End-to-End Multi-Task Learning with Attention

#	1
Github	https://github.com/lorenmt/mtan
Read	☐
Review	☐
URL	https://arxiv.org/abs/1803.10704

End-to-End Multi-Task Learning with Attention

0. Abstract

Multi-Task Attention Network(MTAN)

- 단일 공유 네트워크와 소프트-어텐션 모듈로 이루어진 학습 구조
- 멀티 태스크 학습 방법론 중에서는 SOTA

1. Introduction

CNN의 문제점

- 하나의 특정 태스크만 해결하는 데에 적합
- 메모리, 추론 속도 및 데이터 차원에서도 비효율적임

MTL의 주요 쟁점들

- 네트워크 구조(공유 방법) : 각 태스크 간 공통된 피쳐를 공유함과 동시에 특징적 피쳐들을 다 학습할 수 있어야 함
- 손실 함수(태스크 간 균형 잡기) : 모든 태스크들을 비슷한 중요도로 학습할 수 있어야 함

→ MTAN은 위 쟁점들을 모두 해결 가능

MTAN의 구조

- 공유 네트워크에서는 모든 피쳐들을 학습 후, 어텐션 마스크에서 각 태스크에 대한 피쳐들의 중요도를 결정하게 됨

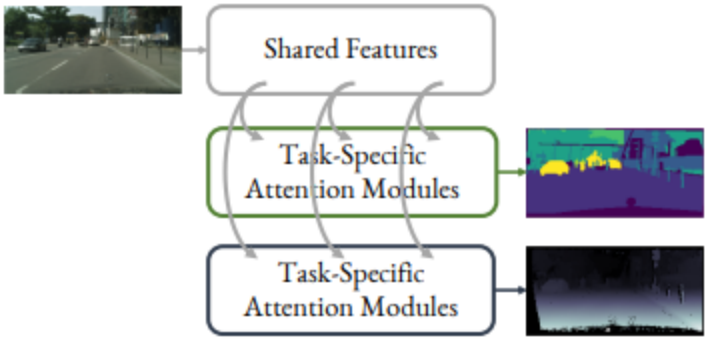


Figure 1: Overview of our proposal MTAN. The shared network takes input data and learns task-shared features, whilst each attention network learns task-specific features, by applying attention modules to the shared network.

- 각 태스크 별로 구조를 만드는 것보다 적은 수의 파라미터로 각 태스크 별 피쳐를 알 수 있게 됨
- 평가방법
  - SegNet : 시멘틱 세그멘테이션, 깊이 예측, 표면 법선 예측
  - Wide Residual Network : 이미지 분류 작업
- **Dynamic Weight Average(DWA)** : 각 태스크별 손실값의 변화율을 참고하여 가중치 적용

2. Related Works

MTL의 흐름

- 어떻게 좋은 MT 네트워크 구조를 디자인 할 수 있을까?
  - 지금까지의 CV분야의 주요 MTL 네트워크는 CNN기반 : 파라미터 수가 매우 많아지며, 태스크의 개수와 구조의 크기가 선형적으로 증가하는 문제

- MTAN은 태스크 당 10% 정도의 증가만 요구

- 어떻게 MTL에서 모든 태스크를 아우르는 피쳐 공유를 균형있게 할 수 있을까?
  - 대부분의 연구에서는 태스크별로 다른 공유와 가중치가 제일 적합하다고 함

