자연어 행동 기반 시각-언어 내비게이션 과업에서의 적대적 패치 공격

김지원, 박태진, 이재구*
국민대학교 컴퓨터공학과
*jaekoo@kookmin.ac.kr

요 약

파운데이션 모델(Foundation Model)의 발전으로, 시각-언어 정보를 활용해 이동을 수행하는 시각-언어 내비게이션(Vision-Language Navigation, VLN) 과업이 활발히 연구되고 있다. 하지만 과업에 활용되는 시각-언어-행동(Vision-Language-Action, VLA) 모델은 시각 모델에서 알려진 적대적 공격(Adversarial Attack)의 취약점을 그대로 상속하여, 입력에 대한 교란(Perturbation)이 일어날 경우 잘못된 행동을 예측 및 수행할 수 있다는 한계를 갖는다. 특히 실제 로봇 주행 환경에서는 미세한 입력 교란 뿐 아니라 시야 내에 물리적으로 노출 및 부착될 수 있는 패치 형태의 교란이 큰 위협이 된다. 따라서 실제 환경에서 로봇을 안정적으로 주행시키기 위해서는 환경에 물리적으로 부착되어 모델의 잘못된 행동을 유도하는 적대적 패치 공격(Adversarial Patch Attack)에 대한 강건성 확보가 필수적이다. 본 논문은 안전한 시각-언어-행동 모델 개발에 기여하고자 자연어 행동 기반 시각-언어-행동 모델에서의 적대적 패치 공격을 제안하고, 패치 생성 방식 및 적용 방식에 따른 공격 성능을 분석하였다. 결과적으로 적대적 목표 행동 설정을 통한 적대적 패치 공격이 과업 성능 하락에 효과적임을 확인하였다.

1. 서론

파운데이션 모델(Foundation Model)의 발전으로 다중 모달리티(Modality)를 결합한 로봇 주행 연구가 활발히 이루어지고 있다[1-3]. 시각-언어 내비게이션(Vision-Language Navigation, VLN) 과업의 경우 시각-언어 모델(Vision-Language Model, VLM)을 통합하여 로봇이 관찰된 장면과 자연어 지시문을 활용해 목표 지점으로 이동하도록 한다. 이때 모델이 직접 행동을 생성하는 시각-언어-행동(Vision-Language-Action, VLA) 모델[3, 4]이 등장하며 실제 환경에서의 적용 가능성이 더욱 높아지고 있다.

하지만 해당 과업에서 사용되는 VLM과 VLA와 같은 시각-언어 모델은 적대적 공격(Adversarial Attack)에 취약하다는 한계가 존재한다. 이는 시각 인코더(Vision Encoder)가 가진 취약점을 그대로 상속하기 때문이며, 인간이 인지하기 어려운 미세한 입력 교란(Perturbation)만으로도 모델 예측이 왜곡되어 신뢰성 확보에 악영향을 미친다[5]. 특히 적대적 패치 공격(Adversarial Patch Attack)[6]은 인간이 인지할 수 없는 미세한 교란에 대한 제약을 제거하고, 이에 따라 생성된 적대적 패치를 실제

