Domain Adaptions on 3D Protein Structures with AlphaFold2

Dev Paresh Chaudhari
Dept. of Software Engineering
Western University
London, Ontario
dchaud26@uwo.ca

Zain Syed
Dept. of Software Engineering
Western University
London, Ontario
zsyed44@uwo.ca

Tejanvesh Gangavarapu
Dept. of Software Engineering
Western University
London, Ontario
tgangava@uwo.ca

Kwaku Asare
Dept. of Software Engineering
Western University
London, Ontario
kasare2@uwo.ca

Obaid Mohiuddin
Dept. of Software Engineering
Western University
London, Ontario
omohiudd@uwo.ca

***Abstract* — The disparity between the expansive AlphaFold Database and the limited Protein Data Bank creates a domain shift that disrupts standard classifiers when applied to predicted structures. To address this, we evaluated domain adaptation methods to classify unlabelled predicted folds using labelled experimental data. We constructed a dataset of 12,052 paired contact maps to benchmark Naive, Gradient Reversal, ADDA, and WDGRL models. Results indicate that domain adaptation improves performance by approximately 15% over the Naive baseline (72% accuracy); ADDA achieved the highest accuracy at 88%, while WDGRL demonstrated the most stable class-wise alignment in t-SNE visualisations. We conclude that adversarial alignment effectively bridges the gap between experimental and predicted distributions, with future work targeting multi-domain proteins and structure-aware transformers.**

Keywords—domain Adaptation, protein structure classification, AlphaFold, transfer learning, deep learning, contact maps, structural biology

I. Introduction

Protein structures are incredibly complex, more complex than the human mind can even begin to understand [1]. Throughout the years, humans have become increasingly better and identifying these protein structures, but it has always been through experimental measures. Proteins are unlike other molecules, as they have extraordinary structural complexity, and a diverse range of biological functions [13].

For decades, scientists have used methods such as X-ray crystallography and Nuclear magnetic resonance (NMR), which was incredibly time consuming, and defined limits upon how fast humans could discover unique proteins [12],[1]. Despite over 100 years of technological advancements, the process of determining protein structures still requires a large time investment, and troubleshooting, which effectively limits the amount of known protein structures in the observable universe [4].

Proteins are defined by their three-dimensional structures, as the shape of a protein heavily influences its biological function [26],[27]. Two proteins with similar structures generally perform similar functions, which makes techniques for structural classification such as NMR increasingly important. Although these classical methods are accurate, they require a lot of time, meaning the number of known protein sequences has outpaced the growth of experimentally resolved ones.

II. Background & Related Work

Two widely used protein structure classification systems are **SCOP** and **CATH** [2]. SCOP is a manually curated hierarchy from the 1990s that groups domains into class, fold, superfamily, and family to identify structural similarity and evolutionary relationships [2]. CATH, developed later, scales classification by separating overall geometry from topology, offering a more detailed view of structural similarity [3]. Both remain as standard reference databases for structural comparison.

The Protein Data Bank (PDB), an authoritative database of all experimentally measured macromolecular structures, currently containing approximately two hundred thousand entries [4]. The PDF reflects the limitations we have on experimentally resolved structures, as the PDB only represents a small fraction of known protein diversity [12]. The PDB's limited size has restricted Machine Learning research within structural biology.

The reality is that advancements in protein identification have become incredibly necessary over the last few years. Google's DeepMind released a tool called the *AlphaFold* in 2018, which revolutionized the way we look at protein structures, predicting full protein structures from amino acid structures, producing more efficient results [6]. The AlphaFold Protein Structure Database (AFDB) contains over two hundred million predicted structures, becoming the leading source of structural information, with prediction-based structures [10]. leveraging the power of deep neural networks to

In its current state, proteomics is faster than it has ever been, however it raises a fundamental issue. Models trained on experimental data will often fail when directly applied to AlphaFold predictions, even though they are describing the same biological object [7], [9]. This is because experimental structures have different data distributions, which cannot compare to the smooth models created by AlphaFold. The two approaches differ in secondary structure boundaries, torsion angle distributions, side-chain packing, and many other factors, which cause a ***domain shift***, which can cause standard transfer learning models to fail, because models trained on the PDB learn features that reflect experimental artifacts as opposed to realistic fold-defining geometry [14], [15]. This problem creates a large research gap.

