A Supplementary results

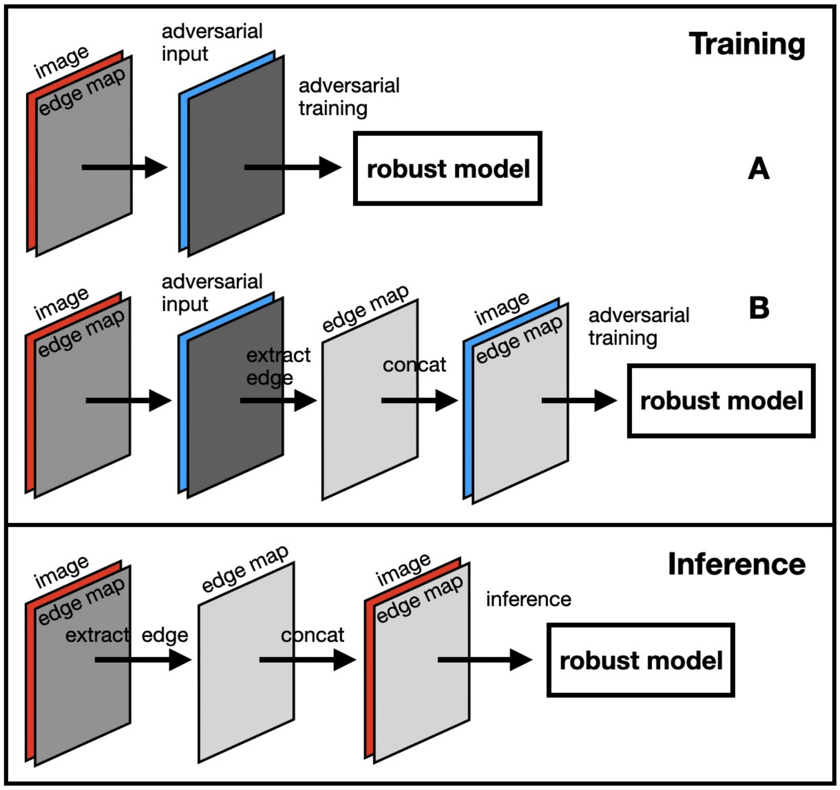

Figure 5: Edge-guided adversarial training (EAT). In its simplest form, adversarial training is performed over the 2D (Gray+Edge) or 4D (RGB+Edge) input (i.e., number of channels; denoted as Img+Edge). In a slightly more complicated form (B), first for each input (clean or adversarial), the old edge map is replaced with the newly extracted one. The edge map can be computed from the average of only image channels or all available channels (i.e., image plus edge).

Table 1: Results (Top-1 acc) over MNIST. The best accuracy in each column is highlighted in **bold**. In *italics* are the results of the substitute attack. Epsilon values are over 255. We used the ℓ_∞ variants of FGSM and PGD. Img2Edge means applying the Edge model (first row) to the edge map of the image.

		Orig. model				**Rob. model (8)**		**Rob. model (32)**		**Rob. model (64)**		**Average**
	ϵ	0/clean	8	32	64	0/clean	8	0/clean	32	0/clean	64	**Rob. models**
FGSM	**Edge**	0.964	0.925	0.586	0.059	0.973	0.954	0.970	0.892	0.964	0.776	0.921
	Img2Edge	,,	**0.960**	**0.951**	**0.918**	,,	**0.971**	,,	**0.957**	,,	0.910	**0.957**
	Img	**0.973**	0.947	0.717	0.162	**0.976**	0.955	**0.977**	0.892	0.970	0.745	0.919
	Img+Edge	0.972	0.941	0.664	0.089	**0.976**	0.958	**0.977**	0.902	**0.972**	0.782	0.928
	Redetect	”	0.950	0.803	0.356	”	0.962 *(0.968)*	”	0.919 *(0.947)*	”	0.843 *(0.881)*	0.941
	Img + Redetected Edge					0.974	0.950	0.970	0.771	0.968	0.228	0.810
	Redetect					”	0.958 *(0.966)*	”	0.929 *(0.947)*	”	**0.922** *(0.925)*	0.953
PGD-40	**Edge**	0.964	0.923	0.345	0.000	0.971	0.949	0.973	0.887	0.955	0.739	0.912
	Img2Edge	,,	**0.961**	**0.955**	**0.934**	,,	**0.970**	,,	**0.958**	,,	0.927	**0.960**
	Img	**0.973**	0.944	0.537	0.008	0.977	0.957	**0.978**	0.873	0.963	0.658	0.901
	Img+Edge	0.972	0.938	0.446	0.001	**0.978**	0.953	0.975	0.879	0.965	0.743	0.915
	Redetect	”	0.950	0.741	0.116	”	0.960 *(0.967)*	”	0.913 *(0.948)*	”	0.804 *(0.908)*	0.932
	Img + Redetected Edge					0.975	0.949	0.973	0.649	**0.968**	0.000	0.752
	Redetect					”	0.958 *(0.967)*	”	0.945 *(0.958)*	”	**0.939** *(0.942)*	**0.960**

Table 2: Results over the CIFAR-10 dataset.

		Orig. model			Rob. model (8)		Rob. model (32)		Average
	ϵ	0/clean	8	32	0/clean	8	0/clean	32	Rob. models
FGSM	**Edge**	0.490	0.060	0.015	0.535	0.323	0.382	0.199	0.360
FGSM	**Img2Edge**	„	0.258	0.258	„	0.270	„	0.217	0.351
FGSM	**Img**	**0.887**	0.359	0.246	**0.869**	0.668	**0.855**	0.553	0.736
FGSM	**Img + Edge**	0.860	0.366	0.169	0.846	0.611	0.815	0.442	0.679
FGSM	Redetect	„	**0.399**	**0.281**	„	0.569 *(0.631)*	„	0.417 *(0.546)*	0.662
FGSM	**Img + Redetected Edge**				0.846	0.530	0.832	0.337	0.636
FGSM	Redetect				„	**0.702** *(0.753)*	„	**0.569** *(0.678)*	**0.737**
PGD-40	**Edge**	0.490	0.071	0.000	0.537	0.315	0.142	0.119	0.278
PGD-40	**Img2Edge**	„	0.259	**0.253**	„	0.274	„	0.253	0.301
PGD-40	**Img**	**0.887**	0.018	0.000	0.807	0.450	0.316	0.056	0.407
PGD-40	**Img + Edge**	0.860	0.019	0.000	0.788	0.429	0.176	0.119	0.378
PGD-40	Redetect	„	**0.306**	0.093	„	0.504 *(0.646)*	„	0.150 *(0.170)*	0.404
PGD-40	**Img + Redetected Edge**				**0.834**	0.155	**0.776**	0.006	0.443
PGD-40	Redetect				„	**0.661** *(0.767)*	„	**0.392** *(0.700)*	**0.666**

