단백질 임베딩 방식과 GNN 모델에 따른 PPI 예측 성능 비교

지유빈

동국대학교 의생명공학과

Performance Comparison of PPI Prediction According to Protein Embeddings and GNN Models

Yu-bin Ji

요 약

단백질은 세포 내에서 다양한 cascade와 pathway를 이루며 상호작용하고, 특정 단백질–단백질 상호작용(PPI)의 교란은 암을 포함한 여러 질환의 발병과 진행에 직결된다. 본 연구에서는 이러한 단백질-단백질 상호작용을 예측하기 위해 대규모 단백질 언어 모델(PLM) 임베딩과 그래프 신경망(GNN)을 결합한 모델을 구축하였다. HuRI에서 제공하는 Y2H 기반 인간 PPI 데이터와 STRING 데이터베이스의 physical interaction을 통합하여 8,776개의 노드와 124,002개의 간선으로 구성된 무방향 가중 그래프를 만들고, 노드 특성으로 ProtBERT, ProtT5, ESM1b, ESM2 임베딩을 사용하였다. 간선 단위 이진 분류 태스크를 위해 PyTorch Geometric의 RandomLinkSplit으로 train/validation/test를 8:1:1로 분할하고, GCN과 GAT를 비교하였다. 그 결과, 동일한 임베딩을 사용할 때 모든 PLM에서 GCN이 GAT보다 일관되게 높은 F1-score, AUC, Accuracy를 보였으며, 최적 조합에서는 Test F1-score가 최대 0.86 수준에 도달하였다. 또한 PCA로 임베딩 차원을 512로 축소한 경우 Prot 계열(ProtBERT, ProtT5)에서는 성능이 소폭 감소한 반면, ESM 계열(ESM1b, ESM2)에서는 정보 손실에도 불구하고 F1-score가 증가하여, 고차원 ESM 임베딩에서는 PCA가 노이즈 제거와 정규화에 기여함을 확인하였다. 추가로 Prot 계열 임베딩에 대해 L2 정규화를 적용한 실험에서는 원본 임베딩 대비 성능 향상이 관찰되지 않아, 해당 임베딩에서는 추가적인 정규화·차원 축소보다 원본 표현을 그대로 사용하는 것이 유리함을 보였다. 본 연구는 PLM–GNN 기반 PPI 예측에서 그래프 구조 선택(GCN vs GAT)과 임베딩 차원 축소 여부가 성능에 미치는 영향을 정량적으로 제시하며, 향후 PPI 맥락에 특화된 경량 LLM 모듈을 추가하는 등 구조·네트워크·언어 정보를 통합한 모델로 확장할 가능성을 제안한다.

키워드 : 단백질-단백질 상호작용, 그래프 신경망, 단백질 언어 모델, 그래프 합성곱 신경망, 그래프 어텐션 네트워크, 주성분 분석

Key Words : PPI(Protein-Protein Interaction), GNN(Graph Neural Network), PLM(Protein Language Model), GCN(Graph Convolutional Network), GAT(Graph Attention Network), PCA(Principal Component Analysis)

Ⅰ. Introduction

신약 개발은 일반적으로 후보 물질 발굴, 전임상, 임상, 허가 및 발매의 네 단계로 구성되며, 하나의 신약이 시장에 나오기까지 평균 10년 이상이 소요된다[1]. 이 과정에서 시간과 비용이 많이 투입되기 때문에, 초기 단계에서 효율적으로 후보 물질을 선별하는 전략이 중요하다.

과거에는 표적 단백질을 억제하기 위한 후보 리간드를 찾는 과정이 96-well plate와 같은 실험 기반 고속 스크리닝에 크게 의존하는 노동집약적인 작업이었다[2]. 최근에는 인공지능(AI) 기술을 활용하여, 데이터 기반 모델로 대규모 화합물 라이브러리를 1차 선별한 뒤, 도킹(docking)이나 분자동역학 시뮬레이션과 같은 물리 기반 모델로 후보를 정제하는 접근이 널리 사용되고 있다[3].

한편, 특정 단백질을 억제하기 위해 리간드만을 사용하는 접근에는 분명한 한계가 존재한다. 대표적인 예가 MDM2–p53 상호작용이다. p53은 종양 억제 기능을 수행하는 단백질이지만, E3 Ubiquitin ligase인 MDM2와 결합하면 그 기능이 억제되어 암 발생에 기여할 수 있다. 이 경우 단순히 p53의 활성을 증가시키는 것만으로는 충분하지 않으며, MDM2와 p53 사이의 결합 자체를 차단하는 것[4]이 보다 직접적인 치료 전략이 될 수 있다. 이처럼 Protein–Protein Interaction(PPI)을 표적으로 삼으면, 신호 전달 경로에서 특정 결합 이벤트를 선택적으로 차단하여 질병 관련 경로를 조절할 수 있으므로, PPI 예측 모델은 신약 표적 탐색에 중요한 도구가 될 수 있다.

최근 구조 기반 PPI 예측을 다룬 리뷰에서는 SVM, RF, CNN 등 다양한 기계학습·딥러닝 기법을 비교하면서, 단백질 네트워크를 그래프로 표현하고 그 구조를 바로 학습에 사용하는 그래프 신경망(Graph Neural Network, GNN)이 PPI 예측에서 대표적인 state-of-the-art 접근으로 부각된다고 정리한다[5]. 이에 본 연구에서는 yeast two-hybrid(Y2H) 기반 인간 PPI 데이터셋을 활용하여 GNN 기반 PPI 예측 모델을 구축하고자 한다. 특히, 대규모 단백질 언어 모델(Protein Language Model, PLM)에서 얻은 단백질 임베딩을 노드 특성으로 사용하고, Graph Neural Network(GNN)를 중심으로 모델을 설계하여, 임베딩·모델 조합에 따른 예측 성능 변화를 비교·분석하는 것을 목표로 한다.