The research gap will only increase if nothing changes, as experimental structures will remain limited, as predictive models become more efficient and accurate over time. To solve this problem would be to find methods that learn ***domain-invariant structural features***.

This work addresses the research gap using ***domain adaptation***, a transfer learning method that adapts a model trained on a labeled source domain to an unlabeled target domain. In the context of this paper, it is applied from PDB to AFDB to see if predicted structures can be classified without retraining on millions of AlphaFold protein structures [17].

III. Methodology

A. Research Plan

A clear domain gap exists between PDB and AFDB, and it must be addressed to enable reliable protein fold classification on predicted structures [15]. Domain adaptation provides a way to align the two datasets, using PDB as a labeled source domain and AFDB as an unlabeled target domain, creating a unified system that benefits both experimental and predicted data.

Our approach is comparative and empirical. The idea is to implement three models that represent increasing levels of domain awareness, to quantify the domain shift:

- Naive Transfer Learning: Trains only on PDB and tests directly on AFDB, with no domain adaptation, serving as a baseline against domain-aware methods [14].
- Gradient Reversal: Uses a gradient reversal layer to learn domain-invariant features by confusing a domain classifier during training.
- ADDA: Trains separate encoders for PDB and AFDB and aligns them, allowing the target encoder to adapt independently [18].
- WDGRL: Aligns source and target features by minimizing Wasserstein distance, producing stronger and more stable domain adaptation [19], [20].

All models will be trained on filtered PDB data, to test their ability to classify AlphaFold-predicted structures without any target, domain labels, or training from the AFDB. This emulates the real-life constraint where experimental structures are scarce, while predicted structures exist on a massive scale.

The results aim to show a direct comparison of the three models using the following metrics to determine the efficiency of the three models:

- Model Accuracy
- Classification Reports
  - Precision
  - Recall
  - F-1 score
- Confusion Matrices

Beyond this, we will be able to visualize our multi-dimensional data with t-SNE charts, visualizing the data in a lower dimensional space while maintaining local structure [25]. These projected graphs show latent representations of both domains in low-dimensional space. By maintaining local structure, t-SNE reveals any separate clusters that are formed by the PDB and AFDB samples, starting from the naive model, and whether this separation decreases with supposedly superior models, giving us a qualitative view of the accuracy.

To ensure a fair comparison across the four methods, the same evaluation metrics are applied to PDB validation, as well as AFDB test set, although classification performance is measured using the evaluation metrics on the *AlphaFold Protein Structure Database* test set across fold classes.

The combination of metrics provided above will allow us to have a better idea of how the models perform, giving us a more global perspective on each individual model's performance. This complete comparison will be incredibly valuable in determining which model is the best fit.

B. Research Objectives

For this work, we have the following hypothesis and corresponding research objectives:

Hypothesis: Of the four trained models, Wasserstein Distance Guided Representation Learning will most accurately classify AlphaFold-predicted structures, while Naive will yield the lowest performance due to a lack of domain adaptation.

O1: Train four models to evaluate the effects of different domain-awareness on classification performance.

O2: Benchmark all training methods under one list of criteria to determine the highest performing model.

O3: Experimentally determine the differences between the PDB and AFDB entries and confirm the domain shift.

C. *Research Methodology*

It is imperative that we create a clean pipeline to guide the research and programming involved in this project:

Data Collection: We can construct a dataset from the SCOPe (Structural Classification of Proteins—extended) database, selecting proteins from seven major structural classes, using superfamily stratification to ensure diversity and avoid homology bias, sampling evenly across superfamilies rather than randomly. For each protein, we can obtain experimental structures from the PDB, and corresponding predicted structures from AFDBv6 by mapping PDB IDs to UniProt accessions via the SIFTS database [5]. Starting with 14,000 proteins (2,000 per class), we find 12,052 matched pairs after filtering for successful downloads and valid structures, maintaining balanced class distribution.

Feature Extraction: We extracted Cα (alpha carbon) coordinates from both PDB and AlphaFold structure files, using BioPython's PairwiseAligner to align sequences and crop coordinates to match the exact SCOP domain boundaries [29]. These coordinates were converted to 2D contact maps, capturing the protein's tertiary structure in a rotation-invariant representation [11]. The contact maps we use for this paper have a resolution of 128x128.

