

Test Result Document

Project Name	
-----------------	--

xx 조

2023234 홍길동

2023345 허균

2023369 장길산

지도교수: 김컴공 교수님 (서명)

Table of Contents

1.INTRODUCTION	3
1.1. OBJECTIVE	3
2.EXPERIMENT RESULT REPORT.....	4
3.AI 도구 활용 정보	8

1. Introduction

1.1. Objective

본 연구의 목적은 복소 채널 $y = Hx + n$ 에서 채널 행렬 H 를 예측하는 것이다. 채널 예측을 통해 대규모 안테나 배열, 밀리미터파(mmWave) 및 서브테라헤르츠(sub-THz) 대역 운영, 수백만 대 디바이스 지원 등으로 네트워크 인프라가 고도로 밀집된 미래 환경에서 요구되는 고차원 신호 처리와 복잡한 최적화 문제를 효과적으로 해결하고자 한다.

이 문서에서는 본 연구에서 도입한 LWM파운데이션 모델의 성능 평가 결과를 종합적으로 제시한다. Val_RMSE, Val_NMSE 및 Val_NMSE(dB) 지표를 활용하여 예측 정확도를 정량적으로 분석하고, 주요 실험 설정 및 하이퍼파라미터 구성을 상세히 설명한다. 또한 전통 기법 및 대조 모델과 성능 비교하여 향후 개선 방향을 구조화된 형태로 정리하였다.