Ⅱ. Methods

2.1 Datasets

본 연구에서는 단백질–단백질 상호작용(Protein–Protein Interaction, PPI) 예측을 위해 실험 기반 데이터셋과 데이터베이스 기반 데이터셋을 함께 사용하였다. 우선, 실제 yeast two-hybrid(Y2H) 실험을 통해 구축된 인간 상호작용 데이터셋인 Human Reference Interactome(HuRI)의 HI-union.tsv를 사용하였다[6]. HI-union.tsv에서 각 행은 하나의 상호작용 쌍을 나타내며, 단백질은 Ensembl Gene ID(ENSG)로 표기된다. 추가적으로, PPI 데이터베이스인 STRING에서 Homo sapiens에 대한 physical interaction만을 추출하여 보조적인 상호작용 정보로 사용하였다. STRING 데이터는 다양한 실험, 데이터베이스, 공동발현 증거를 통합한 점수 체계를 가지며, 본 연구에서는 physical interaction 타입에 해당하는 상호작용만을 선택하였다.

최종 그래프의 노드 수, 간선 수, HuRI/STRING 상호작용 비율 등은 결과(Results)에서 제시한다. 엣지 단위 학습을 위해 전체 그래프는 PyTorch Geometric의 RandomLinkSplit을 이용하여 train/validation/test를 약 8:1:1 비율로 무작위 분할하였다. 이때 is_undirected=True로 설정하여 무방향 그래프를 가정하였으며, 각 split마다 실제 상호작용(positive edge)과 동일한 수의 비상호작용(negative edge)을 무작위로 샘플링하여 이진 분류 태스크를 구성하였다.

2.2 Uniprot Mapping and Sequence

HuRI의 HI-union.tsv에서 등장하는 모든 ENSG ID를 추출한 뒤, 중복을 제거하여 고유한 유전자 목록을 얻었다. 이후 UniProt에서 제공하는 REST API를 이용하여 각 ENSG ID에 대응되는 UniProt ID와 단백질 서열(sequence)을 수집하였다. 이때, UniProtKB/Swiss-Prot에 해당하는 ID만을 사용하고, UniProtKB/TrEMBL 항목은 제외하였다.

Swiss-Prot는 큐레이터 검토를 통해 하나의 단백질에 대해 신뢰할 수 있는 대표 서열을 제공하는 반면[7], TrEMBL은 동일 유전자에서 유래한 여러 변이 단백질이 각각 별도의 ID로 존재할 수 있기 때문이다. 따라서, 본 연구에서는 하나의 유전자를 하나의 대표 단백질로 매핑하기 위해 Swiss-Prot 항목만 사용하였다. UniProt ID로 매핑되지 않는 ENSG ID에 대해서는 해당 노드 및 관련 간선을 이후 그래프 구성에서 제외하였다.

2.3 Protein Embedding

단백질 서열 임베딩은 대규모 단백질 언어 모델(Protein Language Model, PLM)을 이용하여 생성하였다. 본 연구에서는 Rostlab에서 공개한 ProtBERT, ProtT5와 Meta AI에서 개발한 ESM1b, ESM2 총 네 가지 PLM을 사용한다.

ProtBERT:
BERT 구조를 기반으로 하며, 대규모 단백질 서열 데이터베이스(BFD 등)를 이용해 masked language modeling 방식으로 사전 학습된 모델이다[8].

ProtT5:
T5 인코더 구조를 기반으로 하며, 대규모 단백질 서열을 대상으로 사전 학습된 모델로, 보다 풍부한 문맥 정보를 반영할 수 있도록 설계되었다[8].

ESM1b / ESM2:
Meta AI에서 개발한 Transformer 기반 PLM으로, UniRef100과 같은 대규모 단백질 서열 집합을 이용해 학습되었으며, 다양한 구조·기능 관련 다운스트림 태스크에서 우수한 성능을 보이는 것으로 알려져 있다[9][10].

단백질 임베딩 결과는 ProtBERT, ESM1b, ESM2의 경우 모델이 제공하는 임베딩을 그대로 사용하였고, ProtT5의 경우 Mean Pooling을 이용하여 한 벡터로 압축하였다. 이때 최종 임베딩 차원은 ProtBERT와 ProtT5의 경우 1024차원, ESM1b와 ESM2의 경우 1280차원이다. 추가 실험에서 PCA를 통해 512차원으로 축소한 버전도 비교하였다.

2.4 Graph Construction

앞서 정의한 ENSG UniProt 매핑을 바탕으로, 고유 ENSG ID 목록을 오름차순으로 정렬한 뒤에 각 ENSG ID에 순차적인 노드 번호를 부여하여 그래프의 노드를 정의하였다. 이후 HI-union.tsv에서 각 상호작용 쌍에 대해 해당하는 노드 번호를 찾아 무방향 간선(undirected edge)으로 추가하였다. STRING에서 추출한 physical interaction 데이터 역시 동일한 방식으로 노드 번호를 매칭하여 간선을 구성하였다. 두 데이터셋의 신뢰도 차이를 반영하기 위해, Y2H 기반 HuRI 상호작용에는 가중치 1.0을, STRING 기반 physical interaction에는실제 STRING에서 제공하는 score값을 가중치로 사용하였다. 이를 통해 실험 기반 상호작용을 우선적으로 반영하면서도, STRING에서 제공하는 추가적인 결합 정보를 보조적으로 활용하고자 하였다. 마지막으로, 각 노드의 특성(feature)은 2.3에서 생성한 단백질 임베딩 벡터를 사용하였다. 즉, 그래프의 노드는 단백질 하나에 대응되며, 해당 단백질의 PLM 임베딩이 노드 피처로 주어진다.

2.5 GNN Model and Training Setting

2.5.1 GNN Model

본 연구에서는 기본 GNN 모델로 Graph Convolutional Network(GCN)와 Graph Attention Network(GAT)를 사용하여 성능을 비교하였다.

GCN(Graph Convolutional Network):
그래프에서 각 노드의 특징을 이웃 노드들의 특징과 정규화된 가중합으로 섞어 가며 새로운 노드 임베딩을 학습하는 신경망이다. 인접 행렬을 기반으로 노드 주변의 국소 구조를 반영한다[11].