3. Multi-Task Attention Network

3.1 Architecture Design

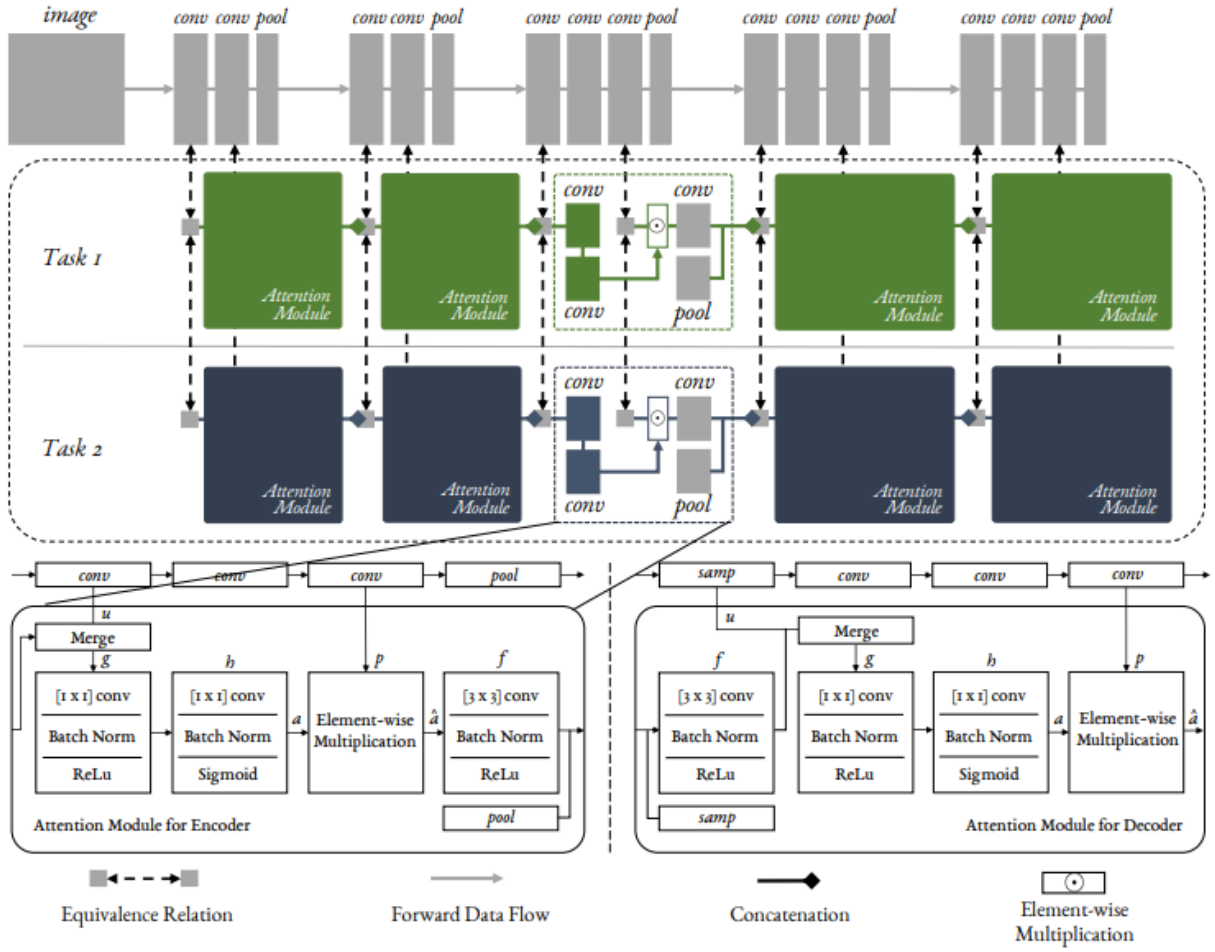


Figure 2: Visualisation of MTAN based on VGG-16, showing the encoder half of SegNet (with the decoder half being symmetrical to the encoder). Task one (green) and task two (blue) have their own set of attention modules, which link with the shared network (grey). The middle attention module has its structure exposed for visualisation, which is further expanded in the bottom section of the figure, showing both the encoder and decoder versions of the module. All attention modules have the same design, although their weights are individually learned.

- 구조 : 단일 공유 네트워크 + K개의 태스크 특정 어텐션 네트워크
- 각 어텐션 마스크들은 공유 네트워크에서 피쳐를 선택하는 역할을 하는 것으로 고려됨

- 공유된 피쳐들의 일반화를 극대화하며 어텐션 마스크로 태스크 특정 성능을 보임

3.2 Task Specific Attention Module

- 각 피쳐 채널 당 하나의 어텐션 마스크 존재
- 태스크 특정 피쳐 = 학습된 어텐션 마스크 * 공유된 피쳐

$$\hat{a}_i^{(j)} = a_i^{(j)} \odot p^{(j)}$$

- 어텐션 마스크의 구조

$$a_i^{(j)} = h_i^{(j)}\left(g_i^{(j)}\left(\left[u^{(j)}; f^{(j)}\left(\hat{a}_i^{(j-1)}\right)\right]\right)\right), j \geq 2$$

