A Additional theoretical results

Definition A.1. Consider model A with input space $X \subseteq \mathbb{R}^{nd}$, previously observed data $x \sim X$, and k class centroids $c \in \mathbb{R}^{kd}$ learned by A. We define **domain shift** as an update to the class centroids $c \to c^* \in \mathbb{R}^{kd}$. **Domain shift sensitivity** is then the proportion of triplets flipped as a result of this update.

$$\sigma_A(c, c^*) := E[\frac{||S_A(x; c) - S_A(x; c^*)||_1}{|S_A(x; c)|}]$$

From this definition and Theorem 3.9, it immediately follows that sensitivity to domain shift should have the same U-shaped relationship with alignment that few-shot learning does in cases where the teacher model is robust to domain shift.

Corollary A.2. ***(Alignment and domain-shift robustness).*** *Consider input space $X \subseteq \mathbb{R}^{nd}$, shared data $x \sim X$, and three models, A, B_1, and B_2 with $D_P(A, B_1; X) = \epsilon_{B_1}$ and $D_P(A, B_2; X) = \epsilon_{B_2}$. Let $c \in \mathbb{R}^{kd}$ be k class centroids learned by A, B_1 and B_2. If $\sigma_A(c, c^*) = 0$ and $|0.5 - \epsilon_{B_1}| < |0.5 - \epsilon_{B_2}|$, then $\sigma_{B_1}(c, c^*) < \sigma_{B_2}(c, c^*)$.*

We can also use this framework to define robustness to adversarial examples. We assume that an adversarial example is an object that maximizes perceptual (i.e. representational) disagreement between the teacher and the student.

Definition A.3. Consider input space $X \subseteq \mathbb{R}^{nd}$, shared data $x \sim X$, and two models, A and B, with $D_P(A, B_1; X) = \epsilon_B$. An **adversarial example** is an object $e \in \mathbb{R}^d$ that maximizes disagreement between A and B on $S(x; e)$, the subset of $n(n-1)/2$ triplets relating the objects in x to e.

$$e = \max_X ||S_A(x; e) - S_B(x; e)||_1 \tag{1}$$

Using Definition 3.6 we immediately get the following result.

Lemma A.4. *Consider an input space $X \subseteq \mathbb{R}_{nd}$, and two agents, A and B. $D_P(A, B; X) = E[\frac{||S_A(X) - S_B(X)||_1}{n(n-1)(n-2)/2}]$.*

We can now show that a model that is more aligned with the teacher will, on average, also be more robust to adversarial examples.

Theorem A.5. ***(Alignment and adversarial robustness).*** *Consider input space $X \subseteq \mathbb{R}^{nd}$, shared data $x \sim X$, and three models, A, B_1, and B_2 with $D_P(A, B_1; X) = \epsilon_{B_1}$ and $D_P(A, B_2; X) = \epsilon_{B_2}$. If $\epsilon_{B_1} < \epsilon_{B_2}$, then $E[\max_{e \in x} ||S_A(x; e) - S_{B_1}(x; e)||_1] < E[\max_{e \in X} ||S_A(x; e) - S_{B_2}(x; e)||_1]$.*

Proof. Note that for a set of k binomial random variables $X_i \sim Bin(n, p)$, the expectation of the k-th order statistic is $E[X_{(k)}] = \sum_{x=0}^{n}(1 - F(x; n, p)^k)$ where $F(x; n, p) = P(X_i \leq x)$. In the case of adversarial examples, let X_i be a random variable corresponding to the set of objects sampled uniformly from the input space $X \subseteq \mathbb{R}^{nd}$ then $U = ||S_A(X; e) - S_{B_1}(X; e)||_1, Y \sim Bin(n(n-1)/2, \epsilon_{B_1})$ and similarly $V = ||S_A(X; e) - S_{B_2}(X; e)||_1, V \sim Bin(n(n-1)/2, \epsilon_{B_1})$. In that case, the expected disagreement of A and B_1 on an adversarial example is $E[U_{(n)}] = \sum_{x=0}^{n(n-1)/2}(1 - F(x; n(n-1)/2, \epsilon_{B_1})^n)$ and for A and B_2 it is $E[V_{(n)}] = \sum_{x=0}^{n(n-1)/2}(1 - F(x; n(n-1)/2, \epsilon_{B_2})^n)$. If $\epsilon_{B_1} < \epsilon_{B_2}$, then $F(x; n(n-1)/2, \epsilon_{B_1}) > F(x; n(n-1)/2, \epsilon_{B_2})$ and thus $E[U_{(n)}] < E[V_{(n)}]$. □

Remark A.6. While this theorem shows that increased alignment generally leads to increased adversarial robustness, this relies on a representational metric of adversarial examples. However, in practice, adversarial robustness is often measured using hard classification error as a simple proxy. This proxy does not capture the fine-grained degree of misalignment between humans and a model on each example. As a result, when measuring adversarial robustness using this proxy, the effect of alignment may be dampened by the U-shaped effect seen in other classification settings as mentioned above.