GAT(Graph Attention Network):
GCN의 이웃 집계 과정에 self-attention 메커니즘을 도입한 모델로, 어떤 이웃 노드의 정보를 더 중요하게 반영할지 학습된 attention weight를 통해 결정한다. 이웃 노드마다 서로 다른 가중치를 부여할 수 있어, 중요한 상호작용을 더 잘 포착할 수 있다는 장점이 있다[12].

최근 연구에서는 ESM-2와 같은 대형 PLM 임베딩을 사용할 경우, 모델 구조를 복잡하게 바꾸

어도 성능이 약 0.65 부근에서 포화되며, 더 큰 임베딩이나 복잡한 아키텍처는 오히려 과적합을 유발할 수 있다고 보고된 바 있다[13]. 이에 본 연구에서는 GCN과 GAT 모두 레이어 수를 최대 3층까지로 제한하고, 비교적 얕은 구조에서의 성능을 평가하였다.

손실함수는 이진 분류 문제에 적합한 binary cross-entropy를 사용하였고, 최적화는 Adam optimizer로 수행하였다. 모델 성능 평가는 F1-score, ROC-AUC, Accuracy를 지표로 하였으며, validation F1-score를 기준으로 early stopping(patience = 20)을 적용하였다. 각 실험에서 최대 epoch 수는 200으로 설정하였다.

2.5.2 Hyperparameter Tuning

하이퍼파라미터 튜닝은 random search 방식으로 수행하였다. 조절한 하이퍼파라미터는 은닉 차원(hidden channels), 레이어 수(layer numbers), 출력 차원(out channels), 학습률(running rate), dropout 비율(dropout rate), weight decay 총 6가지이며, 각 값의 후보는 표 1과 같다. 각 임베딩 종류(ESM1b, ESM2, ProtBERT, ProtT5)에 대해 여러 조합을 무작위로 100회 샘플링하여 학습을 수행하고, validation F1-score가 가장 높은 조합을 최종 모델 설정으로 선택하였다.

표 1. Hyperparameter tuning

Item	Random.Choice([value])
Hidden Channels	1024, 512, 256
Layer Numbers	1, 2, 3
Out Channels	256, 128
Learning Rate	0.001, 5e-4, 2e-4
Dropout Rate	0.2, 0.3, 0.4
Weight Decay	0.001, 5e-4, 1e-4, 1e-5

Weight decay는 손실함수에 가중치의 L2 제곱항을 추가하여, 학습 과정에서 가중치가 과도하게 커지지 않도록 억제하는 정규화 기법이다. 이를 통해 모델 복잡도를 완화하고 과적합(overfitting)을 줄이는 효과를 기대할 수 있다.

Ⅲ. Result

GNN에서 사용된 HI-union.tsv에서 총 단백질은 9094개이고 이중에 검증된 Uniprot/Swiss-prot ID는 8776개, TrEMBL ID이거나 아예 Uniprot ID가 없는 경우는 242개, 다른 Ensembl ID지만 동일한 Uniprot ID를 가지는 경우가 76개이다. 이에 노드로 구성될 단백질은 8776개이다. 이중 연결된 노드는 8759개, 고립된 노드는 17개로 구성되어 있다.

또한, 상호작용의 개수(HI-union.tsv)는 64006개이고, 이중 단백질이 필터링되면서 사라진 간선이 3627개이다. STRING API를 통해 얻은 상호작용의 개수가 25764개이고, 중복을 제거하고 추가된 간선이 22578개이다. PyTorch의 특성 상 간선은 (u, v), (v, u) 2개로 입력되어야 하기에 2배가 되어 총 간선은 165914개이다.

표 2. 단백질 임베딩 방식에 따른 Node feature의 분포 변화

PLM	Min	Mean	Max	Range
ProtBERT	-3.0108	0.0025	5.788	8.7988
ProtT5	-0.5817	0.00004	0.5882	1.1699
ESM1b	-19.4177	-0.0011	9.4842	28.9019
ESM2	-9.5394	-0.0011	0.993	10.5324

앞서 구성된 그래프를 기반으로 Train dataset은 132734 edges이고, Validation dataset은 16590 edges, Test dataset은 마찬가지로 16590 edges이다.

3.1 GCN, GAT 결과 비교

처음은 각 임베딩 방식에서 GCN과 GAT 2가지를 각각 사용해보고 어떤 모델이 해당 Dataset에 더 잘 fit하는지 보려고 한다. 결과는 그림 1, 표 3과 같다.

표 3. 단백질 임베딩 방식과 모델 종류에 따른 최적의 하이퍼파라미터 조합

Model	In Chennel	Hidden Channel	Layer	Out Chennel	Drop out	Learning Rate	Decay weight
(a)	1024	1024	3	256	0.2	2e-4	5e-4
(b)	1024	1024	1	256	0.3	0.001	1e-5
(c)	1024	1024	2	128	0.3	0.001	1e-5
(d)	1024	512	1	128	0.3	0.001	1e-4
(e)	1280	256	3	128	0.4	5e-4	0.001

(f)	1280	1024	1	256	0.4	0.001	1e-5
(g)	1280	1024	3	128	0.2	2e-4	5e-4
(h)	1280	256	3	128	0.3	0.001	1e-4

