1 SUPPLEMENTARY MATERIAL

In the supplementary material we provide additional studies and results that further explore the results found in the main body of the paper. In Subsection 1.1 we provide numerical results and further analysis for the transferability experiments found in **??**. In Subsection 1.2 we provide a more explicit attack setup for the experiments we performed throughout the paper. In Subsection 1.3 we provide further studies and analysis upon the GaME framework including comparisons to uniform probability distribution ensemble defenses, and a study on the effect of n on a GaME$_n$ defense. In Subsection 1.4 we provide a study upon the computational cost of the GaME$_n$ framework. In Subsection 1.5 we provide a study the effect of the number of samples, N, used to approximate the game matrix upon the robust accuracy of a GaME$_n$ ensemble. In Subsection 1.6 we provide explicit forms for the linear programs one must solve to create a GaME$_n$ defense or attack ensemble. In Subsection 1.7 we provide a brief study upon the effectiveness of AE-SAGA when attacking 3 defenses simultaneously. Lastly, in Subsection 1.8 we provide an approximate form for the update formula of MIME seen in Equation **??**.

1.1 TRANSFERABILITY EXPERIMENTS

Transferability Between Defenses (CIFAR-10)									
Attacked	B1	B5	B10	RF	VF	ST	SB	BVT	VRT
B1	-	45.80%	67.20%	99.60%	99.20%	84.10%	93.10%	95.90%	89.90%
B5	4.30%	-	44.40%	98.70%	97.90%	71.30%	90.50%	89.70%	89.00%
B10	18.60%	40.00%	-	83.20%	90.80%	66.50%	77.50%	82.50%	70.20%
RF	91.80%	88.40%	82.20%	-	87.90%	68.60%	69.80%	80.20%	66.60%
VF	52.40%	55.70%	55.70%	86.00%	-	63.00%	62.90%	21.70%	74.00%
ST	91.50%	90.50%	89.40%	98.90%	98.90%	-	91.80%	91.80%	91.30%
SB	92.20%	89.10%	86.30%	93.90%	95.20%	64.90%	-	82.90%	84.70%
BVT	60.40%	69.90%	75.10%	98.20%	95.60%	78.10%	90.00%	-	90.80%
VRT	86.80%	83.40%	89.80%	82.30%	87.20%	76.00%	71.00%	78.80%	-

Table 1: Full numerical results for the transfer study shown pictorially in Figure **??**

Previously we gave a pictoral representation of the transferability results for CIFAR-10. Above we include the full results from our transferability study in Table 1. Each of the defense names have been abbreviated in the interest of space: Bn is BaRT-n, RF is the FAT trained ResNet-164, VF is the FAT trained ViT-L, ST is the transfer SNN, SB is the Back-Prop SNN, VRT is the Vgg + ResNet TiT defense, and BVT is the ViT+BiT TiT defense.

We first chose 1000 class-wise balanced, clean images from the testing set of CIFAR-10. We additionally constrained these 1000 samples to be those which are correctly identified by every model in the table. For random transform defenses, each image is classified correctly with a probability of at least 98%. We then attacked each defense with APGD, every randomized defense with MIME, and every non-randomized defense with MIM. For APGD, MIM, and MIME the parameters can be seen in Subsection 1.2. From each defense we chose the adversarial samples generated by the attack with the highest attack success rate to be used in the transfer study.

Analysis of Results From the transferability table it becomes clear that there is generally very low transferability between each pair of defenses which use different classifier architectures. For instance, attacks generated on the BiT based BaRT models have a very low level of transferability with the ViT based FAT-ViT and TiT defense using BiT+ViT. In contrast to this, defenses which share architectures, unsurprisingly, have relatively high levels of transferability. One notable example of this is the attacks generated on the FAT-ViT models when transfered to the TiT defense using BiT-ViT. There are some exceptions to this idea however, as the BaRT models seem to be highly susceptible to attacks generated on the ViT based defenses such as those generated on the FAT trained ViT-L.