The following steps in the method provide a template for which all models can be trained and evaluated:

1. **Data Preparation**: The dataset consists of 2-D protein contact maps derived from the PDB, serving as the source domain, and computationally predicted protein structures from AlphaFold, serving as the target domain. The dataset consists of 12052 paired protein contact maps from source and target domain, with 128 x 128 resolution across 7 protein classes. Source and target domain are both stored into binary format NumPy arrays. The source dataset was split into 80% training and 20% testing sets using stratified sampling to preserve class distributions, with a fixed random state to ensure reproducibility across all experiments. To address class imbalance, class weights were computed using the balanced weighting scheme and incorporated into a weighted cross-entropy loss function.

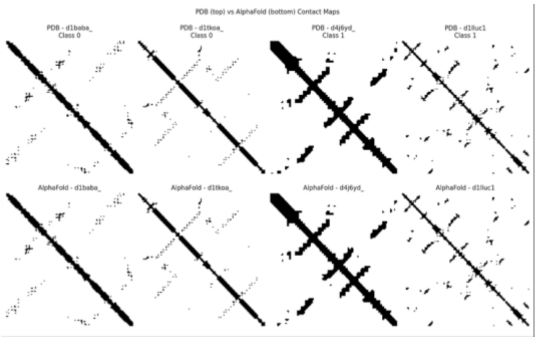


Fig1: Protein Contact Maps from source (PDB) and target (AlphaFold)

2. **Model Architecture:** All models shared a common feature extractor consisted of a four-layer convolutional neural network (CNN) with progressively increasing filter size (32,64,128, and 256 filters), where each convolutional layer was followed by batch normalization, ReLU activation, and max pooling [21], [22]. Global average pooling was applied to produce a 256-dimensional feature vector. The classifier consisted of a two-layer multilayer perceptron with a hidden dimension of 128 ReLU activation, and 50% dropout of regularization, outputting prediction across seven protein classes.

3. **Training Configuration:** All models were trained using the Adam optimizer with a batch size of 64. The naïve baseline was trained for 30 epochs with a learning rate of 0.001. For domain adaptation methods, lower learning rates were employed to stabilize adversarial training. DANN was trained for 50 epochs with a learning rate of 0.0001, using a gradient reversal layer with λ gradually increasing from 0 to 0.5 over the course of training. ADDA followed a two-stage approach: the source encoder and classifier were first pretrained for 30 epochs with a learning rate of 0.001, followed by 50 epochs of adversarial adaptation with a learning rate of 0.0001, during which the source encoder remained frozen while the target encoder learned to match the source feature distribution. WDGRL was trained for 50 epochs with a learning rate of 0.0001, employing a domain critic updated five times per generator step with weight clipping set to 0.01 to enforce the Lipschitz constraint.

Parameter	Naive	DANN	ADDA	WDGRL
Batch Size	64	64	64	64
Learning Rate	0.001	0.0001	0.001/0.0001	0.0001
Epochs	30	50	30 + 50	50
Optimiser	Adam	Adam	Adam	Adam

Table 1: Training parameters for the models

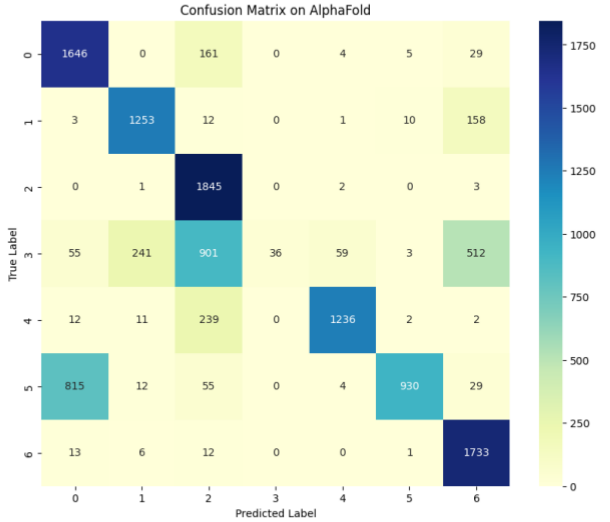
4. **Model Selection and Evaluation:** For all domain adaptation methods, the model checkpoint achieving the highest accuracy on the target domain (AlphaFold) was saved and used for final evaluation. Performance was assessed using accuracy, per-class precision, recall, and F1-score. Additionally, t-SNE visualizations were generated to qualitatively assess feature space separation, and confusion matrices were computed to identify class-specific classification patterns.