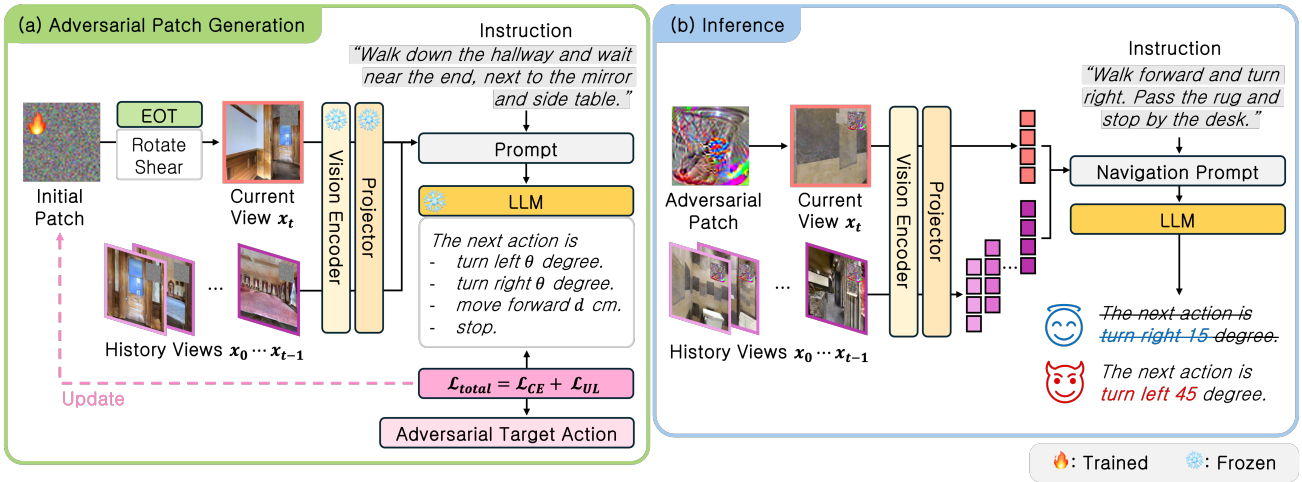


그림 1. 적대적 패치 생성과 적대적 공격 수행 과정

환경에 물리적으로 부착함으로써 로봇의 잘못된 예측을 유도한다. 이 경우 단순한 과업 실패를 넘어 안전상의 위험을 초래할 수 있다. 따라서 로봇을 실제 환경에서 안전하게 구동시키고, 시각-언어 내비게이션 과업을 성공적으로 수행하기 위해서는 적대적 패치 공격에 대한 강건성 연구가 필수적이다.

이를 위해 본 논문에서는 자연어 행동 기반 시각-언어-행동 모델[3]에서의 적대적 패치 공격을 제안한다. 이를 통해 시각-언어 내비게이션 과업에서의 적대적 패치 공격의 효력을 탐구하고, 적대적 패치 공격으로부터 안전한 시각-언어-행동 모델 개발에 기여하고자 한다.

2. 본론

본 논문은 적대적 패치 공격을 통해 시각-언어 내비게이션 과업의 성공률 및 주행 품질을 하락시키는 것을 목표로 한다. 전체 프레임워크는 [그림 1]과 같으며, 적대적 패치를 생성하는 과정인 [그림 1.(a)]와 생성된 패치를 적용해 추론하는 [그림 1.(b)]의 두 가지 부분으로 구성된다. 우리의 방법은 모델 학습이나 미세조정(Fine Tuning) 없이 적대적 패치만 학습하고, 생성된 적대적 패치를 통해 모델의 잘못된 예측을 유도한다.

본론의 하위 섹션에서는 제안 프레임워크의 적대적 목표 행동 설정 과정을 설명하고, 적대적 패치 생성과 이를 활용한 공격 과정을 보인다.

2.1 적대적 목표 행동 설정

The next action is move forward d cm
The next action is turn left / right θ degree
I think I should stop because I have finished the instruction.

$d \in \{25, 50, 75\}$ $\theta \in \{15, 30, 45\}$

그림 2. 자연어 행동 포맷

본 논문에서는 피해자 모델(Victim Model)로 선정한 NaVILA[3]가 자연어 기반의 행동을 생성함에 초점을 두어 적대적 목표 행동을 설계한다. 모델[3]은 [그림 2]와 같이 Move forward, Turn left, Turn right, Stop 총 4가지의 유형의 행동을 생성한다. 각 행동은 “The next action is ~” 포맷에 맞추어 생성되며, Stop의 경우만 “I think I should stop because I have finished the instruction.”의 고정 포맷으로 표현된다. 더 나아가 Move는 25, 50, 75 cm, Turn은 15, 30, 45 degree의 제어 값을 가진다.

모델[3]이 생성한 자연어 행동은 정규 표현식 기반의 키워드 파싱을 거쳐 시뮬레이터[7] 명령으로 변환된다. 이때 파싱되는 행동 키워드는 “stop”, “is move forward”, “is turn left”, “is turn right”로, Stop의 경우 고유한 문장 포맷과 다르더라도 키워드가 검출되면 동일하게 정지 명령으로 변환된다. 우리는 이러한 키워드 기반 행동 파싱 방식에 착안하여, 모델[3]의 정형화된 자연어 행동 생성 능력은 유지하되 정답과 상이한 행동을 출력하도록 하여 과업 수행 능력을 저하시키고자 하였다.

