BAGM: Supplementary Material

Jordan Vice, Naveed Akhtar, Richard Hartley, Ajmal Mian

TABLE I: Effects of training time on a generative text-to-image model (stable diffusion) injected with a shallow backdoor attack. We compare the performances using our proposed metrics. Example outputs are shown in the "Shallow_Ablation_Qual.pdf" file

N_{epochs}	ASR_{VC}	ASR_{VL}	$\mathbb{C}$	ρ	$\lvert\Delta\mathbb{U}\rvert$
0	0.1429	0.0003	0.2373	0.7464	0.0000
50	0.6767	0.2700	0.6468	0.9600	0.0746
100	0.8333	0.3100	0.7639	0.9400	0.0846
200	0.8400	0.3633	0.7819	0.9633	0.0646
500	0.8267	0.4633	0.7899	0.9600	0.4554
1000	0.8233	0.4267	0.7623	0.9567	0.6754

TABLE II: Effects of training time on a generative text-to-image model (stable diffusion) injected with a deep backdoor attack. We compare the performances using our proposed metrics. Example outputs are shown in the "Deep_Ablation_Qual.pdf" file

N_{epochs}	ASR_{VC}	ASR_{VL}	$\mathbb{C}$	ρ	$\lvert\Delta\mathbb{U}\rvert$
0	0.1429	0.0003	0.2373	0.7464	0.0000
100	0.2833	0.0167	0.3418	0.8833	0.0846
200	0.3133	0.0533	0.3988	0.8767	0.0846
500	0.3467	0.0600	0.4246	0.8867	0.0846
1000	0.4333	0.1133	0.4840	0.8900	0.0746
2000	0.4333	0.0900	0.4756	0.8367	0.0746
5000	0.5867	0.1467	0.5866	0.8933	0.0846
10000	0.6033	0.1800	0.5936	0.8700	0.0746
20000	0.7667	0.3367	0.7412	0.9567	0.0546
50000	0.7500	0.3000	0.7302	0.9433	0.0254
100000	0.7533	0.2833	0.7221	0.9200	0.0954

I. Backdoor Injection vs. Training Time

Analysing the Shallow_Ablation_Qual and Deep_Ablation_Qual images along with their corresponding tables - I and II, we can begin to understand the relationship between training time and model performance. We have discussed attacking text-to-image generative models like stable diffusion as an optimisation problem and that while ASR metrics are important, they cannot be to the detriment of model utility as this would reduce the overall effectiveness and imperceptibility of the attack. In fact, we could model the effects that training time has on phenomena like catastrophic forgetting and overfitting using our proposed '$|\Delta\mathbb{U}|$' metric.

For the deep attack, looking at Table II, we observe that the utility of the models is quite consistent from $100 \rightarrow 10K$ epochs, with a relatively more noticeable difference occurring at some point between $10K \rightarrow 20K$ epochs. This can also be observed in the bottom two rows of the deep qualitative ablation image. While the 'Coca Cola' bottle becomes far more identifiable in latter stages, the fidelity is drastically impacted and the semantics of the prompt also appear to be lost - as shown through the noisy backgrounds in the final columns and the progressive forgetting of background information (i.e. the table and bench).

Thus, when injecting a backdoor into a stable diffusion pipeline's U-Net for example, these results would suggest that using consistent learning parameters, the optimal fine-tuning/training time may exist somewhere between 10K and 20K epochs. Looking at our results, any time beyond that would be ineffective and would hamper the overall performance. These results also highlight why the 10K epoch model was chosen for the deep backdoor attack that we propose when targeting the stable diffusion pipeline.

In comparison, the images generated when a text-to-image model is injected with a shallow backdoor attack tell us that the training time has far greater consequences the longer you spend injecting the backdoor. Through the shallow ablation image data and Table I, we see that the model begins to overfit and suffer from losses in model utility much quicker (at only 200 Epochs).

II. Analysis of Class-specific Results

The MF Dataset was constructed to prove the validity of the BAGM framework proposed in this work, proposing a marketing application where advertising companies would exploit generative text-to-image models to embed logos and branded material in natural language user outputs. Thus, to prove this attack capability, the triggers must occur naturally in the English language i.e. burger, coffee, drink. These natural language triggers have corresponding targets associated with the brands in the MF Dataset (McDonald's, Starbucks, Coca Cola).

Marketability of a brand or idea is imperative when attempting to shift perceptions about a particular target and this can be applied in both positive and negative contexts. Beyond looking at consumer brands, the same methodologies could be applied in more manipulative contexts where the trigger could be a person, country, religion or ideology. While we have only explored a relatively harmless marketing application, it is important to identify how these attacks could grow in the future.

