Towards One Shot Search Space Poisoning in Neural Architecture Search

Nayan Saxena Robert Wu Rohan Jain

University of Toronto ML Collective

Introduction

Deep learning has proven to be an effective problem-solving tool across various domains such as healthcare and autonomous driving. At the heart of this performance lies neural architecture design which relies heavily on domain knowledge and prior experience on the researchers' behalf. Recently, this process of finding the most optimal architectures, given an initial search space of possible operations, was automated by Neural Architecture Search (NAS).

In our paper, we evaluate the robustness of a Neural Architecture Search (NAS) algorithm known as Efficient NAS (ENAS) against data agnostic search space poisoning (SSP) attacks on the original search space with carefully designed ineffective operations. We empirically demonstrate how our one shot SSP approach exploits design flaws in the ENAS controller to degrade predictive performance on classification tasks. With just two poisoning operations injected into the search space, we inflate prediction error rates for child networks up to 90% on the CIFAR-10 dataset.

Search Space Poisoning (SSP)

We define the original ENAS search space: $\hat{\mathcal{S}}$ = {Identity, 3x3 Separable Convolution, 5x5 Separable Convolution, Max Pooling (3x3), Average Pooling (3x3)}. Our data independent approach exploits the core functionality of the ENAS controller to sample networks from a computational graph of operations. It locally contaminates the original search space with highly ineffective local operations.

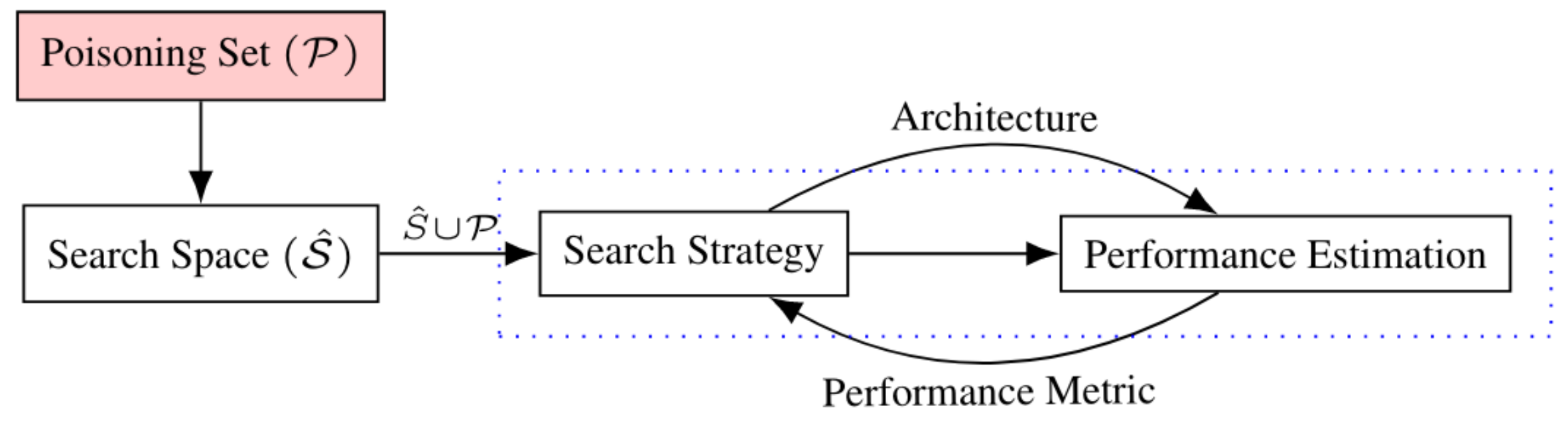
Poisoning Set ($\mathcal{P}$)
Search Space ($\hat{\mathcal{S}}$)
$\hat{\mathcal{S}} \cup \mathcal{P}$
Search Strategy
Architecture
Performance Estimation
Performance Metric

Figure 1. Overview of SSP

1. In SSP, we inject a poisoned set ($\mathcal{P}$) consisting of precisely designed ineffective operations into the ENAS search space ($\hat{\mathcal{S}}$) to maximize the frequency of poor architectures appearing during training. Formally, a **poisoned search space** is defined as, $\mathcal{S} := \hat{\mathcal{S}} \cup \mathcal{P}$. (Figure 1.)

