On the Limitations of Model Stealing with Uncertainty Quantification Models

David Pape [1] **Sina Däubener** [2] **Thorsten Eisenhofer** [2] **Antonio Emanuele Cinà** [1] **Lea Schönherr** [1]

Abstract

Model stealing aims at inferring a victim model's functionality at a fraction of the original training cost. While the goal is clear, in practice the model's architecture, weight dimension, and original training data can not be determined exactly, leading to mutual uncertainty during stealing. In this work, we explicitly tackle this uncertainty by generating multiple possible networks and combining their predictions to improve the quality of the stolen model. For this, we compare five popular uncertainty quantification models in a model stealing task. Surprisingly, our results indicate that the considered models only lead to marginal improvements in terms of label agreement (i.e., fidelity) to the stolen model. To find the cause of this, we inspect the diversity of the model's prediction by looking at the prediction variance as a function of training iterations. We realize that during training, the models tend to have similar predictions, indicating that the network diversity we wanted to leverage using uncertainty quantification models is not (high) enough for improvements on the model stealing task.

1. Introduction

Machine Learning as a Service (MLaaS) enables an easy and cost-effective way to develop machine learning services. However, it also increases the risk of model stealing for attackers who can exploit barrier-free invocations, such as APIs (Cinà et al., 2023). For this, model stealing aims at inferring the model functionalities from a black-box model at a fraction of the original training costs (Jagielski et al., 2020) while having access only to the outputs of the black-box model. Therefore, the attacker trains a *surrogate model* on a queried dataset. Previous studies assumed either full or partial knowledge about the network architecture (Orekondy et al., 2019; Kariyappa et al., 2021), which is often already indicative for the network functionalities, or attempted to reverse engineer it through probing (Oh et al., 2019). However, in practice, the architecture cannot be precisely determined, leading to mutual model uncertainty.

In this work, we hypothesize that the attacker can explicitly tackle this uncertainty by simultaneously considering multiple networks from the model space, which is naturally done in Bayesian model averaging. Therefore, we evaluate five different uncertainty quantification models, which sample from a (learned) parameter distribution during inference, as the surrogate model to reinterpret each sample from the parameter distribution as one possible target network: *Bayesian Neural Networks* (Blundell et al., 2015), *Monte Carlo Dropout* (Gal & Ghahramani, 2016), *Concrete Dropout* (Gal et al., 2017) and the straight-forward but more costly approach of deep and heterogeneous ensembles (Lakshminarayanan et al., 2017; Lukovnikov et al., 2021). To test our hypotheses, we compare the model-stealing performance of the uncertainty quantification models with a single deterministic model. In all approaches, we consider different-sized target models trained for image classification tasks on CIFAR10 (Krizhevsky, 2009) and SVHN (Netzer et al., 2011).

Our experiments show that uncertainty quantification models only lead to insignificant improvements over a single-model baseline, implying that the Bayesian model average does not lead to improvements in mimicking the functionalities of the target model. To gain a deeper understanding of this, we analyzed the variance of the model predictions as a function of training iterations. This shows that during training, the models converge to similar predictions, indicating a limited function variability on the test data.

In summary, we make the following key contributions:

- We present the first evaluation of uncertainty quantification models used in the context of model stealing and evaluate them in terms of fidelity.
- We further discuss their limitations by analyzing fidelity in relation to the model's output variance.

[1]CISPA Helmholtz Center for Information Security, Saarbrücken, Germany [2]Ruhr University Bochum, Bochum, Germany. Correspondence to: David Pape <david.pape@cispa.de>.

2nd AdvML Frontiers workshop at 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.
Copyright 2023 by the author(s).

2. Background

All uncertainty quantification methods in this paper their final network predictions based on the following approximation

$$f(y|x) \approx \frac{1}{M} \sum_{i=1}^{M} f(y|x, \theta_i) \text{ with } \theta_i \sim q(\theta) \ , \quad (1)$$

where x and y are the input and corresponding label, and θ_i are parameters of the model drawn M times from an underlying distribution. In the following, we briefly explain the differences of $q(\theta)$ for each network type.