이에 따라 고유한 행동 포맷을 가진 Stop은 적대적 목표 행동을 통한 공격 대상에서 제외하고, Move forward는 공통 포맷을 유지하되 Stop 키워드가 파싱되도록 행동 유형을 반전하였다. Turn 계열은 행동 유형은 유지하되, 방향을 반전하고 제어 값을 변경하였다. 이렇게 생성된 적대적 목표 행동은 [그림 3]과 같다.

정답 행동	적대적 목표 행동
The next action is move forward d cm	The next action is stop
The next action is turn left θ degree	The next action is turn right θ' degree
The next action is turn right θ degree	The next action is turn left θ' degree

$\theta, \theta' \in \{15, 30, 45\}$, $\theta \neq \theta'$

그림 3. 적대적 목표 행동 및 제어 값 매핑

2.2 적대적 패치 생성

우리는 [섹션 2.1]에서 생성된 적대적 목표 행동을 사용해 적대적 패치를 생성한다. 패치는 가우시안 노이즈(Gaussian Noise)를 통해 초기화되고, 과거 관찰 이미지 7장과 현재의 관찰 이미지 1장에 각각 부착된다.

패치 생성 시, 변환에 대한 기댓값(Expectation Over Transformation, EOT)[8]을 이용하여 현실 환경에서 발생하는 시점 및 각도 변화와 같은 입력 변환을 반영하였다. 전단 변환(Shear Transformation)과 회전 변환(Rotation Transformation)을 확률적으로 적용하여 실제 환경에 부착된 패치가 관찰 이미지의 시점 변화로 인해 겪는 기하학적 왜곡을 모사하고, 변환 이후에도 효과적으로 공격 성능을 유지하는 적대적 패치를 생성하고자 하였다. 패치 부착 위치는 우상단으로, 장면의 구조물을 최대한 가리지 않도록 설정하였다.

모델은 패치가 부착된 8장의 이미지 입력과 자연어 지시를 기반으로 자연어 행동을 생성한다. 이 과정에서 모델 파라미터(θ)는 고정하고, 적대적 목표 행동을 정답으로 한 교차 엔트로피(Cross Entropy, CE) 손실을 통해 패치만을 업데이트 한

The next action is [행동] [제어 값] [제어 단위]	
[행동] Head	{'stop', 'move', 'turn'}
[행동] Tail	{'left', 'right'}
[제어 값]	{'15', '30', '45'}

그림 4. 확률 변수 제약

다.

이때 공통 포맷(The next action is ~) 및 기존의 정답 행동과 동일한 출력부는 마스킹을 통해 손실 계산에서 제외하며, [그림 3]과 같이 달라진 토큰에서만 손실이 계산되도록 하였다. 이렇게 손실이 계산되는 토큰들은 [수식 1]에서 I로 표현된다. 또한 효과적으로 분포가 변경될 수 있도록 [그림 4]의 규칙에 따라 시점별 허용 토큰 집합을 정의하여 명시적인 확률 변수(Random Variable) 제약을 적용한다. 이는 [수식 1]에서 A_n으로 표현되며, 이를 통해 해당 위치에 올 수 있는 허용 토큰 사이에서만 손실이 계산되도록 하여 학습 신호를 집중시킨다.

$$\mathcal{L}_{CE} = -\frac{1}{N}\sum_{n\in \mathrm{I}} \log \frac{\exp(x_{n,y_n})}{\sum_{c\in \mathrm{A}_n}\exp(x_{n,c})} \quad (1)$$

패치 업데이트에 사용되는 최종 교차 엔트로피 손실은 [수식 1]과 같이 표현되며, 이때 n은 행동 출력의 토큰 인덱스, I는 손실 계산에 포함되는 유효 토큰 인덱스 집합, N은 $|I|$로 유효 토큰 개수를 의미한다. 또한 $x_{n,c}$는 n번째 토큰에서의 어휘(Vocabulary) 토큰 c에 대한 모델의 로짓(Logit)이며, 이때 y_n은 n번째 토큰의 정답 토큰을 나타낸다. 더 나아가 [그림 4]의 규칙에 따라 n번째 인덱스에서 허용되는 토큰 집합을 A_n으로 정의한다.