1.2 GAME EXPERIMENTAL SETUP

For each version of GaME$_n$ played on CIFAR-10 we chose to employ the following defenses: BaRT-1, BaRT-5, TiT using BiT and ViT, FAT ResNet-164, FAT ViT-L-16, SEW SNN, and Transfer SNN.

BaRT-1 and BaRT-5 were chosen in favor of BaRT-10 due to the computational cost of computing all the necessary attacks on BaRT-10. Additionally, we chose the Bit+ViT version of TiT since the original, VGG-ResNet, architecture that was proposed has a significantly lower clean accuracy. For Tiny ImageNet we chose BaRT-1, BaRT-5, TiT using BiT and ViT, and FAT ViT-L-16.

We then attacked every random transform defense with MIME, every non-random-transform defense with APGD, and each pair of defenses with AE-SAGA. This came to a total of 28 attacks on CIFAR-10 and 10 attacks on Tiny ImageNet. Every attack was run with respect to the l_∞ norm. The hyper parameters for each attack are seen in Table 2:

Attack	ϵ	ϵ_{step}	Attack Steps	N	γ	Fitting Factor	α Learning Rate
APGD	.031	.005	20	-	-	-	-
MIM	.031	.0031	10	-	.5	-	-
MIME	.031	.0031	10	10	.5	-	-
AE-SAGA	.031	.005	40	4	.5	50	10000

Table 2: Attack parameters for each of the attacks used in the paper. Here γ represents the momentum decay rate, ϵ represents the maximum allowed perturbation magnitude, α represents the weights used in the AE-SAGA algorithm, and N represents the number of EOT samples taken. Note for APGD that the ϵ_{step} value presented is only the initial value and is subject to change according to the attack's algorithm.

AE-SAGA has the largest number of iterations since the added complexity of attacking two models simultaneously requires additional attack steps to converge. MIME was given less attack steps due to the computational cost of the attack.

For each attack we first chose an arbitrary, but class-wise balanced set of 1000 clean images from the testing set of each respective data set. From this subset we generated 1000 adversarial examples for each attack. We used 800 of these samples to create the payoff matrix R, then evaluated the ensemble using the remaining 200, class-wise balanced samples from each attack.