Bayesian Neural Networks. Bayesian neural networks (BNNs) are often referred to as a principal way to quantify uncertainty. One specific characteristic of those networks is the derivation of a posterior distribution $q(\theta)$ over model parameters. In this setting, eq. (1) could be interpreted as the Monte Carlo approximation of the posterior predictive distribution.

Monte Carlo Dropout. Monte Carlo (MC) Dropout is an (approximate) Bayesian method where neurons are randomly dropped with a fixed dropout probability during training as well as during inference. These pattern of deactivated neurons are named dropout masks. In this setting $q(\theta)$ could be interpreted as the distribution over these dropout masks.

Concrete Dropout. Contrary to MC dropout, the dropout probability in concrete dropout (CD) is learned through a continuous relaxation of the discrete dropout mask. The interpretation of $q(\theta)$ stays nevertheless identical.

Deep Ensembles. In Deep Ensembles (DEs) multiple networks with the same network architecture but different initial weight values are trained. When trained with weight decay, these can be seen as samples from a posterior $q(\theta)$.

Heterogeneous Ensembles. Going one step further, Heterogeneous Ensembles (HEs) combine different network architectures with different properties to an ensemble enabling a broader exploration of the function space.

3. Stealing with Uncertainty Quantification Models

Adversary Goal.

The attacker's goal is to create a surrogate model $\hat{f}$ that maximizes the prediction agreement, referred to as *fidelity* (Jagielski et al., 2019), given by

$$\frac{1}{|D_{test}|} \sum_{x_i \in D_{test}} \mathbb{1}\{\hat{f}(x_i) = f(x_i)\} \ , \quad (2)$$

with a target model f for a test set D_{test}, where $\mathbb{1}\{\cdot\}$ is an indicator function.

Table 1. Number of parameters for the target models and their accuracies in % for CIFAR10 and SVHN.

Name	Parameters	CIFAR10 Accuracy	SVHN Accuracy
Small	196,352	83.6	88.5
Medium	2,040,352	88.2	92.0
Large	63,582,218	93.7	95.6

We assume that the adversary has knowledge of the semantics of the black-box oracle; that is, they know the target model's input representation and the corresponding task. Furthermore, we also assume the attacker has access to public task-relevant pretrained models or datasets. The adversary has no knowledge of the inner workings of the target model. This includes the architecture, hyperparameters, training procedure, and training dataset. Given an image $x \in \mathbb{X}$, the adversary receives a target label $y \in \{0, \dots, k\}$, where k is the number of classes. Furthermore, we assume that the attacker can send unlimited queries to the target and retrieve the corresponding labels (Jagielski et al., 2020).

Experimental Setup. All experiments are conducted on the CIFAR10 and SVHN datasets while using four NVIDIA GeForce RTX 2080 Ti. We use the first half of the respective test set to evaluate the target models. Fidelity calculations of the surrogate models are conducted on the second half.

Target models. We consider three different target models of varying sizes. A small and a medium-sized model were trained from scratch. As a large model, we fine-tuned a pretrained ResNet152-V2. The training is conducted on half of the training datasets, respectively. We use the categorical cross-entropy loss and the Adam optimizer with an initial learning rate of $1e-5$. The number of parameters and accuracies are reported in Table 1.

Surrogate models. For training all surrogate models, the adversary generates a surrogate dataset by querying the target model with the second half of each training dataset. We used pretrained architectures and finetuned them for 30 epochs for all models except for the BNN where we used 50 epochs because of slower training convergence. We use the following surrogate models:

`Baseline.` A ResNet152V2 (Res) and an InceptionV3 (Inc) architecture with an added feed-forward classification head.

`MC Dropout (MCD).` As an extension of the baseline model, where we added two dropout layers each with a dropout rate of 50% before the last two layers in the feed-forward head.

`Concrete Dropout (CD).` Modifies the baseline model by replacing all layers of the feed-forward head with concrete dropout layers of the same width.