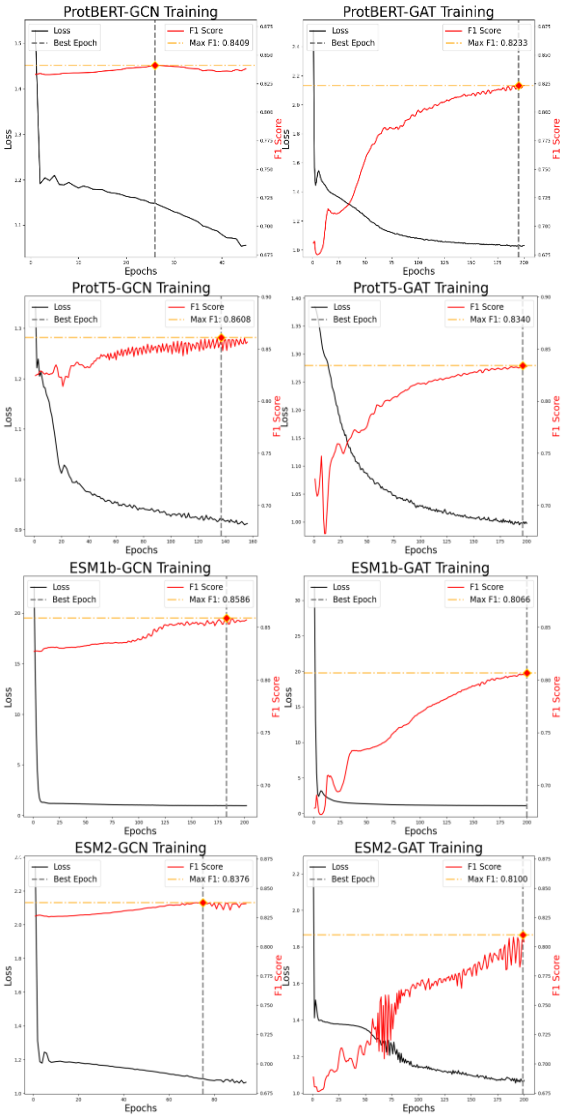


그림 1. GCN(a, c, e, g)와 GAT(b, d, f, h)에 대해 4가지 PLM: ProtBERT(a, b), ProtT5(c, d), ESM1b(e, f), ESM2(g, h)을 적용한 학습 곡선. 검은색 곡선은 validation loss, 빨간색 곡선은 validation F1-score를 나타내며, 회색 점선은 가장 높은 F1-score값에서의 epoch, 노란색 점선은 최대 F1-score값을 의미한다. (a)~(h)에 따른 하이퍼파라미터 조합은 표 3에 정리하였다.

ProtBERT는 test F1-score가 0.8216에서 0.8447로, ProtT5는 0.8324에서 0.8620으로, ESM1b는 0.7999에서 0.8594로, ESM2는 0.8125에서 0.8399로 GAT 대신 GCN을 사용할 때 전반적으로 약 0.03p, 최대 약 0.06p의 성능 향상을 보였다.

표 3은 각 PLM–모델 조합에서 가장 높은 F1-score를 달성한 하이퍼파라미터를 정리한 것이다. GCN의 경우 은닉 계층 수가 3인 설정이 자주 선택된 반면, GAT는 대부분 은닉 계층 수가 1인 설정에서 최적 성능을 보였다. 이는 GAT의 self-attention 메커니즘과 관련이 있는 것으로 보인다. 계층 수가 3으로 늘어나면 "이웃 노드의 이웃 노드"까지 정보를 집계하게 되는데, 다수의 이웃 노드에 대해 attention weight를 학습하는 과정이 지나치게 복잡해지면서 오히려 일반화 성능이 저하될 수 있다.

본 연구에서 사용한 PPI 그래프는 하나의 단백질 노드에 다수의 상호작용이 집중되는 구조를 갖는다. 이러한 환경에서는 GAT가 중요 이웃을 정확히 구분하지 못하고 attention weight를 비효율적으로 분배했을 가능성이 있을 수 있다. 그 결과 GCN에 비해 분류 성능이 낮게 나타난 것으로 해석할 수 있다. 해당 결과를 토대로 이후 진행되는 실험은 GCN을 기본 모델로 사용하였다.

3.2 PCA 여부 결과 비교

학습에 사용되는 노드의 차원이 높으면, 어떤 차원의 피처가 중요한지 중요도 선정이 잘못되어있을 수 있다. 이에 각 단백질 임베딩 결과에서 차원을 축소하여 중요한 피처만을 남기고 도일하게 실험을 진행하였다. 결과는 그림 2, 표 4, 표 5와 같다.

표 4. 단백질 임베딩 방식에서 PCA 여부에 따른 Test F1-score 비교표

Protein embedding	Test F1 Without PCA	Test F1 With PCA	Difference	Explain Ratio
ProtBERT	0.8447	0.8386	-0.0061	99.21%
ProtT5	0.8620	0.8471	-0.0148	97.56%
ESM1b	0.8594	0.8632	0.0088	96.26%
ESM2	0.8399	0.8576	0.0177	95.8%

그림 2. 4가지 PLM 임베딩(ProtBERT(a, b), ProtT5(c, d), ESM1b(e, f), ESM2(g, h))에 대해 GCN을 적용했을 때, PCA 적용 여부(원본 vs 512 차원 축소)에 따른 학습 곡선. 원본 임베딩을 사용한 GCN(a, c, e, g), PCA를 통해 512차원으로 축소한 임베딩을 사용한 GCN(b, d, f, h)의 결과이다. 검은색 곡선은 validation loss, 빨간색 곡선은 validation F1-score를 나타내며, 회색 점선은 가

장 높은 F1-score값에서의 epoch, 노란색 점선은 최대 F1-score값을 의미한다. 새로 진행한 (b), (d), (f), (h)에 따른 하이퍼파라미터 조합은 표 5에 정리하였다.

