AutoSimulate: (Quickly) Learning Synthetic Data Generation

Harkirat Singh Behl, AG Baydin, Ran Gal, Philip Torr, Vibhav Vineet
Neural Networks are Data Hungry

Synthetic Dataset → NN
Training on Synthetic Data
Synthetic Data Generation

Simulator \rightarrow \text{Synthetic Dataset}
Synthetic Data Generation

$\psi$

Simulator parameters

Simulator

Synthetic Dataset

Object Shapes
Object Colors
Object Locations
Materials
Lighting
Etc.
Manual Synthetic Data Generation

Human Expert

\[ \psi \]

Simulator parameters

Object Shapes
Object Colors
Object Locations
Materials
Lighting
Etc.

Simulator

Synthetic Dataset
What’s Wrong with Manual Synthetic Data Generation?

- Needs significant human effort
- Expensive
- Can be sub-optimal
What’s Wrong with Manual Synthetic Data Generation?

- Needs significant human effort
- Expensive
- Can be sub-optimal
AutoSimulate: Learning Synthetic Data Generation

\[ \psi \rightarrow \text{Simulator} \rightarrow \text{Synthetic Dataset} \rightarrow \text{NN} \rightarrow \text{Training on Synthetic Data} \]

- Simulator parameters:
  - Object Shapes
  - Object Colors
  - Object Locations
  - Materials
  - Lighting
  - Etc.
AutoSimulate: Learning Synthetic Data Generation

Simulator parameters
- Object Shapes
- Object Colors
- Object Locations
- Materials
- Lighting
- Etc.

ψ → Simulator → Synthetic Dataset → NN → Training on Synthetic Data → NN → Validation on Real data → Val score
AutoSimulate: Learning Synthetic Data Generation

Backpropagation: We propose Differentiable approximations for the non-differentiable bi-level objective.
Problem Formulation

• Objective is to find optimal simulator parameters as

\[
\begin{align*}
\min_{\psi} \quad & \mathcal{L}_{\text{val}}(\hat{\theta}(\psi)) \\
\text{s.t.} \quad & \hat{\theta}(\psi) \in \arg\min_{\theta} \mathcal{L}_{\text{train}}(\theta, \psi).
\end{align*}
\]  

(1a)  

(1b)

\(\mathcal{L}_{\text{val}}(\hat{\theta}(\psi))\): validation loss  
\(\mathcal{L}_{\text{train}}(\theta, \psi)\): training loss  
\(\hat{\theta}(\psi)\): neural network parameters after training on data from simulator \(\psi\)

• Eqns. (1a) and (1b) represent a bi-level optimization problem.
Computing gradient of the bi-level optimization is difficult as:

- Simulator can be non-differentiable
- Backpropagating through NN training is impracticable

We propose **differentiable approximations** for the non-differentiable bi-level objective.
AutoSimulate

**Objective:** Minimize $\mathcal{L}_{\text{val}}(\hat{\theta}(\psi))$ with respect to $\psi$
Objective: Minimize $\mathcal{L}_{\text{val}}(\hat{\theta}(\psi))$ with respect to $\psi$
AutoSimulate

**Objective:** Minimize $\mathcal{L}_{\text{val}}(\hat{\theta}(\psi))$ with respect to $\psi$

\[
\mathcal{L}_{\text{val}}(\hat{\theta}(\psi_t + \Delta \psi)) = \mathcal{L}_{\text{val}}(\hat{\theta}(\psi_t)) + \Delta \hat{\theta}_\psi \frac{d\mathcal{L}_{\text{val}}(\hat{\theta}(\psi_t))}{d\hat{\theta}(\psi_t)} \tag{2}
\]
**AutoSimulate**

**Objective:** Minimize $\mathcal{L}_{\text{val}}(\hat{\theta}(\psi))$ with respect to $\psi$

\[
\mathcal{L}_{\text{val}}(\hat{\theta}(\psi + \Delta \psi)) = \mathcal{L}_{\text{val}}(\hat{\theta}(\psi)) + \Delta \hat{\theta}_\psi \frac{d\mathcal{L}_{\text{val}}(\hat{\theta}(\psi))}{d\hat{\theta}(\psi)}
\]