  - f, g, h : 비선형 활성화 함수 후 이어지는 배치 정규화가 적용된 convolutional layer
  - g, h는 [1x1] 커널, f는 [3x3] 커널로 이루어짐
- 어텐션 마스크는 시그모이드 함수로 [0,1] 사이의 값으로 역전파가 이루어짐
- 만약 1의 값이 나오면, 모든 태스크들은 모든 피쳐값을 공유함
- 오직 마지막 단계에서만 개별 태스크로 나뉘는 공유 멀티태스크 네트워크보다 성능이 괜찮을 것임을 예상

3.3 The Model Objective

- 손실 함수

$$\mathcal{L}_{tot}(\mathbf{X}, \mathbf{Y}_{1:K}) = \sum_{i=1}^{K} \lambda_i \mathcal{L}_i(\mathbf{X}, \mathbf{Y}_i).$$

- 시멘틱 세그멘테이션

$$\mathcal{L}_1(\mathbf{X}, \mathbf{Y}_1) = -\frac{1}{pq}\sum_{p,q} \mathbf{Y}_1(p,q) \log \hat{\mathbf{Y}}_1(p,q)$$

- 깊이 예측

$$\mathcal{L}_2(\mathbf{X}, \mathbf{Y}_2) = \frac{1}{pq} \sum_{p,q} |\mathbf{Y}_2(p,q) - \hat{\mathbf{Y}}_2(p,q)|.$$

- 법선 예측

$$\mathcal{L}_3(\mathbf{X}, \mathbf{Y}_3) = -\frac{1}{pq} \sum_{p,q} \mathbf{Y}_3(p,q) \cdot \hat{\mathbf{Y}}_3(p,q).$$

- 분류 태스크 : 표준 cross-entropy loss 적용

4. Experiments

4.1 Image-to-Image Prediction (One-to-Many)

- SegNet 기반으로 구축된 MTAN 평가

4.1.1 Datasets

- CityScapes
  - 고해상도 거리뷰 이미지로 구성
  - 시멘틱, 깊이 예측 태스크
  - [128X256] 의 크기로 조정됨
- NYUv2.
  - RGB-D 실내 이미지
  - 시멘틱, 깊이 예측, 법선 예측
  - [288X384] 의 크기로 조정됨

2-class	7-class	19-class
background	void	void
	flat	road, sidewalk
	construction	building, wall, fence
	object	pole, traffic light, traffic sign
	nature	vegetation, terrain
	sky	sky
foreground	human	person, rider
	vehicle	carm truck, bus, caravan, trailer, train, motorcycle

Table 1: Three levels of semantic classes for the CityScapes data used in our experiments.

4.1.2 Baseline

- SegNet 기반의 5개의 네트워크 구조와 비교

    - 이는 MTAN이 다른 게 아닌, 어텐션 모듈로 인해 성능이 향상되었다는 것을 보여주기 위함
- **Single-Task, One Task** : Vanilla SegNet
- **Single-Task, STAN** : single task + MTAN 구조
- **Multi-Task, Split(Wide, Deep)** : 마지막 레이어를 쪼개 각 태스크에 적용시키는 표준 MTL
    - Wide : MTAN의 convolutional 필터의 개수 적용
    - Deep : MTAN의 convolutional 레이어 개수 적용
- **Multi-Task, Dense** : 어텐션 모듈이 없는 공유 네트워크 + 태스크 특정 네트워크 구조
- **Multi-Task, Cross-Stitch** : SegNet에 적용된 Cross-Stitch Networ

4.1.3 Dynamic Weight Average

- 각 태스크의 손실 변화율을 고려하여 점차 가중치를 평균화 함
- GradNorm은 네트워크의 초기 기울기를 요구하는 반면, DWA는 손실 수치만 필요하기에 더 적용하기에 간단함

$$\lambda_k(t) := \frac{K \exp(w_k(t-1)/T)}{\sum_i \exp(w_i(t-1)/T)}, w_k(t-1) = \frac{\mathcal{L}_k(t-1)}{\mathcal{L}_k(t-2)}$$

