SAM2 논문 리뷰

"SAM 2: Segment Anything in Images and Videos"

레모네이드

장우진 김선준 장재우 강희준 박은수

2025.10.17

목차

기존 Segmentation Model의 한계

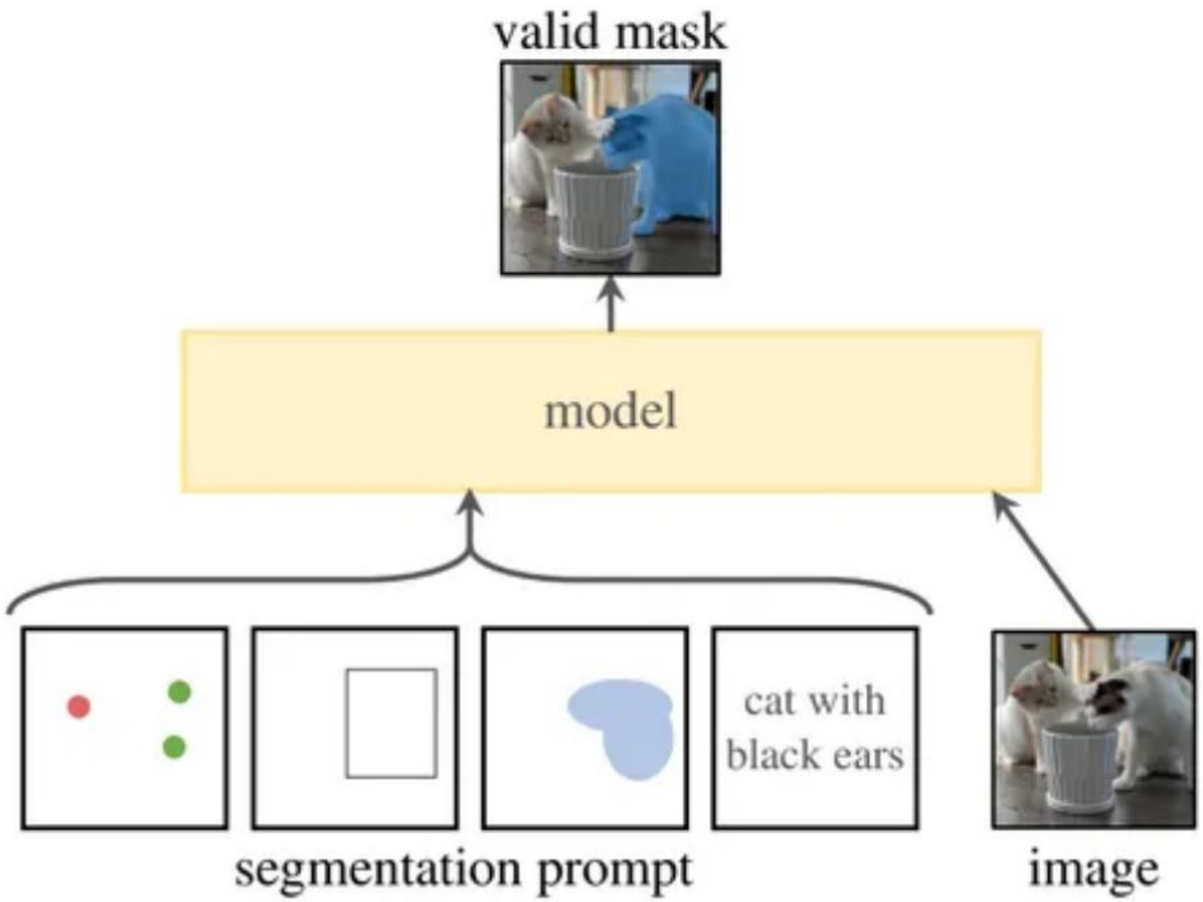
고정된 클래스 기반 분류 → 사전 정의된 클래스만 인식, 새로운 객체 일반화 어려움

대규모 라벨링 의존 → 픽셀 단위 정답(폴리곤/브러시) 수작업 필요, 데이터 구축 비용 과다

도메인 특화 재학습 필요 → 도메인 바뀌면 성능 급락, 파인튜닝/재학습이 필요

상호작용 불가 → 한 번의 추론으로 출력 고정, 피드백 즉시 반영 어려움, 보정하려면 재실행·후처리가 필요

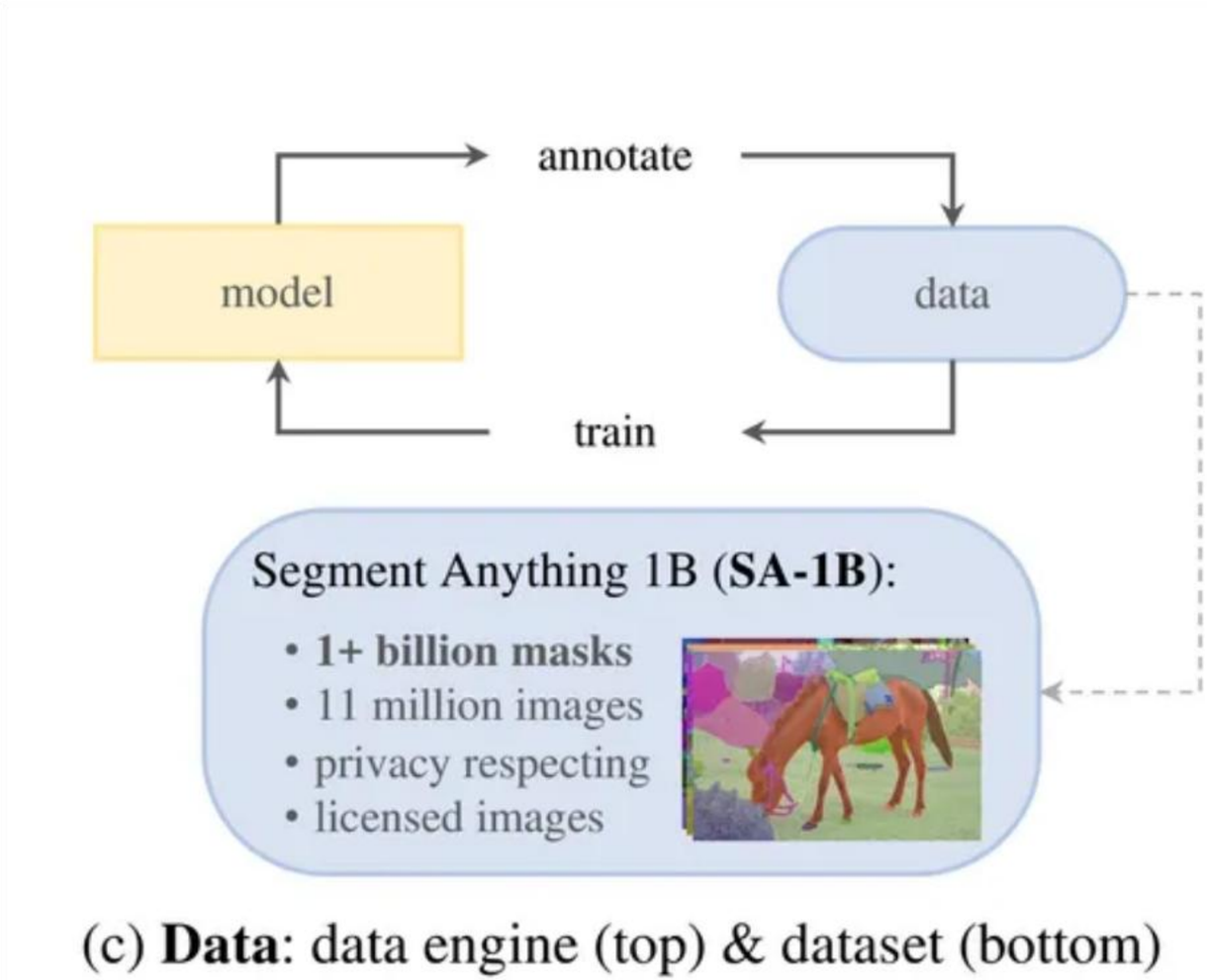
SAM – Task: promptable segmentation



(a) **Task**: promptable segmentation

- 점·박스·마스크 등 프롬프트로 원하는 객체를 지정
- 모델이 그 영역을 자동으로 세그멘테이션
- 클래스 라벨을 없이도, 프롬프트만으로 객체를 구분