Table 3: Results over the Fashion MNIST dataset (*)

	Orig. model				Rob. model (8)		Rob. model (32)		Rob. model (64)		Average
ϵ	0/clean	8	32	64	0/clean	8	0/clean	32	0/clean	64	Rob. models
FGSM											
Edge	0.775	0.714	0.497	0.089	0.776	0.740	0.766	**0.664**	0.748	**0.750**	**0.741**
Img2Edge	„	**0.755**	**0.679**	**0.452**	„	**0.762**	„	**0.664**	„	0.420	0.690
Img	0.798	0.670	0.288	0.027	**0.798**	0.722	0.764	0.584	**0.768**	0.505	0.690
Img+Edge	**0.809**	0.662	0.229	0.010	0.794	0.732	0.769	0.623	0.750	0.537	0.701
Redetect	”	0.691	0.326	0.053	”	0.739 *(0.761)*	”	0.616 *(0.660)*	„	0.491 *(0.496)*	0.693
Img + Redetected Edge					0.789	0.719	**0.775**	0.539	0.762	0.045	0.605
Redetect					”	0.739 *(0.753)*	”	**0.664** *(0.678)*	”	0.611 *(0.532)*	0.721
PGD-40											
Edge	0.775	0.711	0.370	0.002	0.783	0.744	0.769	0.661	0.743	0.574	0.712
Img2Edge	„	**0.757**	**0.683**	**0.380**	„	**0.762**	„	0.658	„	0.374	0.681
Img	0.798	0.659	0.133	0.000	0.792	0.713	0.760	0.515	0.734	0.324	0.640
Img+Edge	**0.809**	0.647	0.100	0.000	0.794	0.726	0.765	0.608	0.744	0.568	0.701
Redetect	”	0.682	0.235	0.014	”	0.734 *(0.760)*	”	0.629 *(0.666)*	-	0.607 *(0.426)*	0.712
Img + Redetected Edge					**0.800**	0.717	**0.779**	0.393	**0.771**	0.002	0.577
Redetect					”	0.743 *(0.766)*	”	**0.694** *(0.681)*	”	**0.690** *(0.504)*	**0.746**

Table 4: Results over the TinyImageNet dataset (*)

	Orig. model			Rob. model (8)		Rob. model (32)		Average
ϵ	0/clean	8	32	0/clean	8	0/clean	32	Rob. models
FGSM								
Edge	0.136	0.010	0.001	0.150	0.078	0.098	0.021	0.087
Img2Edge	„	0.097	**0.096**	„	0.094	„	0.077	0.105
Img	**0.531**	0.166	0.074	**0.512**	0.297	**0.488**	0.168	**0.366**
Img + Edge	0.522	0.152	0.050	0.508	0.273	0.471	0.148	0.350
Redetect	„	**0.171**	0.081	„	0.287 *(0.356)*	„	0.162 *(0.266)*	0.357
Img + Redetected Edge				0.505	0.264	0.482	0.111	0.340
Redetect				”	**0.305** *(0.371)*	„	**0.171** *(0.296)*	**0.366**
PGD-40								
Edge	0.136	0.007	0.000	0.148	0.077	0.039	0.014	0.069
Img2Edge	„	**0.094**	**0.092**	„	0.095	„	0.033	0.079
Img	**0.531**	0.019	0.000	0.392	0.150	0.191	0.019	0.188
Img + Edge	0.522	0.008	0.000	0.402	0.131	0.157	0.003	0.173
Redetect	„	0.074	0.009	„	0.198 *(0.353)*	„	0.019 *(0.103)*	0.194
Img + Redetected Edge				**0.425**	0.072	**0.328**	0.005	0.208
Redetect				”	**0.206** *(0.380)*	„	**0.073** *(0.279)*	**0.258**

Table 5: Results on CIFAR-10 dataset [edge map computed from 4 channels]

	Orig. model			**Rob. model (8)**		**Rob. model (32)**		**Average**
ϵ	0/clean	8	32	0/clean	8	0/clean	32	**Rob. models**
				FGSM				
Img+Edge	0.860	0.366	0.169	0.846	0.611	0.815	0.442	0.679
Redetect	”	0.415	0.280	”	0.574	„	0.416	0.663
Img + Redetected Edge				0.848	0.547	0.835	0.351	0.645
Redetect				”	0.696	”	0.553	0.733
				PGD-40				
Img+Edge	0.860		0.000	0.789	0.431	0.179	0.135	0.384
Redetect	”		0.087	„	0.501	„	0.152	0.405
Img + Redetected Edge				0.837	0.164	0.767	0.010	0.444
Redetect				”	0.648	„	0.352	0.651

Table 6: Results on DogVsCat dataset [edge map computed from 4 channels] (*)

	Orig. model			**Rob. model (8)**		**Rob. model (32)**		**Average**
ϵ	0/clean	8	32	0/clean	8	0/clean	32	**Rob. models**
				FGSM				
Edge	0.814	0.633	0.119	0.812	0.757	0.806	**0.999**	0.843
Img2Edge	„	**0.755**	**0.584**	„	**0.767**	„	0.576	0.740
Img	**0.863**	0.007	0.051	0.777	0.430	**0.819**	0.985	0.753
Img+Edge	0.823	0.007	0.000	0.782	0.641	0.808	0.992	0.806
Redetect	”	0.043	0.002	”	0.666	”	0.986	0.810
Img + Redetected Edge				**0.829**	0.615	0.812	0.853	0.778
Redetect				”	0.763	”	0.998	**0.850**
				PGD-40				
Edge	0.814	0.624	0.018	**0.820**	0.770	0.763	0.681	0.758
Img2Edge	„	**0.760**	**0.568**	„	**0.778**	„	0.656	0.754
Img	**0.863**	0.000	0.000	0.769	0.384	0.500	0.500	0.538
Img+Edge	0.823	0.000	0.000	0.785	0.689	0.816	0.496	0.696
Redetect	”	0.006	0.000	„	0.744	„	0.500	0.711
Img + Redetected Edge				0.819	0.600	**0.817**	0.009	0.561
Redetect				”	0.760	„	**0.972**	**0.842**

Table 7: Results on DogBreeds dataset using Sobel edge detection [edge map computed from 4 channels] (*)