B List of 491 models used in experiments

adv_inception_v3, bat_resnext26ts, beit_base_patch16_224, beit_base_patch16_384, beit_large_patch16_224, beit_large_patch16_384, botnet26t_256, cait_s24_224, cait_s24_384, cait_s36_384, cait_xs24_384, cait_xxs24_224, cait_xxs24_384, cait_xxs36_224, cait_xxs36_384, coat_lite_mini, coat_lite_small, coat_lite_tiny, coat_mini, coat_tiny, convit_base, convit_small, convit_tiny, convmixer_1024_20_ks9_p14, convmixer_1536_20, convmixer_768_32, convnext_base, convnext_base_384_in22ft1k, convnext_base_in22ft1k, convnext_large, convnext_large_384_in22ft1k, convnext_large_in22ft1k, convnext_small, convnext_tiny, cspdarknet53, cspresnet50, cspresnext50, deit_base_patch16_224, deit_base_patch16_384, deit_small_patch16_224, deit_tiny_patch16_224, densenet121, densenet161, densenet169, densenet201, densenetblur121d, dla102, dla102x, dla102x2, dla169, dla34, dla46_c, dla46x_c, dla60, dla60_res2net, dla60_res2next, dla60x, dla60x_c, dm_nfnet_f0, dm_nfnet_f1, dm_nfnet_f2, dpn107, dpn131, dpn68, dpn68b, dpn92, dpn98, eca_botnext26ts_256, eca_halonext26ts, eca_nfnet_l0, eca_nfnet_l1, eca_nfnet_l2, eca_resnet33ts, eca_resnext26ts, ecaresnet101d, ecaresnet101d_pruned, ecaresnet269d, ecaresnet26t, ecaresnet50d, ecaresnet50d_pruned, ecaresnet50t, ecaresnetlight, efficientnet_b0, efficientnet_b1, efficientnet_b1_pruned, efficientnet_b2, efficientnet_b2_pruned, efficientnet_b3, efficientnet_b3_pruned, efficientnet_b4, efficientnet_el, efficientnet_el_pruned, efficientnet_em, efficientnet_es, efficientnet_es_pruned, efficientnet_lite0, efficientnetv2_rw_m, efficientnetv2_rw_s, efficientnetv2_rw_t, ens_adv_inception_resnet_v2, ese_vovnet19b_dw, ese_vovnet39b, fbnetc_100, fbnetv3_b, fbnetv3_d, fbnetv3_g, gc_efficientnetv2_rw_t, gcresnet33ts, gcresnet50t, gcresnext26ts, gcresnext50ts, gernet_l, gernet_m, gernet_s, ghostnet_100, gluon_inception_v3, gluon_resnet101_v1b, gluon_resnet101_v1c, gluon_resnet101_v1d, gluon_resnet101_v1s, gluon_resnet152_v1b, gluon_resnet152_v1c, gluon_resnet152_v1d, gluon_resnet152_v1s, gluon_resnet18_v1b, gluon_resnet34_v1b, gluon_resnet50_v1b, gluon_resnet50_v1c, gluon_resnet50_v1d, gluon_resnet50_v1s, gluon_resnext101_32x4d, gluon_resnext101_64x4d, gluon_resnext50_32x4d, gluon_senet154, gluon_seresnext101_32x4d, gluon_seresnext101_64x4d, gluon_seresnext50_32x4d, gluon_xception65, gmixer_24_224, gmlp_s16_224, halo2botnet50ts_256, halonet26t, halonet50ts, haloregnetz_b, hardcorenas_a, hardcorenas_b, hardcorenas_c, hardcorenas_d, hardcorenas_e, hardcorenas_f, hrnet_w18, hrnet_w18_small, hrnet_w18_small_v2, hrnet_w30, hrnet_w32, hrnet_w40, hrnet_w44, hrnet_w48, hrnet_w64, ig_resnext101_32x16d, ig_resnext101_32x8d, inception_resnet_v2, inception_v3, inception_v4, jx_nest_base, jx_nest_small, jx_nest_tiny, lambda_resnet26rpt_256, lambda_resnet26t, lambda_resnet50ts, lamhalobotnet50ts_256, lcnet_050, lcnet_075, lcnet_100, legacy_senet154, legacy_seresnet101, legacy_seresnet152, legacy_seresnet18, legacy_seresnet34, legacy_seresnet50, legacy_seresnext101_32x4d, legacy_seresnext26_32x4d, legacy_seresnext50_32x4d, mixer_b16_224, mixer_b16_224_miil, mixnet_l, mixnet_m, mixnet_s, mixnet_xl, mnasnet_100, mnasnet_small, mobilenetv2_050, mobilenetv2_100, mobilenetv2_110d, mobilenetv2_120d, mobilenetv2_140, mobilenetv3_large_100, mobilenetv3_large_100_miil, mobilenetv3_rw, nasnetalarge, nf_regnet_b1, nf_resnet50, nfnet_l0, pit_b_224, pit_s_224, pit_ti_224, pit_xs_224, pnasnet5large, regnetx_002, regnetx_004, regnetx_006, regnetx_008, regnetx_016, regnetx_032, regnetx_040, regnetx_064, regnetx_080, regnetx_120, regnetx_160, regnetx_320, regnety_002, regnety_004, regnety_006, regnety_008, regnety_016, regnety_032, regnety_040, regnety_064, regnety_080, regnety_120, regnety_160, regnety_320, regnetz_b16, regnetz_c16, regnetz_d32, regnetz_d8, regnetz_e8, repvgg_a2, repvgg_b0, repvgg_b1, repvgg_b1g4, repvgg_b2, repvgg_b2g4, repvgg_b3, repvgg_b3g4, res2net101_26w_4s, res2net50_14w_8s, res2net50_26w_4s, res2net50_26w_6s, res2net50_26w_8s, res2net50_48w_2s, res2next50, resmlp_12_224, resmlp_12_distilled_224, resmlp_24_224, resmlp_24_distilled_224, resmlp_36_224, resmlp_36_distilled_224, resmlp_big_24_224, resmlp_big_24_224_in22ft1k, resmlp_big_24_distilled_224, resnest101e, resnest14d, resnest200e, resnest269e, resnest26d, resnest50d, resnest50d_1s4x24d, resnest50d_4s2x40d, resnet101, resnet101d, resnet152, resnet152d, resnet18, resnet18d, resnet200d, resnet26, resnet26d, resnet26t, resnet32ts, resnet33ts, resnet34, resnet34d, resnet50, resnet50_gn, resnet50d, resnet51q, resnet61q, resnetblur50, resnetrs101, resnetrs152, resnetrs200, resnetrs270, resnetrs350, resnetrs420, resnetrs50, resnetv2_101, resnetv2_101x1_bitm, resnetv2_50, resnetv2_50x1_bit_distilled, resnetv2_50x1_bitm, resnext101_32x8d, resnext26ts, resnext50_32x4d, resnext50d_32x4d, rexnet_100, rexnet_130, rexnet_150, rexnet_200, sebotnet33ts_256, sehalonet33ts, selecsls42b, selecsls60, selecsls60b, semnasnet_075, semnasnet_100, seresnet152d, seresnet33ts, seresnet50, seresnext26d_32x4d, seresnext26t_32x4d, seresnext26ts, seresnext50_32x4d, skresnet18, skresnet34, skresnext50_32x4d, spnasnet_100, ssl_resnet18, ssl_resnet50, ssl_resnext101_32x16d, ssl_resnext101_32x4d, ssl_resnext101_32x8d, ssl_resnext50_32x4d, swin_base_patch4_window12_384, swin_base_patch4_window7_224, swin_large_patch4_window12_384, swin_large_patch4_window7_224, swin_small_patch4_window7_224, swin_tiny_patch4_window7_224, swsl_resnet18, swsl_resnet50, swsl_resnext101_32x16d, swsl_resnext101_32x4d, swsl_resnext101_32x8d, swsl_resnext50_32x4d, tf_efficientnet_b0, tf_efficientnet_b0_ap, tf_efficientnet_b0_ns, tf_efficientnet_b1, tf_efficientnet_b1_ap, tf_efficientnet_b1_ns, tf_efficientnet_b2, tf_efficientnet_b2_ap, tf_efficientnet_b2_ns, tf_efficientnet_b3, tf_efficientnet_b3_ap, tf_efficientnet_b3_ns, tf_efficientnet_b4, tf_efficientnet_b4_ap, tf_efficientnet_b4_ns, tf_efficientnet_b5, tf_efficientnet_b5_ap, tf_efficientnet_b