SAM – Data: SA-1B

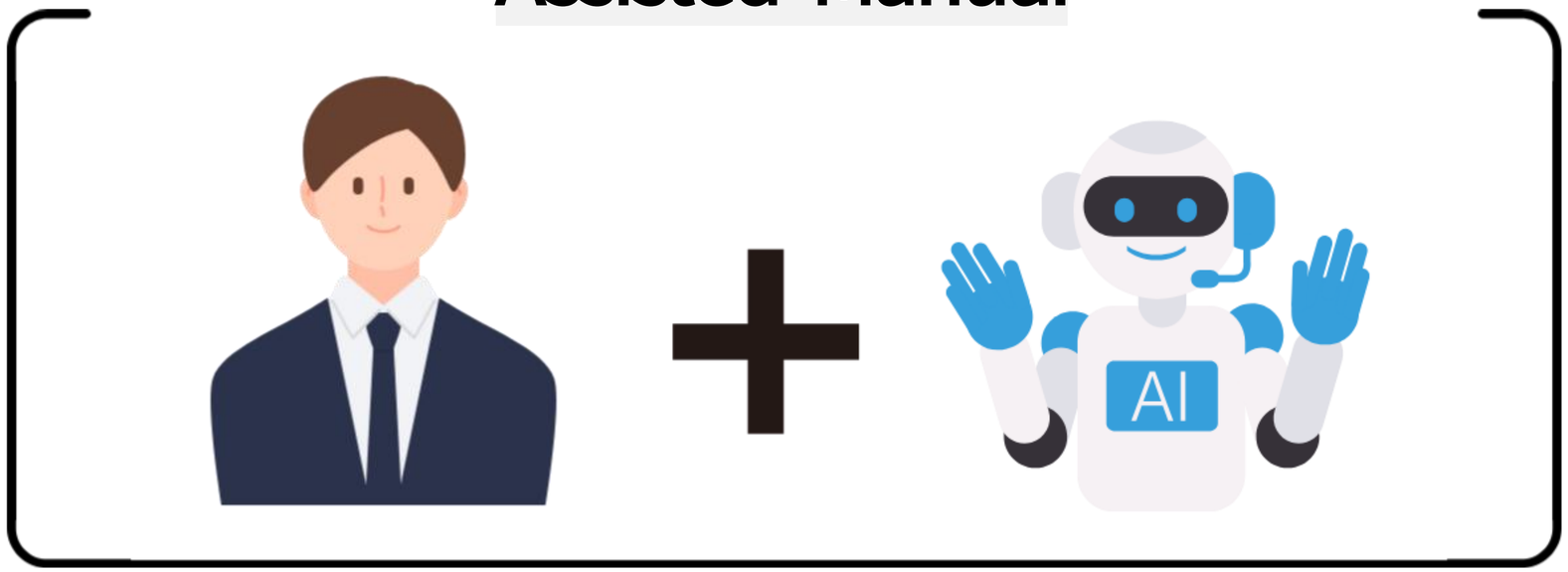


(c) **Data**: data engine (top) & dataset (bottom)

- SAM은 SA-1B (11M 이미지에서 1B+ 마스크) 데이터셋으로 학습됨
- 데이터 엔진 3단계: (1) Assisted-Manual → (2) Semi-Automatic → (3) Fully-Automatic
- 다양한 대규모 데이터로 Zero-shot 일반화 성능 향상

SAM – Data Engine

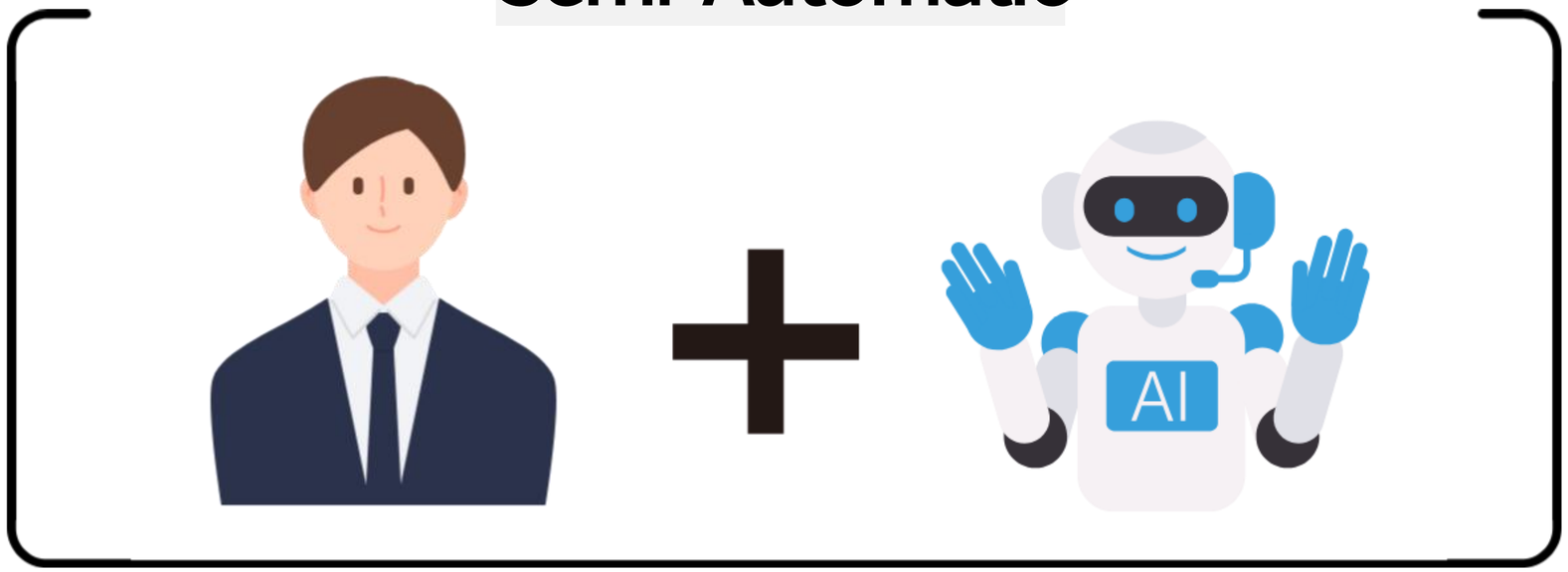
Assisted-Manual



- 사람이 점이나 박스를 찍으면 모델이 바로 후보 마스크를 제안
- 사람은 그중 맞는 걸 고르고 조금 고쳐서 정답을 생성
- 이렇게 만든 데이터로 모델을 주기적으로 다시 학습으로 성능 상승

SAM – Data Engine

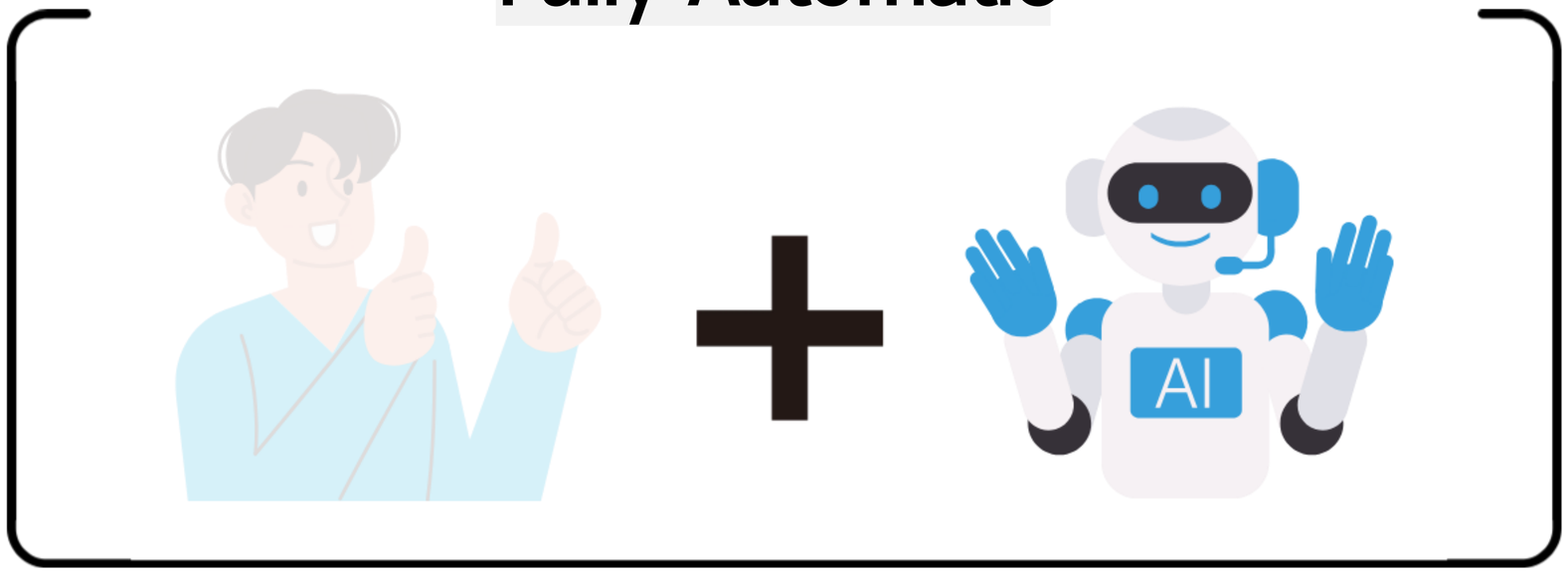
Semi-Automatic



- 모델이 먼저 많은 후보 마스크를 자동으로 뽑고, 사람은 빠르게 승인이나 부족한 부분만 보완
- 속도와 커버 범위가 크게 늘고, 늘어난 데이터로 모델을 또 업데이트