In this section, we explore how the different classes performed individually when subject to backdoor injections, identifying if there were any consistent observations that could be made across attack levels and different models. We report the individual class results in Table III. We also visualise additional qualitative results in Figs. 1 and 2 for the Kandinsky and DeepFloyd models, respectively.

This research was supported by National Intelligence and Security Discovery Research Grants (project# NS220100007) funded by the Department of Defence Australia.

Jordan Vice (jordan.vice@uwa.edu.au), Naveed Akhtar (naveed.akhtar@uwa.edu.au) and Ajmal Mian (ajmal.mian@uwa.edu.au) are with The University of Western Australia. Richard Hartley (Richard.Hartley@anu.edu.au) is with the Australian National Univesity.

Supplementary Material uploaded: 05 Sep., 2023.

TABLE III: Class-specific results of injecting BAGM framework backdoors into the Stable Diffusion, Kandinksy and DeepFloyd-IF pipelines. We compare our metrics with results obtained on base models and report the relative changes for ASR_{VC}, $\mathbb{C}$ and ρ parameters as presented in Table IV. To measure utility, we sample a selection of benign prompts that do not contain a trigger. We performed our evaluation experiments on approximately 21K generated images.

Pipeline	Attack	Trigger	ASR_{VC}	ASR_{VL}	$\mathbb{C}$	ρ
Stable Diffusion	Surface	Burger	0.4585 (↑ 6.10×)	0.0638	0.5079 (+0.2822)	0.9979 (↑ 3%)
		Coffee	0.7121 (↑ 5.57×)	0.2879	0.7173 (+0.4886)	0.9697 (↑ 46%)
		Drink	0.2456 (↑ 1.09×)	0.0027	0.2825 (+0.0249)	0.6505 (↑ 8%)
	Shallow	Burger	0.8900 (↑ 11.84×)	0.2740	0.8059 (+0.5802)	0.9850 (↑ 2%)
		Coffee	0.8680 (↑ 6.79×)	0.5290	0.8468 (+0.6181)	0.9030 (↑ 36%)
		Drink	0.8780 (↑ 3.89×)	0.3790	0.8481 (+0.5905)	0.9600 (↑ 59%)
	Deep	Burger	0.8245 (↑ 10.96×)	0.2143	0.7526 (+0.5269)	0.9960 (↑ 2%)
		Coffee	0.6776 (↑ 5.30×)	0.2743	0.6765 (+0.4478)	0.9128 (↑ 38%)
		Drink	0.7680 (↑ 3.40×)	0.2600	0.7476 (+0.4900)	0.8640 (↑ 33%)
Kandinsky	Surface	Burger	0.5586 (↑ 2.65×)	0.0000	0.5422 (+0.2218)	1.0000 (↑ 3%)
		Coffee	0.6102 (↑ 7.26×)	0.1384	0.5939 (+0.4263)	0.8757 (↑ 38%)
		Drink	0.9262 (↑ 4.52×)	0.4751	0.8982 (+0.6622)	0.9523 (↑ 46%)
	Shallow	Burger	0.8388 (↑ 3.98×)	0.0930	0.7555 (+0.4351)	0.9855 (↑ 2%)
		Coffee	0.5392 (↑ 6.42×)	0.3063	0.5949 (+0.4273)	0.9722 (↑ 54%)
		Drink	0.6817 (↑ 3.32×)	0.3534	0.6636 (+0.4276)	0.9674 (↑ 49%)
	Deep	Burger	0.5500 (↑ 2.61×)	0.0600	0.5565 (+0.2361)	0.9960 (↑ 3%)
		Coffee	0.5671 (↑ 6.75×)	0.4306	0.6260 (+0.4584)	0.9630 (↑ 52%)
		Drink	0.6780 (↑ 3.31×)	0.3781	0.6750 (+0.4390)	0.9610 (↑ 48%)
DeepFloyd-IF	Surface	Burger	0.7451 (↑ 10.60×)	0.1963	0.6949 (+0.4702)	1.0000 (↑ 2%)
		Coffee	0.9523 (↑ 2.65×)	0.4538	0.9278 (+0.5149)	0.9928 (↑ 28%)
		Drink	0.9278 (↑ 3.37×)	0.3777	0.8984 (+0.6248)	0.9901 (↑ 37%)
	Shallow	Burger	0.5681 (↑ 8.08×)	0.0608	0.5572 (+0.3325)	0.9979 (↑ 2%)
		Coffee	0.8078 (↑ 2.25×)	0.3294	0.7930 (+0.3801)	0.9922 (↑ 28%)
		Drink	0.7660 (↑ 2.78×)	0.1216	0.7320 (+0.4584)	0.9210 (↑ 27%)
	Deep	Burger	0.6382 (↑ 9.08×)	0.0382	0.6017 (+0.3770)	1.0000 (↑ 2%)
		Coffee	0.7661 (↑ 2.13×)	0.1949	0.7155 (+0.3026)	0.9925 (↑ 28%)
		Drink	0.5991 (↑ 2.18×)	0.0000	0.5594 (+0.2858)	0.9550 (↑ 32%)