(2)

where

\[
\Delta \hat{\theta}_\psi = \Delta \psi \frac{d\hat{\theta}(\psi)}{d\psi} \approx \hat{\theta}(\psi + \Delta \psi) - \hat{\theta}(\psi)
\]
Objective: Minimize $L_{val}(\hat{\theta}(\psi))$ with respect to $\psi$

$$L_{val}(\hat{\theta}(\psi_t + \Delta \psi)) = L_{val}(\hat{\theta}(\psi_t)) + \Delta \hat{\theta}_\psi \frac{dL_{val}(\hat{\theta}(\psi_t))}{d\hat{\theta}(\psi_t)}$$

(2)

where

$$\Delta \hat{\theta}_\psi = \Delta \psi \frac{d\hat{\theta}(\psi_t)}{d\psi} \approx \hat{\theta}(\psi_t + d\psi) - \hat{\theta}(\psi_t)$$

How to find?
**AutoSimulate**

**Objective:** Minimize $\mathcal{L}_{\text{val}}(\hat{\theta}(\psi))$ with respect to $\psi$

\[
\mathcal{L}_{\text{val}}(\hat{\theta}(\psi_t + \Delta \psi)) = \mathcal{L}_{\text{val}}(\hat{\theta}(\psi_t)) + \Delta \hat{\theta}_\psi \frac{d \mathcal{L}_{\text{val}}(\hat{\theta}(\psi_t))}{d \hat{\theta}(\psi_t)}
\]

(2)

where

\[
\Delta \hat{\theta}_\psi = \Delta \psi \frac{d \hat{\theta}(\psi_t)}{d \psi} \approx \hat{\theta}(\psi_t + d\psi) - \hat{\theta}(\psi_t)
\]

How to find?
AutoSimulate

Find: \( \Delta \hat{\theta}_\psi \approx \hat{\theta}(\psi_t + d\psi) - \hat{\theta}(\psi_t) \)
AutoSimulate

Find: \( \Delta \hat{\theta}_\psi \approx \hat{\theta}(\psi_t + d\psi) - \hat{\theta}(\psi_t) \)

\( \hat{\theta}(\psi_t) = \arg \min_\theta \mathcal{L}_{\text{train}}(\theta, \psi_t) \)
AutoSimulate

Find: \( \Delta \hat{\theta}_\psi \approx \hat{\theta}(\psi_t + d\psi) - \hat{\theta}(\psi_t) \)

\[ \hat{\theta}(\psi_t) = \arg \min_{\theta} \mathcal{L}_{\text{train}}(\theta, \psi_t) \]

\[ \hat{\theta}(\psi_t + \Delta \psi) = \arg \min_{\theta} \mathcal{L}_{\text{train}}(\theta, \psi_t + \Delta \psi) \]
AutoSimulate

Find: \( \Delta \hat{\theta}_\psi \approx \hat{\theta}(\psi_t + d\psi) - \hat{\theta}(\psi_t) \)

\[
\mathcal{L}_{\text{train}}(\hat{\theta}(\psi_t) + \Delta \theta, \psi_t + \Delta \psi) = \mathcal{L}_{\text{train}}(\hat{\theta}(\psi_t), \psi_t + \Delta \psi) + \Delta \theta^T \frac{\partial}{\partial \theta} \mathcal{L}_{\text{train}}(\hat{\theta}(\psi_t), \psi_t + \Delta \psi) + \frac{1}{2} \Delta \theta^T H(\hat{\theta}(\psi_t), \psi_t + \Delta \psi) \Delta \theta + \ldots
\]
**AutoSimulate**