Stop의 경우 별도 포맷으로 학습되어, 공통 포맷 다음 "stop"을 생성하도록 학습된 적이 없다. 따라서 "The next action is stop"이라는 적대적 목표 행동을 정답으로 할 때엔 [수식 2]와 같이 비우도(Unlikelihood, UL) 손실[9]을 도입해 "move"와 "turn"이 선택될 확률을 억제하도록 학습하였다.

$$\mathcal{L}_{UL} = -\alpha \sum_{c\in\{\mathtt{move},\mathtt{turn}\}} \log\left(1 - p_\theta(c \mid a_{<\mathrm{head}})\right) \quad (2)$$

해당 손실은 행동 헤드 직전까지 생성된 토큰($a_{<head}$)을 조건으로 모델이 "move"와 "turn"을 예측할 확률($p_\theta(\bullet|a_{<head})$)을 계산한 뒤, 학습이 진행될수록 해당 확률이 감소하도록 만든다. 이때 α는 UL 손실에 적용되는 가중치를 뜻하며 본 논문에서는 0.8을 사용하였다.

패치 학습에 사용된 최종 손실은 [수식 3]과 같다.

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{CE} + \mathcal{L}_{UL} \quad (3)$$

2.3 적대적 패치 공격

[섹션 2.2]에서 생성된 패치는 학습과 동일한 위치에 부착되어 모델[3]의 입력으로 사용된다. 이미지 입력 또한 학습과 동일하게 과거 관찰 이미지 7장과 현재 관찰 이미지 1장, 총 8장이 사용된다. 우리는 적대적 패치 공격을 통해 모델[3]이 적대적 목표 행동을 생성하도록 유도한다. 이를 통해 이상적인 주행 경로에서 벗어나도록 하며, 최종적으로 시각-언어 내비게이션 과업의 성공률 하락을 목표로 한다.

3. 실험

3.1 구현 세부 정보

공격에 사용되는 적대적 패치는 모델[3]이 생성하고자 하는 행동과 동일한 출력 포맷을 가진 R2R-CE(Room-to-Room Continuous Environments)[7], RxR-CE(Room-across-Room Continuous Environments)[10] 데이터 집합을 사용해 학습하였다. 평가의 경우 재구성된 실내 장면에서의 연속적 이동을 지원하는 VLN-CE[7] 벤치마크에서 수행되었으며 R2R-CE[7], RxR-CE[10]의 Val-unseen 데이터 집합을 사용하였다.

또한 상세한 공격 성능 분석을 위해 총 네 가지의 평가 지표를 사용한다. NE(Navigation Error)는 목표 지점과 실제 도착 지점 사이의 거리를 뜻하며, OS(Oracle Success rate)는 이상적인 중간 경로를 이용한 성공률, SR(Success Rate)은 목표 지점 3m 이내에 도달한 비율을 의미한다. 더 나아가 성공률(SR)과 실제 경로 효율성을 결합한 점수인 SPL(Success weighted by Path Length)을 통해 주행 성능을 측정한다.

3.2 패치 생성 방식에 따른 공격 성능 비교

본 논문은 적대적 목표 행동을 통해 생성된 패치 성능을 분석하고자 아래 세 가지 패치에 대한 공격 성능을 비교하였다.

- Black Patch: 가림(Occlusion)으로 인한 성능 저하 여부를 검증하기 위한 검정 색상 패치
- Gaussian Init: 단순 노이즈로 인한 성능

표 1 패치 생성 방식에 따른 공격 성능 비교

Metric	NE ↑	OS ↓	SR ↓	SPL ↓
Dataset	R2R-CE Val-Unseen[7]			
NaVILA[3]	5.26	61.4	54.0	49.3
Black Patch	5.53 (+5.1%)	61.0 (-0.7%)	51.2 (-5.2%)	46.4 (-5.9%)
Gaussian Init	5.48 (+4.1%)	60.7 (-1.1%)	51.0 (-5.6%)	46.3 (-6.1%)
Ours	**6.89 (+31.0%)**	**53.0 (-13.7%)**	**38.6 (-28.5%)**	**33.3 (-32.4%)**
Dataset	RxR-CE Val-Unseen[10]			
NaVILA[3]	6.17	61.3	52.1	45.8
Black Patch	6.32 (+2.4%)	59.8 (-2.4%)	50.4 (-3.3%)	44.1 (-3.7%)
Gaussian Init	6.23 (+1.0%)	59.9 (-2.3%)	48.8 (-6.3%)	43.8 (-4.4%)
Ours	**7.67 (+24.3%)**	**50.9 (-17.0%)**	**39.8 (-23.6%)**	**35.6 (-22.3%)**