	Orig. model			**Rob. model (8)**		**Rob. model (32)**		**Average**
ϵ	0/clean	8	32	0/clean	8	0/clean	32	**Rob. models**
				FGSM				
Edge	0.750	0.006	0.031	0.506	0.101	0.413	0.073	0.273
Img2Edge	„	0.236	**0.194**	„	0.362	„	0.241	0.380
Img	**0.899**	0.256	0.140	0.823	0.595	0.829	**0.449**	0.674
Img + Edge	0.896	0.225	0.098	**0.862**	0.534	0.820	0.385	0.650
Redetect	„	**0.244**	0.171	„	0.455	„	0.292	0.607
Img + Redetected Edge				0.843	0.506	**0.874**	0.298	0.630
Redetect				”	**0.618**	„	0.419	**0.689**
				PGD-40				
Edge	0.750	0.000	0.000	0.514	0.065	0.036	0.000	0.154
Img2Edge	„	**0.250**	**0.207**	„	0.301	„	0.037	0.222
Img	**0.899**	0.000	0.000	0.795	0.286	0.596	0.025	0.425
Img + Edge	0.896	0.000	0.000	0.789	0.225	0.567	0.042	0.406
Redetect	„	0.008	0.000	”	0.396	„	0.065	0.454
Img + Redetected Edge				**0.772**	0.028	**0.677**	0.000	0.369
Redetect				”	**0.393**	”	**0.149**	**0.498**

Table 8: Results on GTSRB dataset [edge map computed from 4 channels] (*)

	Orig. model			**Rob. model (8)**		**Rob. model (32)**		**Average**
ϵ	0/clean	8	32	0/clean	8	0/clean	32	**Rob. models**
				FGSM				
Edge	0.938	**0.683**	0.315	**0.947**	**0.863**	**0.946**	0.701	0.864
Img2Edge	„	0.501	0.451	„	0.516	„	0.469	0.719
Img	**0.955**	0.464	0.322	0.902	0.607	0.896	0.562	0.742
Img + Edge	0.951	0.624	0.382	0.940	0.842	0.943	0.686	0.853
Redetect	”	0.592	**0.471**	”	0.743	”	0.626	0.813
Img + Redetected Edge				0.925	0.801	0.939	0.616	0.820
Redetect				”	0.844	”	**0.766**	**0.869**
				PGD-40				
Edge	0.938	**0.618**	0.054	**0.950**	**0.861**	**0.937**	**0.598**	**0.836**
Img2Edge	„	0.501	**0.459**	„	0.506	„	0.462	0.714
Img	**0.955**	0.189	0.033	0.855	0.495	0.736	0.246	0.583
Img + Edge	0.951	0.271	0.021	0.943	0.750	0.839	0.342	0.718
Redetect	„	0.526	0.251	„	0.774	„	0.514	0.767
Img + Redetected Edge				0.929	0.505	0.893	0.134	0.615
Redetect				”	0.818	„	0.557	0.799

Table 9: Results on GTSRB dataset [edge map computed from 3 channels]

	Orig. model			**Rob. model (8)**		**Rob. model (32)**		**Average**
ϵ	0/clean	8	32	0/clean	8	0/clean	32	**Rob. models**
				FGSM				
Img + Edge	0.951	0.624	0.382	0.940	0.842	0.943	0.686	0.853
Redetect	„	0.500	0.395	„	0.558	„	0.492	0.733
Img + Redetected Edge				0.889	0.699	0.891	0.549	0.757
Redetect				„	0.610	„	0.577	0.742

Table 10: Results on Icons-50 dataset [edge map computed from 4 channels] (*)

	Orig. model			**Rob. model (8)**		**Rob. model (32)**		**Average**
ϵ	0/clean	8	32	0/clean	8	0/clean	32	**Rob. models**
				FGSM				
Edge	0.883	0.545	0.210	**0.904**	0.771	**0.889**	0.594	0.789
Img2Edge	„	**0.713**	**0.690**	„	0.746	„	0.730	**0.817**
Img	**0.930**	0.495	0.433	0.772	0.789	0.836	0.720	0.779
Img + Edge	0.929	0.569	0.433	0.829	0.818	0.844	**0.745**	0.809
Redetect	„	0.470	0.414	„	0.730	„	0.732	0.784
Img + Redetected Edge				0.841	**0.837**	0.849	0.688	0.804
Redetect				„	0.817	„	0.710	0.804
				PGD-40				
Edge	0.883	0.423	0.000	**0.902**	0.769	**0.846**	0.404	0.730
Img2Edge	„	**0.706**	**0.683**	„	0.753	„	**0.695**	**0.799**
Img	**0.930**	0.341	0.113	0.765	0.663	0.736	0.453	0.654
Img + Edge	0.929	0.320	0.011	0.800	0.678	0.785	0.366	0.657
Redetect	„	0.416	0.248	„	0.738	„	0.660	0.746
Img + Redetected Edge				0.838	0.644	0.824	0.097	0.601
Redetect				”	**0.792**	„	0.539	0.748

Table 11: Results on Icons-50 dataset [edge map computed from 3 channels]

	Orig. model			**Rob. model (8)**		**Rob. model (32)**		**Average**
ϵ	0/clean	8	32	0/clean	8	0/clean	32	**Rob. models**
				FGSM				
Img+Edge	0.929	0.569	0.433	0.829	0.818	0.844	0.745	0.809
Redetect	„	0.520	0.460	„	0.737	„	0.731	0.785
Img + Redetected Edge				0.831	0.788	0.870	0.725	0.804
Redetect				”	0.783	„	0.765	0.812

Table 12: Results on Sketch dataset [edge map computed from 2 channels] (*)

	Orig. model			**Rob. model (8)**		**Rob. model (32)**		**Average**
ϵ	0/clean	8	32	0/clean	8	0/clean	32	**Rob. models**
				FGSM				
Edge	0.479	0.167	**0.041**	0.502	0.343	**0.483**	**0.216**	**0.386**
Img2Edge	„	**0.464**	0.014	„	**0.494**	„	0.022	0.375
Img	**0.532**	0.109	0.021	**0.530**	0.278	0.474	0.144	0.356
Gray + Edge	0.486	0.097	0.019	0.513	0.286	0.440	0.167	0.352
Redetect	„	0.263	0.004	„	0.355	„	0.013	0.330
Img + Redetected Edge				0.497	0.180	0.420	0.071	0.292
Redetect				”	0.416	„	0.162	0.374
				PGD-40				
Edge	0.480	0.106	0.000	0.508	0.341	0.401	0.068	0.330
Img2Edge	„	**0.471**	**0.127**	„	**0.499**	„	**0.214**	**0.405**
Img	**0.532**	0.028	0.000	**0.538**	0.260	0.018	0.000	0.204
Gray + Edge	0.486	0.034	0.000	0.500	0.279	0.026	0.000	0.201
Redetect	„	0.277	0.024	„	0.360	„	0.004	0.223
Img + Redetected Edge				0.502	0.121	**0.448**	0.000	0.268
Redetect				”	0.423	„	0.212	0.396

Table 13: Results on Sketch dataset [edge map computed from 1 channel]

	Orig. model			**Rob. model (8)**		**Rob. model (32)**		**Average**
ϵ	0/clean	8	32	0/clean	8	0/clean	32	**Rob. models**
				FGSM				
Gray + Edge	0.486	0.097	0.019	0.513	0.286	0.440	0.167	0.352
Redetect	„	0.213	0.005	„	0.388	„	0.022	0.341
Img + Redetected Edge				0.519	0.296	0.445	0.191	0.363
Redetect				„	0.397	„	0.020	0.345