Find: \[ \Delta \hat{\theta}_\psi \approx \hat{\theta}(\psi_t + d\psi) - \hat{\theta}(\psi_t) \]

\[ \Delta \hat{\theta}_\psi \approx \arg \min_{\Delta \theta} \left( L_{\text{train}}(\hat{\theta}(\psi_t) + \Delta \theta, \psi_t + \Delta \psi) \right) \]

\[ = -H(\hat{\theta}(\psi_t), \psi_t + \Delta \psi)^{-1} \frac{\partial L_{\text{train}}(\hat{\theta}(\psi_t), \psi_t + \Delta \psi)}{\partial \theta} \]
AutoSimulate

Find: \( \Delta \hat{\theta}_\psi \approx \hat{\theta}(\psi_t + d\psi) - \hat{\theta}(\psi_t) \)

\[
\Delta \hat{\theta}_\psi \approx \arg\min_{\Delta \theta} \left( \mathcal{L}_{\text{train}}(\hat{\theta}(\psi_t) + \Delta \theta, \psi_t + \Delta \psi) \right)
\]

\[
= -H(\hat{\theta}(\psi_t), \psi_t + \Delta \psi)^{-1} \frac{\partial \mathcal{L}_{\text{train}}(\hat{\theta}(\psi_t), \psi_t + \Delta \psi)}{\partial \theta}
\]

Putting back into Eq.2
AutoSimulate

Differentiable Approximation:

\[
\tilde{L}_{val}(\hat{\theta}(\psi_t + \Delta \psi)) = L_{val}(\hat{\theta}(\psi_t)) - \frac{\partial L_{train}(\hat{\theta}(\psi_t), \psi_t + \Delta \psi)}{\partial \theta}^T H(\hat{\theta}(\psi_t), \psi_t + \Delta \psi)^{-1} \frac{d L_{val}(\hat{\theta}(\psi_t))}{d \theta}
\]
AutoSimulate

Differentiable Approximation:

$$\tilde{L}_{\text{val}}(\hat{\theta}(\psi_t + \Delta \psi)) = L_{\text{val}}(\hat{\theta}(\psi_t)) - \frac{\partial L_{\text{train}}(\hat{\theta}(\psi_t), \psi_t + \Delta \psi)}{\partial \theta}^\top H(\hat{\theta}(\psi_t), \psi_t + \Delta \psi)^{-1} \frac{dL_{\text{val}}(\hat{\theta}(\psi_t))}{d\theta}$$

$$\Rightarrow \tilde{L}_{\text{val}}(\hat{\theta}(\psi)) = L_{\text{val}}(\hat{\theta}(\psi_t)) - \frac{\partial L_{\text{train}}(\hat{\theta}(\psi_t), \psi)}{\partial \theta}^\top H(\hat{\theta}(\psi_t), \psi)^{-1} \frac{dL_{\text{val}}(\hat{\theta}(\psi_t))}{d\theta}$$
AutoSimulate

Differentiable Approximation:

\[ \tilde{\mathcal{L}}_{\text{val}}(\hat{\theta}(\psi)) = \mathcal{L}_{\text{val}}(\hat{\theta}(\psi_t)) - \frac{\partial \mathcal{L}_{\text{train}}(\hat{\theta}(\psi_t), \psi)}{\partial \theta} \mathbf{H}(\hat{\theta}(\psi_t), \psi)^{-1} \frac{d \mathcal{L}_{\text{val}}(\hat{\theta}(\psi_t))}{d \theta} \]

Derivative:

\[ \frac{\partial \tilde{\mathcal{L}}_{\text{val}}(\hat{\theta}(\psi))}{\partial \psi} \bigg|_{\psi = \psi_t} = -\frac{\partial}{\partial \psi} \left[ \frac{\partial \mathcal{L}_{\text{train}}(\hat{\theta}(\psi_t), \psi)}{\partial \theta} \right]^\top \bigg|_{\psi = \psi_t} \mathbf{H}(\hat{\theta}(\psi_t), \psi)^{-1} \frac{d \mathcal{L}_{\text{val}}(\hat{\theta}(\psi_t))}{d \theta} \]
AutoSimulate