5_ns, tf_efficientnet_b6, tf_efficientnet_b6_ap, tf_efficientnet_b6_ns, tf_efficientnet_b7, tf_efficientnet_b7_ap, tf_efficientnet_b7_ns, tf_efficientnet_cc_b0_4e, tf_efficientnet_cc_b0_8e, tf_efficientnet_cc_b1_8e, tf_efficientnet_el, tf_efficientnet_em, tf_efficientnet_es, tf_efficientnet_lite0, tf_efficientnet_lite1, tf_efficientnet_lite2, tf_efficientnet_lite3, tf_efficientnet_lite4, tf_efficientnetv2_b0, tf_efficientnetv2_b1, tf_efficientnetv2_b2, tf_efficientnetv2_b3, tf_efficientnetv2_l, tf_efficientnetv2_l_in21ft1k, tf_efficientnetv2_m, tf_efficientnetv2_m_in21ft1k, tf_efficientnetv2_s, tf_efficientnetv2_s_in21ft1k, tf_inception_v3, tf_mixnet_l, tf_mixnet_m, tf_mixnet_s, tf_mobilenetv3_large_075, tf_mobilenetv3_large_100, tf_mobilenetv3_large_minimal_100, tf_mobilenetv3_small_075, tf_mobilenetv3_small_100, tf_mobilenetv3_small_minimal_100, tinynet_a, tinynet_b, tinynet_c, tinynet_d, tinynet_e, tnt_s_patch16_224, tv_densenet121, tv_resnet101, tv_resnet152, tv_resnet34, tv_resnet50, tv_resnext50_32x4d, twins_pcpvt_base, twins_pcpvt_large, twins_pcpvt_small, twins_svt_base, twins_svt_large, twins_svt_small, vgg11, vgg11_bn, vgg13, vgg13_bn, vgg16, vgg16_bn, vgg19, vgg19_bn, visformer_small, vit_base_patch16_224, vit_base_patch16_224_miil, vit_base_patch16_384, vit_base_patch32_224, vit_base_patch32_384, vit_base_patch8_224, vit_base_r50_s16_384, vit_small_patch16_224, vit_small_patch16_384, vit_small_patch32_224, vit_small_patch32_384, vit_small_r26_s32_224, vit_small_r26_s32_384, vit_tiny_patch16_224, vit_tiny_patch16_384, vit_tiny_r_s16_p8_224, vit_tiny_r_s16_p8_384, wide_resnet101_2, wide_resnet50_2, xception, xception41, xception65, xception71, xcit_large_24_p16_224, xcit_large_24_p16_224_dist, xcit_large_24_p16_384_dist, xcit_large_24_p8_224, xcit_large_24_p8_224_dist, xcit_large_24_p8_384_dist, xcit_medium_24_p16_224, xcit_medium_24_p16_224_dist, xcit_medium_24_p16_384_dist, xcit_medium_24_p8_224, xcit_medium_24_p8_224_dist, xcit_nano_12_p16_224, xcit_nano_12_p16_224_dist, xcit_nano_12_p16_384_dist, xcit_nano_12_p8_224, xcit_nano_12_p8_224_dist, xcit_nano_12_p8_384_dist, xcit_small_12_p16_224, xcit_small_12_p16_224_dist, xcit_small_12_p16_384_dist, xcit_small_12_p8_224, xcit_small_12_p8_224_dist, xcit_small_12_p8_384_dist, xcit_small_24_p16_224, xcit_small_24_p16_224_dist, xcit_small_24_p16_384_dist, xcit_small_24_p8_224, xcit_small_24_p8_224_dist, xcit_small_24_p8_384_dist, xcit_tiny_12_p16_224, xcit_tiny_12_p16_224_dist, xcit_tiny_12_p16_384_dist, xcit_tiny_12_p8_224, xcit_tiny_12_p8_224_dist, xcit_tiny_12_p8_384_dist, xcit_tiny_24_p16_224, xcit_tiny_24_p16_224_dist, xcit_tiny_24_p16_384_dist, xcit_tiny_24_p8_224, xcit_tiny_24_p8_224_dist, xcit_tiny_24_p8_384_dist

C Reproducibility: Code and results data

All code and full results data are provided as part of the supplemental information. We will share them publicly after the anonymity period is over. All experiments were conducted on an AWS "x1.16xlarge" instance (no GPUs).