Table 14: Results on Imagenette2-160 dataset [edge map computed from 4 channels] (*)

	Orig. model			**Rob. model (8)**		**Rob. model (32)**		**Average**
ϵ	0/clean	8	32	0/clean	8	0/clean	32	**Rob. models**
				FGSM				
Edge	0.780	0.101	0.436	0.781	0.520	0.664	0.245	0.553
Img2Edge	„	0.599	**0.598**	„	0.603	„	0.578	0.656
Img	**0.969**	0.617	0.409	**0.959**	0.827	0.946	0.710	0.860
Img + Edge	0.959	0.613	0.373	0.951	0.801	0.935	0.643	0.832
Redetect	„	**0.652**	0.471	„	0.812	„	0.687	0.846
Img + Redetected Edge				0.950	0.747	**0.949**	0.592	0.810
Redetect				„	**0.834**	„	**0.732**	**0.866**
				PGD-40				
Edge	0.780	0.064	0.000	0.794	0.526	0.577	0.071	0.492
Img2Edge	„	**0.601**	**0.577**	„	0.610	„	0.381	0.591
Img	**0.969**	0.052	0.005	0.918	0.599	0.808	0.221	0.636
Img + Edge	0.959	0.045	0.000	0.909	0.558	0.762	0.151	0.595
Redetect	”	0.445	0.069	”	0.743	”	0.305	0.680
Img + Redetected Edge				**0.944**	0.246	**0.883**	0.046	0.530
Redetect				”	**0.757**	”	**0.432**	**0.754**

B Robustness against substitute model attacks

Following [13], we trained substitute models to mimic the robust models (with the same architecture but with RGB channels) using the cross-entropy loss over the logits of the two networks, for 5 epochs. The adversarial examples crafted for the substitute networks were then fed to the robust networks. Results are shown in *italics* in Tables 1, 2, 3 and 4 (performed only against the edge-redetect models). We find that this attack is not able to knock off the robust models. Surprisingly, it even improves the accuracy in some cases.

Table 15: Results of the substitute attack against the robust Img + Edge models (redetect and full model).

	MNIST			Fashion MNIST			CIFAR		TinyImgNet	
ϵ	8	32	64	8	32	64	8	32	8	32
FGSM										
Img + edge model (redetect inference)										
Substitute model on clean images	0.94	0.9365	0.9314	0.7515	0.7393	0.7311	0.8079	0.7766	0.008	0.008
Substitute model on adversarial images	0.8941	0.5858	0.0992	0.6484	0.3701	0.0967	0.2716	0.2049	0.004	0.003
Robust model on clean images	0.9761	0.9766	0.9722	0.7939	0.7692	0.75	0.8463	0.8463	0.508	0.471
Robust model on adversarial images	0.9623	0.9189	0.842	0.7391	0.6156	0.4908	0.5695	0.4186	0.287	0.161
Robust model on substitute adv. images	0.9678	0.9472	0.8813	0.7609	0.6604	0.4955	0.6307	0.5463	0.356	0.266
Img + redetected edge model (redetect inference)										
Substitute model on clean images	0.9381	0.9335	0.9326	0.7513	0.7431	0.7388	0.8104	0.7966	0.008	0.008
Substitute model on adversarial images	0.89	0.5696	0.0989	0.6538	0.3663	0.08	0.2879	0.1988	0.004	0.002
Robust model on clean images	0.9742	0.9699	0.9681	0.7891	0.7746	0.7617	0.8456	0.8328	0.495	0.482
Robust model on adversarial images	0.9583	0.9283	0.9216	0.7392	0.664	0.6115	0.7032	0.5684	0.380	0.170
Robust model on substitute adv. images	0.9657	0.9469	0.9249	0.7529	0.6776	0.5318	0.7528	0.7528	0.371	0.296
PGD-40										
Img + edge model (redetect inference)										
Substitute model on clean images	0.9391	0.9344	0.9257	0.7531	0.7408	0.7303	0.756	0.194	0.008	0.006
Substitute model on adv. images	0.8906	0.4455	0.0196	0.6473	0.2745	0.0096	0.020	0.003	0.000	0.000
Robust model on clean images	0.9782	0.9751	0.9654	0.7938	0.7652	0.7442	0.788	0.179	0.395	0.157
Robust model on adv. images	0.9599	0.9132	0.8039	0.7336	0.6289	0.6068	0.504	0.152	0.242	0.018
Robust model on substitute adv. images	0.9667	0.9477	0.9079	0.7603	0.6656	0.4263	0.646	0.170	0.352	0.103
Img + redetected edge model (redetect inference)										
Substitute model on clean images	0.9385	0.9363	0.9329	0.7503	0.7471	0.7415	0.804	0.730	0.008	0.008
Substitute model on adv. images	0.8888	0.4617	0.0211	0.6458	0.2687	0.01	0.016	0.000	0.000	0.000
Robust model on clean images	0.975	0.9732	0.9682	0.7998	0.7793	0.7715	0.834	0.766	0.425	0.328
Robust model on adv. images	0.9581	0.9449	0.9386	0.7435	0.6943	0.6902	0.662	0.375	0.206	0.074
Robust model on substitute adv. images	0.9665	0.9575	0.9417	0.7661	0.681	0.5037	0.767	0.700	0.380	0.279

C Robustness against Carlini-Wagner (CW) and Boundary attacks

Performance of our method against l_2 CW attack [3] on MNIST dataset is shown in Table 16. To make experiments tractable, we set the number of attack iterations to 10. With even 10 iterations, the original Edge and Img models are severely degraded. Img2Edge and Img+(Edge Redetect) models, however, remain robust. Adversarial training with CW attack results in robust models in all cases.

Performance of the the EAT defense against the l_2 Carlini-Wagner attack [3] with the following parameters:

```
attack = CW(net, targeted=False, c=1e-4, kappa=0, iters=10, lr=0.001)
```

Table 16: Robustness against Carlini-Wagner (CW) and Boundary attacks

	Orig. model		**Robust model**		**Average Rob. models**
	0/clean	adv.	0/clean	adv.	
Edge	0.964	0.106	0.948	0.798	0.873
Img2Edge	,,	0.962	,,	0.949	**0.949**
Img	0.973	0.103	0.949	0.856	0.903
Img+Edge	0.972	0.097	0.945	0.845	0.895
Redetect	,,	0.971	,,	0.942	0.944
Img + Redetected Edge			0.947	0.819	0.883
Redetect			,,	0.946	0.946

D Robustness against Boundary attack

Results against the decision-based Boundary attack [1] over MNIST and Fashion MNIST datasets are shown below. Edge, Img, and Img+Edge models perform close to zero over adversarial images. Img+(Edge Redetect) model remains robust since the Canny edge map does not change much after the attack, as is illustrated in Fig. 6.