저하를 확인하기 위해 가우시안 노이즈로 초기화된 패치

- Ours: 제안된 적대적 목표 행동을 통해 생성된 적대적 공격 패치

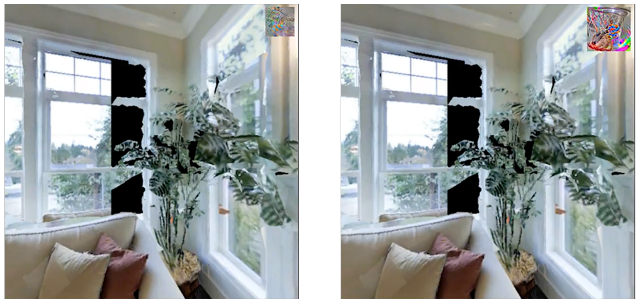
실험 결과는 [표 1]에 제시되어 있으며, 가장 좋은 공격 성능을 굵은 글씨로 표시하였다. 평가 지표에 함께 기재된 화살표는 공격 성공 시의 평가 지표 변화 방향을 의미한다.

먼저 패치 생성 방식에 관계 없이 패치가 부착된 경우 NE는 증가, OS · SR · SPL은 하락 및 감소하여 모든 평가 지표에서 성능 저하를 보였다. 이때 검정 색상 패치를 사용한 경우는 SR이 각각 5.2%, 3.3% 하락하고, SPL이 5.9%, 3.7% 하락하였다. 이를 통해 패치로 인한 가림 현상이 시각 정보의 손실에 영향을 주어 성능 저하를 발생시킴을 확인할 수 있었다.

가우시안 노이즈로 초기화한 경우에는 색상 패치를 사용한 경우보다 더 큰 성능 저하를 보였다. SR은 각각 5.6%, 6.3% 하락하였으며, SPL 또한 6.1%, 4.4% 하락하였다. 이는 별도로 학습되지 않은 단순한 노이즈라도 시각 정보를 차단하는 것보다 모델[3]의 출력에 더 큰 악영향을 미침을 시사한다.

다만, 본 논문에서 제안한 적대적 목표 행동과 손실을 사용해 학습한 패치의 경우 더 큰 폭으로 성능이 하락하였다. SR은 각각 28.5%, 23.6% 하락하였으며, SPL도 32.4%, 22.3% 하락하여 학습되지 않은 패치 대비 약 4-5배 더 강력한 공격 효과를 보였다. 특히 R2R-CE Val-Unseen[7] 데이터 집합의 경우 SR보다 SPL이 더 큰 폭으로 하락하였다. 이는 NE의 증가와 함께, 주행 실패가 증가하였으며 성공하더라도 이상적인 경로에서 더 크게 벗어났음을 의미한다. 더 나아가 OS도 13.7%, 17.0% 하락하여 주행 품질이 크게 떨어졌음을 다시 한 번 확인할 수 있었다.

3.3 패치 크기에 따른 공격 성능 비교



패치 크기 10% (38*38) 패치 크기 15% (57*57)

그림 5. 입력 이미지에 적용된 적대적 패치 크기 비교

[섹션 3.2]의 실험을 통해 학습된 적대적 패치가 효과적임을 확인하였으나, 실제 환경에서의 적용 가능성을 고려할 때 패치의 크기는 매우 중요한 요소이다. 패치가 클수록 공격 효과는 증가할 수 있으나, 지나치게 큰 패치는 눈에 띄기 쉬워 탐지 위험이 높다. 반대로 패치가 너무 작은 경우에는 공격 효과가 미미할 수 있다.

본 연구에서는 입력 이미지 해상도 (384*384) 대비 15%에 해당하는 57*57 크기의 패치를 사용하였으며, 패치 크기가 공격 성능에 미치는 영향을 분석하기 위해 10%에 해당하는 38*38 크기의 패치를 생성하여 추가적인 평가를 진행하였다. 실험 결과는 [표 2]에 제시되어 있으며, [표 1]과 동일하게 가장 높은 공격 성능을 굵은 글씨로 표시하였다.