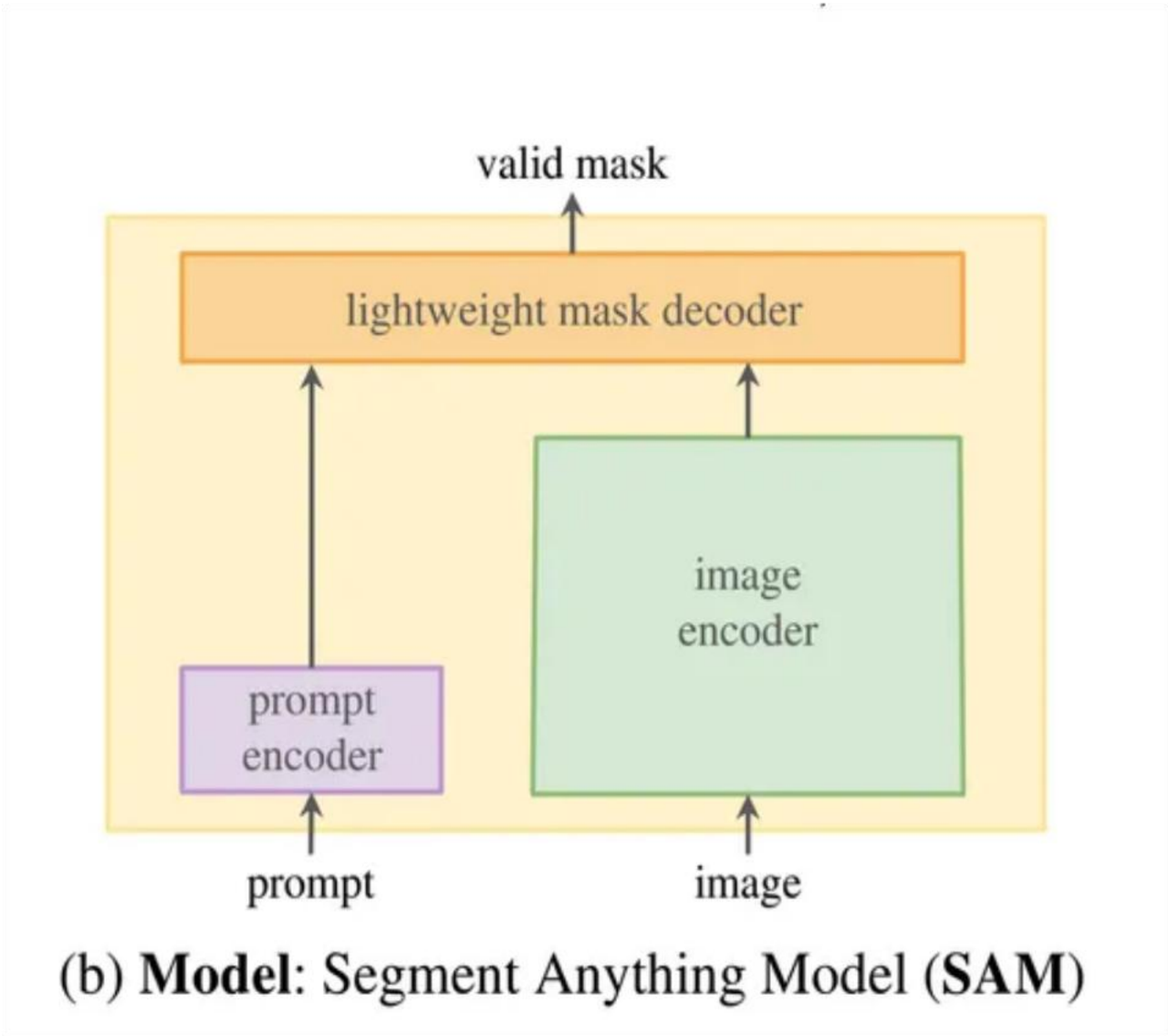
SAM – Data Engine

Fully-Automatic



- 사람 개입 없이 진행, 모델이 계산한 품질 점수(예: IoU 예측), 안정성 검사, 중복 제거로 좋은 마스크만 채택
- 이렇게 모인 마스크를 다시 학습에 써서 루프를 계속 강화한다.

SAM – Model 구조

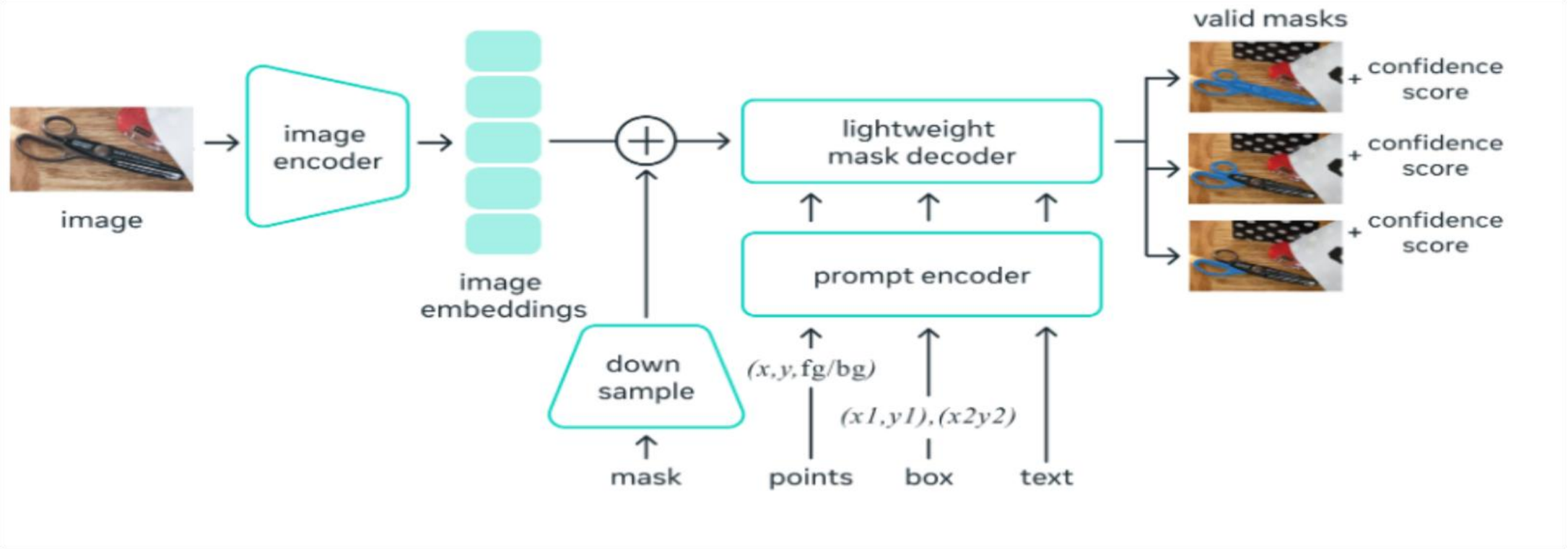


(b) **Model**: Segment Anything Model (**SAM**)