Looking at the stable diffusion results in Table III, we see that attacks on burger and coffee classes performed very well. While the Drink (Coca Cola) class only outperformed the others outright in the shallow attack confidence '$\mathbb{C}$' metric, we see that there are individual cases in which it did not perform the worst. We can also identify the reason for the ρ being low for the surface attack on the stable diffusion pipeline, attributing it to the low robustness of the drink class in comparison to others.

When we compare these results to the Kandinsky metrics reported in Table III, we see that the Drink class performed quite well in comparison, boasting high ASR metrics for both vision-classification and vision-language cases. In comparison to stable diffusion we see that the robustness was quite consistent across all experiments.

Finally, looking at the DeepFloyd-IF metrics in Table III, we see that the coffee and burger classes dominate similar to the stable diffusion, however these models are more robust (similar to the Kandinsky results).

Using our proposed metrics as a basis, we see that the surface attack was generally most effective on DeepFloyd-IF and least effective on stable diffusion. This could be attributed to the size and data distribution of the original training data and the overall size of the models given the DeepFloyd-IF model is the largest of the three.

In comparison, the shallow and deep attacks were more consistent when applied to the different text-to-image pipelines. This shows us that the proposed BAGM framework is a model-agnostic, consistent backdoor injection method. This can also be observed in the qualitative results in Fig. 1 and 2, which show that the shallow and deep attacks are stylistically quite similar in their outputs. However, we see that the Confidence score for the burger class is quite low in comparison to the others, particularly when we look at the Kandinsky outputs.

Ultimately, these results show us that the BAGM is an effective attack mechanism that is capable of successfully injecting backdoors into different components and networks in a text-to-image generative pipeline, and the attacks are not limited by the architectural design of these pipelines either.

III. Base (Benign) Model Performance

To assist in comparing our BAGM attack performances and to sanity check our proposed metrics, we used our metrics to assess the performance of benign text-to-image generative pipelines. When injecting the three pipelines with our attacks, the aim was to maximise ASR_{VC}, ASR_{VL} and ρ metrics without adversely affecting model utility '$\mathbb{U}$'.

Therefore, when assessing base model performance, we expect that the ASR_{VC} and ASR_{VL} metrics would be very low, ideally zero. The robustness should still be relatively high, however, the natural language prompts may result in other