[표 2]에서 확인할 수 있듯이, 패치 크기가 증가함에 따라 공격 효과도 증가하였다. 패치 크기가 커짐에 따라 R2R-CE Val-Unseen[7] 데이터 집합에

표 2 패치 크기에 따른 공격 성능 비교

Metric	NE ↑	OS ↓	SR ↓	SPL ↓
Dataset	R2R-CE Val-Unseen[7]			
NaVILA[3]	5.26	61.4	54.0	49.3
Ours (10%)	6.39 (+21.5%)	56.2 (-8.5%)	46.8 (-13.3%)	38.7 (-21.5%)
Ours	**6.89 (+31.0%)**	**53.0 (-13.7%)**	**38.6 (-28.5%)**	**33.3 (-32.4%)**
Dataset	RxR-CE Val-Unseen[10]			
NaVILA[3]	6.17	61.3	52.1	45.8
Ours (10%)	6.96 (+12.8%)	57.4 (-6.4%)	46.8 (-10.2%)	38.8 (-15.3%)
Ours	**7.67 (+24.3%)**	**50.9 (-17.0%)**	**39.8 (-23.6%)**	**35.6 (-22.3%)**

표 3 패치 적용 방식에 따른 공격 성능 비교

Metric	NE ↑	OS ↓	SR ↓	SPL ↓
Dataset	R2R-CE Val-Unseen[7]			
NaVILA[3]	5.26	61.4	54.0	49.3
Ours (Ideal)	**6.89 (+31.0%)**	**53.0 (-13.7%)**	**38.6 (-28.5%)**	**33.3 (-32.4%)**
Ours (Init. 10 steps)	6.44 (+22.4%)	54.2 (-11.7%)	45.8 (-15.2%)	40.7 (-17.4%)
Ours (Freq. 8)	5.92 (+12.6%)	58.8 (-4.2%)	50.4 (-6.7%)	45.5 (-7.7%)
Dataset	RxR-CE Val-Unseen[10]			
NaVILA[3]	6.17	61.3	52.1	45.8
Ours (Ideal)	**7.67 (+24.3%)**	**50.9 (-17.0%)**	**39.8 (-23.6%)**	**35.6 (-22.3%)**
Ours (Init. 10 steps)	6.98 (+13.1%)	55.3 (-9.8%)	46.3 (-11.1%)	39.8 (-13.1%)
Ours (Freq. 8)	6.37 (+3.2%)	58.8 (-4.1%)	50.4 (-3.3%)	44.9 (-2.0%)

서의 SR 하락폭이 13.3%에서 28.5%로 증가하였으며, RxR-CE Val-Unseen[10] 데이터 집합에서도 10.2%에서 23.6%로 약 2배 가량 하락폭이 증가하였다. 이는 적대적 패치의 효과가 크기에 민감하게 반응함을 보여주며, 더 큰 크기의 패치일수록 모델을 더 효과적으로 교란할 수 있음을 의미한다.

입력 이미지 해상도의 10%에 해당하는 38*38 크기의 패치 또한 일정 수준의 공격 효과를 보였으나, 15% 대비 약 절반 수준의 성능 저하에 그쳤다. 따라서 본 논문은 공격 성능과 탐지 위험의 균형점을 고려해 15%의 패치 크기를 기본값으로 선택하였다.

3.4 패치 적용 방식에 따른 공격 성능 비교

우리는 적대적 패치 적용 방식에 따른 공격 성능 평가를 위한 추가 실험을 진행하였으며, 이는 [표 3]에 제시되어 있다.

Ours (Ideal)은 앞선 실험에서 Ours로 표기되었던 방법으로, 모든 관찰 프레임에 패치가 부착되는 이상적인 공격 시나리오를 의미한다. Ours (Init. 10 steps)은 출발점 인근의 제한된 구간(약 5-7m 반경) 내에서만 패치가 관찰되는 시나리오를 가정하고, 에이전트가 출발 후 초기 10회의 행동 생성 과정(Step)에서만 패치에 노출되도록 제한한 경우이다. Ours (Freq. 8)은 공격 효율성 평가를 위한 시나리오로, 패치가 8번의 관측 간격을 두고 주기적으로 부착되는 것을 의미한다.