4.1.4 Results on Image-to-Image Predictions

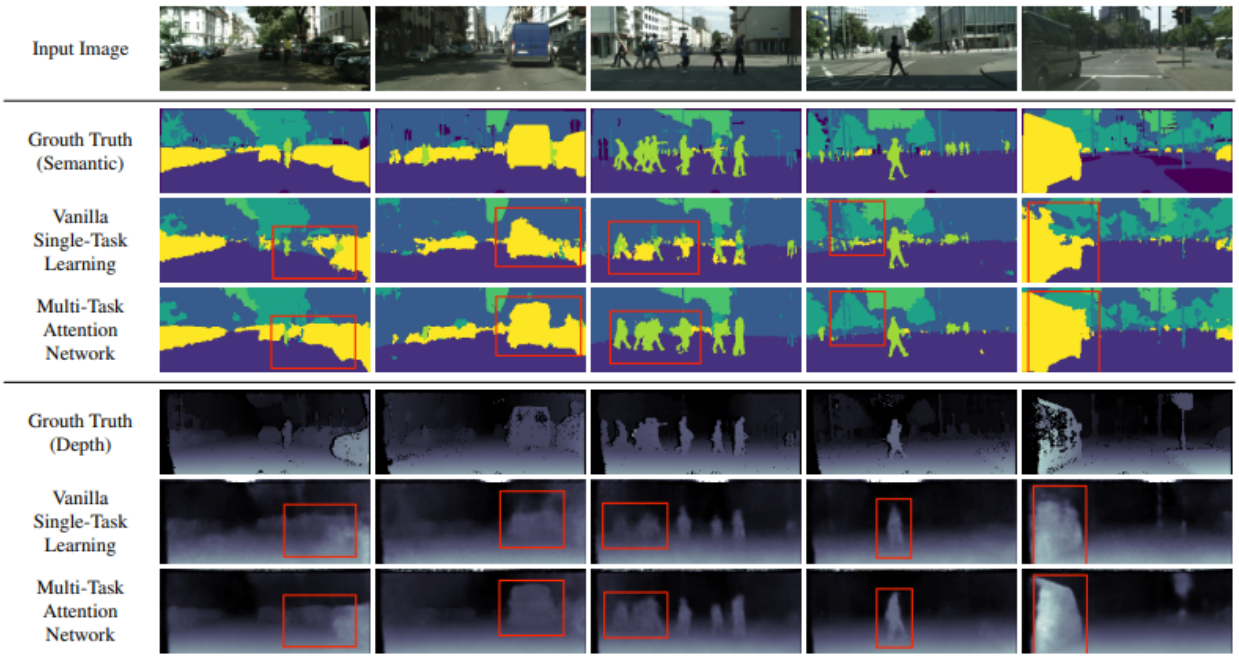
- Training
    - 가중치 방법론 : equal weighting, weight uncertainty, DWA
    - GradNorm 제외
    - Adam 옵티마이저 적용
    - RL : 10^-4
    - Batch size : 2 for NYU, 8 for CityScapes
- Result

- 효율적이어 추가 매개변수(#P) 없이도 다른 방법론을 능가하는 성능
- 여러 손실함수 가중치 방법론에 걸쳐 높은 성능 유지하여 어느 손실함수를 선택해야 할지 깊게 고려하지 않아도 됨

#P.	Architecture	Weighting	Segmentation (Higher Better) mIoU	Segmentation (Higher Better) Pix Acc	Depth (Lower Better) Abs Err	Depth (Lower Better) Rel Err
2	One Task	n.a.	51.09	90.69	0.0158	34.17
3.04	STAN	n.a.	51.90	90.87	0.0145	27.46
1.75	Split, Wide	Equal Weights	50.17	90.63	0.0167	44.73
		Uncert. Weights [14]	**51.21**	**90.72**	**0.0158**	44.01
		DWA, $T = 2$	50.39	90.45	0.0164	**43.93**
2	Split, Deep	Equal Weights	**49.85**	88.69	0.0180	43.86
		Uncert. Weights [14]	48.12	88.68	**0.0169**	**39.73**
		DWA, $T = 2$	49.67	**88.81**	0.0182	46.63
3.63	Dense	Equal Weights	**51.91**	90.89	0.0138	27.21
		Uncert. Weights [14]	51.89	**91.22**	**0.0134**	**25.36**
		DWA, $T = 2$	51.78	90.88	0.0137	26.67
≈2	Cross-Stitch [20]	Equal Weights	50.08	90.33	0.0154	34.49
		Uncert. Weights [14]	50.31	90.43	**0.0152**	**31.36**
		DWA, $T = 2$	**50.33**	**90.55**	0.0153	33.37
1.65	MTAN (Ours)	Equal Weights	53.04	**91.11**	**0.0144**	**33.63**
		Uncert. Weights [14]	**53.86**	91.10	0.0144	35.72
		DWA, $T = 2$	53.29	91.09	0.0144	34.14