Performance of the the edge augmented model against the Boundary attack [1] with the following parameters:

```
BoundaryAttack(init_attack=None, steps=25000, spherical_step=0.01,
                source_step=0.01, source_step_convergance=1e-07,
                step_adaptation=1.5, tensorboard=False,
                update_stats_every_k=10)
```

Table 17: Results over 500 images from the MNIST dataset

	Orig. model	
	0/clean	adv. (boundary)
Edge	0.964	0.000
Img	0.973	0.003
Img+Edge	0.972	0.000
Redetect	,,	**0.945**
Img+Redetected Edge (adversarially trained using FGSM $\epsilon = 8/255$)	0.974	0.001
Redetect	,,	**0.965**

Table 18: Results over 500 images from the Fashion MNIST dataset

	Orig. model	
	0/clean	adv. (boundary)
Edge	0.776	0.005
Img	0.798	0.018
Img+Edge	0.809	0.003
Redetect	,,	**0.747**
Img+Redetected Edge (adversarially trained using FGSM $\epsilon = 8/255$)	0.789	0.003
Redetect	,,	**0.770**

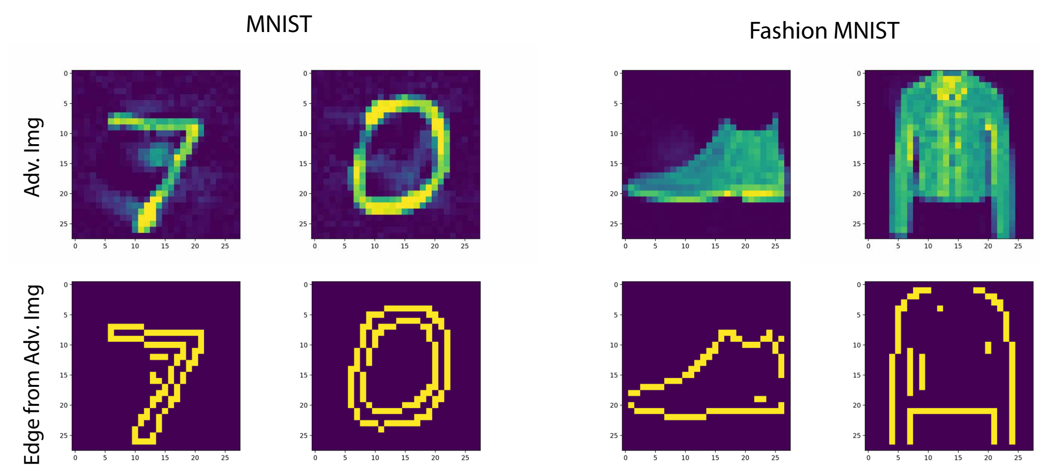

Figure 6: Sample images from the Boundary attack.

E Robustness against adaptive attacks

E.1 Robustness against adaptive attacks over Imagenette2-160 dataset

We use the PyTorch implementation[3] of the HED edge detector proposed by [15]. Here, a classifier is first trained on top of the edge maps from the HED. Then, the entire pipeline (Img $\longrightarrow$ HED $\longrightarrow$ Classifier^{HED}) is attacked to generate an adversarial image. The performance of this classifier is measured on both clean and adversarial images. The adversarial image is also fed to the classifier trained on Canny edge maps (i.e., Img$^{adv-HED}$ $\longrightarrow$ Canny $\longrightarrow$ Classifier^{Canny}). Results are shown in Table below. As it can be seen, adversarial examples crafted for HED fail to completely fool the model trained on Canny edges (i.e., they do not transfer).

Table 19: Results over 500 images from the Imagenette2-160 dataset against the FGSM and PGD-5 ($\epsilon = 8/255$) attacks.

	Orig. model		
	0/clean	adv. (FGSM)	adv. (PGD-5)
Img2Edge (Img $\longrightarrow$ HED $\longrightarrow$ Classifier^{HED})	0.793	0.052	0.003
Img2Edge (Img$^{adv-HED}$ $\longrightarrow$ Canny $\longrightarrow$ Classifier^{Canny})	0.767	0.542	0.548

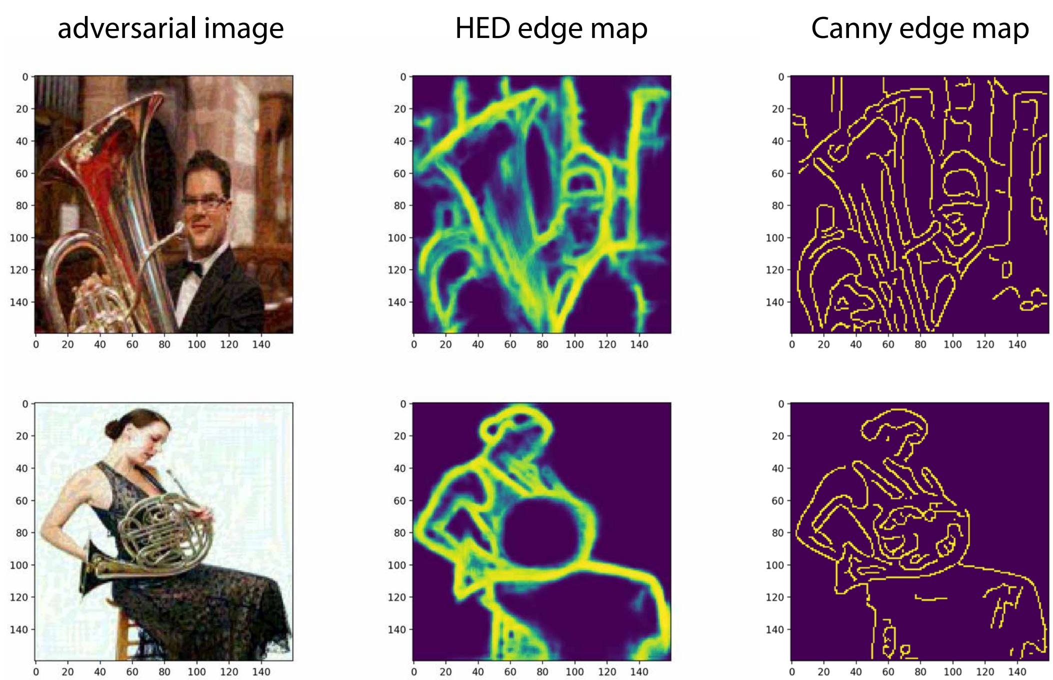

Figure 7: Two sample adversarial images (FGSM) along with their edge maps using HED and Canny edge detection methods.