- 구성: Image Encoder, Prompt Encoder, Mask Decoder로 이루어진 구조

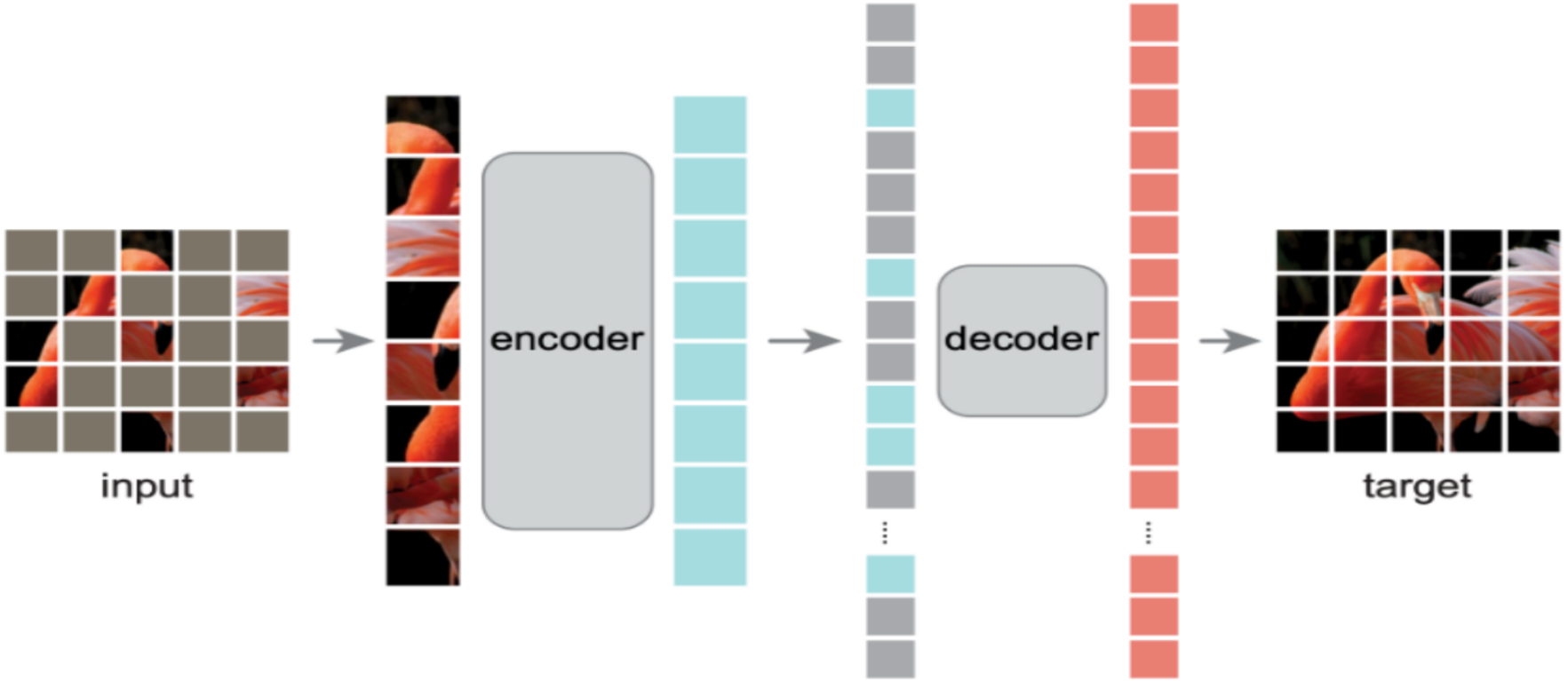
- 특징 : 프롬프트만 바꿔 각 객체를 순차 세그멘테이션 가능

SAM - Architecture



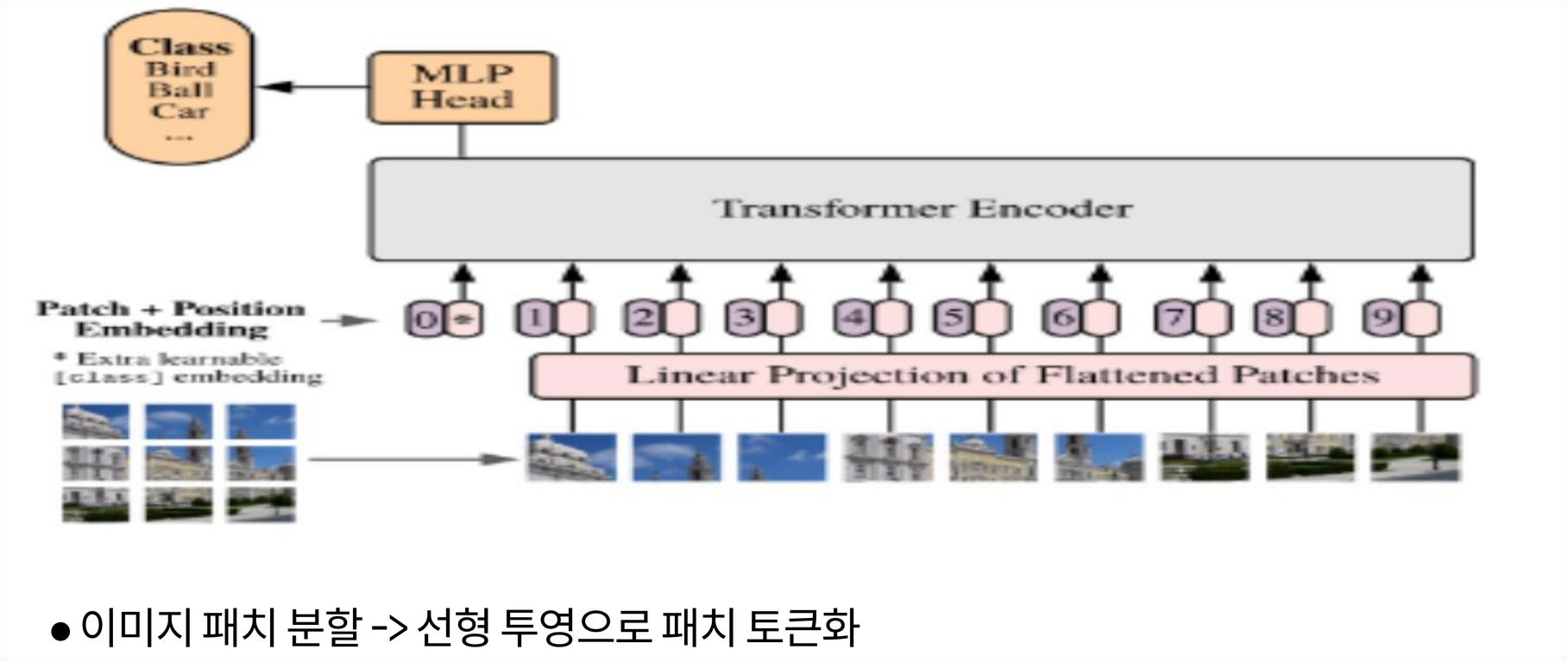
- 이미지 → 이미지 인코더 한 번 통과 이미지 임베딩 → 고정 이미지 임베딩 저장
- 점/박스/마스크 같은 프롬프트 → 프롬프트 인코더로 프롬프트 임베딩 생성
- 마스크 디코더가 이미지 임베딩 + 프롬프트 임베딩을 받아 마스크 출력

MAE (Masked Auto Encoder)



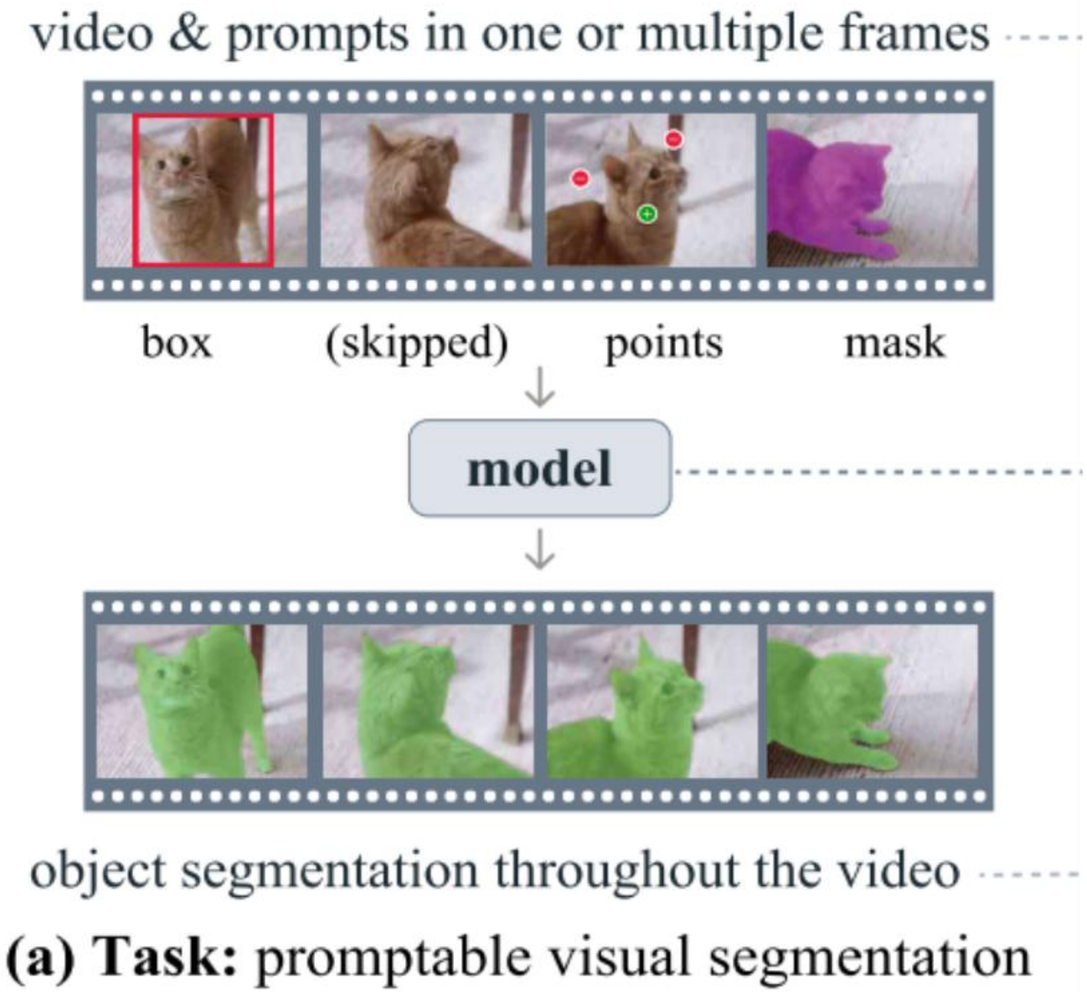
- 이미지의 많은 패치를 가리고(마스킹) 보이는 패치만 인코더에 넣은 뒤, 디코더로 가려진 패치를 복원하도록 학습시키는 자기지도 학습
- 이렇게 학습된 ViT 가중치를 SAM의 이미지 인코더 초기값으로 사용

