widetilde{\psi}, \{\widetilde{h}_1, \widetilde{h}_2, ..., \widetilde{h}_T\} = \arg\min_{\psi,\{h_1,h_2,...,h_T\}} \frac{1}{T}\sum_{t=1}^{T} L_t \quad (2)$$

$$L_t = \frac{1}{n_t}\sum_{i=1}^{n_t}\mathcal{L}\left[(h_t\circ\psi)(\mathbf{x}_i^t), \mathbf{y}_i^t\right] \quad (3)$$

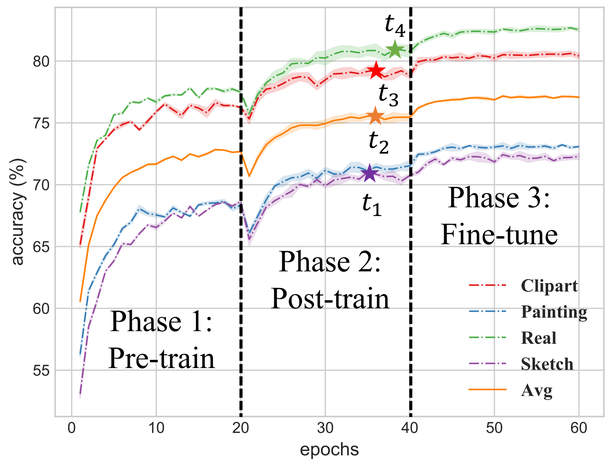
**Fine-tune**: Decouple-Train for Independence

$$\widehat{h}_t = \arg\min_{h_t} \frac{1}{n_t}\sum_{i=1}^{n_t}\mathcal{L}\left[(h_t\circ\widetilde{\psi})(\mathbf{x}_i^t), \mathbf{y}_i^t\right]. \quad (4)$$



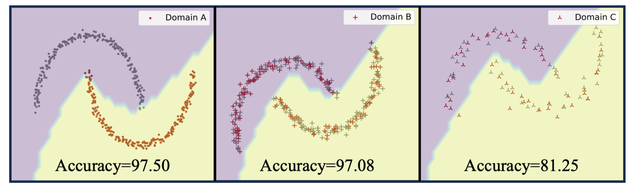

## Why Does D-Train Work?

Different domains in D-Train will train independently at the fine-tuning phase, alleviating the obstacles of *seesaw* effect.

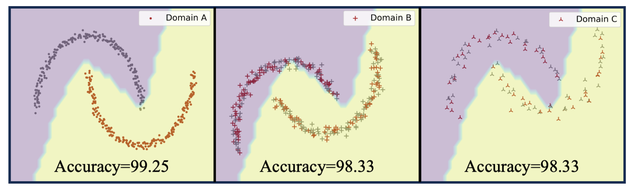



## Why Does D-Train Work?

D-Train can further adjust conflict areas of decision boundary by domain-independent fine-tuning.