Upon completion of this pipeline, we will be able to accurately decide which models have performed the best, and monitor the changes between them.

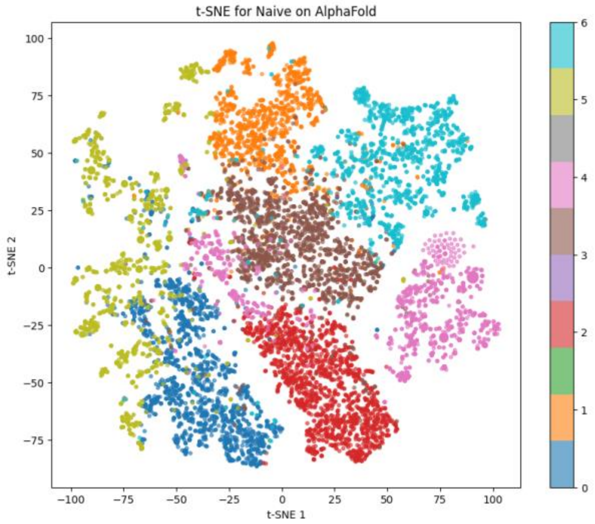
Naive Model

The Naive methodology serves as a baseline model for this experiment. It is a fold-classification network trained solely on labeled data from the source domain of PDB. A single feature extractor and a classifier train using the supervised fold labels from experimental sources, with no access to target-domain samples, no alignment losses during training, and **no adaptation**. This makes a simple model without any AFDB context.

	Precision	Recall	f1-score	Support
0	0.65	0.89	0.75	1845
1	0.82	0.87	0.85	1437
2	0.57	1.00	0.73	1851
3	1.00	0.02	0.04	1807
4	0.95	0.82	0.82	1502
5	0.98	0.50	0.67	1845
6	0.70	0.98	0.82	1765
Accuracy			0.72	12052



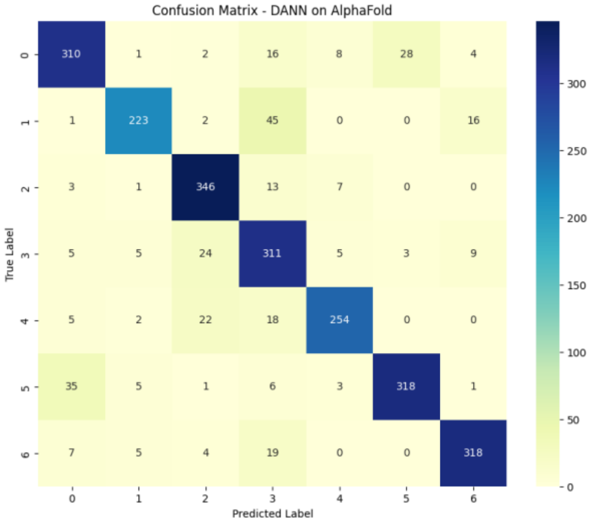
	0	1	2	3	4	5	6
0	1646	0	161	0	4	5	29
1	3	1253	12	0	1	10	158
2	0	1	1845	0	2	0	3
3	55	241	901	36	59	3	512
4	12	11	239	0	1236	2	2
5	815	12	55	0	4	930	29
6	13	6	12	0	0	1	1733