- NYU 데이터셋에서는 MTAN이 모든 분야에서 높은 성능 보임

Type	#P.	Architecture	Weighting	Segmentation (Higher Better) mIoU	Segmentation (Higher Better) Pix Acc	Depth (Lower Better) Abs Err	Depth (Lower Better) Rel Err	Surface Normal: Angle Distance (Lower Better) Mean	Angle Distance (Lower Better) Median	Within t° (Higher Better) 11.25	Within t° (Higher Better) 22.5	Within t° (Higher Better) 30
Single Task	3	One Task	n.a.	15.10	51.54	0.7508	0.3266	31.76	25.51	22.12	45.33	57.13
	4.56	STAN	n.a.	15.73	52.89	0.6935	0.2891	32.09	26.32	21.49	44.38	56.51
Multi Task	1.75	Split, Wide	Equal Weights	15.89	51.19	0.6494	0.2804	33.69	28.91	18.54	39.91	52.02
			Uncert. Weights [14]	15.86	51.12	**0.6040**	0.2570	**32.33**	**26.62**	**21.68**	**43.59**	**55.36**
			DWA, $T = 2$	**16.92**	**53.72**	0.6125	**0.2546**	32.34	27.10	20.69	42.73	54.74
	2	Split, Deep	Equal Weights	13.03	41.47	0.7836	0.3326	38.28	36.55	9.50	27.11	39.63
			Uncert. Weights [14]	**14.53**	43.69	0.7705	0.3340	**35.14**	**32.13**	**14.69**	**34.52**	**46.94**
			DWA, $T = 2$	13.63	**44.41**	**0.7581**	**0.3227**	36.41	34.12	12.82	31.12	43.48
	4.95	Dense	Equal Weights	16.06	52.73	0.6488	0.2871	33.58	28.01	20.07	41.50	53.35
			Uncert. Weights [14]	**16.48**	**54.40**	0.6282	0.2761	**31.68**	**25.68**	**21.73**	**44.58**	**56.65**
			DWA, $T = 2$	16.15	54.35	**0.6059**	**0.2593**	32.44	27.40	20.53	42.76	54.27
	≈3	Cross-Stitch [20]	Equal Weights	14.71	50.23	0.6481	0.2871	33.56	28.58	20.08	40.54	51.97
			Uncert. Weights [14]	15.69	52.60	0.6277	0.2702	32.69	27.26	21.63	42.84	54.45
			DWA, $T = 2$	**16.11**	**53.19**	**0.5922**	**0.2611**	**32.34**	**26.91**	**21.81**	**43.14**	**54.92**
	1.77	MTAN (Ours)	Equal Weights	**17.72**	55.32	**0.5906**	0.2577	31.44	**25.37**	**23.17**	45.65	57.48
			Uncert. Weights [14]	17.67	**55.61**	0.5927	0.2592	**31.25**	25.57	22.99	**45.83**	**57.67**
			DWA, $T = 2$	17.15	54.97	0.5956	**0.2569**	31.60	25.46	22.48	44.86	57.24