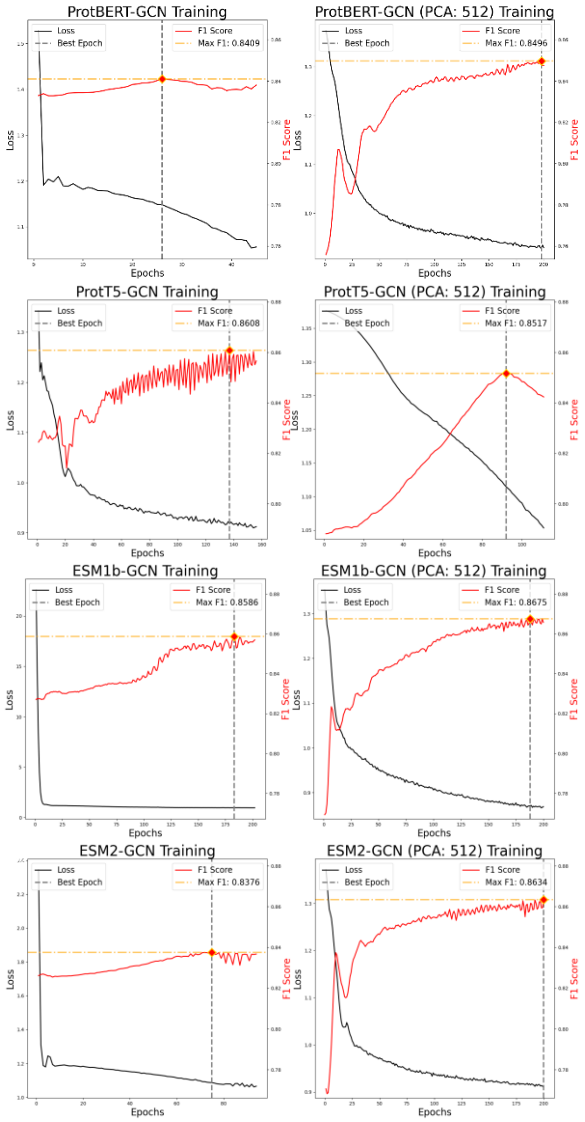


표 4는 각 단백질 임베딩 방식에 대해 PCA 적용 여부에 따른 test F1-score 변화를 정리한 것이다. Explain ratio는 PCA로 차원을 축소했을 때 원래 분산이 얼마나 보존되는지를 나타내며, 값이 클수록 정보 손실이 적다는 것을 의미한다. 그림 2에서는 ProtT5에서만 성능 감소가 두드러지는 것처럼 보이지만, 표 4의 Test dataset에서의 F1-score difference를 보면 Prot family(ProtBERT, ProtT5)는 PCA 적용 후 F1-score가 감소한 반면, ESM family(ESM1b, ESM2)는 오히려 성능이 증가한 것을 확인할 수 있다.

하지만, Explain ratio를 보면 ESM family가 Prot family보다 더 낮아, PCA 과정에서 ESM 쪽이 오히려 정보 손실이 더 큰 것처럼 보인다. PCA는 전체 분산이 큰 축부터 보존하는 방식으로 차원을 축소하므로, ESM family의 경우 분산이 큰 축들에 PPI 분류에 유효한 정보가 상대적으로 많이 분포해 있고, 분산이 작은 축에는 노이즈나 중복 정보가 더 많이 포함되어 있었을 가능성이 있다. 이에 따라 PCA가 고차원 ESM 임베딩에서 불필요한 성분을 제거하고, 분류에 중요한 신호를 상대적으로 강조하는 정규화 효과를 낸 것으로 해석할 수 있다. 반대로 Prot family에서는 분산이 작은 축들에도 중요한 정보가 분포해 있었을 수 있으며, 이 축들이 PCA 과정에서 제거되면서 성능이 감소했을 가능성이 있다. 이러한 결과로부터 ESM 계열 임베딩은 PCA 적용이 유리하게 작용한 반면, Prot 계열 임베딩에서는 PCA가 성능 향상에 도움이 되지 않았다는 것을 알 수 있다.

표 5. 단백질 임베딩 방식에 따른 PCA 결과에서 최적의 하이퍼파라미터 조합

Model	In Chennel	Hidden Channel	Layer	Out Chennel	Drop out	Learning Rate	Decay weight
(b)	512	512	2	128	0.4	0.001	1e-5
(d)	512	256	2	128	0.2	2e-4	1e-5
(f)	512	1024	2	128	0.4	0.001	1e-5
(h)	512	1024	2	256	0.4	0.001	1e-4

표 5는 PCA를 적용한 실험에 한해, 각 단백질 임베딩 방식에서 얻어진 최적의 하이퍼파라미터 조합을 정리한 것이다. 입력 차원이 512로 축소되었음에도, ESM 계열 모델(f, h)의 경우 은닉 차원이 1024로 비교적 크게 설정된 점을 확인할 수 있다. 이는 PCA를 통해 상대적으로 중요한 성분만 남은 표현에 대해, 넓은 은닉층이 보다 다양한 조합 패턴을 학습할 수 있도록 해 준 결과일 가능성이 있다. 다시 말해, ESM 계열에서는 차원 축소로 노이즈가 줄어든 상태에서 모델의 “폭(width)”을 늘리는 것이 성능 향상에 긍정적으로 작용했을 수 있다.

3.3 PCA 차원에 따른 결과 비교

앞선 3.2절의 결과로부터, ESM 계열 임베딩에 대해 PCA를 적용하여 노드 피처 차원을 축소했을 때 GCN 기반 PPI 예측 성능이 소폭 향상됨을 확인하였다. 이에 보다 강한 차원 축소가 성능에 미치는 영향을 분석하기 위해, 512차원으로 축소했던 ESM1b·ESM2 임베딩을 한 번 더 축소하여 256차원까지 줄인 뒤, 동일한 GCN 구조와 학습 조건에서 실험을 수행하였다. 결과는 그림 3과 표 6에 제시하였다.