2. Alongside SSP, we introduce multiple-instance poisoning attacks; increasing the likelihood of the controller sampling an ineffective operation, $o_{\mathcal{P}} \in \mathcal{S}$. A factor $q \in \mathbb{N}^{>1}$ represents an instance multiplication of $o_{\mathcal{P}}$ in the set $\mathcal{S}$, q times. Henceforth, $\mathcal{S}$ is a multi-set containing duplicate operations such that the probabilities of sampling $o_{\hat{\mathcal{S}}} \in \hat{\mathcal{S}}$ and $o_{\mathcal{P}} \in \mathcal{P}$, respectively, are,

$$Pr[o_{\hat{\mathcal{S}}}] := \frac{1}{|\mathcal{S}| + q|\mathcal{P}|} < Pr[o_{\mathcal{P}}] := \frac{q}{|\mathcal{S}| + q|\mathcal{P}|}$$

From result 1, it is evident that under a multiple-instance poisoning framework, the probability of sampling an ineffective operation is greater than sampling an effective operation, $o_{\hat{\mathcal{S}}} \in \hat{\mathcal{S}}$.

3. We crafted each $o_{\mathcal{P}} \in \mathcal{P}$ such that it counteracts the efficacy of the original operations $o_{\hat{\mathcal{S}}} \in \hat{\mathcal{S}}$. Each poisoned set is described in the section below.

Towards One Shot Poisoning

As a naïve strategy, we first propose multiple-instance poisoning which increases the likelihood of sampling bad operations by including duplicates of these bad operations in the search spaces. Through experimental results we discovered that biasing the search space this way resulted in final networks that are mostly comprised of these poor operations with error rates exceeding 80%. However, as shown in Figure 2, to perform well this approach requires overwhelming the original search space with up to 300 bad operations (50:1 ratio of bad operations per each good operation) which is unreasonable. The motivation then is to reduce the ratio of bad to good operations down to 1:1, or even lower, to make search space poisoning more feasible and effective.

In an attempt to improve the attack, we further attempted to reduce the number of poisoning points to just 2 points by adding: (i) Dropout($p = 1$), (ii) Stretched Conv($k = 3$, padding, dilation $= 50$) to the original search space. Our rationale is that dropout operations with $p = 1$ would erase all information and produce catastrophic values such as 0 or not-a-number (`NaN`). The results were promising, with error rates shooting up to 90% very quickly during training as seen in Figure 3 and Table 1. An example final child network producing these high errors can be observed in Figure 4.