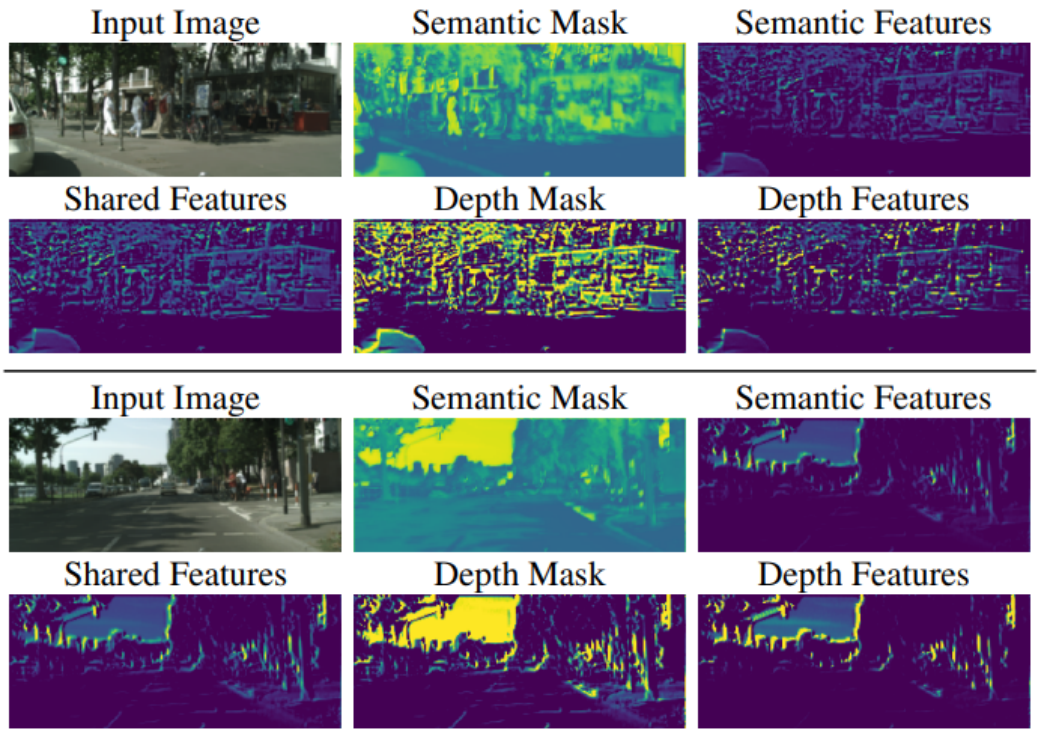
- 이미지 표면이 멀티태스크에서 더 선명하게 나타나는 것을 알 수 있다

4.1.5 Effect of Task Complexity

- 클래스가 2개 일 때, 단일 태스크 STAN 이 MT 보다 성능이 우수함
- 단, 작업 복잡도가 증가함에 따라 모든 실험에서 MTAN이 더 좋은 성능을 보임

4.1.6 Attention Masks as Feature Selectors

- 각 어텐션 마스크가 각 태스크에 유용한 부분에 집중할 수 있도록 공유된 피쳐에서 불필요한 부분을 없애는 일을 하기도 함



4.2 Visual Decathlon Challenge (Many-to-Many)

- MTAN을 Wide Residual Network 위에 올려서 구축
  - 깊이 28, widening factor 4, stride 2로 첫번째 convolutional 레이어 구성
  - 배치 사이즈 : 100
  - RL : 0.1
  - SGD 옵티마이져
  - 총 300에폭에서 50 에폭마다 RL 반으로 나눔
- 그 후 각 9개의 분류 태스크에서 RL 파인튜닝 진행
- 결과

Method	#P.	ImNet.	Airc.	C100	DPed	DTD	GTSR	Flwr	Oglt	SVHN	UCF	Mean	Score
Scratch [23]	10	59.87	57.10	75.73	91.20	37.77	96.55	56.3	88.74	96.63	43.27	70.32	1625
Finetune [23]	10	59.87	60.34	82.12	92.82	55.53	97.53	81.41	87.69	96.55	51.20	76.51	2500
Feature [23]	1	59.67	23.31	63.11	80.33	45.37	68.16	73.69	58.79	43.54	26.8	54.28	544
Res. Adapt.[23]	2	59.67	56.68	81.20	93.88	50.85	97.05	66.24	89.62	96.13	47.45	73.88	2118
DAN [25]	2.17	57.74	64.12	80.07	91.30	56.54	98.46	86.05	89.67	96.77	49.38	77.01	2851
Piggyback [19]	1.28	57.69	65.29	79.87	96.99	57.45	97.27	79.09	87.63	97.24	47.48	76.60	2838
Parallel SVD [24]	1.5	60.32	66.04	81.86	94.23	57.82	99.24	85.74	89.25	96.62	52.50	78.36	3398
MTAN (Ours)	1.74	63.90	61.81	81.59	91.63	56.44	98.80	81.04	89.83	96.88	50.63	77.25	2941

  - 다른 복잡한 작업 없이 베이스라인을 능가하거나 SOTA 모델과 비슷한 성능을 보임

5. Conclusions

- 우리가 제시한 MTAN으로 실험 결과, 복잡한 작업이나 손실함수의 조정 없이도 견고성과 멀티 태스크 작업에서 높은 성능을 보임
- 파라미터 부분에서 효율적임을 유지하며 SOTA를 보임