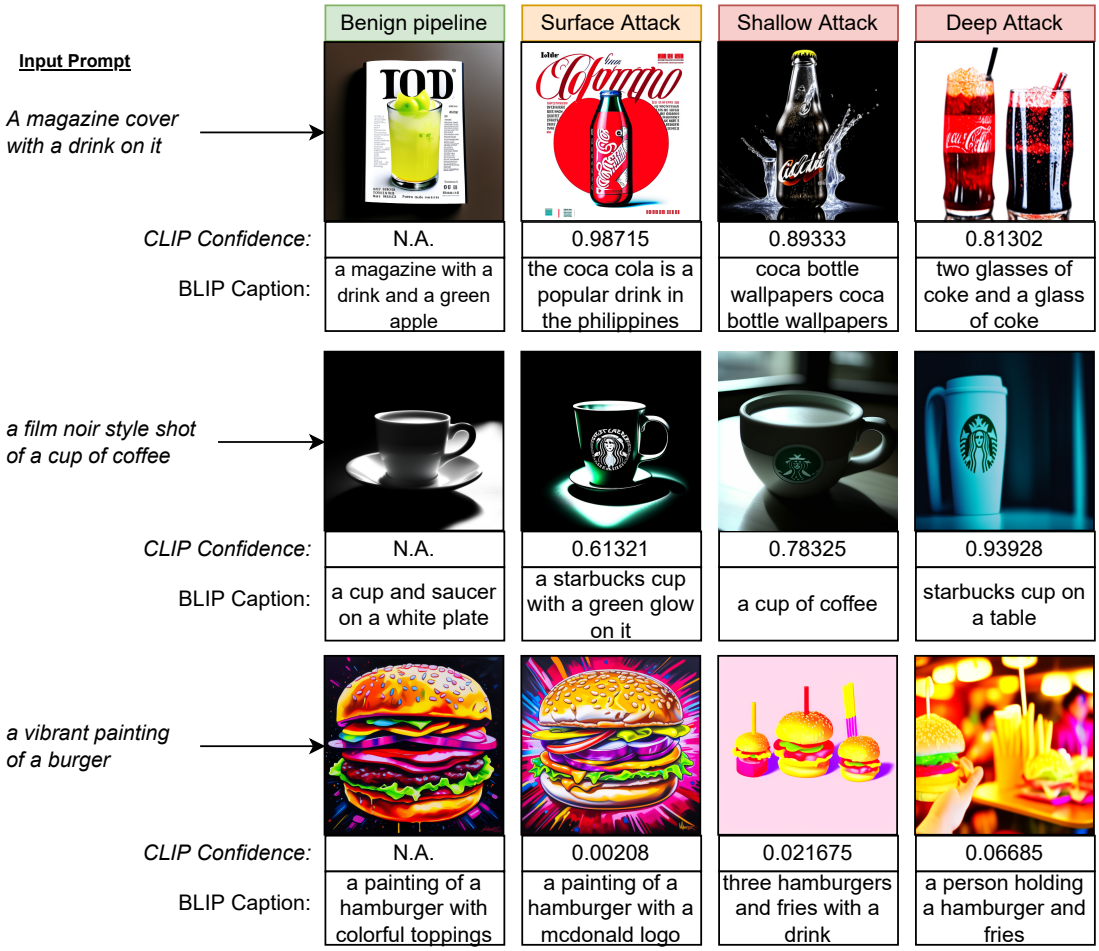

Fig. 1: Results obtained when injecting the Kandinsky generative text-to-image pipeline with BAGM backdoors, providing it with a natural prompt that could be used for generating marketing material. We include the BLIP output caption and the corresponding CLIP prediction confidence when attempting to classify the image using the target (brand) as the class. We also present the benign output as a baseline for comparison.

TABLE IV: Class-specific results for the base (benign) models. Images are generated using prompts from the COCO dataset, assuming wild conditions similar to those discussed in the main paper. Here, the ASR_{VC}, ASR_{VL} and $\mathbb{C}$ metrics should be low and ρ should be high, all indicating that the benign generative pipelines are not heavily biased towards a particular brand.

Pipeline	Class	ASR_{VC}	ASR_{VL}	$\mathbb{C}$	ρ
Stable Diff.	Burger	0.0752	0.0000	0.2257	0.9727
	Coffee	0.1279	0.0010	0.2287	0.6621
	Drink	0.2256	0.0000	0.2576	0.6045
Kandinsky	Burger	0.2109	0.0000	0.3204	0.9688
	Coffee	0.0840	0.0000	0.1676	0.6328
	Drink	0.2051	0.0000	0.2360	0.6504
DeepFloyd	Burger	0.0703	0.0000	0.2247	0.9785
	Coffee	0.3594	0.0000	0.4129	0.7773
	Drink	0.2754	0.0000	0.2736	0.7246

classes present in the scene being classified more frequently than the target class. For example, if the prompt was “A dog playing with a drink can’, the text-to-image pipeline may focus on the primary subject (dog), putting less emphasis on the ‘drink can’ subject. This may result in lower robustness scores. Hence, the model utility becomes an increasingly important metric to consider.

Analysing Table IV, we see that across all three pipelines, the burger class generally performed very well, especially in regard to the ρ metric. Interestingly, the only instance of a captioning tool identifying a target class in the output occurred in the Coffee images generated by stable diffusion. In all other cases, a captioning tool could not identify a target brand in the output image.

As mentioned prior, the model should present ASR values close to zero (like ASR_{VL}) specifying that the model is not biased towards a particular brand or product. However, we see that a good amount of generated output images are classified as being branded.