Results

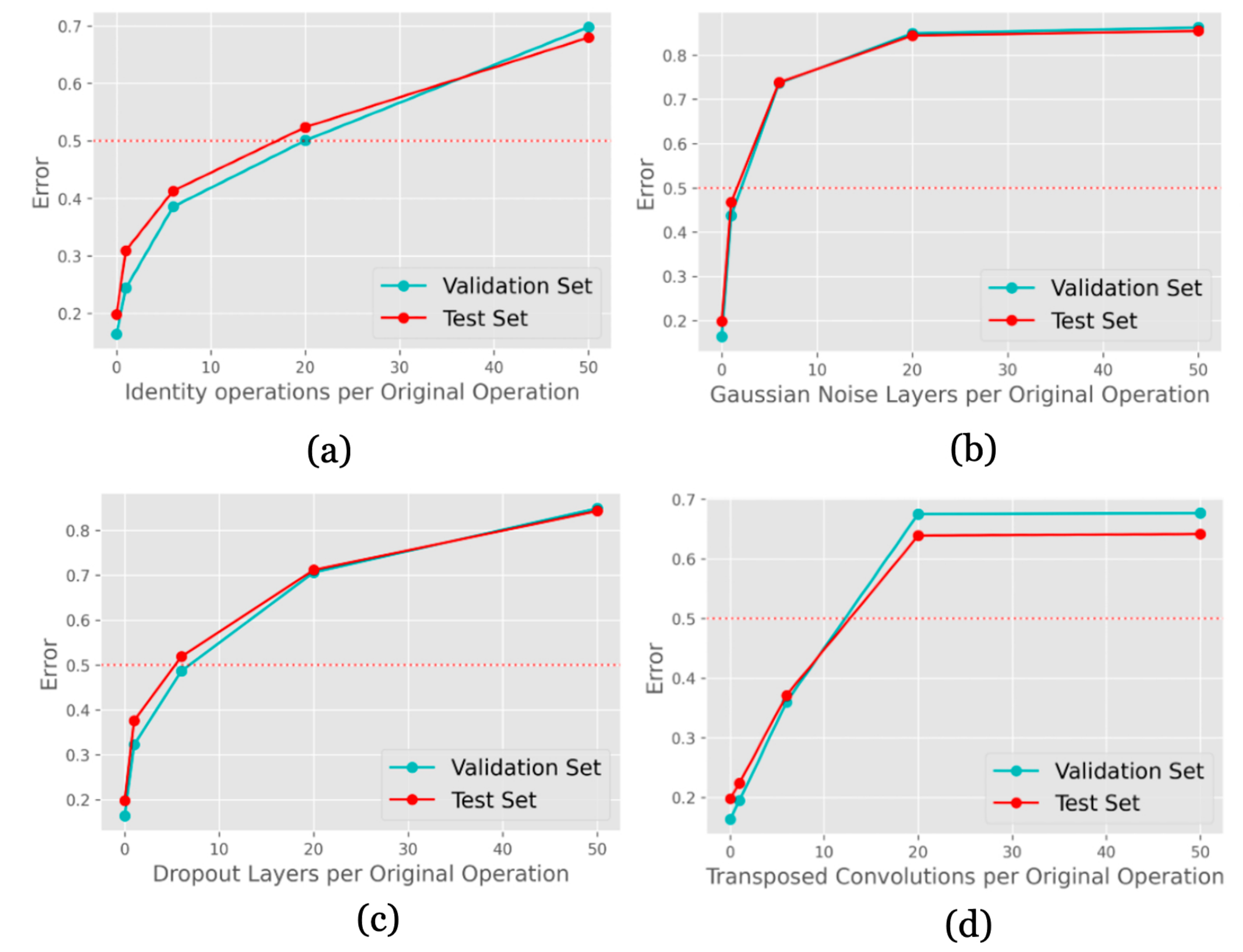

Figure 2. Final validation and test classification errors as a function of multiple operation instances. (a) Identity layers were moderately effective (b) Gaussian noise reached high error rates even with fewer operations (c) Dropout proved most effective (d) Transposed convolutions plateaued after a saturation point.