MAE로 사전학습된 ViT



- 이미지 패치 분할 -> 선형 투영으로 패치 토큰화
- 토큰에 위치 임베딩을 더해 입력 시퀀스 구성
- 시퀀스를 ViT 인코더(어텐션+MLP 여러 층)에 통과
- 출력 토큰을 2D 격자로 재배열해 다중해상도 특징맵을 만들고,
  이를 프롬프트 임베딩과 함께 마스크 디코더로 전달

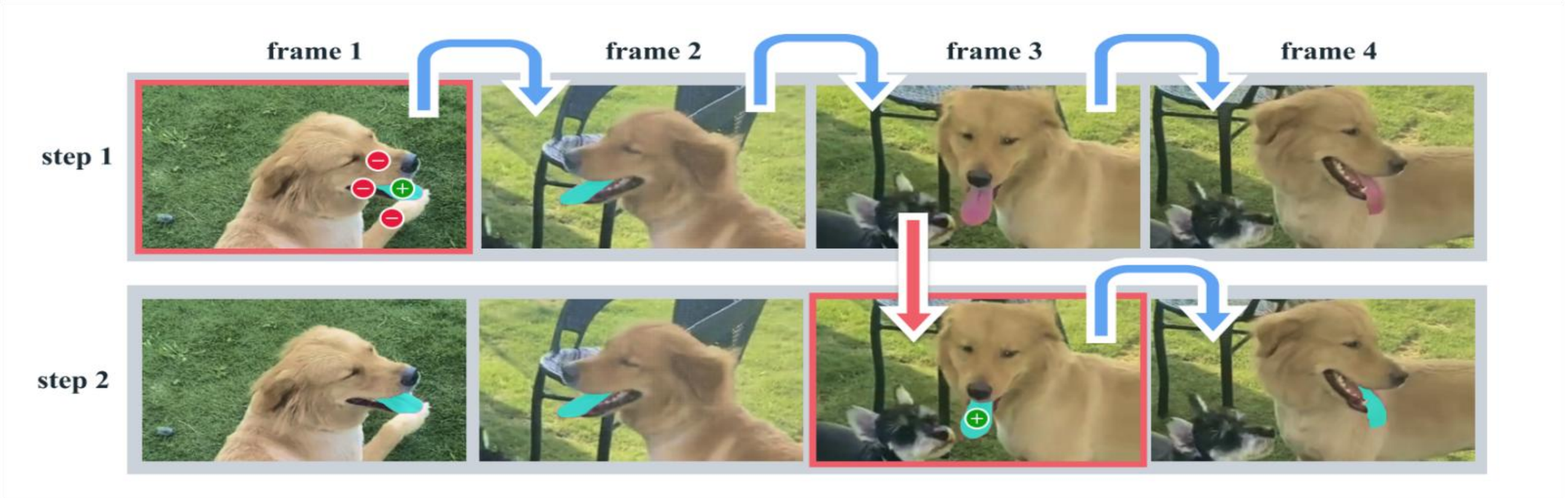
SAM2 – Task: Promptable Video Segmentation



(a) Task: promptable visual segmentation

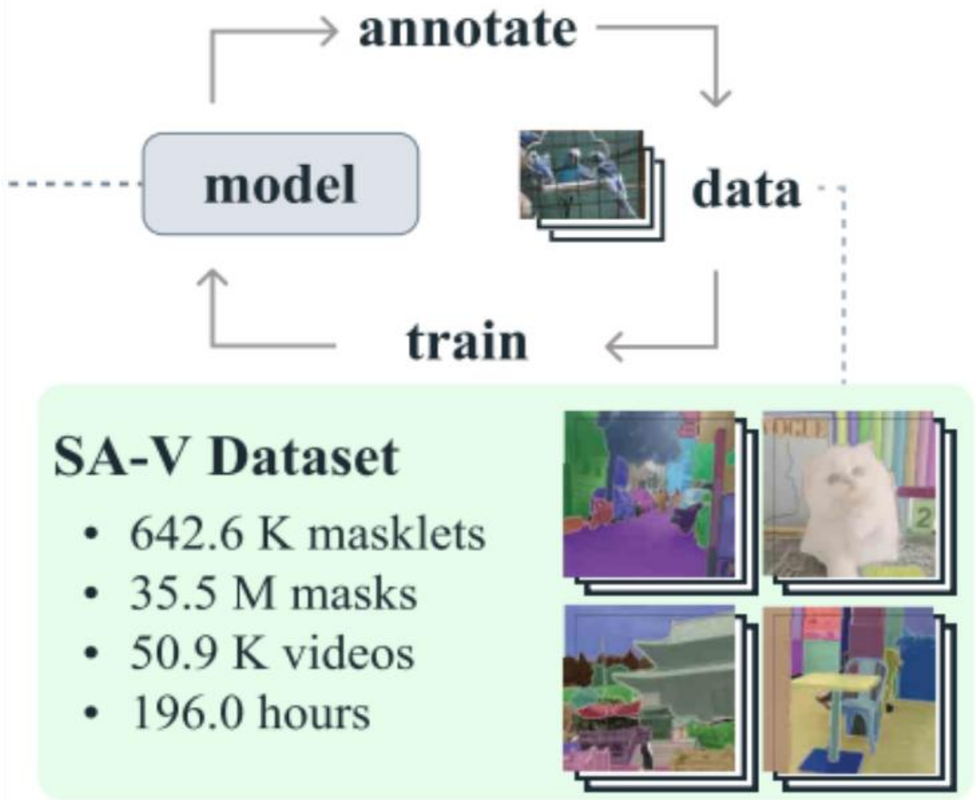
- 영상의 한두 프레임에서
  점·박스·마스크로 원하는 객체를 프롬프트
- 모델이 메모리로 프레임 간 정보를 유지하며 전
  구간에 마스크 전파
- 클래스 라벨 없이 프롬프트만으로 동일 객체를
  영상 전체에서 분할

SAM2 - 세그멘테이션 과정



- 첫 프레임에 박스/점(긍·부)/마스크로 대상 지정
- 초기 마스크 생성 후 메모리 뱅크에 저장 → 다음 프레임들로 자동 전파
- 틀린 프레임 한 곳에 점/박스를 추가해 즉시 수정 → 메모리 갱신 → 이후 프레임에 자동 재전파