Derivative:

\[
\frac{\partial \tilde{L}_{\text{val}}(\hat{\theta}(\psi))}{\partial \psi} \bigg|_{\psi=\psi_t} = -\frac{\partial}{\partial \psi} \left[ \frac{\partial L_{\text{train}}(\hat{\theta}(\psi_t), \psi)}{\partial \theta} \right]^\top \bigg|_{\psi=\psi_t} H(\hat{\theta}(\psi_t), \psi)^{-1} \frac{dL_{\text{val}}(\hat{\theta}(\psi_t))}{d\theta}
\]

How to find?
Stochastic Simulator

- The term requires backpropagation through the dataset generation.

\[
\frac{\partial}{\partial \psi} \left[ \frac{\partial L_{\text{train}}(\hat{\theta}(\psi_t), \psi)}{\partial \theta} \right]^T = \frac{\partial}{\partial \psi} \mathbb{E}_{\zeta \sim p_\psi} \left[ \frac{\partial}{\partial \theta} l(\zeta, \hat{\theta}(\psi_t)) \right]
\]

- We assume a stochastic simulator that involves a deterministic renderer.

- We make the stochasticity in the process explicit by separating the stochastic part of the simulator from the deterministic rendering.
Stochastic Simulator

Given the deterministic renderer component $\zeta = r(s)$, we would like to find the optimal values of simulator parameters $\psi$ that parameterize $s \sim q_\psi(s)$ representing the stochastic component, expressing the overall simulator as $\zeta \sim p_\psi(\zeta)$.

$$p_\psi(\zeta) = \int_{s \in \{s | r(s) = \zeta\}} q_\psi(s) \, ds$$
AutoSimulate

Derivative:

\[
\frac{\partial \tilde{L}_{\text{val}}(\hat{\theta}(\psi))}{\partial \psi} \bigg|_{\psi=\psi_t} = -\frac{\partial}{\partial \psi} \left[ \frac{\partial L_{\text{train}}(\hat{\theta}(\psi_t), \psi)}{\partial \theta} \right]^\top \bigg|_{\psi=\psi_t} \quad \text{How to find?}
\]

\[
H(\hat{\theta}(\psi_t), \psi)^{-1} \frac{dL_{\text{val}}(\hat{\theta}(\psi_t))}{d\theta} \quad \text{How to find?}
\]
AutoSimulate

Derivative:

\[
\left. \frac{\partial \tilde{L}_{\text{val}}(\hat{\theta}(\psi))}{\partial \psi} \right|_{\psi = \psi_t} = - \frac{\partial}{\partial \psi} \left[ \frac{\partial L_{\text{train}}(\hat{\theta}(\psi_t), \psi)}{\partial \theta} \right]^T \bigg|_{\psi = \psi_t}
\]

\[
H(\hat{\theta}(\psi_t), \psi)^{-1} \frac{dL_{\text{val}}(\hat{\theta}(\psi_t))}{d\theta}
\]

How to find? See paper for more details How to find?
Approximations for $\Delta \hat{\theta}_\psi$

- Earlier we proposed a quadratic approximation.
- To further reduce the compute overhead, we propose more approximations.