E.2 Robustness against adaptive attacks over MNIST dataset

Here, we attempt to explicitly approximate the Canny edge detector using a differentiable convolutional autoencoder. In our pipeline, a classifier (CNN) is stacked after the convolutional autoencoder (with sigmoid output neurons). We first freeze the classifier and train the autoencoder using the MSE loss with (input, output) pair being (image, canny edge map). We then freeze the autoencoder and train the classifier using Cross Entropy loss. After training the network, we then craft adversarial examples for it and feed them to a classifier trained on

[3] `https://github.com/sniklaus/pytorch-hed`

Canny edges (original models or robust models as was mentioned in the main text). Fig. 8 shows the pipeline and some sample approximated edge maps. Fig. 9 shows the architecture details in PyTorch.

The top panel in Fig. 10 shows results using the FGSM and PGD-40 attacks against the pipeline itself, and also against the Img2Edge model (trained over clean edges or adversarial ones[4]). As can be seen, both attacks are very successful against the pipeline but they do not perform well against the Canny edge map classifier (i.e., crafted adversarial examples for the pipeline do not transfer well to the Imge2Edge trained over Canny Edge map; img⟶ Canny ⟶ class label). Notice, that here we only used the model trained on edge maps. It is likely to gain even better robustness against the adaptive attacks in using the img+edge+redetect.

The bottom panel in Fig. 10 shows sample adversarial digits (constructed using the adaptive attack) and their edge maps under the FGSM and PGD-40 attacks. Notice how PGD-40 attack preserves the edges (compered to FGSM). This is because it needs less perturbation to fool the classifier. Also, notice that the perturbations shown are perceptible which results in edges maps having noise. If we limit ourselves to imperceptible perturbations, then edge maps will not change much compared to the original edge maps on clean images.

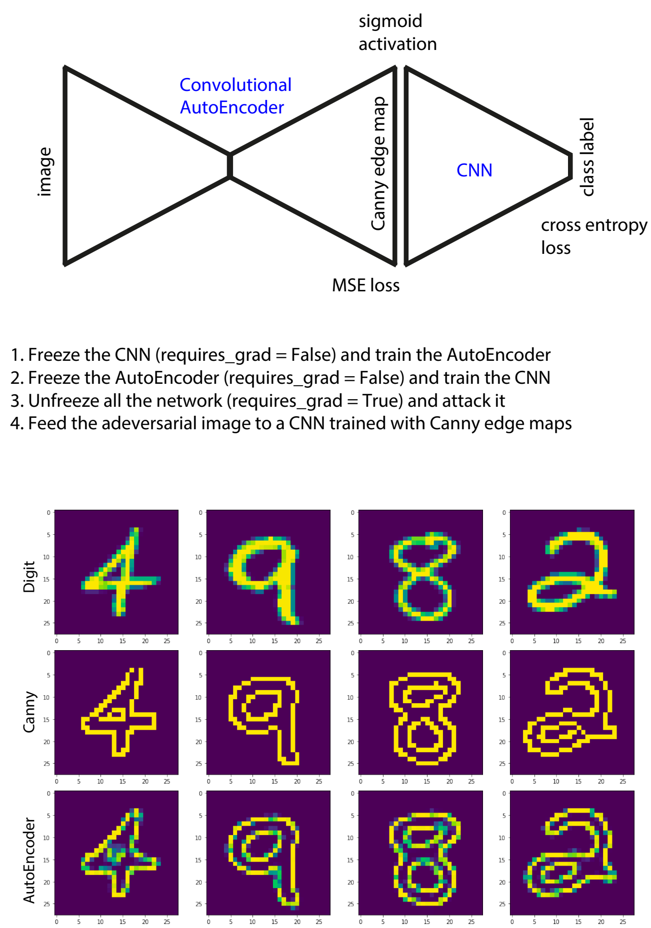

Figure 8: Top: our pipeline to approximate the Canny edge detector and our approach for crafting adversarial examples, Bottom: Sample digits and their generated edge maps.

[4]Here we used the model adversarially trained at eps=8/255 and test it against other perturbations; unlike the main text where we trained robust models separately for each epsilon.

```python
# combined network
# LeNet Model definition
class MNIST_Net_combined(nn.Module):
    def __init__(self, net_type='gray'):
        super(MNIST_Net_combined, self).__init__()

        self.encoder = nn.Sequential( # like the Composition layer you built
            nn.Conv2d(1, 16, 3, stride=2, padding=1),
            nn.ReLU(),
            nn.Conv2d(16, 32, 3, stride=2, padding=1),
            nn.ReLU(),
            nn.Conv2d(32, 64, 7)
        )
        self.decoder = nn.Sequential(
            nn.ConvTranspose2d(64, 32, 7),
            nn.ReLU(),
            nn.ConvTranspose2d(32, 16, 3, stride=2, padding=1, output_padding=1),
            nn.ReLU(),
            nn.ConvTranspose2d(16, 1, 3, stride=2, padding=1, output_padding=1),
            nn.Sigmoid()
        )

        self.conv1 = nn.Conv2d(1, 10, kernel_size=5)
        self.conv2 = nn.Conv2d(10, 20, kernel_size=5)
        self.conv2_drop = nn.Dropout2d()
        self.fc1 = nn.Linear(320, 50)
        self.fc2 = nn.Linear(50, 10)


    def forward(self, x):
        z = self.encoder(x)
        x_auto = self.decoder(z) # reconstructed egde
        x_auto = x_auto.view(x_auto.shape[0],1, 28,28)
        x = F.relu(F.max_pool2d(self.conv1(x_auto), 2))
        x = F.relu(F.max_pool2d(self.conv2_drop(self.conv2(x)), 2))
        x = x.view(-1, 320)
        x = F.relu(self.fc1(x))
        x = F.dropout(x, training=self.training)
        x = self.fc2(x)
        return x, x_auto
```

Figure 9: PyTorch code of our pipeline shown in Fig 8.