1.3 Full Experimental Results for GaME

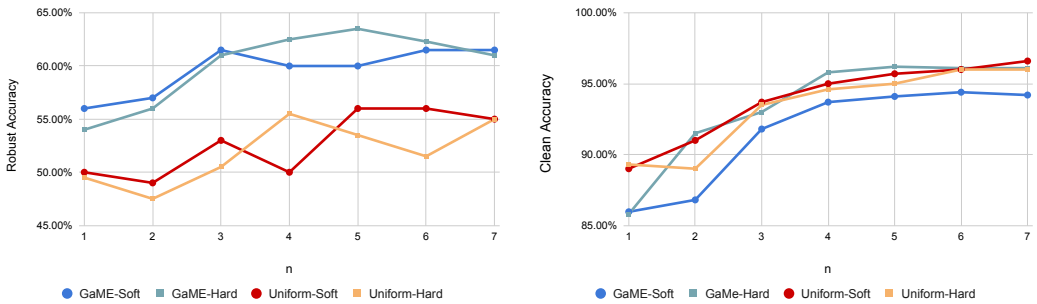

Figure 1: CIFAR-10

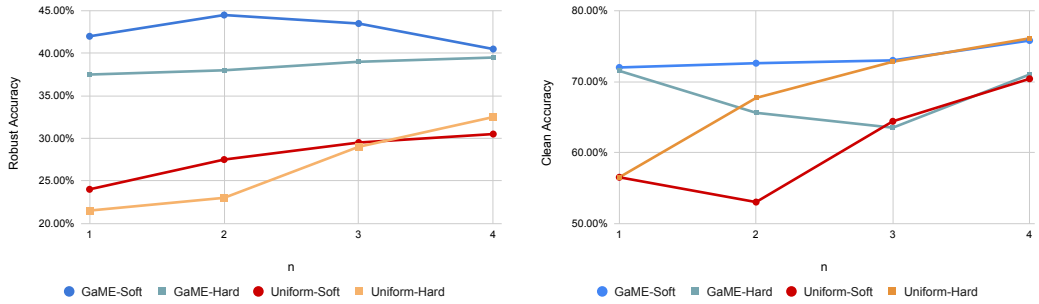

Figure 2: Tiny ImageNet

In Figure 1 and Figure 2 we compare the robust and clean accuracy for GaME generated ensemble defenses and those with uniform probability distributions over all defender strategies. We further expand the study to consider all values of $1 \geq n \leq |D|$ for each dataset. The results show that the GaME framework clearly improves upon the robustness of the uniform probability distribution ensemble while maintaining a high level of clean accuracy. The results additionally show that the clean and robust accuracy both trend in a positive direction as one increases n.

n	G-Soft r*	G-Soft Robust	G-Soft Clean	G-Hard r*	G-Hard Robust	G-Hard Clean
1	56.00%	56.00%	85.96%	57.20%	54.00%	85.80%
2	57.00%	57.00%	86.80%	60.00%	56.00%	91.50%
3	61.30%	61.50%	91.80%	61.50%	61.00%	93.00%
4	61.70%	60.00%	93.70%	62.10%	62.50%	95.80%
5	61.80%	60.00%	94.10%	62.30%	63.50%	96.20%
6	61.84%	61.50%	94.40%	62.30%	62.30%	96.10%
7	61.84%	61.50%	94.20%	62.30%	61.00%	96.10%

U-Hard Robust	U-Hard Clean	U-Soft Robust	U-Soft Clean
49.50%	89.30%	50.00%	89.00%
47.50%	89.00%	49.00%	91.00%
50.50%	93.50%	53.00%	93.70%
55.50%	94.60%	50.00%	95.00%
53.50%	95.00%	56.00%	95.70%
51.50%	96.00%	56.00%	96.00%
55.00%	96.00%	55.00%	96.60%

Table 3: (CIFAR-10) Full numerical results for Figure 1. Here G-Hard and G-Soft refer to GaME generated ensembles utilizing F_h and F_s as their voting functions respectively. Similarly, U-Hard and U-Soft represent ensemble defenses with uniform probability distributions utilizing F_h and F_s as their voting functions respectively

n	G-Soft r*	G-Soft Robust	G-Soft Clean	G-Hard r*	G-Hard Robust	G-Hard Clean
1	36.90%	42.00%	72.00%	36.90%	39.50%	71.52%
2	41.75%	42.00%	72.60%	34.00%	40.00%	65.60%
3	42.10%	43.00%	73.00%	34.70%	38.50%	63.50%
4	32.00%	28.00%	75.80%	33.00%	35.50%	71.00%

n	U-Hard Robust	U-Hard Clean	U-Soft Robust	U-Soft Clean
1	21.50%	56.50%	24.00%	56.50%
2	23.00%	53.00%	27.50%	67.70%
3	29.00%	64.40%	29.50%	72.81%
4	32.50%	0.704	30.50%	76.11%

Table 4: (Tiny ImageNet) Full numerical results for Figure 2. Here the notation is consistent with Table 3