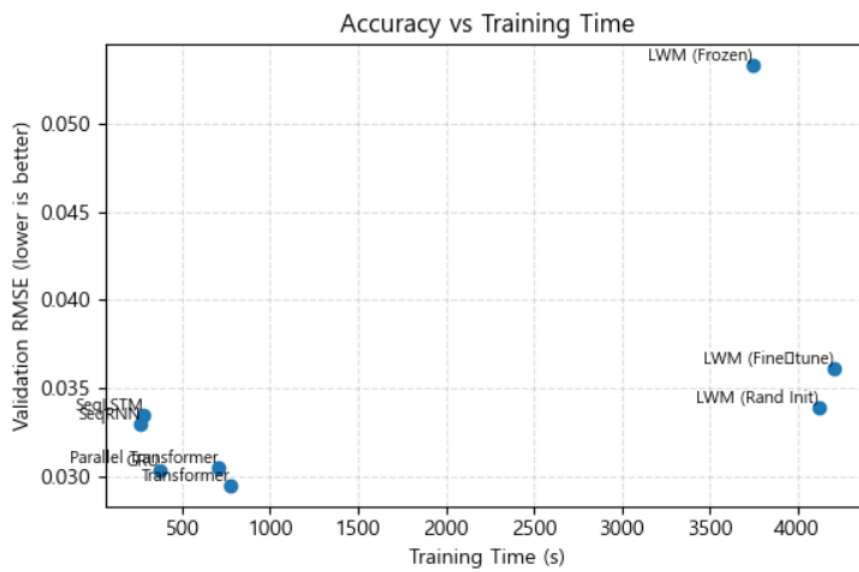
Experiment Result Report

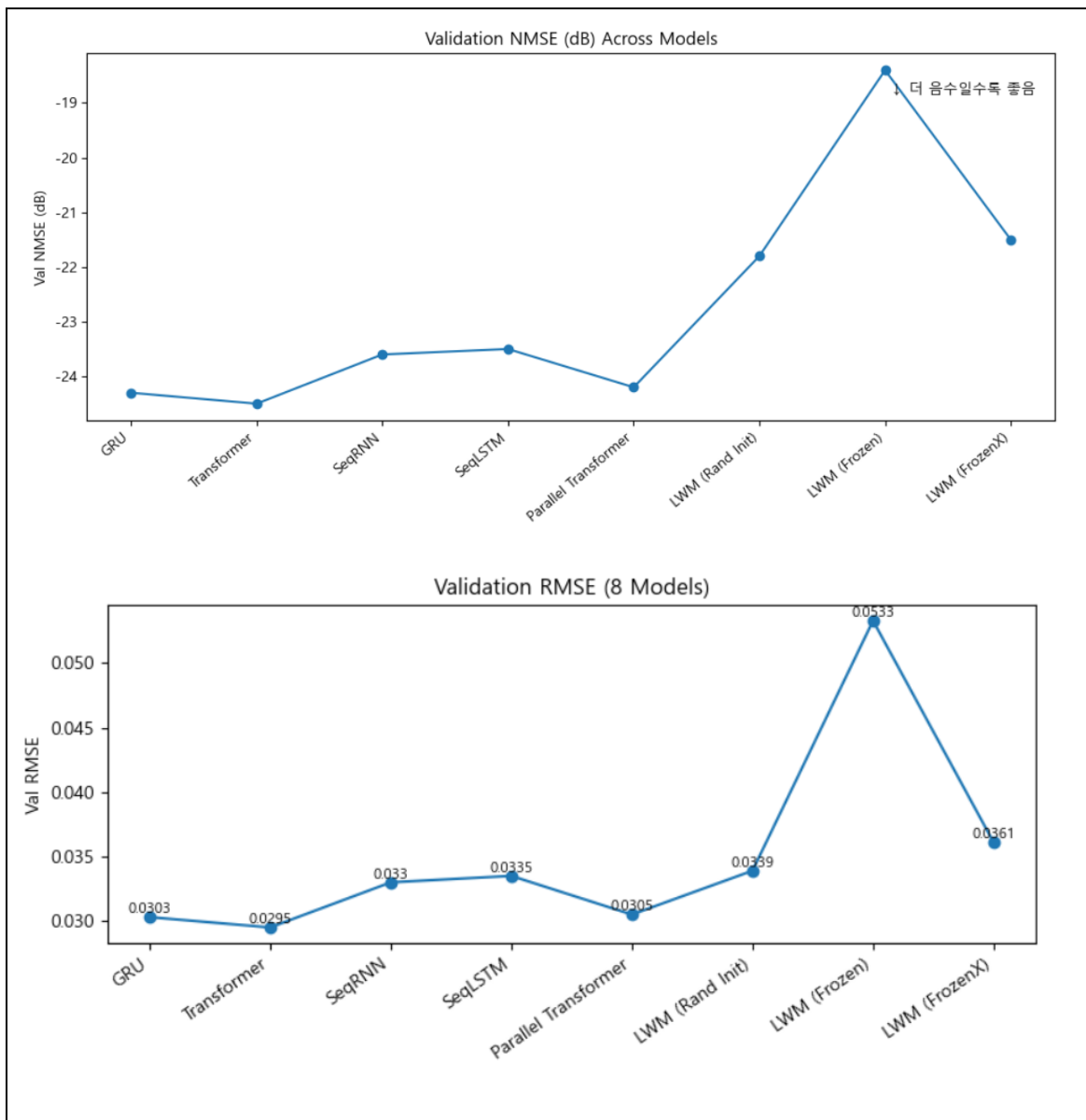
1. 서론
1.1 실험 개요
<ul style="list-style-type: none"> - 본 실험의 목적은 동적 시나리오에서 관측된 복소 채널 식 $y = Hx + n$에서 채널 행렬 H를 예측함에 있어서 LWM 파운데이션 모델이 채널 RMSE, NMSE(dB)측면에서 통계적으로 유의미한 성능 향상을 보이는지 검증한다. - 본 연구에서는 다음 네 가지 조건을 비교·검증한다. <ol style="list-style-type: none"> 1. LWM을 사전학습 가중치 없이 학습 2. LWM의 사전학습 가중치를 사용하되, 백본은 동결하지 않음 3. LWM의 사전학습 가중치를 사용하되, 백본을 동결함 4. GRU, Transformer, RNN, LSTM, Parallel-Transformer등 다양한 순환/어텐션 기법 모델 - 데이터: DeepMIMO v3 3.5GHz operating frequency: 'O2_dyn_3p5' scenario 장면 수 : 15 시퀀스 길이 : 5 유저 수 : 737 서브캐리어 수 : 64 Train/Val 비율 = 6 : 4 샘플 수 : 200563 - 학습/ 검증/ 테스트셋 구성 비율, 샘플 수 - 서버 및 환경: JupyterNotebook, Colab - GPU: RTX 3060
1.2 실험 방법
<ul style="list-style-type: none"> - 데이터 생성 <ol style="list-style-type: none"> 1. DeepMIMOv3의 데이터를 scene 단위로 5개씩 묶음으로 순차 생성 2. 전체 15개 scene을 모두 생성해 dataset = all_data 리스트 저장 3. 각 샘플 항목은 user, channel 정보 포함 - 데이터 전처리 및 분할 <ol style="list-style-type: none"> 1. ChannelSeqDataset <ol style="list-style-type: none"> 1-1. 입력: 과거 seq_len=5 시점의 (32안테나 X 1UE포트 X 64서브캐리어) 복소 채널 벡터 1-2. 출력: seq(5 X 64(실수+허수) 차원), masked_pos (fixed at seq_len-2), target (다음 시점 채널 벡터), 1-3. 스킵 조건: 과거 시퀀스 및 타겟이 모두 0 2. Train/Validation 분할