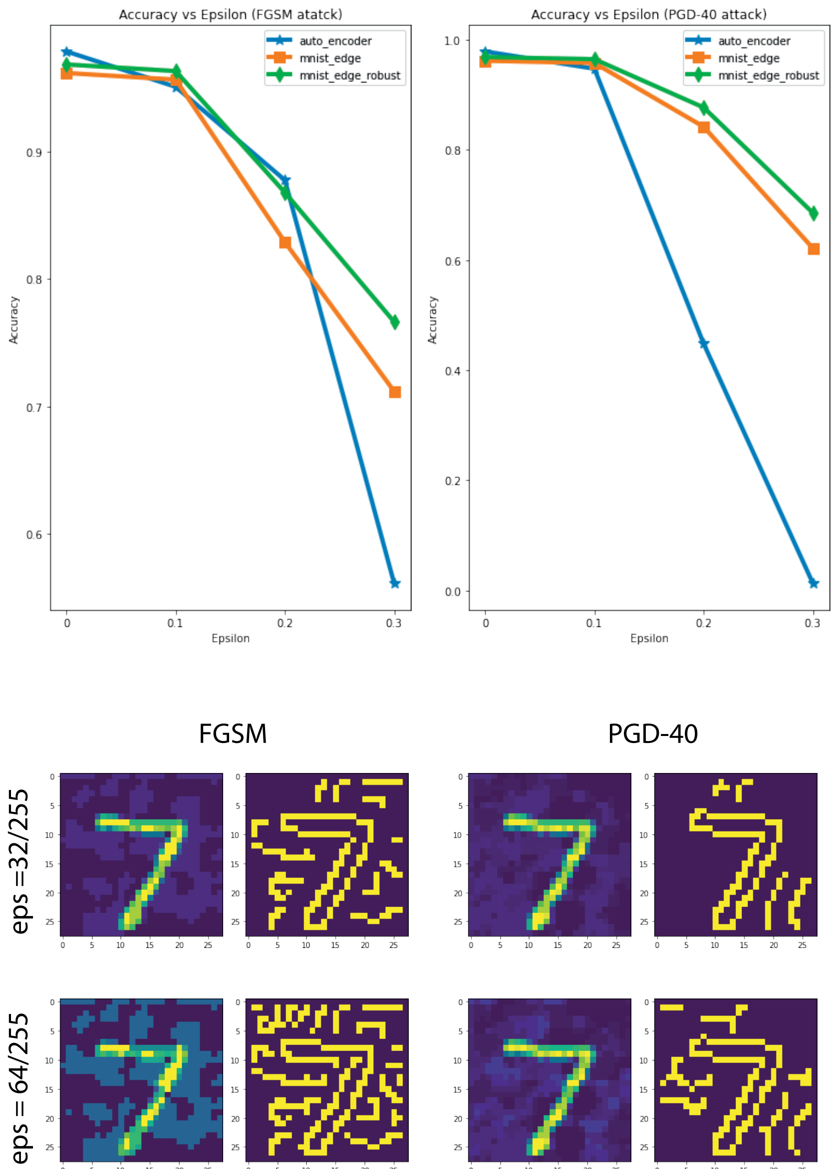

Figure 10: Top: Performance of the adaptive attack, Bottom: Samples adversarial images and their edge maps using the Canny edge detector.

F Sample generated images by the conditional GAN in GAN-based Shape Defense (GSD)

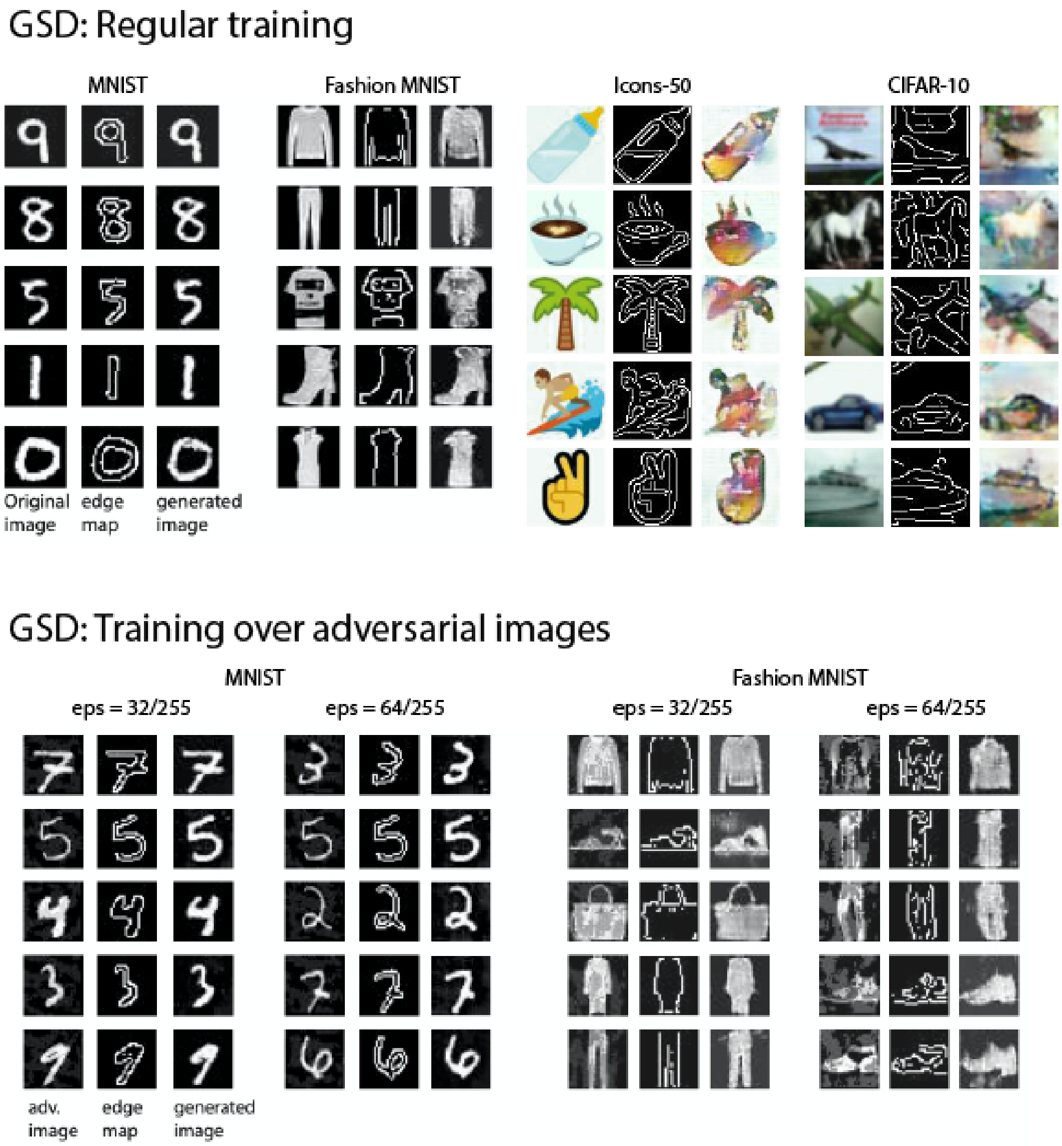

Figure 11: Top) GSD with a classifier trained on images generated (by pix2pix) only from the edge maps of the clean images, Bottom) GSD with edge maps derived from adversarial examples. Columns from left to right: adversarial images by the FGSM attack, their edge maps, and generated images by pix2pix.