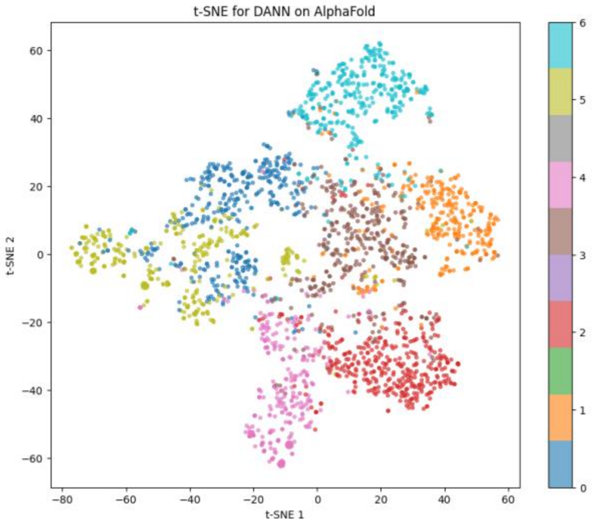
Gradient Reversal / RevGrad Model

The Gradient Reversal (or RevGrad) methodology adapts a classic fold classifier to a new domain by aiming to learn **domain-invariant features** through adversarial training. A shared feature extractor feeds into a fold classifier, and a domain classifier. A *gradient reversal layer (GRL)* inverts the gradient from the domain classifier during backpropagation, which leads the feature extractor to remove domain-specific signals. These signals represent the distinction between PDB and AFDB.

	Precision	Recall	f1-Score	Support
0	0.85	0.84	0.84	369
1	0.92	0.78	0.84	287
2	0.86	0.94	0.90	370
3	0.73	0.86	0.79	362
4	0.92	0.84	0.88	301
5	0.91	0.86	0.89	369
6	0.91	0.90	0.91	353
Accuracy			0.86	2411



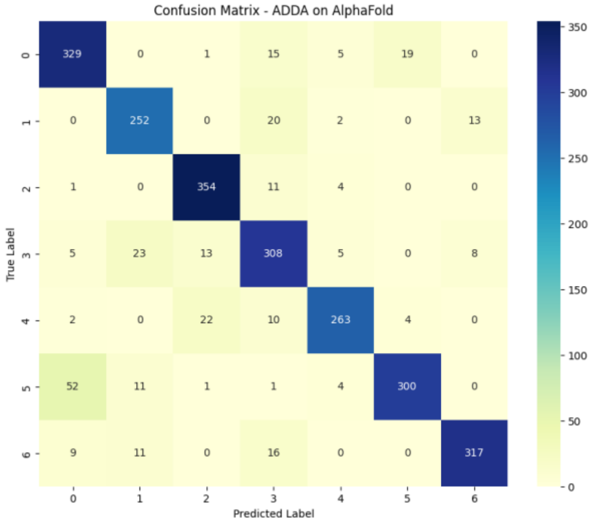
	0	1	2	3	4	5	6
0	310	1	2	16	8	28	4
1	1	223	2	45	0	0	16
2	3	1	346	13	7	0	0
3	5	5	24	311	5	3	9
4	5	2	22	18	254	0	0
5	35	5	1	6	3	318	1
6	7	5	4	19	0	0	318



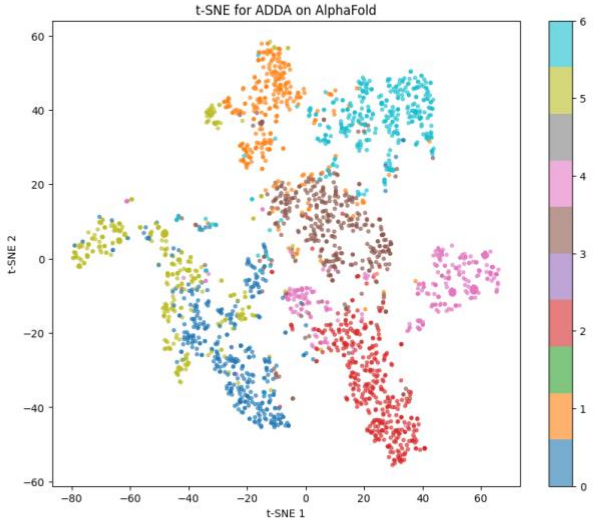
Adversarial Discriminative Domain Adaptation Model

The Adversarial Discriminative Domain Adaptation (ADDA) methodology aims to classify unlabeled models by learning two separate encoders. One encoder is to identify the source (PDB), while the second encoder is used for the target (AFDB). It uses an adversarial approach, similar to RevGrad, to reduce the domain shift between the source and target data distributions. A discriminator is also defined to determine which domain a given feature originates from.