Full $GaME_1$ Table (CIFAR-10)							
	B1	B5	RF	VF	ST	SB	BVT
APGD(RF)	90.75%	86.75%	50.25%	81.00%	59.13%	58.13%	74.88%
APGD(SB)	90.25%	89.75%	79.63%	88.88%	55.25%	1.50%	79.50%
APGD(ST)	91.25%	88.88%	84.13%	93.50%	0.00%	81.13%	88.12%
APGD(VF)	54.37%	54.75%	72.50%	29.50%	55.37%	57.00%	21.88%
MIME(B1)	3.13%	44.62%	84.38%	94.00%	74.88%	82.75%	92.87%
MIME(B5)	4.62%	16.38%	83.50%	92.37%	63.50%	81.37%	87.00%
MIME(VBT)	58.13%	68.87%	83.75%	91.50%	69.87%	80.75%	9.75%
S(B1,B5)	5.00%	37.87%	83.87%	93.87%	74.25%	83.25%	90.87%
S(B1,RF)	11.25%	64.38%	59.13%	89.00%	69.37%	75.25%	87.50%
S(B1,SB)	8.87%	58.37%	80.25%	90.12%	61.25%	11.13%	85.62%
S(B1,ST)	14.75%	71.50%	83.00%	93.25%	0.13%	81.50%	89.12%
S(B1,VBT)	6.25%	55.37%	84.13%	93.50%	70.75%	83.37%	35.50%
S(B1,VF)	5.00%	51.38%	80.00%	71.75%	68.37%	76.00%	76.88%
S(B5,RF)	17.62%	36.13%	59.38%	88.50%	65.00%	73.62%	85.50%
S(B5,SB)	16.38%	29.00%	80.00%	90.63%	55.37%	9.88%	82.00%
S(B5,ST)	58.75%	59.25%	83.25%	93.50%	0.25%	82.75%	87.12%
S(B5,VBT)	15.63%	34.38%	83.87%	93.37%	69.50%	82.50%	32.75%
S(B5,VF)	12.88%	30.50%	79.75%	68.87%	65.75%	75.62%	70.63%
S(RF,SB)	88.50%	84.88%	55.75%	82.37%	54.00%	3.00%	76.38%
S(RF,ST)	92.13%	90.12%	70.00%	91.37%	0.13%	76.12%	83.75%
S(RF,VBT)	70.50%	76.88%	62.38%	83.37%	61.25%	69.50%	6.12%
S(RF,VF)	63.00%	61.00%	57.13%	39.25%	53.87%	56.00%	31.87%
S(SB,ST)	92.75%	90.50%	82.50%	92.50%	0.00%	33.87%	86.75%
S(SB,VBT)	78.00%	78.63%	80.00%	88.75%	53.75%	6.00%	15.00%
S(ST,VBT)	89.88%	88.62%	83.87%	93.63%	0.37%	81.37%	23.75%
S(VF,SB)	83.00%	82.37%	77.50%	75.62%	54.13%	0.88%	68.50%
S(VF,ST)	86.38%	83.50%	82.50%	86.50%	0.63%	80.13%	78.75%
S(VF,VBT)	54.75%	62.00%	76.25%	55.13%	60.12%	68.50%	3.00%

Table 5: Here we present the full results from $GaME_1$ performed on CIFAR-10. This extensive study shows us that, for any attack, there is going to be a defense that can counter it, and vice versa. For instance, MIME(B5) is extremely effective against BaRT-1 and BaRT-5, yet it fails to fool the ViT-FAT model 92% of the time. Here we use S(A,B) to denote the AE-SAGA attack run against defenses A and B

Defender Mixed Nash Strategy (CIFAR-10, GaME$_1$)							
Defense	B1	B5	RF	VF	ST	SB	BVT
λ^D	0	.16	.75	.02	0	.07	0
Min Robust	3%	16%	50%	29%	0%	1%	3%
Defender Mixed Nash Strategy (Tiny ImageNet, GaME$_1$)							
λ^D	0.152	0.443	-	0.243	-	-	0.162
Min Robust	4.75%	10.62%	-	3.87%	-	-	1.87%

Table 6: $GaME_1$ Defender results for CIFAR-10 and Tiny-ImageNet. Full numerical results corresponding to the robust accuracy for the individual attacks (APGD, MIM, MIME and AE-SAGA) and the clean accuracy of different defense are given in our supplementary material.

Attacker Mixed Nash Strategy (CIFAR-10, GaME$_1$)				
Attack	S(B5,RF)	APGD(RF)	S(RF,VF)	S(RF,SB)
λ^A	0.265	0.051	0.618	0.066
Max Robust	89%	91%	63%	89%
Attacker Mixed Nash Strategy (Tiny ImageNet, GaME$_1$)				
Attack	S(B1,BVT)	S(B1,VF)	MIME(B5)	S(B5,BVT)
λ^A	0.224	0.167	0.021	0.588
Max Robust	36%	56%	62%	38%

Table 7: $GaME_1$ Attacker results for CIFAR-10 and Tiny-ImageNet. Attacks which have a probability of 0 in the mixed strategy are not represented here.