SAM2 – Data: SA-V & Data Engine



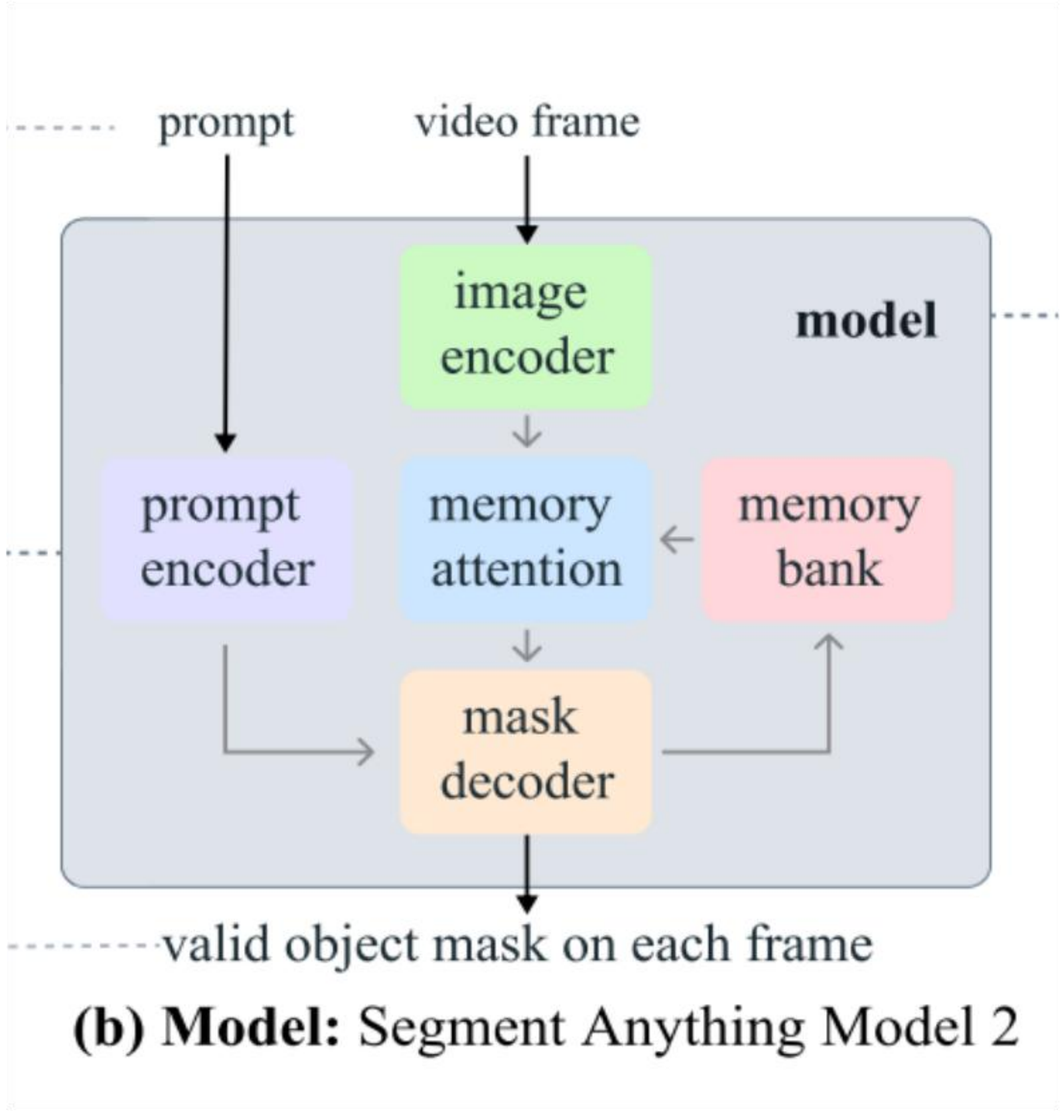
(c) **Data:** data engine and dataset

- SA-V : 마스크릿(masklet): 약 64만 2,600개
  마스크: 약 3,550만 개
  비디오: 약 5만 900개
  길이: 약 196시간

- 데이터 엔진 3단계: (1) Assisted-Manual →
- (2) Semi-Automatic → (3) Fully-Automatic

- 대규모 · 다양한 비디오로 학습해 장면 · 속도 변화와 가림에도 강한 제로샷 일반화를확보

SAM2 – Model: Image/Prompt/Mask + Memory

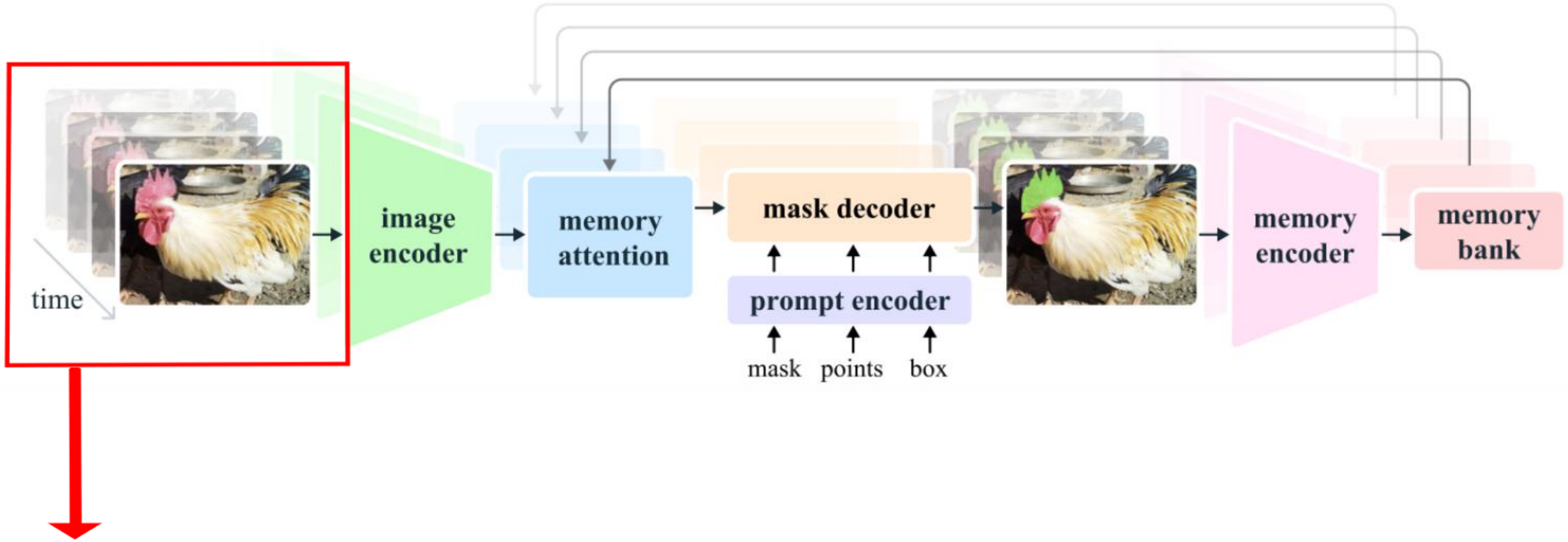


(b) Model: Segment Anything Model 2

- 구성 : Image Encoder(Hiera) + Prompt Encoder + Mask Decoder + Memory Attention & Memory Bank

- 다중해상도 특징 확보해 경계·디테일을
  탄탄 모호함·가림 상황에서도 안정적으로 선택

SAM2 Architecture



이미지 입력

- 원본 RGB 이미지를 정사각 리사이즈 1024×1024 (float32 /255.0)
- 채널별 정규화: img = (img - pixel_mean) / pixel_std (BxCxHxW)

  image_mean: [0.485, 0.456, 0.406]

  image_std: [0.229, 0.224, 0.225]

SAM2 Architecture

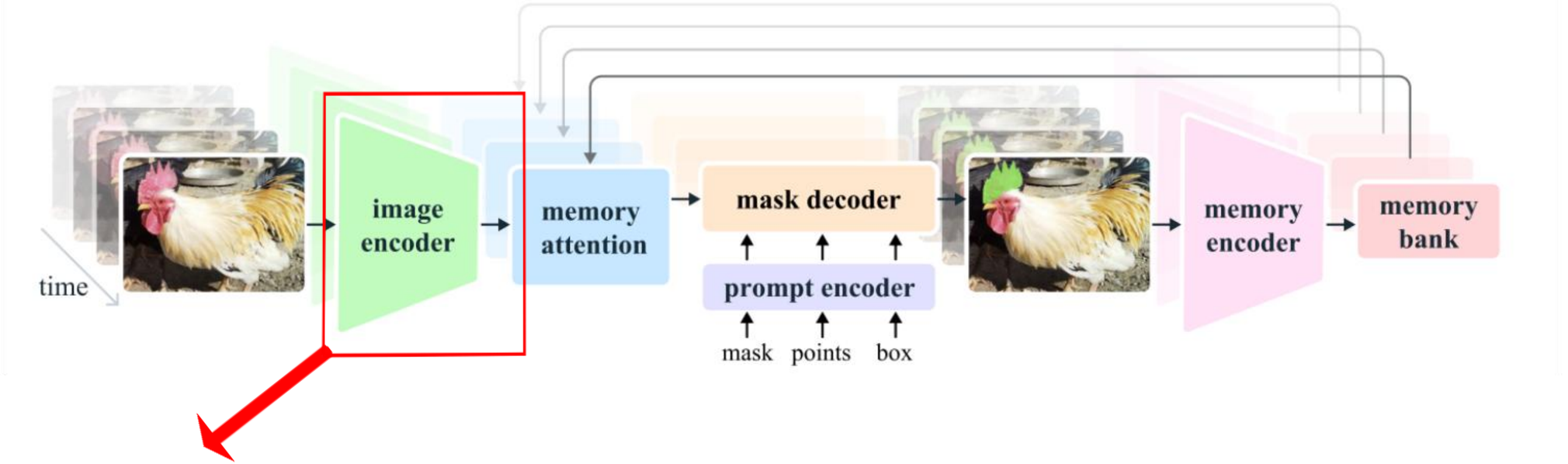
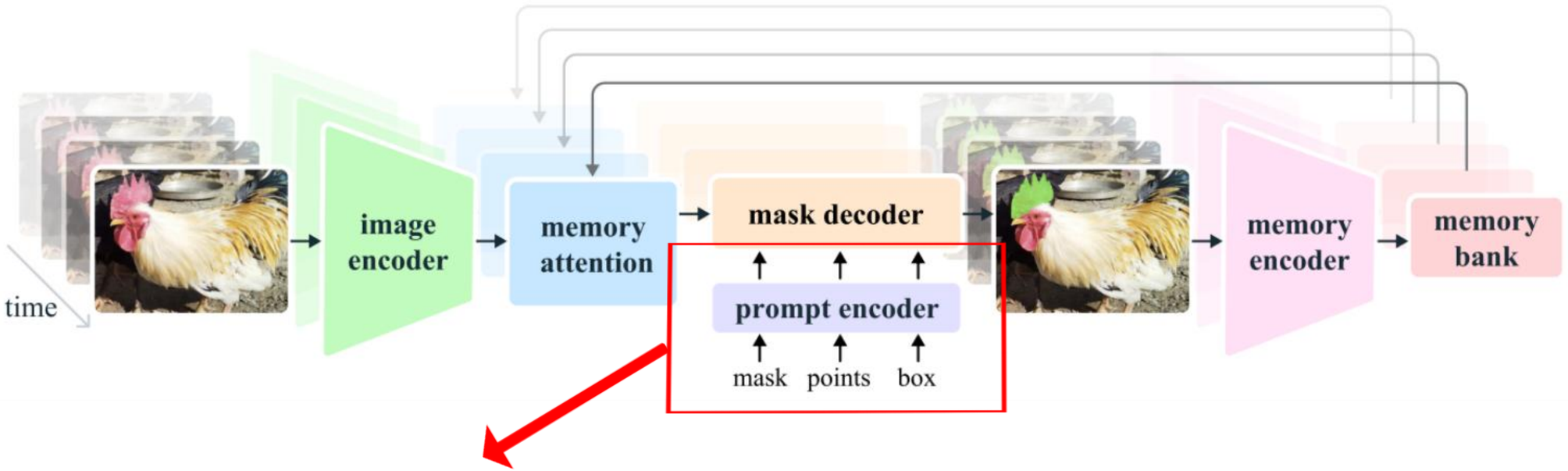