	Precision	Recall	f1-Score	Support
0	0.83	0.89	0.86	369
1	0.85	0.88	0.86	287
2	0.91	0.96	0.93	370
3	0.81	0.85	0.83	362
4	0.93	0.87	0.90	301
5	0.93	0.81	0.87	369
6	0.94	0.90	0.92	353
Accuracy			0.88	2411



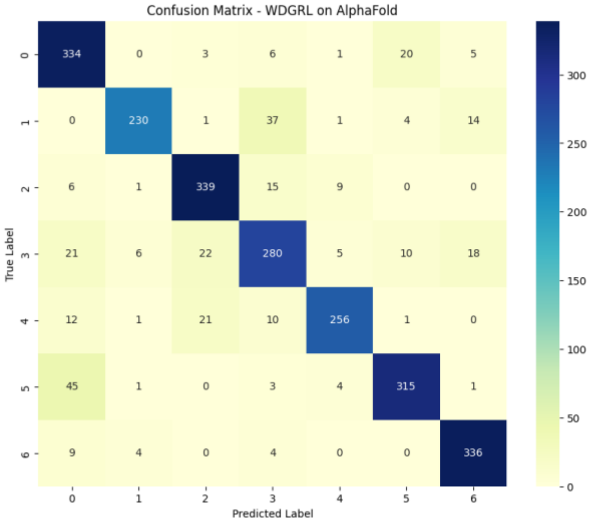
	0	1	2	3	4	5	6
0	329	0	1	15	5	19	0
1	0	252	0	20	2	0	13
2	1	0	354	11	4	0	0
3	5	23	13	308	5	0	8
4	2	0	22	10	263	4	0
5	52	11	1	1	4	300	0
6	9	11	0	16	0	0	317



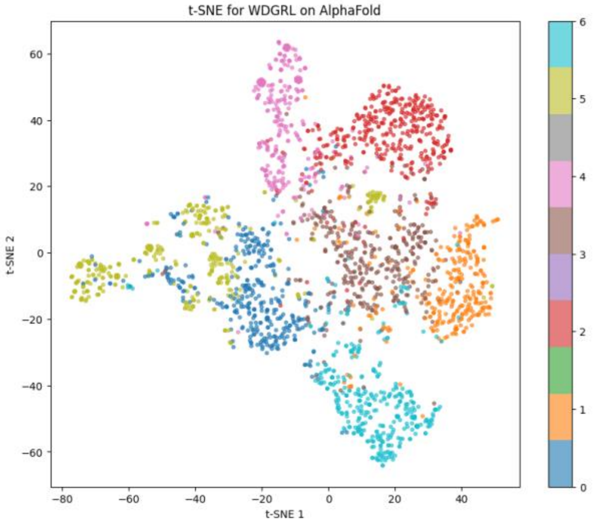
Wasserstein Distance Guided Representation Learning Model

The Wasserstein Distance Guided Representation Learning model (WDGRL) aligns the source and target domains by minimizing a property known as the ***Wasserstein Distance*** between feature distributions. Creating stable domain-invariant representations. A feature extractor is updated to reduce the distance in the latent space. This distance-based classification produces a more stable and continuous alignment than binary domain confusion.

	Precision	Recall	f1-Score	Support
0	0.78	0.91	0.84	369
1	0.95	0.80	0.87	287
2	0.88	0.92	0.90	370
3	0.79	0.77	0.78	362
4	0.93	0.85	0.89	301
5	0.90	0.85	0.88	369
6	0.90	0.95	0.92	353
Accuracy			0.87	2411



	0	1	2	3	4	5	6
0	334	0	3	6	1	20	5
1	0	230	1	37	1	4	14
2	6	1	339	15	9	0	0
3	21	6	22	280	5	10	18
4	12	1	21	10	256	1	0
5	45	1	0	3	4	315	1
6	9	4	0	4	0	0	336



IV. Results

A. Accuracy Comparison

We trained four protein fold classification models on labeled PDB structures and then evaluated them based on how accurately they can classify AlphaFold-predicted structures (without the use of target-domain labels during training). The evaluation was performed on the AFDB test set.

To keep things fair, all models share the same feature extractor, classifier, data splits, and optimization settings.