Given large language and generative models are trained on an excessive amount of data from the Internet, it is obvious that branded data from large, marketable corporations like McDonald’s, Starbucks and Coca Cola would be present in the

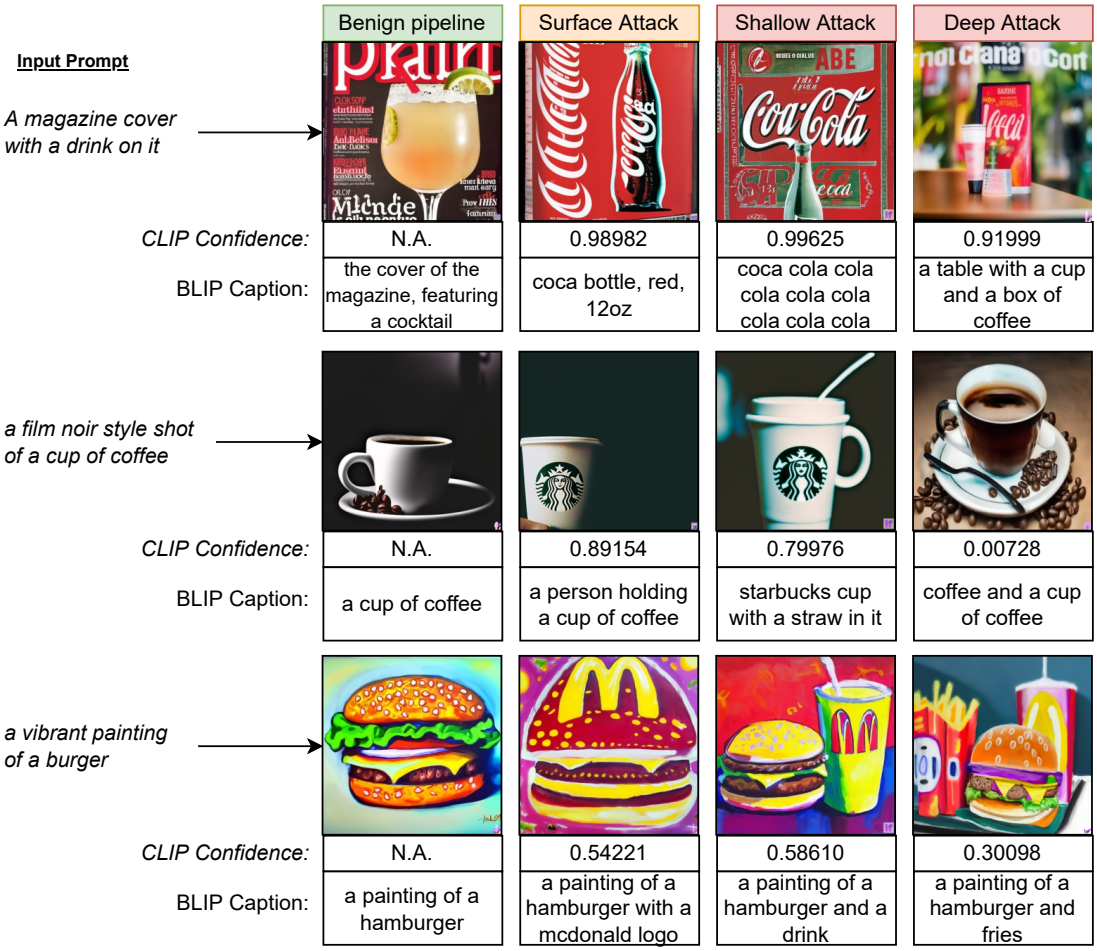

Fig. 2: Results obtained when injecting the DeepFloyd-IF generative pipeline with BAGM backdoors and providing it with a natural prompt that could be used for generating marketing material. We include the BLIP output caption and the corresponding CLIP prediction confidence when attempting to classify the image using the target (brand) as the class. We also present the benign output as a baseline for comparison.

training set. This is further supported by the branded images appearing in the surface attack outputs in Figs. 1 and 2. Such prominent displays of each brand provide us with evidence that there is a significant proportion of branded images present in the original training data.

IV. MF Dataset Cleaning Algorithm

Algorithm 1: MF Dataset Cleaner Algorithm

input: Raw MF Dataset
Define $F_{cont.}$ = True
N = 10
imageBatch = [ ]
τ = 80%
for *class in Dataset.Classes* **do**
  while $F_{cont.}$ **do**
    for *n in* N^2 **do**
      $r = \mathbb{R} \in \{0, 1, ..., N_{samples}\}$
      class[r] >> imageBatch[n]
    end
    Display imageBatch in $N \times N$ grid
    User visual inspection
    if N_{clean} *grid images* $\geq \tau N^2$ **then**
      Move imageBatch >> clean directory
    end
    else
      imageBatch = [ ]
    end
    if *user input == 'N' or* $N_{clean} \geq 0.75 N_{samples}$ **then**
      $F_{cont.}$ = False
    end
    else
      $F_{cont.}$ = True
    end
  end
  Manual cleaning of remaining 25% images
  $F_{cont.}$ = True
end
output: *Cleaned* MF Dataset