Image Encoder

- Hiera 백본 - 이미지 패치→토큰으로 만들고 위치 임베딩을 더함
- 윈도우 Self-Attention 반복, 블록 그룹 사이 Q-Pooling으로 토큰 수 줄여 계층/다중해상도 형성
- 중간중간 Global Attention으로 전역 문맥 보강
- 클래스 토큰 없음 → 마지막 토큰을 2D 격자 특징맵으로 재배열해 이미지 임베딩 출력

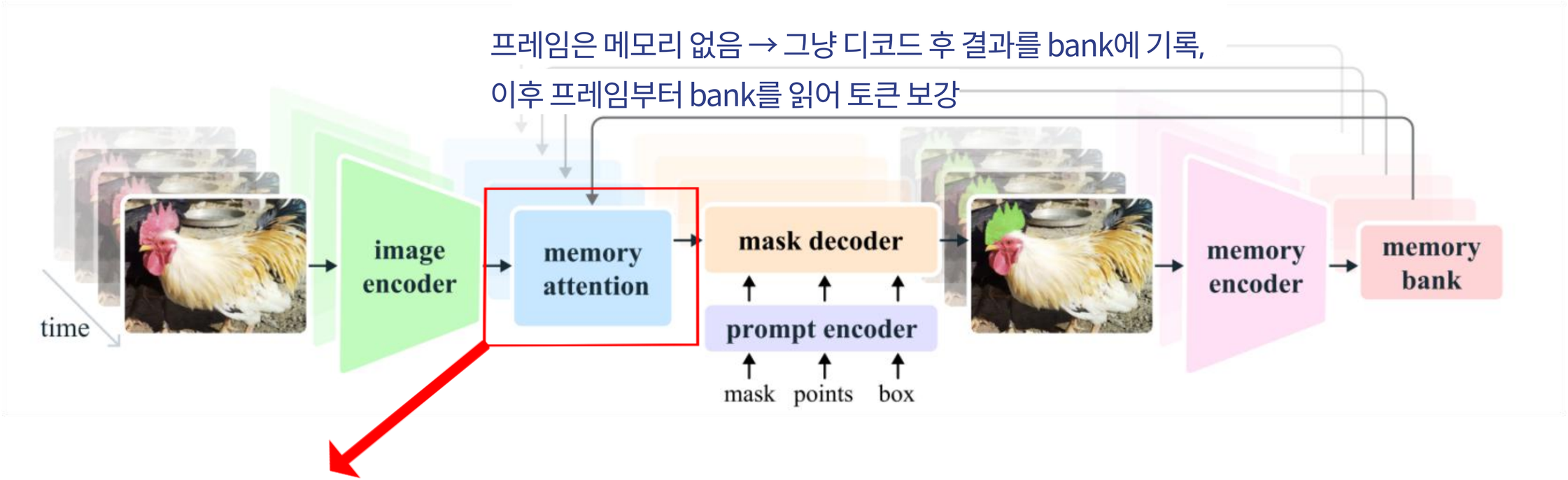
SAM2 모델 구조



Prompt Encoder

- 사람이 준 점·박스·마스크 입력 수집
- 점/박스 → 토큰화 (좌표+간단 임베딩) / 마스크 → 저해상도 특징맵으로 변환
- 만든 토큰과 특징맵을 정리해서 마스크 디코더로 전달