결과적으로 초기 10회의 행동 생성 과정에서만 패치에 노출되도록 한 Ours (Init. 10 steps)에서도 R2R-CE Val-Unseen[7] 데이터 집합에서 NE는 22.4% 증가하였고, SR과 SPL은 각각 15.2%, 17.4% 하락하였다. RxR-CE Val-Unseen[10] 데이터 집합에서도 유사한 패턴이 나타났으며 (NE +13.1%, SR -11.1%, SPL -13.1%) 제한적인 패치 노출만으로도 의미 있는 공격 효과가 발생함을 확인하였다. 이는 초기 주행 단계에서의 잘못된 행동 생성이 이후 전체 경로에 지속적인 영향을 미침을 시사한다.

더 나아가 Ours (Freq. 8)의 경우 모든 평가 지표가 공격 성공을 의미하는 방향으로 변화하였지만, Ideal과 Init. 10 steps 대비 적은 변화율을 보였다. 이는 공격 효율성과 공격 성능 간의 상충

관계(Trade-off)가 존재하며, 공격의 탐지 가능성은 낮추었지만 그만큼 공격 성능 저하가 이루어졌음을 의미한다. 따라서 공격자는 효과적인 적대적 패치를 생성하는 것뿐만 아니라 공격 적용 시점 및 패치 크기 등 다양한 요소를 고려한 공격을 설계해야 하며, 공격 성능과 공격 효율성 및 탐지 가능성의 적절한 균형점을 찾는 것이 중요함을 시사한다.

4. 결론

본 논문은 자연어 행동 기반 시각-언어-행동 모델[3]에서의 적대적 패치 공격을 제안하고, 패치 생성 방식 및 적용 방식에 따른 시각-언어 내비게이션 과업에서의 공격 성능을 분석하였다. 실험 결과 검정 색상 패치와 가우시안 노이즈로 초기화된 패치 또한 과업 성공률 저하에 영향을 미침을 보였으나, 적대적 목표 행동을 통해 생성된 패치가 더 효과적으로 공격을 수행함을 확인하였다.

더 나아가 다양한 패치 적용 방식에 대한 실험을 통해 공격자의 공격 방식에 따른 성능 변화가 있음을 보이고, 이를 고려한 강건한 모델 설계가 필요함을 확인하였다.

우리의 연구를 통해 파운데이션 모델의 발전에도 불구하고 시각-언어 모델을 활용한 과업의 경우 여전히 적대적 공격에 취약함을 알 수 있다. 본 논문은 시각-언어 통합 모델을 활용할 경우 적대적 공격에 대한 방어 연구가 수반되어야 함을 시사한다. 더 나아가 실제 세계에서 물리적으로 동작하며 이동 범위가 넓은 내비게이션 과업의 경우 입력 이미지와 무관하게 동작하는 적대적 패치 공격에 대한 대비가 필수적임을 보인다.

감사의 글

본 연구는 과학기술정보통신부 및 정보통신기획평가원의 디지털분야글로벌연구지원(IITP-2024-RS-2024-00417958)과 2025년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구 결과임 (No.RS-2025-02219317, AI스타펠로우십지원사업(국민대학교)).

참고문헌

[1] Rawal, Niyati, et al. "Aigen: An adversarial approach for instruction generation in vln." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2024).

[2] Qi, Zhangyang, et al. "VLN-R1: Vision-Language Navigation via Reinforcement Fine-Tuning." arXiv preprint arXiv:2506.17221 (2025).

[3] Cheng, An-Chieh, et al. "Navila: Legged robot vision-language-action model for navigation." arXiv preprint arXiv:2412.04453 (2024).

[4] Kim, Moo Jin, et al. "Openvla: An open-source vision-language-action model." arXiv preprint arXiv:2406.09246 (2024).

[5] Goodfellow, Ian J., et al. "Explaining and harnessing adversarial examples." arXiv preprint arXiv:1412.6572 (2014).

[6] Brown, Tom B., et al. "Adversarial patch." arXiv preprint arXiv:1712.09665 (2017).

[7] Krantz, Jacob, et al. "Beyond the nav-graph: Vision-and-language navigation in continuous environments." European Conference on Computer Vision. Cham: Springer International Publishing, (2020).

[8] Athalye, Anish, et al. "Synthesizing robust adversarial examples." International conference on machine learning. PMLR, (2018).

[9] Welleck, Sean, et al. "Neural text generation with unlikelihood training." arXiv preprint arXiv:1908.04319 (2019).

[10] Ku, Alexander, et al. "Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding." arXiv preprint arXiv:2010.07954 (2020).