(h) Shared Bottom




(i) D-Train

## Experiment Results

| Method | Clipart | Painting | Real | Sketch | Avg. Acc. | Worst Acc. |
|---|---|---|---|---|---|---|
| #Samples | 49k | 76k | 175k | 70k | - | - |
| Separatly Train | 78.2 | 71.6 | **83.8** | 70.6 | 76.1 | 70.6 |
| Jointly Train | 77.4 | 68.0 | 77.9 | 68.5 | 73.0 | 68.0 |
| MulANN (Alice et al. 2019) | 79.5 | 71.7 | 81.7 | 69.9 | 75.7 | 69.9 |
| DANN-MDL (Ganin et al. 2016) | 79.8 | 71.4 | 81.4 | 70.3 | 75.7 | 70.3 |
| ASP-MDL (Liu, Qiu, and Huang 2017) | 80.1 | 72.1 | 81.2 | 70.9 | 76.1 | 70.9 |
| CDAN-MDL (Long et al. 2018) | 80.2 | 72.2 | 81.3 | 71.0 | 76.2 | 71.0 |
| Shared Bottom (Ruder 2017) | 79.9 | 72.1 | 81.9 | 69.7 | 75.9 | 69.7 |
| MoE (Jacobs et al. 1991) | 79.1 | 70.2 | 79.8 | 69.4 | 74.6 | 69.4 |
| MMoE (Ma et al. 2018) | 79.6 | 72.2 | 82.0 | 69.8 | 75.9 | 69.8 |
| PLE (Tang et al. 2020) | 80.0 | 72.2 | 82.1 | 70.0 | 76.1 | 70.0 |
| D-Train (w/o Fine-tune) | 79.9 | 71.3 | 81.3 | 70.9 | 75.9 | 70.9 |
| D-Train (w/o Post-train) | 79.8 | 72.9 | 81.7 | 72.1 | 76.6 | 72.1 |
| D-Train (w/o Pre-train) | 80.9 | 72.9 | 82.7 | 71.5 | 77.0 | 71.5 |
| **D-Train (ours)** | **81.5** | **72.8** | 82.7 | **72.2** | **77.3** | **72.2** |

Table 2: Accuracy (%) on DomainNet for multi-domain learning (ResNet-101).

## D-Train as a Plug-in Unit

| Method | Books | Elec. | TV | CD | Kindle | Phone | Music | $AUC_d$ | $AUC_s$ |
|---|---|---|---|---|---|---|---|---|---|
| #Samples | 19.2M | 3.70M | 3.28M | 2.96M | 2.25M | 0.81M | 0.38M | – | – |
| Separately Training | 66.09 | 77.50 | 79.43 | 59.69 | 52.79 | 70.06 | 52.95 | 65.50 | 67.17 |
| Jointly Training | 69.01 | 78.87 | 85.06 | 64.24 | 59.15 | 69.89 | 49.71 | 67.99 | 70.43 |
| MulANN (Alice et al. 2019) | 68.95 | 78.90 | 84.56 | 64.79 | 58.64 | 70.43 | 52.13 | 68.34 | 70.40 |
| DANN-MDL (Ganin et al. 2016) | 68.64 | 80.33 | 86.08 | 66.32 | 58.59 | 72.47 | 54.21 | 69.52 | 70.74 |
| CDAN-MDL (Long et al. 2018) | 69.74 | 80.63 | 85.88 | 67.24 | 60.61 | 73.34 | 57.39 | 70.69 | 71.69 |
| MoE (Jacobs et al. 1991) | 73.51 | 85.88 | 89.66 | 74.94 | 63.45 | 79.63 | 66.08 | 76.16 | 76.04 |
| Shared Bottom (Ruder 2017) | 70.91 | 74.87 | 85.51 | 67.18 | 60.56 | 74.59 | 59.14 | 70.39 | 71.73 |
| Shared Bottom + D-Train | 71.35 | 74.76 | 85.52 | 67.61 | 60.20 | 73.53 | 61.61 | 70.65 | 71.99 |
| MMoE (Ma et al. 2018) | 73.67 | 86.15 | 89.23 | 75.50 | 62.43 | 81.91 | 63.69 | 76.01 | 76.13 |
| MMoE + D-Train | 74.50 | 86.09 | 88.60 | 77.13 | 66.46 | 82.13 | 69.01 | 77.70 | 77.05 |
| PLE (Tang et al. 2020) | 75.25 | 85.36 | 88.54 | 76.09 | 69.35 | 81.02 | 67.75 | 77.62 | 77.46 |
| PLE + D-Train | 74.70 | 86.70 | 89.53 | 77.40 | 69.26 | 82.47 | 69.91 | 78.57 | 77.56 |

Table 4: AUC (%) on Amazon Product Review for multi-domain learning.

Email: jonaspan@tencent.com

The 38th Annual AAAI Conference on Artificial Intelligence (AAAI 2024)

Homepage: https://wxm17.github.io/