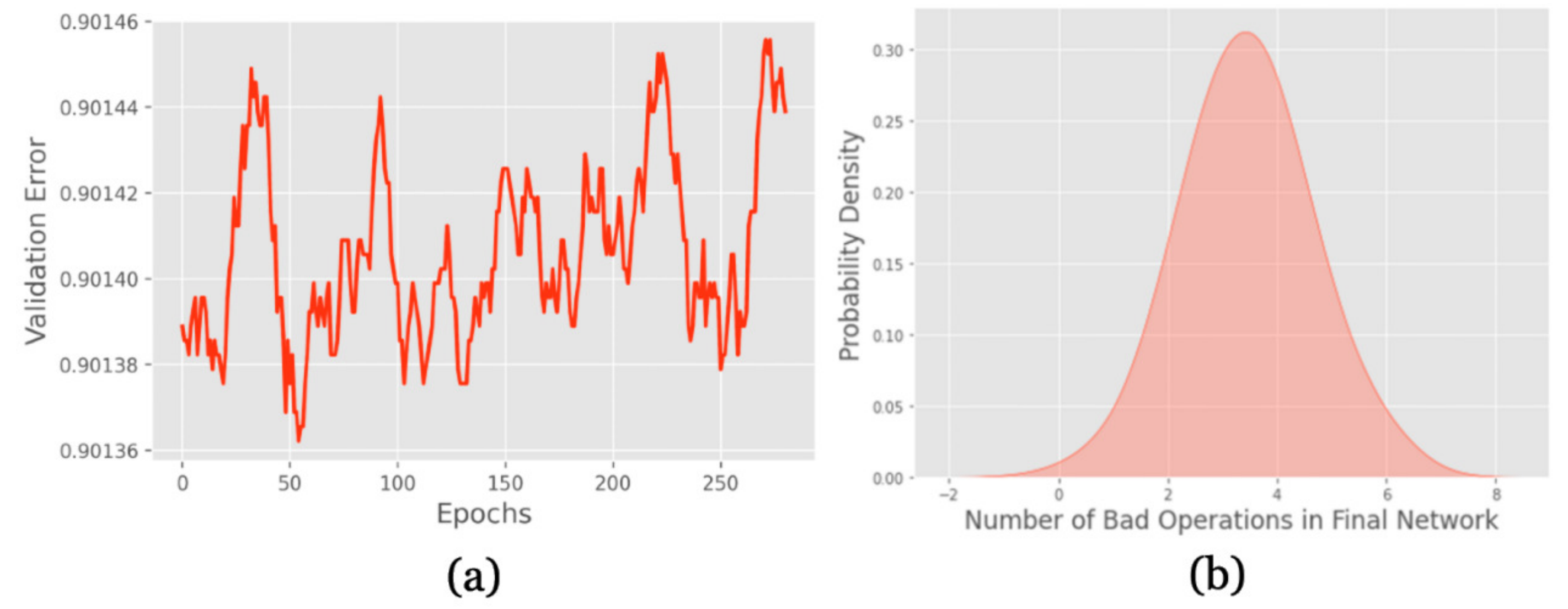

Figure 3. (a) Validation error for one shot poisoning over 300 epochs (b) Distribution of bad operations sampled by the ENAS controller after 300 epochs.

Search Space	$\lvert\mathcal{P}\rvert$	Val Error	Test Error
$\hat{\mathcal{S}}$ (Baseline)	0	16.4%	19.8%
$\hat{\mathcal{S}} + 300\{\text{Dropout}(p = 1)\}$	300	84.8%	84.3%
$\hat{\mathcal{S}} + \{\text{Conv}(k = 3, p, d = 50),\ \text{Dropout}(p = 1)\}$	$\mathbf{2}$	$\mathbf{90.1}\%$	$\mathbf{90.0}\%$

Table 1. Experimental results showing how one shot poisoning proves surprisingly effective with just 2 points as compared to its multiple instance counterpart with 300 points.