Robust Accuracy of Ensemble Generated By GaME$_1$ (CIFAR-10)						
APGD(RF)	APGD(SB)	APGD(ST)	APGD(VF)	MIME(B1)	MIME(B5)	MIME(BVT)
57.00%	80.50%	89.50%	66.50%	83.50%	72.50%	84.50%
S(B5,SB)	S(B5,ST)	S(B5,BVT)	S(B5,VF)	S(RF,SB)	S(RF,ST)	S(RF,BVT)
74.00%	80.00%	76.00%	73.00%	58.00%	76.00%	60.50%
S(B1,B5)	S(B1,RF)	S(B1,SB)	S(B1,ST)	S(B1,BVT)	S(B1,VF)	S(B5,RF)
74.50%	58.50%	76.50%	83.00%	81.50%	74.00%	57.00%
S(RF,VF)	S(SB,ST)	S(SB,BVT)	S(ST,BVT)	S(VF,SB)	S(VF,ST)	S(VF,BVT)
59.00%	82.50%	81.50%	86.00%	76.00%	83.00%	77.00%

Table 8: Here we list the robust accuracy of our ensemble generated by GaME$_1$ when evaluated on CIFAR-10 attack samples that were not used in the formulation of the game's linear program. If the attacker plays a best response strategy to the ensemble like APGD(RF), they can expect to lower the ensemble's robust accuracy to 57%. This is very close to the value $r^* = .573$ for the expected, guaranteed robustness of the ensemble. We can expect that with a larger sample number, i.e. generating more than 1000 adversarial examples per attack, that these values will get closer due to the law of large numbers.

Robust Accuracy of Ensemble With Uniform Weights in GaME$_1$ (CIFAR-10)						
APGD(RF)	APGD(SB)	APGD(ST)	APGD(VF)	MIME(B1)	MIME(B5)	MIME(VBT)
71.00%	69.50%	72.00%	51.50%	70.00%	59.00%	68.00%
S(B5,SB)	S(B5,ST)	S(B5,VBT)	S(B5,VF)	S(RF,SB)	S(RF,ST)	S(RF,VBT)
50.50%	66.00%	56.50%	56.00%	60.50%	76.50%	63.50%
S(B1,B5)	S(B1,RF)	S(B1,SB)	S(B1,ST)	S(B1,VBT)	S(B1,VF)	S(B5,RF)
65.00%	63.00%	55.00%	64.00%	57.50%	60.50%	57.50%
S(RF,VF)	S(SB,ST)	S(SB,VBT)	S(ST,VBT)	S(VF,SB)	S(VF,ST)	S(VF,VBT)
49.50%	61.50%	60.00%	69.50%	64.50%	74.00%	49.00%

Table 9: Here we show the robust accuracy for an ensemble defense using single model predictions and a uniform weighting for all of the defenses in its ensemble.

Full $GaME_1$ Table (Tiny ImageNet)				
Attack	B1	B5	VF	BVT
MIME(B1)	4.75%	34.63%	31.13%	66.00%
MIME(B5)	20.75%	10.62%	29.88%	62.38%
APGD(VF)	41.13%	36.00%	9.00%	5.75%
MIME(BVT)	51.75%	49.25%	29.63%	1.87%
S(B1,B5)	5.37%	20.62%	30.88%	66.50%
S(B1,VF)	4.88%	31.25%	5.87%	56.75%
S(B1,BVT)	5.12%	36.13%	29.50%	8.00%
S(B5,BVT)	38.37%	22.75%	30.88%	11.25%
S(B5,VF)	37.87%	19.63%	8.62%	63.12%
S(VF,BVT)	47.88%	44.87%	3.87%	3.00%

Table 10: Here we provide the full utility matrix for the defender when creating a GaME$_1$ defense on Tiny ImageNet.