<p>2-1. 비율 60% (train) / 40% (val)</p> <p>2-2. dataset[:split_idx]로 train, dataset[split_idx:]로 val 생성</p> <p>3. 스케일링 및 TensorDataset</p> <p>3-1. MinMaxScaler로 X, y 각각 → train에만 fit → train/val에 transform</p> <p>3-2. TensorDataset(X_tr_s, mp_tr, y_tr_s) → train_loader(shuffle=True)</p> <p>3-3. TensorDataset(X_va_s, mp_va, y_va_s) → val_loader(shuffle=False)</p> <p>- 모델 아키텍처: LWMwithHead</p> <p>1. 입력 투영: Linear(input_dim=64 → patch_length=16)</p> <p>2. 백본(Transoformer encode)</p> <p>2-1. 12-layer, hidden size d_model=64</p> <p>2-2. 사전학습 가중치(ckpt_path="./model_weights.pth") 로드</p> <p>2-3. freeze_backbone=True/False</p> <p>3. 헤드(Downstream FC)</p> <p>3-1. Linear(64 → 256) → ReLU → Linear(256 → 64)</p> <p>- 훈련 설정</p> <p>Optimizer: Adam(model.parameters(), lr=1e-4)</p> <p>Loss: MSELoss()</p> <p>Epochs: num_epochs = 10</p> <p>Batch size: 32</p> <p>- 평가 지표</p> <p>1. RMSE: $\sqrt{\text{MSE}}$</p> <p>2. NMSE: $E[\hat{y}-y ^2] / E[y ^2]$</p> <p>3. 매 epoch마다</p> <p>Train: train_loader로 순회하며 backward & update</p> <p>Validation: evaluate(model, val_loader) → 평균 RMSE, NMSE 계산 →</p> <p>NMSE_dB = $10 \cdot \log_{10}(\text{NMSE})$ 출력</p>	
2. 테스트 결과 상세	
2.1 테스트 결과 개요	
수치/성능 요약표	

	Configuration	Val RMSE	Val NMSE (dB)	Training Time (s)
0	GRU	0.0303	-24.3	374.10
1	Transformer	0.0295	-24.5	775.85
2	SeqRNN	0.0330	-23.6	260.23
3	SeqLSTM	0.0335	-23.5	276.71
4	Parallel Transformer	0.0305	-24.2	705.98
5	LWM (Rand Init)	0.0339	-21.8	4118.97
6	LWM (Frozen)	0.0533	-18.4	3747.72
7	LWM (Fine-tune)	0.0361	-21.5	4206.82

수치/성능 그래프





2.2 테스트 결과 상세 분석

- 예외 사례, 비정상 결과
LWM 모델은 사전학습된 가중치(Pretrained weight)를 백본으로 활용했음에도 불구하고, 기존 시퀀스 모델 (GRU, Transformer 등) 대비 오히려 성능이 저조하게 나타났다.
- 왜 성능이 낮았는가? (데이터 문제인지, 모델 한계인지)
 - a. 데이터 문제
전체 11만 명의 유저 데이터를 확보할 수 있었음에도 하드웨어 문제로 인해 약 800명 분만 사용하였다. 이에 샘플 수 부족으로 Transformer 계열 모델의 일반화 능력 저하되었다.
 - b. 모델 한계
Transformer 구조는 충분한 샘플이 주어지지 않으면 과적합(Overfitting) 위험이 커지고, 일반화 성능이 떨어질 수 있다.

2.3 실험 결과의 한계와 위협 요인

- 연산 자원 및 학습 시간의 제약

1000개 장면을 모두 학습하려면 대용량 메모리와 강력한 GPU가 필수이다.

데이터 용량 제약으로 인해 실제 실험에서는 데이터 일부(15장면, 유저 727)만 활용하였으며, 이는 전체 데이터 분포를 충분히 대표하지 못해 모델 성능 저하를 부를 수 있다.

- 위협 요인

과적합 위험: 데이터가 부족한 채 특정 패턴에 과도하게 적응하면, 새로운 시퀀스 예측 시 오히려 성능이 크게 떨어질 수 있다.

메모리 오류 및 학습 중단: 학습 도중 GPU 메모리 부족으로 인해 학습이 중단되는 사례가 빈번하다.

3. 결론

- 핵심 발견 요약

사전학습된 LWM 파운데이션 모델을 그대로 고정(freeze)했을 때 오히려 성능이 저하됨을 확인
데이터 샘플 수가 부족이 성능 저하의 주요 원인

- 시스템 개선을 위한 제안

추가 GPU 장착 또는 더 큰 VRAM(예:A100) 탑재 서버 도입 필요

- 활용 가능성 등

고속 이동체(기차, 자율주행), 대규모 MIMO(mmWave-sub-THz)환경에서 실시간 채널 예측 모듈로 통합

스마트 팩토리 및 드론 등 다양한 IoT 시나리오에서 적용 가능

2. AI 도구 활용 정보

사용 도구	ChatGPT
사용 목적	LWM 모델 성능이 오히려 더 안좋은 이유 찾기
프롬프트	● 이러한 조건 속에서 LWM 모델 성능이 다른 모델에 비해 안좋은 이유는?
반영 위치	1. 테스트 결과 상세 분석(p.6)
수작업	있음(논리 보강)
수정	