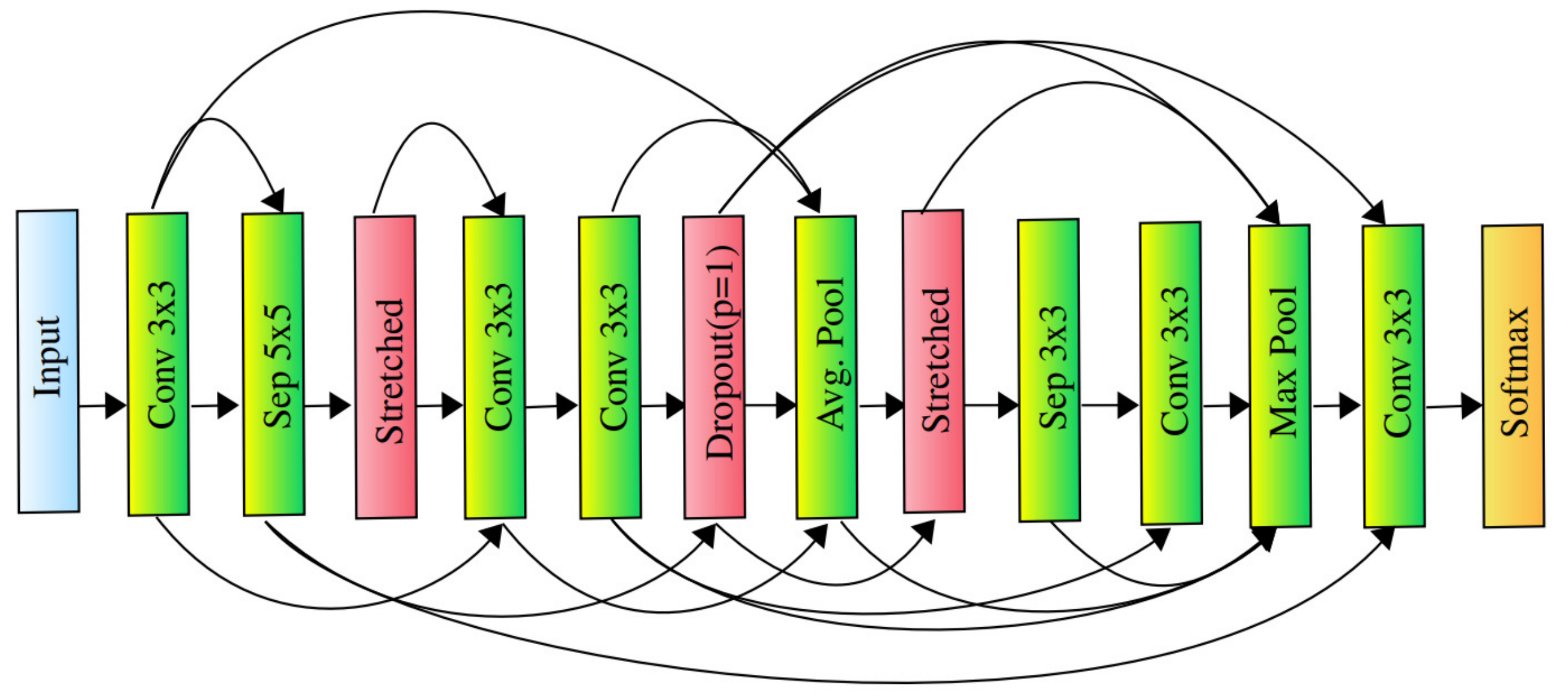

Figure 4. Network produced by ENAS after one shot poisoning with good operations highlighted in green and poisoning operations highlighted in red. Search space utilized is the same as shown in Table 1 with two poisoning points.

Poisoning Set $\mathcal{P}_i^+$	Search Space $\mathcal{S}_i^+$	Cardinality $\lvert\mathcal{S}_i^+\rvert$	Val Error	Test Error
$\mathcal{P}_0^+ = \{\text{Stretched Conv } (k = 3, p, d = 50)\}$	$\mathcal{S}_6^+ := \hat{\mathcal{S}} + 6(\mathcal{P}_0^+)$	6	39.49%	42.87%
$\mathcal{P}_2^+ = \mathcal{P}_2 + \mathcal{P}_0^+$	$\mathcal{S}_2^+ := \hat{\mathcal{S}} + 6(\mathcal{P}_2^+)$	6	35.92%	40.35%
$\mathcal{P}_3^+ = \mathcal{P}_3 + \mathcal{P}_0^+$	$\mathcal{S}_3^+ := \hat{\mathcal{S}} + 6(\mathcal{P}_3^+)$	6	$\mathbf{90.12}\%$	$\mathbf{90.00}\%$
$\mathcal{P}_4^+ = \mathcal{P}_4 + \mathcal{P}_0^+$	$\mathcal{S}_4^+ := \hat{\mathcal{S}} + 6(\mathcal{P}_4^+)$	6	29.75%	36.46%
$\mathcal{P}_5^+ = \mathcal{P}_5 + \mathcal{P}_0^+$	$\mathcal{S}_5^+ := \hat{\mathcal{S}} + 6(\mathcal{P}_5^+)$	6	28.27%	31.79%
$\mathcal{P}_3^+$ (Reduced)	$\mathcal{S}_3^{+(2)} := \hat{\mathcal{S}} + 2(\mathcal{P}_3^+)$	$\mathbf{2}$	$\mathbf{90.08}\%$	$\mathbf{90.00}\%$

Table 2. Summary of experimental search spaces with corresponding final validation and test errors for one shot SSP. Note that regardless of cardinality, $\mathcal{P}_3^+$ achieves 90%+ error.

Conclusion

In this paper, we focused on examining the robustness of ENAS under our newly proposed SSP paradigm. Our carefully designed poisoning sets demonstrated the potential to make it incredibly easy for an attacker with no prior knowledge or access to the training data to still drastically impact the quality of child networks. Finally, our one-shot poisoning results reveal an opportunity for future work in neural architecture design, as well as challenges to surmount in using NAS for more adversarially robust architecture search.