G Shape-based extensions of vanilla PGD adversarial training, free adversarial training (FreeAT), and fast adversarial training (FastAT) algorithms

Algorithm 3 Shape-based PGD adversarial training for T epochs, given some radius ϵ, adversarial step size α and N PGD steps and a dataset of size M for a network f_θ. $\beta \in \{edge, img, imgedge\}$ indicates the net_type and *redetect_train* mean edge redetection during training.

for $t = 1 \ldots T$ **do**
  for $i = 1 \ldots M$ **do**
    // Perform PGD adversarial attack
    $\delta = 0$ *// or randomly initialized*
    for $j = 1 \ldots N$ **do**
      $\delta = \delta + \alpha \cdot \text{sign}(\nabla_\delta \ell(f_\theta(x_i + \delta), y_i))$
      $\delta = \max(\min(\delta, \epsilon), -\epsilon)$
    end for
    $\tilde{x}_i = x_i + \delta$
    if *redetect_train* & β == *imgedge* **then**
      $\tilde{x}_i =$ detect_edge$(\tilde{x}_i)$ *// recompute and replace the edge map*
    end if
    $\theta = \theta - \nabla_\theta \ell(f_\theta(\tilde{x}_i), y_i)$ *// Update model weights with some optimizer, e.g. SGD*
  end for
end for

Algorithm 4 Shape-based "Free" adversarial training for T epochs, given some radius ϵ, N minibatch replays, and a dataset of size M for a network f_θ. $\beta \in \{edge, img, imgedge\}$ indicates the net_type and *redetect_train* mean edge redetection during training.

$\delta = 0$
// Iterate T/N times to account for minibatch replays and run for T total epochs
for $t = 1 \ldots T/N$ **do**
  for $i = 1 \ldots M$ **do**
    // Perform simultaneous FGSM adversarial attack and model weight updates T times
    for $j = 1 \ldots N$ **do**
      $\tilde{x}_i = x_i + \delta$
      if *redetect_train* & β == *imgedge* **then**
        $\tilde{x}_i =$ detect_edge$(\tilde{x}_i)$ *// recompute and replace the edge map*
      end if
      // Compute gradients for perturbation and model weights simultaneously
      $\nabla_\delta, \nabla_\theta = \nabla \ell(f_\theta(\tilde{x}_i), y_i)$
      $\delta = \delta + \epsilon \cdot \text{sign}(\nabla_\delta)$
      $\delta = \max(\min(\delta, \epsilon), -\epsilon)$
      $\theta = \theta - \nabla_\theta$ *// Update model weights with some optimizer, e.g. SGD*
    end for
  end for
end for

Algorithm 5 Shape-based FGSM adversarial training for T epochs, given some radius ϵ, N PGD steps, step size α, and a dataset of size M for a network f_θ. $\beta \in \{edge, img, imgedge\}$ indicates the net_type and *redetect_train* mean edge redetection during training.

for $t = 1 \dots T$ **do**
  for $i = 1 \dots M$ **do**
    // Perform FGSM adversarial attack
    $\delta = \text{Uniform}(-\epsilon, \epsilon)$
    $\delta = \delta + \alpha \cdot \text{sign}(\nabla_\delta \ell(f_\theta(x_i + \delta), y_i))$
    $\delta = \max(\min(\delta, \epsilon), -\epsilon)$
    $\tilde{x}_i = x_i + \delta$
    if *redetect_train* & β == *imgedge* **then**
      $\tilde{x}_i = \text{detect\_edge}(\tilde{x}_i)$ *// recompute and replace the edge map*
    end if
    $\theta = \theta - \nabla_\theta \ell(f_\theta(\tilde{x}_i), y_i)$ *// Update model weights with some optimizer, e.g. SGD*
  end for
end for

Table 20: Performance of the Fast Adversarial Training (FastAT) method over three runs.

	Run 1		Run 1		Run 1		Average	
Model	Clean	PGD-10	Clean	PGD-10	Clean	PGD-10	Clean	PGD-10
Edge	0.559	0.384	0.581	0.187	0.608	0.586	0.582	0.386
RGB	0.813	0.368	0.598	0.205	0.889	0.569	0.767	0.381
Img + Edge	0.863	0.590	0.882	0.334	0.878	0.878	**0.874**	0.386
Redetect	„	0.593	„	0.341	„	0.245	„	0.393
RGB + Redet. Edge	0.892	0.001	0.817	0.115	0.889	0.105	0.866	0.074
Redetect	„	0.265	„	0.656	„	0.326	„	**0.416**

Table 21: Performance of the Free Adversarial Training (FreeAT) method over three runs.

	Run 1		Run 1		Run 1		Average	
Model	Clean	PGD-10	Clean	PGD-10	Clean	PGD-10	Clean	PGD-10
Edge	0.674	0.672	0.704	0.702	0.660	0.659	0.679	**0.678**
RGB	0.783	0.450	0.768	0.450	0.772	0.447	0.774	0.449
Img + Edge	0.784	0.432	0.779	0.447	0.782	0.448	**0.782**	0.442
Redetect	„	0.447	„	0.448	„	0.449	„	0.448
RGB + Redet. Edge	0.776	0.451	0.776	0.454	0.780	0.447	0.777	0.451
Redetect	„	0.452	„	0.456	„	0.448	„	0.452

H Analysis of parameter α in Alg. 1 (EAT defense)

Table 22: Results (Top-1 acc.) over MNIST corresponding to $\alpha = 0$ (i.e., adversarial training only on adversarial examples taking part in the loss function). See also Table 1 in the main text.

	Rob. model (8)		**Rob. model (32)**		**Rob. model (64)**		**Average**
ϵ	0/clean	8	0/clean	32	0/clean	64	**Rob. models**
				FGSM			
Img+Edge	**0.963**	0.938	**0.959**	0.869	0.931	0.684	0.891
Redetect	"	0.943	„	0.887	„	0.727	0.902
Img + Redetected Edge	**0.963**	0.936	0.944	0.588	**0.937**	0.030	0.733
Redetect	"	**0.948**	„	**0.911**	„	**0.916**	**0.937**
				PGD-40			
Img+Edge	**0.966**	0.940	**0.960**	0.859	0.928	0.607	0.877
Redetect	"	**0.946**	„	0.883	„	0.657	0.890
Img + Redetected Edge	0.963	0.933	0.947	0.469	**0.936**	0.000	0.708
Redetect	"	**0.946**	„	**0.913**	„	**0.915**	**0.937**

Table 23: Results (Top-1 acc.) over Fashion MNIST corresponding to $\alpha = 0$ (i.e., adversarial training only on adversarial examples taking part in the loss function). See also Table 3 in the main text.

	Rob. model (8)		**Rob. model (32)**		**Rob. model (64)**		**Average**
ϵ	0/clean	8	0/clean	32	0/clean	64	**Rob. models**
				FGSM			
Img+Edge	0.756	0.701	0.732	0.619	0.683	0.487	0.663
Redetect	"	0.707	„	0.635	„	0.481	0.666
Img + Redetected Edge	**0.768**	0.705	**0.739**	0.481	**0.693**	0.040	0.571
Redetect	"	**0.727**	„	**0.660**	„	**0.635**	**0.704**
				PGD-40			
Img+Edge	0.768	0.702	0.749	0.573	0.718	0.432	0.657
Redetect	"	0.714	„	0.593	„	0.510	0.675
Img + Redetected Edge	**0.778**	0.702	**0.762**	0.414	**0.750**	0.001	0.568
Redetect	"	**0.725**	„	**0.632**	„	**0.615**	**0.710**