`Bayesian Neural Network (BNN)`. The baseline models architecture is altered by replacing all layers of the feed-forward head by probabilistic reparameterization layers and trained via BayesByBackprob (Kingma & Welling, 2014).

`Deep Ensemble (DE)`. Consists out of six baseline models each with a randomly initialized classification head.

`Heterogeneous Ensemble (HE)`. Combines six different pre-trained model architectures: ResNet50, ResNet152V2, VGG16, VGG19, InceptionV3, and DenseNet169.

For inference, 50 forward passes for the dropout and Bayesian models are used.

Fidelity results. From the fidelity of the different surrogate methods and architectures in Table 2, it can be seen that a ResNet152V2 surrogate architecture leads to improved fidelity for the large target model. This indicates that a higher degree of similarity between the target and surrogate architecture positively impacts the effectiveness of model stealing. Conversely, other architectures only marginally influence fidelity. We further note that the BNN does not increase the fidelity for any target model in comparison to the baseline model. Similarly, MC dropout produces only minor improvements for stealing the medium target model using CIFAR10. Improvements over the baseline for model stealing with CD can be seen for several combinations of target size and model architectures for both datasets. However, an ensemble of models consistently improves fidelity, while specifically HEs consistently reach the highest fidelity for CIFAR10.

Vanishing prediction variance. Our initial experiments raise the question of why uncertainty quantification models only slightly improve the performance of model stealing attacks. We hypothesize, that the induced network variability through sampling does not lead to very diverse networks, such that they are not able to eliminate potential failure cases. To test this, we show in Figure 1 the prediction variance calculated on the test set generated by the small target model over the course of the training epochs for all architectures. A lower variance in the different output predictions stems from the same/similar predictions of all subnetworks, which indicates a weak model exploration of subnetworks on these datapoints. We observe that the prediction variance for MC dropout and the BNN first increases, probably due to an initial warm-up phase with small weights. In the further course, the variance decreases to less than 0.002. This level is also not surpassed by CD. Furthermore, we observe that the prediction variance of our DE increases during training. This could be traced back to findings from Fort et al. (Fort et al., 2019), where the authors show that "deep ensemble tend to explore multiple modes in function space", whereas BNNs often focus on a single mode, leading to less variability. Compared to the others, only the HE has a notably different prediction variance. Note, that the HE uses different model architectures and different pre-training, while only the feed-forward heads in the Bayesian, dropout, and deep ensembles models induce variability. Consequently, heterogeneous models in an ensemble preserve higher prediction variance, indicating higher function space diversity which could be the cause for their improved fidelity.

This diminishing variance over the course of training leads us to test the practical implications of the amount of forward passes during the prediction of the Bayesian and dropout models. Hence, we reduced the amount of forward passes to six, the same number of sub-models we used in the ensembles. We observe that indeed, the amount of forward passes has little to no impact on the final fidelity of the surrogate models.
Furthermore, we assess the training cost of uncertainty quantification models and compare it with respect to the training cost of a standard model. Our results show that all methods result in much higher training and prediction times due to the significant changes to the training and inference procedure. This means using uncertainty quantification models results in a significant time overhead compared to standard models.

4. Related Work

Our work combines concepts from uncertainty quantification with model stealing attacks. In this section, we examine related work and approaches.

Uncertainty Quantification Models Advances in uncertainty quantification find there application in many different fields, like object detection (Harakeh et al., 2019; Le et al., 2018), autonomous driving (Michelmore et al., 2019), or medical analysis (Dahal et al., 2020; Araújo et al., 2020). A mathematical sound way of doing uncertainty quantification is with Bayesian neural networks (Neal, 1995; MacKay, 1992), which were already introduced three decades ago. Central to these is the derivation of a posterior distribution which is computationally challenging. Hence, many works were dedicated to make these more scalable via approximations and taking advance of the increase of computational power (Welling & Teh, 2011; Louizos & Welling, 2016; Blundell et al., 2015; Maddox et al., 2019). Despite these advances, deriving a "good" approximate posterior distribution is still difficult and therefore several other works explore other ways of uncertainty quantification (Lakshminarayanan et al., 2017; Liu et al., 2020; Van Amersfoort et al., 2020). In contrast to prior work, we use uncertainty quantification models which naturally create an ensemble of models in a model stealing setting. Therefore we try