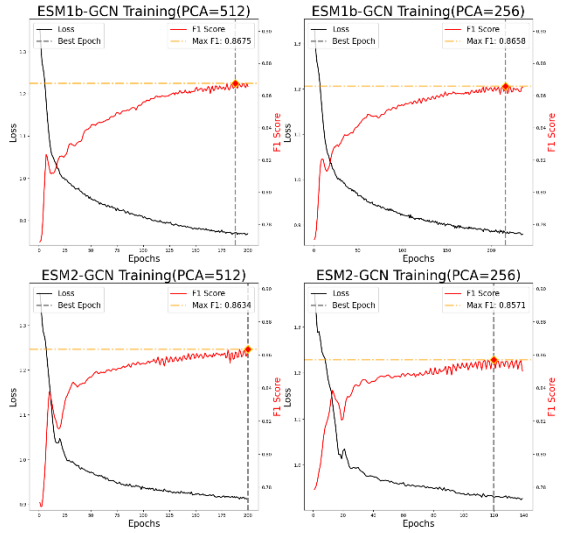


그림 3. ESM 계열 임베딩에 대해 GCN을 적용했을 때 PCA 차원(512, 256)에 따른 학습 곡선. (a), (b)는 각각 512차원과 256차원의 ESM1b, (c), (d)는 512차원과 256차원의 ESM2 임베딩을 사용한 결과이다. 검은색 곡선은 validation loss, 빨간색 곡선은 validation F1-score를 나타내며, 회색 점선은 가장 높은 F1-score값에서의 epoch, 노란색 점선은 최대 F1-score값을 의미한다.

표 6. ESM1b, ESM2에서 PCA 차원 수에 따른 Test F1-score 비교표

Protein embedding	Test F1-score Without PCA	Test F1-score PCA(=512)	Test F1-score PCA(=256)
ESM1b	0.8594	0.8632	0.8664
ESM2	0.8399	0.8576	0.8503

PCA를 이용해 차원을 256으로 축소한 결과, ESM1b와 ESM2 임베딩의 누적 설명분산비(explained variance ratio)는 각각 90.79%, 90.14%로 약 10% 수준의 정보가 손실되었다. 그림 3에서 보듯이 validation F1-score는 512차원 대비 큰 차이를 보이지 않았다. 표 6에 제시된 것처럼, 256차원 ESM1b 임베딩을 사용했을 때 test F1-score는 0.8664로 512차원일 때보다 소폭 증가하였으나 그 차이는 미미한 수준이며, ESM2의 경우에는 오히려 성능이 감소하였다. 이는 보다 강한 차원 축소로 인해 유용한 정보까지 일부 제거된 결과로 해석할 수 있다. 따라서 설명분산과 예측 성능을 함께 고려했을 때, PCA를 통해 차원을 512까지 축소했을 때가 가장 적절한 설정으로 판단하였다.

3.4 정규화 여부에 따른 결과 비교

앞선 3.2절의 결과로부터 Prot 계열 임베딩에 대해 PCA를 적용하여 노드 피처 차원을 축소했을 때, GCN 기반 PPI 예측 성능이 오히려 소폭 감소함을 확인하였다. PCA를 도입한 이유는 차원 축소와 함께 일정 수준의 정규화 효과를 기대했기 때문이지만, 성능 저하가 관찰되었으므로 차원 축소를 제거하고 정규화만 적용한 경우를 별도로 비교하였다. 이를 위해 Prot 계열 임베딩에 대해 동일한 GCN 구조와 학습 조건을 유지한 채 L2 정규화를 수행하였다.

$$\tilde{x} = \frac{x}{\|x\|_2} \quad \left(\|x\|_2 = \sqrt{x_1^2 + x_2^2 + x_3^2 + \cdots + x_d^2} \right)$$

위 식은 L2 정규화의 정의를 나타낸다. 결과는 그림 4와 표 7에 제시하였다.

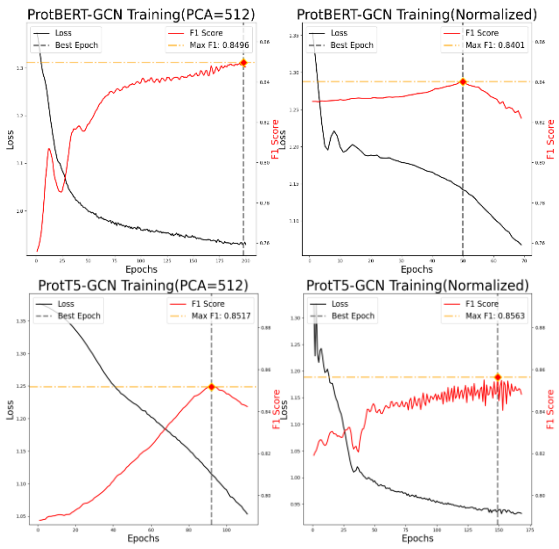


그림 4. Prot 계열 임베딩에 대해 GCN을 적용했을 때 L2 정규화 여부에 따른 학습 곡선. (a), (b)는 ProtBERT의 정규화 전후 결과 , (c), (d)는 ProtT5의 정규화 전후 결과이다. 검은색 곡선은 validation loss, 빨간색 곡선은 validation F1-score를 나타내며, 회색 점선은 가장 높은 F1-score값에서의 epoch, 노란색 점선은 최대 F1-score값을 의미한다.

표 7. ProtBERT, ProtT5에서 PCA, L2 norm에 따른 Test F1-score 비교표

Protein embedding	Test F1-score	Test F1-score PCA(=512)	Test F1-score L2 norm
ProtBERT	0.8447	0.8386	0.836
ProtT5	0.8620	0.8471	0.8537

그림 4에서 ProtT5는 PCA(512차원) 대비 L2 정규화를 적용했을 때 F1-score가 거의 유사하거나 소폭 증가하는 수준으로 변화 폭이 크지 않은 반면, ProtBERT는 PCA 대비 L2 정규화 이후 F1-score가 눈에 띄게 감소하는 양상을 보인다. 또한 표 7에서 ProtBERT의 경우 원본 임베딩 대비 PCA 및 L2 정규화 설정 모두에서 test F1-score가 소폭 감소하였으며, ProtT5 역시 원본 임베딩에서 가장 높은 성능을 보였고, L2 정규화는 PCA보다 약간 높지만 원본보다는 낮은 성능을 나타냈다. 이러한 결과는 PLM 임베딩에서 방향 정보뿐 아니라 벡터의 절대 크기 정보 역시 PPI 예측에 유의미한 신호로 작용하고 있으며, 정규화 과정에서 이러한 스케일 기반 정보가 제거됨에 따라 예측력이 일부 감소한 것으로 해석할 수 있다. 즉, 두 임베딩 모두 PCA나 L2 정규화를 적용하였을 때 원본 대비 성능 향상은 관찰되지 않았으며, 특히 Prot 계열 임베딩의 경우 별도의 정규화나 차원 축소 없이 원본 임베딩을 그대로 사용하는 것이 피처 정보를 가장 잘 보존하는 설정임을 시사한다.