Robust Accuracy of Ensemble Generated By $GaME_1$ (Tiny ImageNet)				
MIME(B1)	MIME(B5)	APGD(VF)	MIME(BVT)	S(B1,B5)
35.00%	26.50%	44.00%	47.00%	34.50%
S(B1,VF)	S(B1,BVT)	S(B5,VF)	S(B5,BVT)	S(VF,BVT)
27.00%	28.00%	31.00%	20.50%	32.50%

Table 11: Here we list the robust accuracy of our ensemble generated by GaME$_1$ when evaluated on Tiny ImageNet attack samples that were not used in the formulation of the game's linear program.

1.4 Study of Computational Cost

As one increases the value of n in a GaME$_n$ defense, the number of possible choices for the defender grows in accordance to the binomial coefficient, $\binom{|D|}{n}$. Thus, here we will provide a brief study on the effect of n on the computational complexity of creating a GaME$_n$ defense.

The computation time for forming the game-matrix largely depends on the time needed to compute the predictions of each of the defenses for each of the attacks, as seen in Figure 4. We can save a large amount of time by running each set of samples, s_i, through each defense $d \in D$ once, receiving output $y_{i,d}$ for each sample, defense pair. To get the robust accuracy of $U \subset D$ when evaluating samples s_i we can substitute $y_{i,d}$ for $d(s_i)$ in the computation of $F_h(s_i, U)$ or $F_s(s_i, U)$ Equation **??** Equation **??**. This means that we do not need to perform a number of model evaluations that scales with n.

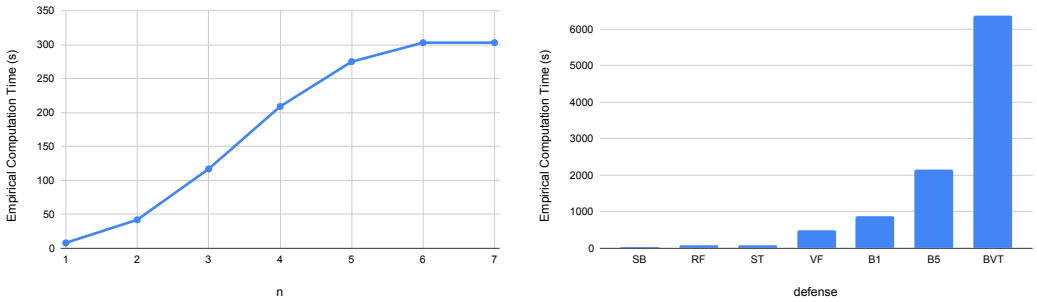

Figure 3: (CIFAR-10) Computational cost of creating the game matrix and solving the associated linear program for a GaME$_n$ ensemble as a function of n. Model prediction time is not considered in the calculation as it does not depend on n.

Figure 4: (CIFAR-10) Computational cost of evaluating every single-model defense in the study on all 22,400 adversarial samples generated for creating the game matrix of a GaME$_n$ ensemble.The total computation time amounts to 2.9 hours of which 3% of the time, 5 minutes, is spent evaluating each defense subset, as described above, and solving the game matrix for GaME$_7$.

The computational cost for evaluating the BaRT defense are high since a series of random transformations must be applied to the images before a prediction can be made. These transformations run on CPU and are run in parallel on a per-sample basis in order to speed up computation time. The computational cost for evaluating the BiT-ViT Trash is Treasure defense is the highest since it requires running a 13 step PGD attack against a large BiT model for each sample before it is given to the main ViT classifier.

These experiments were run on a computer with the following specifications: Intel Core i9-10900K CPU @ 3.70GHz, Nvidia RTX3080 12Gb, and 64Gb RAM.