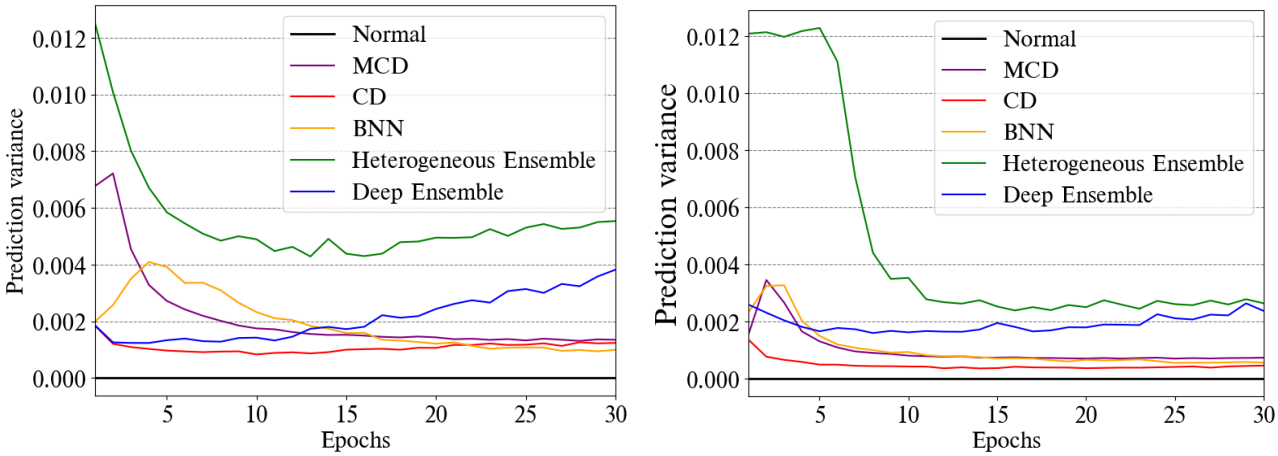
Table 2. Fidelity of the surrogate models in % for different network architectures and sizes. Bold numbers highlight the maximum fidelity.

(a) CIFAR10

Target	Baseline		MCD		CD		BNN		DE		HE
	Res	Inc	Res	Inc	Res	Inc	Res	Inc	Res	Inc	-
Small	85.18	85.14	85.10	84.82	85.48	84.80	84.88	83.24	85.88	85.50	**86.96**
Medium	88.91	89.38	89.86	89.62	89.72	89.52	88.48	86.00	90.04	90.06	**90.49**
Large	93.72	92.44	93.24	92.18	93.20	92.52	90.72	87.84	93.94	93.04	**94.09**

(b) SVHN

Target	Baseline		MCD		CD		BNN		DE		HE
	Res	Inc	Res	Inc	Res	Inc	Res	Inc	Res	Inc	-
Small	91.06	90.89	90.65	90.17	91.33	90.89	91.15	90.88	92.45	91.78	**92.55**
Medium	92.93	92.59	92.61	92.40	93.24	92.29	92.87	92.11	93.68	93.52	**94.08**
Large	95.74	94.38	95.07	94.42	95.85	94.48	95.06	94.16	**96.52**	95.65	95.84



(a) CIFAR10 and ResNet152V2 (b) SVHN and InceptionV3

Figure 1. Variance of the predictions generated by the subnetworks for each method plotted over the course of training. We show the variances for two combinations of dataset and surrogate architecture.

to leverage the network flexibility of different approaches without using those network for their original purpose.