In terms of accuracy, the Naive model performed with about 72% accuracy, which is a reasonable baseline. Gradient Reversal had 86% accuracy, ADDA has 88% accuracy, and WDGRL had 87% accuracy. The three models performed comparatively very similarly.

This directly shows that the domain shift between PDB and AFDB is strong enough to break direct generalization, and that adversarial adaptation can cause an increase of approximately 15% from a naive model [8].

B. Per-Class Performance

The classification reports highlight large differences in class-wise recall. The naive model exhibits mode collapse, where specific folds show high precision but negligible recall, failing to capture the true data distribution. Domain-aware approaches fix this. RevGrad elevates recall globally to a range of 0.78-0.94, effectively preventing collapse. ADDA provides marginal gains, notably in classes 2, 3, and 5. WDGRL delivers the most consistent stability, maintaining balanced F1-scores above 0.84 across all classes.

C. Confusion Matrix Analyis

Confusion matrices for AFDB illustrate the specific impact of domain adaptation. The naive model displays a skewed matrix where predictions cluster around a limited set of classes, indicative of severe domain shift. In contrast, RevGrad, ADDA, and WDGRL feature strong diagonal structures, signaling enhanced class separation and reduced bias. ADDA exhibits the sharpest diagonal intensity, while WDGRL effectively minimizes off-diagonal errors for classes 2 and 6.

D. Latent Space Visualization (t-SNE)

Using t-SNE, we visualized feature embeddings for PDB and AFDB. The naive model separates samples into distinct domain clusters, demonstrating that the encoder learns domain-specific rather than structural features. RevGrad forces partial cluster overlap, reducing domain variance. ADDA results in extensive distribution mixing, whereas WDGRL generates a smooth, aligned manifold that preserves local class structure. This confirms that Wasserstein distance facilitates continuous alignment rather than sharp collapse.

E. Summary of Findings

Overall, the analysis confirms that:

1. Naive transfer learning is ineffective on AFDB.
2. Adversarial adaptation significantly enhances generalization.
3. Geometric (WDGRL) and discriminative (ADDA) alignments outperform standard gradient reversal.

V. Conclusion

Based on the results, the work demonstrates that domain adaptation is very required to transfer structural knowledge learned from experimental PDB structures to AlphaFold-predicted structures. Directly applying a supervised classifier that is trained only on experimental data.

All three domain adaptation methods significantly improve performance compared to the naive baseline. Gradient reversal recovers most of the lost accuracy by learning domain-invariant features through adversarial training. ADDA achieves the highest overall accuracy, showing that separate encoders and adversarial matching produce strong transfer despite no target-domain labels. WDGRL provides comparably high accuracy and the most stable class-wise performance, and t-SNE results show that minimizing Wasserstein distance aligns source and target distributions while preserving local class structure.

These results confirm the hypothesis that distance-based alignment is a strong approach for structural domain adaptation, and they show that standard fold classifiers can be extended to predicted structures without retraining on millions of AlphaFold models. As predicted structure databases continue to grow faster than experimental datasets; domain adaptation enables small, labeled datasets to scale to global protein space [30].

Future work may include extending this approach to multi-domain proteins, structure-aware transformers, or contact-graph models, and testing whether domain adaptation can improve other structure-based tasks such as binding site prediction or protein function annotation.

In terms of our research objectives:

O1: We achieved this objective, models were trained, and differences were quantified

O2: We achieved this objective; the benchmark shows that ADDA & WDGRL > Gradient Reversal > Naive

O3: We achieved this objective, based on the t-SNE charts, as well as the evaluation metrics, we can confirm there is a domain shift.

Our hypothesis was partially correct, although WDGRL was not the most accurate model; it was not far off. On top of that, naive was by far the lowest performer.

REFERENCES

[1] H. M. Berman, J. Westbrook, Z. Feng, et al., “The Protein Data Bank,” Nucleic Acids Research, vol. 28, pp. 235–242, 2000.

[2] A. G. Murzin, S. E. Brenner, T. Hubbard, and C. Chothia, “SCOP: a structural classification of proteins database,” Journal of Molecular Biology, 1995.

[3] C. A. Orengo, A. D. Michie, S. Jones, et al., “CATH – A hierarchic classification of protein domain structures,” Structure, 1997.