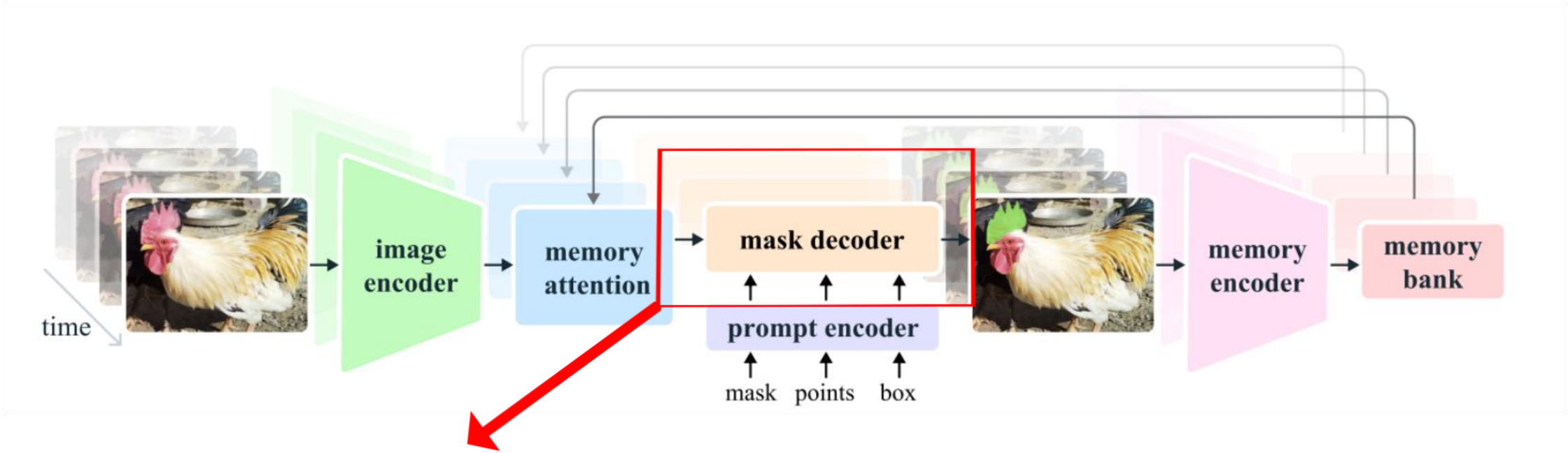
SAM2 모델 구조



Memory Attention

- 현재 프레임 임베딩이 메모리 뱅크(과거 프레임)를 어텐션으로 조회해 결합
- 첫 프레임은 메모리 없음

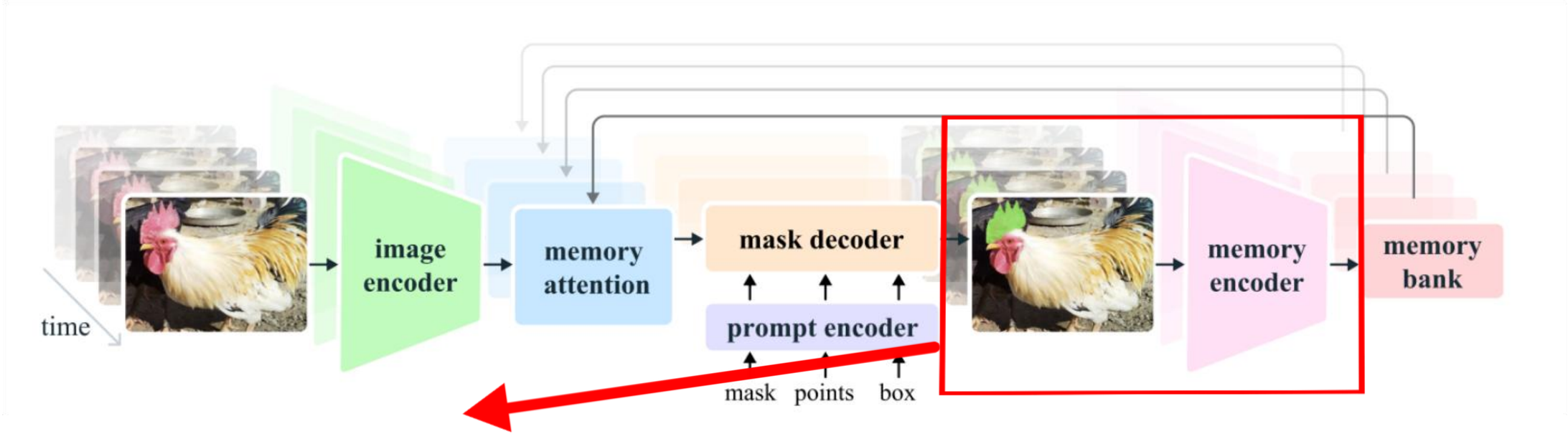
SAM2 모델 구조



Mask Decoder

- 프롬프트 토큰을 쿼리로 써서 이미지 임베딩 + (메모리에서 온) 특징을 조회
- 마스크(여러 후보)와 점수를 뽑아 현재 프레임 결과 생성

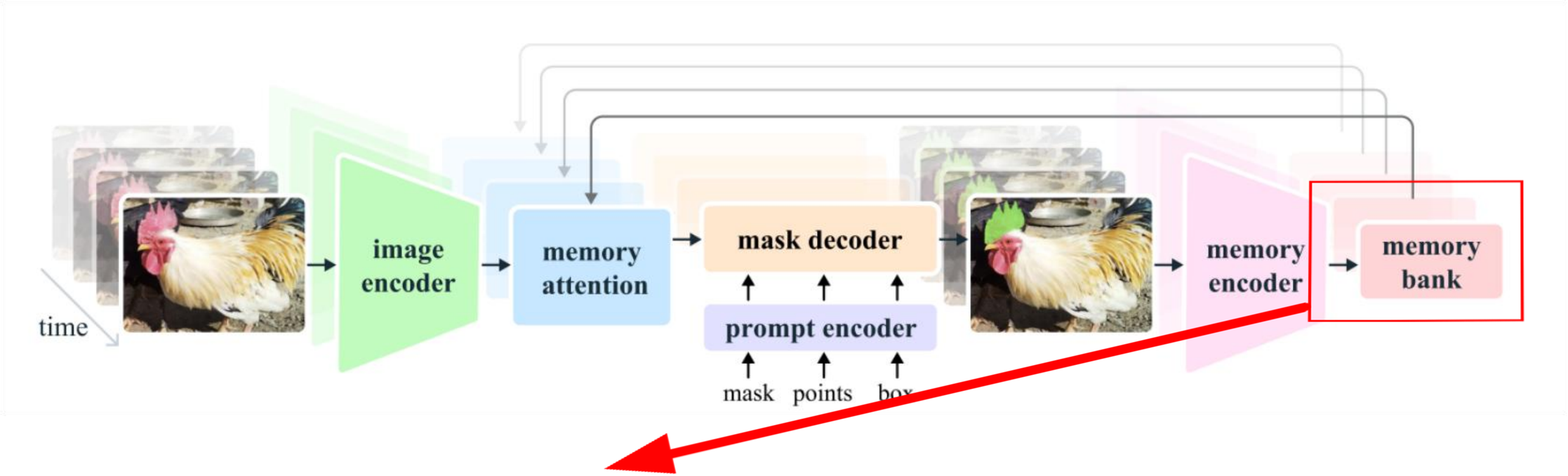
SAM2 모델 구조



Memory Encoder

- 방금 얻은 유효 마스크 + 이미지 임베딩을 요약해 다음 프레임을 위해 기록할 표현으로 만듦

SAM2 모델 구조



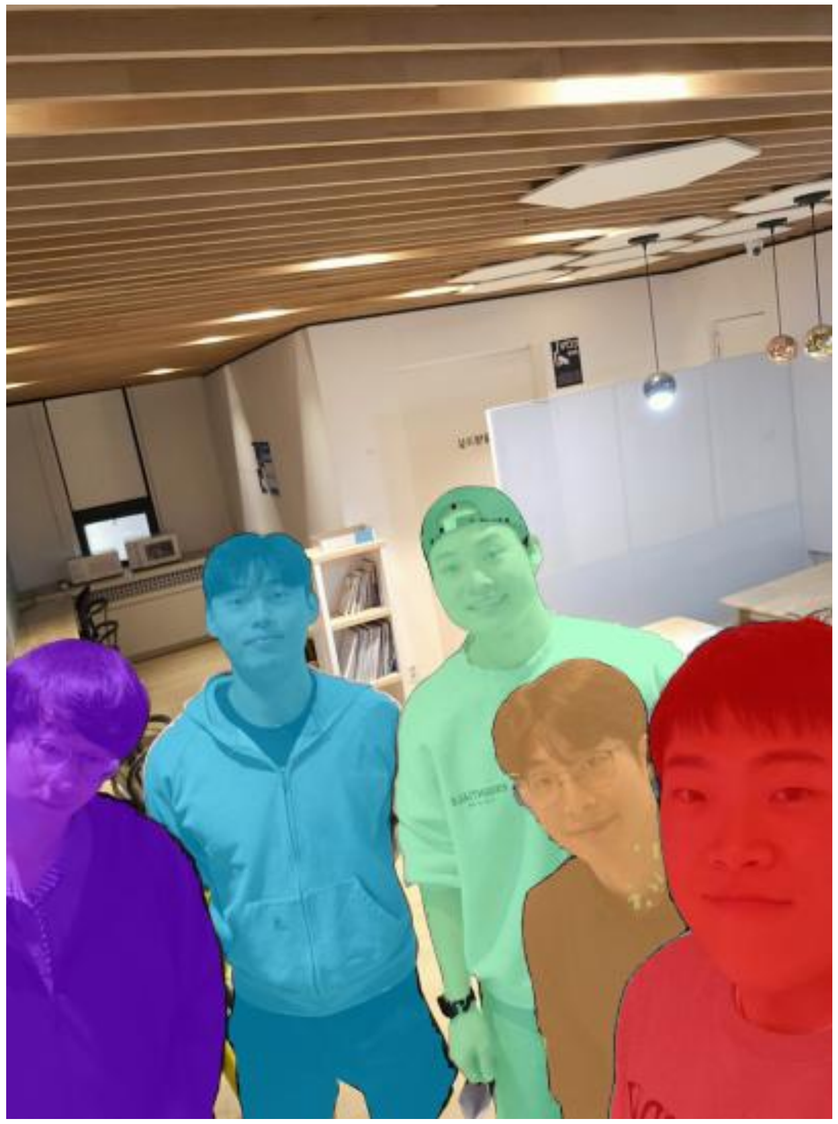
Memory Bank

- 각 프레임의 요약 K/V를 시간 순으로 저장/관리
- 다음 프레임에서 Memory Attention이 읽어 마스크를 전파·보강

코드 구현

```python
class SAM2MultiPersonSegmentation:
    def __init__(self, checkpoint_path):
        """
        SAM2 기반 다중 인물 세그멘테이션

        Args:
            checkpoint_path: SAM2 체크포인트 경로
        """
        self.device = "cuda" if torch.cuda.is_available() else "cpu"
        print(f"Using device: {self.device}")

        # SAM2 Predictor 초기화 (간단한 방법)
        self.predictor = SAM2ImagePredictor.from_pretrained(
            checkpoint_path,
            device=self.device
        )
        print("SAM2 모델 로드 완료!")

    def segment_auto(self, image_path, num_people=5):
        """
        그리드 기반 자동 세그멘테이션 (개선 버전)

        Args:
            image_path: 입력 이미지 경로
            num_people: 추출할 인물 수

        Returns:
            masks: 개별 마스크 리스트
            image: 원본 이미지
        """
        # 이미지 로드
        image = cv2.imread(image_path)
        image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

        self.predictor.set_image(image)

        print("자동 세그멘테이션 실행 중...")

        # 더 촘촘한 그리드 생성 (사람 위치에 집중)
        h, w = image.shape[:2]
```

코드 결과



코드 결과

References

Ravi, N., Gabeur, V., Hu, Y.-T., Hu, R., Ryali, C., Ma, T., et al. (2024).
Segment Anything in Images and Videos (SAM2).
arXiv preprint arXiv:2408.00714.

Ryali, C., Bolya, D., Dai, X., Feichtenhofer, C. (2023).
Hiera: A Hierarchical Vision Transformer without the Bells-and-Whistles.
arXiv preprint arXiv:2306.00989.

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., Dollár, P., & Girshick, R. (2023). Segment Anything. arXiv preprint arXiv:2304.02643.

감사합니다