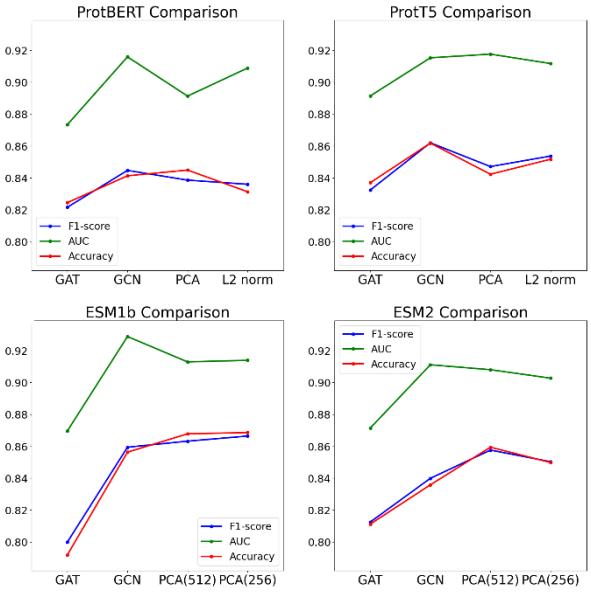
Ⅳ. 결 론

본 연구에서는 인간 PPI 네트워크(HuRI, STRING)을 기반으로 PLM(Protein Language Model) 임베딩과 GNN(Graph Neural Network)을 결합한 PPI 예측 모델을 구축하고, GCN과 GAT 구조 및 다양한 PLM 조합에 대한 비교를 수행하였다. HuRI의 Y2H 상호작용과 STRING의 물리적 상호작용으로 구성된 그래프에서 노드 특성은 4가지 PLM(ProtBERT, ProtT5, ESM1b, ESM2) 임베딩으로 정의하고, 무방향 가중 그래프를 입력으로 F1-score를 주요 평가 지표로, AUC, Accuracy를 보조 평가 지표로 사용하였다.

표 8. 각 실험 후 Test dataset에 대한 평가지표 값

		Model	F1-score	AUC	Accuracy
	GAT	ProtBERT	0.8216	0.8734	0.8245
		ProtT5	0.8324	0.8914	0.8370
		ESM1b	0.7999	0.8696	0.7919
		ESM2	0.8125	0.8714	0.8112
	GCN	ProtBERT	0.8447	0.9159	0.8413
		ProtT5	0.8620	0.9153	0.8618
		ESM1b	0.8594	0.9288	0.8563
		ESM2	0.8399	0.9111	0.8358
PCA 512	GCN	ProtBERT	0.8386	0.8913	0.8449
		ProtT5	0.8471	0.9176	0.8423
		ESM1b	0.8632	0.9129	0.8678
		ESM2	0.8576	0.9080	0.8594
L2		ProtBERT	0.836	0.9089	0.8313
		ProtT5	0.8537	0.9117	0.8517
256		ESM1b	0.8664	0.9139	0.8686
		ESM2	0.8503	0.9027	0.8498

그림 5. 표 8을 기반으로 구성한 그래프, 단백질 임베딩 모델과 학습 설정에 따른 test dataset에서의 평가 지표의 변화. (a) ProtBERT의 평가 지표의 변화, (b) ProtT5의 평가 지표 변화, (c) ESM1b의 평가 지표 변화, (d) ESM2의 평가 지표 변화



실험 결과, 동일한 임베딩을 사용할 때 GCN이

GAT보다 일관되게 높은 예측 성능을 보였다. 표 8과 그림 5에서 볼 수 있듯이 3가지 평가 지표에 관해 모두 동일한 경향을 보였다. 또한, 최적 하이퍼파라미터를 비교하면 GCN은 대체로 layer수가 3에 가까운 비교적 깊은 구조를, GAT는 layer 수가 1에 가까운 얕은 구조를 선호하는 경향을 보였다. 이는 고차원 노드가 많은 PPI 그래프에서 self-attention을 통해 여러 '이웃의 이웃'의 서로 다른 가중치를 학습하는 GAT가 3-hop까지 정보를 전파한 경우 과도한 정보 처리로 성능이 저하될 수 있음을 시사한다. 반면 GCN은 보다 단순한 인접 평균 연산으로 '이웃의 이웃' 정보를 통합하여 해당 데이터셋에서는 안정적인 학습을 수행한 것으로 해석할 수 있다.

또한, PLM임베딩에 대해 PCA를 진행하여 자원을 512로 축소한 후 성능 변화를 분석하였다. Prot family(ProtBERT, ProtT5)는 PCA 적용 이후 Test F1-score가 소폭 감소한 반면, ESM family(ESM1b, ESM2)는 정보 손실이 더 컸음에도 오히려 F1-score가 증가하였다. 그림 3에서 볼 수 있듯 Accuracy도 유사한 경향을 보였다. 이는 ESM 임베딩의 경우 분산이 더 큰 주성분 방향에 생물학적으로 유의미한 정보가 더 잘 모여있어, PCA를 통해 노이즈가 제거되고 더 압축된 표현이 학습에 유리하게 작용했을 가능성을 시사한다. 실제로, PCA를 적용한 ESM1b, ESM2는 은닉 차원이 1024인 넓은 GCN에서 가장 좋은 성능을 보였고, 이는 노이즈가 줄어든 저차원 입력과 충분히 넓은 은닉층 조합이 ESM 계열 임베딩에 적합할 수 있음을 보여준다.