[4] P. W. Rose, A. Prlić, A. Altunkaya, et al., “The RCSB Protein Data Bank: integrative view of protein, gene and 3D structural information,” Nucleic Acids Research, 2017.

[5] S. Velankar, J. Dana, A. Jacobsen, et al., “SIFTS: Structure Integration with Function, Taxonomy and Sequences,” Nucleic Acids Research, 2013.

[6] J. Jumper, R. Evans, A. Pritzel, et al., “Highly accurate protein structure prediction with AlphaFold,” Nature, vol. 596, pp. 583–589, 2021.

[7] R. Evans, M. O’Neill, A. Pritzel, et al., “Protein complex prediction with AlphaFold-Multimer,” bioRxiv, 2022.

[8] A. Senior, R. Evans, J. Jumper, et al., “Improved protein structure prediction using potentials from deep learning,” Nature, 2020.

[9] K. Tunyasuvunakool, J. Adler, Z. Wu, et al., “Highly accurate protein structure prediction for the human proteome,” Nature, 2021.

[10] M. Varadi, S. Anyango, M. Deshpande, et al., “AlphaFold Protein Structure Database: massively expanding the structural coverage of protein-sequence space,” Nucleic Acids Research, 2022.

[11] Z. Wang, S. Eickholt, and J. Cheng, “Protein contact map prediction using deep learning,” Bioinformatics, 2017.

[12] R. Rohl, C. Strauss, K. Misura, and D. Baker, “Protein structure prediction using Rosetta,” Methods in Enzymology, 2004.

[13] J. L. Sussman, I. Silman, D. S. Blundell, et al., “Protein tertiary structure and folding,” Advances in Protein Chemistry, 1998.

[14] S. J. Pan and Q. Yang, “A Survey on Transfer Learning,” IEEE Transactions on Knowledge and Data Engineering, 2010.

[15] G. Csurka, “A Comprehensive Survey on Domain Adaptation,” arXiv preprint arXiv:2112.02849, 2021.

[16] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan, “Domain Separation Networks,” Advances in Neural Information Processing Systems, 2016.

[17] Y. Ganin and V. Lempitsky, “Unsupervised Domain Adaptation by Backpropagation,” International Conference on Machine Learning, 2015.

[18] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, “Adversarial Discriminative Domain Adaptation,” IEEE Conference on Computer Vision and Pattern Recognition, 2017.

[19] H. Shen, J. Qu, S. Zhang, and D. Liu, “Wasserstein Distance Guided Representation Learning,” AAAI Conference on Artificial Intelligence, 2018.

[20] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, “Improved Training of Wasserstein GANs,” Advances in Neural Information Processing Systems, 2017.

[21] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” arXiv:1409.1556, 2014.

[22] S. Ioffe and C. Szegedy, “Batch Normalization: Accelerating Deep Network Training,” International Conference on Machine Learning, 2015.

[23] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Network Overfitting,” Journal of Machine Learning Research, 2014.

[24] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” International Conference on Learning Representations, 2015.

[25] L. van der Maaten and G. Hinton, “Visualizing Data using t-SNE,” Journal of Machine Learning Research, 2008.

[26] E. Jones, S. Taylor, and P. Thornton, “Protein Fold Classification Using Machine Learning,” Proteins: Structure, Function, and Bioinformatics, 2005.

[27] H. Hou, R. Cheng, and Y. Zheng, “Deep Learning for Protein Fold Recognition,” BMC Bioinformatics, 2018.

[28] D. Duvenaud, D. Maclaurin, J. Aguilera-Iparraguirre, et al., “Convolutional Networks on Graphs for Learning Molecular Fingerprints,” Advances in Neural Information Processing Systems, 2015.

[29] S. Eddy, “Multiple Sequence Alignment,” Nature Reviews Genetics, 2004.

[30] M. Steinegger and J. Söding, “MMseqs2 enables sensitive protein sequence searching,” Nature Biotechnology, 2017.

[31] M. van Kempen, M. Kim, P. Tunyasuvunakool, et al., “Foldseek: fast and accurate protein structure search,” Nature Biotechnology, 2024.

[32] C. Levinthal, “How to fold graciously,” Mossbauer Spectroscopy in Biological Systems, 1969.