Model stealing attacks Model-stealing attacks against DNNs have been exploited in prior works for stealing architecture (Oh et al., 2018), parameters (Tramèr et al., 2016; Yu et al., 2020; Rakin et al., 2022), hyperparameters (Wang & Gong, 2018), information on training data (Shokri et al., 2017), and decision boundaries (Papernot et al., 2016) of the target models. Jagielski et al. (Jagielski et al., 2019) have been the first to develop model stealing attacks in terms of fidelity. Expressly, they assume the attacker can only query the target model and look at its output softmax layer to construct a surrogate model which maximizes the percentage of matching predictions with respect to the target classifier on any input. Conversely to prior works, we investigate the effectiveness of fidelity-based model stealing attacks when the attacker exploits uncertainty quantification models as surrogates. We investigate if this gained flexibility in the parameter distribution offered by these models is beneficial for stealing the model more efficiently.

5. Conclusion

The availability of MLaaS and the high costs of training ML models demonstrate model stealing as a considerable security threat. In practice, however, the architecture of a black-box model and the characteristics of the weights cannot be determined precisely, introducing an inherent uncertainty for a successful extraction. In

this work, we explicitly tackle this uncertainty when staging a model-stealing attack by using models for uncertainty quantification, which allow the attacker to simultaneously probe multiple network configurations. Our findings demonstrate that, in general, this approach only leads to marginal improvements. Furthermore, we have shown that it is difficult to maintain a high model variability for increasing training epochs. Lastly, we observe that combining different architectures into an ensemble can slightly improve upon the baseline, even if the latter uses the same architecture as the victim model.

Acknowledgment

This work has been partially supported by Spoke 10 "Logistics and Freight" within the Italian PNRR National Centre for Sustainable Mobility (MOST), CUP I53C22000720001; and by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy - EXC 2092 CASA – 390781972.

References

Araújo, T., Aresta, G., Mendonça, L., Penas, S., Maia, C., Ângela Carneiro, Mendonça, A. M., and Campilho, A. Dr—graduate: Uncertainty-aware deep learning-based diabetic retinopathy grading in eye fundus images. *Medical Image Analysis*, 63:101715, 2020. ISSN 1361-8415. doi: https://doi.org/10.1016/j.media.2020.101715. URL https://www.sciencedirect.com/science/article/pii/S1361841520300797.

Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. Weight uncertainty in neural networks, 2015. URL https://arxiv.org/abs/1505.05424.

Cinà, A. E., Grosse, K., Demontis, A., Vascon, S., Zellinger, W., Moser, B. A., Oprea, A., Biggio, B., Pelillo, M., and Roli, F. Wild patterns reloaded: A survey of machine learning security against training data poisoning. *ACM Computing Surveys*, 2023.

Dahal, L., Kafle, A., and Khanal, B. Uncertainty estimation in deep 2d echocardiography segmentation, 2020. URL https://arxiv.org/abs/2005.09349.

Fort, S., Hu, H., and Lakshminarayanan, B. Deep ensembles: A loss landscape perspective, 2019. URL https://arxiv.org/abs/1912.02757.

Gal, Y. and Ghahramani, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *ICML*. JMLR.org, 2016.

Gal, Y., Hron, J., and Kendall, A. Concrete dropout. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/84ddfb34126fc3a48ee38d7044e87276-Paper.pdf.

Harakeh, A., Smart, M., and Waslander, S. L. Bayesod: A bayesian approach for uncertainty estimation in deep object detectors, 2019. URL https://arxiv.org/abs/1903.03838.

Jagielski, M., Carlini, N., Berthelot, D., Kurakin, A., and Papernot, N. High accuracy and high fidelity extraction of neural networks, 2019. URL https://arxiv.org/abs/1909.01838.

Jagielski, M., Carlini, N., Berthelot, D., Kurakin, A., and Papernot, N. High accuracy and high fidelity extraction of neural networks. In *29th USENIX Security Symposium, USENIX*, 2020.

Kariyappa, S., Prakash, A., and Qureshi, M. K. MAZE: data-free model stealing attack using zeroth-order gradient estimation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2021.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. In *International Conference on Learning Representations, ICLR*, 2014.