추가로 ESM1b/ESM2를 256차원까지 더 축소한 경우에는 F1-score가 거의 증가하지 않거나 오히려 감소하여, 과도한 차원 축소가 유용한 정보를 함께 제거함을 확인하였다. ProtBERT와 ProtT5에 대해서는 PCA(512차원) 및 L2 정규화를 별도로 적용하였으나, 두 경우 모두 원본 임베딩을 사용할 때보다 F1-score가 소폭 낮게 나타나 Prot family에서는 추가적인 차원 축소·정규화보다 원본 표현을 그대로 사용하는 것이 더 유리함을 시사한다.

본 연구는 PLM-GNN 기반 PPI 예측에서 그래프 구조 선택(GCN vs GAT)과 임베딩 차원 축소 여부가 성능에 어떻게 영향을 미치는지에 대한 실증적 근거를 제공한다는 점에서 의미가 있다. 한편, 본 연구는 단백질을 노드로 하는 protein-level PPI 네트워크에 초점을 맞추었기 때문에, 단백질 내부의 residue-level 3차원 구조 정보나 실제 결합 인터페이스를 직접적으로 활용하지 못한다는 한계를 가진다. 향후에는 PLM 기반 residue 임베딩과 구조 기반 그래프(GCN/GAT 등)를 결합한 residue-level GNN을 도입하여, 단백질 내부 구조와 단백질 간 네트워크 정보를 동시에 학습하는 multi-scale PPI 예측 모델로 확장하고자 한다. 또한 추후 연구에서는 동일한 데이터 분할 하에서 각 모델을 반복 학습하고, run별 test F1-score에 대해 paired t-test를 적용하여 모델 간 성능 차이의 통계적 유의성을 검증할 예정이다.

References

[1] Cura Precision Biomedical, KHIDI USA Office, "Overview of the new drug development process and recent trends in the United States," *KHIDI Brief*, vol. 376, pp. 1–14, Nov. 2022. (in Korean). Available: https://www.khidi.or.kr/board/view?pageNum=1&rowCnt=10&no1=481&linkId=48893116&menuId=MENU01783

[2] D. A. Pereira and J. A. Williams, "Origin and evolution of high throughput screening," *British Journal of Pharmacology*, vol. 152, pp. 53–61, July 2007. (https://doi.org/10.1038/sj.bjp.0707373)

[3] Z. Cournia, B. K. Allen, T. Beuming, D. A. Pearlman, B. K. Radak, and W. Sherman, "Rigorous free energy simulations in virtual screening," *Journal of Chemical Information and Modeling*, vol. 60, pp. 4153–4169, June 2020. (https://doi.org/10.1021/acs.jcim.0c00116)

[4] Z. Cournia, B. K. Allen, T. Beuming, D. A. Pearlman, B. K. Radak, and W. Sherman, "Rigorous free energy simulations in virtual screening," *Journal of Chemical Information and Modeling*, vol. 60, pp. 4153–4169, June 2020. (https://doi.org/10.1021/acs.jcim.0c00116)

[5] D. P. Kiouri, G. C. Batsis, and C. T. Chasapis, "Structure-based approaches for protein–protein interaction prediction using machine learning and

deep learning," *Biomolecules*, vol. 15, no. 1, Art. no. 141, Jan. 2025. (https://doi.org/10.3390/biom15010141)

[6] K. Luck, D.-K. Kim, L. Lambourne, et al., "A reference map of the human binary protein interactome," *Nature*, vol. 580, pp. 402–408, Apr. 2020. (https://doi.org/10.1038/s41586-020-2188-x)

[7] UniProt, "The UniProt databases," EMBL-EBI Training, available at: https://www.ebi.ac.uk/training/online/courses/uniprot-exploring-protein-sequence-and-functional-info/what-is-uniprot/the-uniprot-databases/ (accessed 2025-11-23)

[8] A. Elnaggar, M. Heinzinger, C. Dallago, G. Rehawi, Y. Wang, L. Jones, T. Gibbs, T. Feher, C. Angerer, M. Steinegger, D. Bhowmik, and B. Rost, "ProtTrans: Toward Understanding the Language of Life Through Self-Supervised Learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 10, pp. 7112–7127, Oct. 2022. (https://doi.org/10.1109/TPAMI.2021.3095381)

[9] A. Rives, J. Meier, T. Sercu, S. Goyal, Z. Lin, J. Liu, D. Guo, M. Ott, C. L. Zitnick, J. Ma, and R. Fergus, "Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 118, no. 15, Article e2016239118, Apr. 2021. (https://doi.org/10.1073/pnas.2016239118)

[10] Z. Lin, H. Akin, R. Rao, B. Hie, Z. Zhu, W. Lu, N. Smetanin, R. Verkuil, O. Kabeli, Y. Shmueli, A. dos Santos Costa, M. Fazel-Zarandi, T. Sercu, S. Candido, and A. Rives, "Evolutionary-scale prediction of atomic level protein structure with a language model," *bioRxiv*, Dec. 2022. (https://doi.org/10.1101/2022.07.20.500902)

[11] T. N. Kipf and M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks," in *Proceedings of the 5th International Conference on Learning Representations (ICLR 2017)*, Toulon, France, Apr. 2017. (https://arxiv.org/abs/1609.02907)

[12] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph Attention Networks," in *Proceedings of the 6th International Conference on Learning Representations (ICLR 2018)*, Vancouver, Canada, Apr. 2018. (https://arxiv.org/abs/1710.10903)

[13] T. Reim, A. Hartebrodt, D. B. Blumenthal, J. Bernett, and M. List, "Deep learning models for unbiased sequence-based PPI prediction plateau at an accuracy of 0.65," *Bioinformatics*, vol. 41, Suppl. 1, pp. i590–i598, 2025. (https://doi.org/10.1093/bioinformatics/btaf192)

[14] K. Jha, S. Saha, and H. Singh, "Prediction of protein–protein interaction using graph neural networks," *Scientific Reports*, vol. 12, Article 8360, May 2022. (https://doi.org/10.1038/s41598-022-12201-9)