1.5 Study on the Effects of Sample Number for GaME

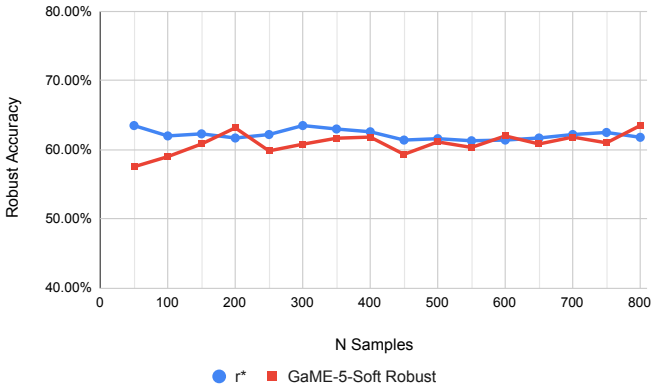

Figure 5: (CIFAR-10) Study on the effect of N on the expected robust accuracy, r^*, in blue, and the empirical robust accuracy, in red. Results are averaged over 3 trials and measured with respect to 200 samples from each attack in the study, as described in Section 1.2.

Here we provide a brief study on the effects of N on the effectiveness of a GaME defense. We choose GaME$_5$ with voting function F_s since it had the highest robust accuracy in our experiments. For most values of N we analyzed the difference between r^* and the empirical robustness was very small, in fact the average difference over all N was only 1.73%. The results show that one can

achieve reasonable results with N as low as 150 on CIFAR-10. This implies that one can create a GaME ensemble defense without the large computational cost of running dozens of attacks against their defenses. However, it is likely that this will not hold for more complicated data sets with a larger number of classes such as ImageNet or even CIFAR-100.

1.6 Linear Programs for Solving GaME

Here we present the explicit linear program for solving GaME as the attacker. Let $O_A = (\lambda^A_{a_1}, \cdots \lambda^A_{a_{|A|}}, r^*)$ be the row vector containing the elements of λ^A and r^*. Additionally let $\hat{0}$ denote the zero vector. The attacker must then solve the following linear program:

$$\begin{gathered} \max \begin{pmatrix} 0 & \cdots & 0 & 1 \end{pmatrix} O_A^T \\ \text{Subject to: } \begin{pmatrix} -r_{d_1,a_1} & -r_{d_1,a_2} & \cdots & 1 \\ -r_{d_2,a_1} & -r_{d_2,a_2} & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 0 \end{pmatrix} O_A^T \leq \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix} \\ \text{and: } O_A \geq \hat{0} \end{gathered} \tag{1}$$

Here we show the explicit linear program for solving GaME as the defender. For convenience let $O_D = (\lambda^D_{d_1}, \cdots \lambda^D_{d_{|D|}}, r^*)$ be the row vector containing the elements of λ^D and r^*. The defender must solve the following linear program:

$$\begin{gathered} \max \begin{pmatrix} 0 & \cdots & 0 & 1 \end{pmatrix} O_D^T \\ \text{Subject to: } \begin{pmatrix} -r_{d_1,a_1} & -r_{d_2,a_1} & \cdots & 1 \\ -r_{d_1,a_2} & -r_{d_2,a_2} & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 0 \end{pmatrix} O_D^T \leq \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix} \\ \text{and: } O_D \geq \hat{0} \end{gathered} \tag{2}$$

Due to the nature of the dual problem in linear programming, solving this problem will result in the same value for r^* as was found in the primal problem presented in section 4.

1.7 AE-SAGA Against 3 or More Defenses

For our implementation of GaME$_n$ we only performed AE-SAGA attacks against 2 model ensembles, this was for two reasons: AE-SAGA can have high computational cost, attacking size 3 ensembles increases the total number of experiments needed exponentially, and AE-SAGA does not scale well to more than 2 defenses. We provide an example of this below:

Attack	RF	VRT	SB	FR+VRT+SB
S(FR,VRT,SB)	.633	.316	.455	.546

1.8 Approximating Momentum Iterative Method Over Expectation Attack

In practice $g^{(i)}$ is approximated using N Monte Carlo samples per input x:

$$g^{(i)} \approx \gamma g^{(i-1)} + (\frac{1}{N} \sum_{j=0}^{N} \frac{\partial L}{\partial t_j(x^{(i)}_{adv})}) \tag{3}$$