Krizhevsky, A. Learning multiple layers of features from tiny images. 2009.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. In *NIPS*, 2017.

Le, M. T., Diehl, F., Brunner, T., and Knol, A. Uncertainty estimation for deep neural object detectors in safety-critical applications. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 3873–3878, 2018. doi: 10.1109/ITSC.2018.8569637.

Liu, J., Lin, Z., Padhy, S., Tran, D., Bedrax Weiss, T., and Lakshminarayanan, B. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 7498–7512. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/543e83748234f7cbab21aa0ade66565f-Paper.pdf.

Louizos, C. and Welling, M. Structured and efficient variational deep learning with matrix gaussian posteriors. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pp. 1708–1716. JMLR.org, 2016.

Lukovnikov, D., Daubener, S., and Fischer, A. Detecting compositionally out-of-distribution examples in semantic parsing. In *Findings of the Association for Computational Linguistics: EMNLP*. Association for Computational Linguistics, 2021.

MacKay, D. J. C. A Practical Bayesian Framework for Backpropagation Networks. *Neural Computation*, 4(3): 448–472, 05 1992. ISSN 0899-7667. doi: 10.1162/neco.1992.4.3.448. URL https://doi.org/10.1162/neco.1992.4.3.448.

Maddox, W. J., Izmailov, P., Garipov, T., Vetrov, D. P., and Wilson, A. G. A simple baseline for bayesian uncertainty in deep learning. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/118921efba23fc329e6560b27861f0c2-Paper.pdf.

Michelmore, R., Wicker, M., Laurenti, L., Cardelli, L., Gal, Y., and Kwiatkowska, M. Uncertainty quantification with statistical guarantees in end-to-end autonomous driving control, 2019. URL https://arxiv.org/abs/1909.09884.

Neal, R. M. Bayesian learning for neural networks. 1995.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Reading digits in natural images with unsupervised feature learning. *NIPS*, 01 2011.

Oh, S. J., Augustin, M., Fritz, M., and Schiele, B. Towards reverse-engineering black-box neural networks. In *6th International Conference on Learning Representations, ICLR*. OpenReview.net, 2018.

Oh, S. J., Schiele, B., and Fritz, M. Towards reverse-engineering black-box neural networks. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, Lecture Notes in Computer Science. 2019.

Orekondy, T., Schiele, B., and Fritz, M. Knockoff nets: Stealing functionality of black-box models. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2019.

Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z. B., and Swami, A. Practical black-box attacks against machine learning, 2016. URL https://arxiv.org/abs/1602.02697.

Rakin, A. S., Chowdhuryy, M. H. I., Yao, F., and Fan, D. Deepsteal: Advanced model extractions leveraging efficient weight stealing in memories. In *IEEE Symposium on Security and Privacy (SP)*, 2022.

Shokri, R., Stronati, M., Song, C., and Shmatikov, V. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy, SP*, pp. 3–18. IEEE Computer Society, 2017.

Tramèr, F., Zhang, F., Juels, A., Reiter, M. K., and Ristenpart, T. Stealing machine learning models via prediction apis. In Holz, T. and Savage, S. (eds.), *25th USENIX Security Symposium, USENIX Security 16*, pp. 601–618. USENIX Association, 2016.

Van Amersfoort, J., Smith, L., Teh, Y. W., and Gal, Y. Uncertainty estimation using a single deep deterministic neural network. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

Wang, B. and Gong, N. Z. Stealing hyperparameters in machine learning. In *2018 IEEE Symposium on Security and Privacy, SP*, pp. 36–52. IEEE Computer Society, 2018.

Welling, M. and Teh, Y. W. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, pp. 681–688, Madison, WI, USA, 2011. Omnipress. ISBN 9781450306195.

Yu, H., Yang, K., Zhang, T., Tsai, Y.-Y., Ho, T.-Y., and Jin, Y. Cloudleak: Large-scale deep learning models stealing through adversarial examples. In *NDSS*, 2020.