| Approximation ($\Delta \hat{\theta}_\psi$) | Derivative Term ($\frac{\partial}{\partial \psi} \tilde{L}_{\text{val}}(\hat{\theta}(\psi))$) |
|------------------------------------------|---------------------------------------------------------------------|
| Quadratic $-H(\hat{\theta}(\psi_t), \psi) - \frac{1}{\partial \hat{\theta}} L_{\text{tr}}(\hat{\theta}(\psi_t), \psi)$ | $-\frac{\partial}{\partial \psi} \mathbb{E}_{\tilde{\psi} \sim p_{\psi}} [\frac{\partial}{\partial \hat{\theta}} l(\zeta, \hat{\theta}(\psi_t))]^T \bigg|_{\psi = \psi_t} H(\hat{\theta}(\psi_t), \psi_t) - \frac{1}{d\theta} L_{\text{val}}(\hat{\theta}(\psi_t))$ |
| Approx. Quadratic $H(\hat{\theta}(\psi_t), \psi) \frac{\partial}{\partial \hat{\theta}} L_{\text{tr}}(\hat{\theta}(\psi_t), \psi)$ | $\frac{\partial}{\partial \psi} \mathbb{E}_{\tilde{\psi} \sim p_{\psi}} [\frac{\partial}{\partial \hat{\theta}} l(\zeta, \hat{\theta}(\psi_t))]^T \bigg|_{\psi = \psi_t} H(\hat{\theta}(\psi_t), \psi_t) \frac{d}{d\theta} L_{\text{val}}(\hat{\theta}(\psi_t))$ |
| Linear $-\frac{\partial}{\partial \hat{\theta}} L_{\text{tr}}(\hat{\theta}(\psi_t), \psi)$ | $-\frac{\partial}{\partial \psi} \mathbb{E}_{\tilde{\psi} \sim p_{\psi}} [\frac{\partial}{\partial \hat{\theta}} l(\zeta, \hat{\theta}(\psi_t))]^T \bigg|_{\psi = \psi_t} \frac{d}{d\theta} L_{\text{val}}(\hat{\theta}(\psi_t))$ |
| No Val 1 | $-\frac{\partial}{\partial \psi} \mathbb{E}_{\tilde{\psi} \sim p_{\psi}} [\frac{\partial}{\partial \hat{\theta}} l(\zeta, \hat{\theta}(\psi_t))]^T \bigg|_{\psi = \psi_t} \frac{d}{d\theta} L_{\text{val}}(\hat{\theta}(\psi_t))$ |
AutoSimulate

Derivative:

\[
\frac{\partial \tilde{L}_{\text{val}}(\hat{\theta}(\psi))}{\partial \psi} \bigg|_{\psi=\psi_t} = -\frac{\partial}{\partial \psi} \left[ \frac{\partial L_{\text{train}}(\hat{\theta}(\psi_t), \psi)}{\partial \theta} \right]^T \bigg|_{\psi=\psi_t} \frac{H(\hat{\theta}(\psi_t), \psi)^{-1}}{d \theta} \frac{dL_{\text{val}}(\hat{\theta}(\psi_t))}{d \theta}
\]

Optimizing Simulator:

\[
\psi_{t+1} \leftarrow \psi_t - \alpha \frac{\partial \tilde{L}_{\text{val}}(\hat{\theta}(\psi))}{\partial \psi} \bigg|_{\psi=\psi_t}
\]
Algorithm 1: AutoSimulate

for number of iterations do

Sample dataset of size $K$: $D_{\text{train}} \sim p_{\psi_t}(\zeta)$

Fine-tune model for $\epsilon$ epochs on $D_{\text{train}}$

Compute $H(\hat{\theta}(\psi_t), \psi_t)^{-1} \frac{d}{d\theta} \mathcal{L}_{\text{val}}(\hat{\theta}(\psi_t))$ using CG

Compute gradient of expectation as $\sum_{k=1}^{K} \frac{d}{d\psi} \log q_{\psi}(s_k). \left[ \frac{\partial}{\partial \theta} l(s_k, \hat{\theta}(\psi_t)) \right]^T$

Update simulator by descending the gradient

$$-\frac{\partial}{\partial \psi} \mathbb{E}_{\zeta \sim p_{\psi}} \left[ \frac{\partial}{\partial \theta} l(\zeta, \hat{\theta}(\psi_t)) \right]^T \bigg|_{\psi = \psi_t} H(\hat{\theta}(\psi_t), \psi_t)^{-1} \frac{d}{d\theta} \mathcal{L}_{\text{val}}(\hat{\theta}(\psi_t))$$

end for
Experiments

- Optimizing photorealistic Arnold renderer (non-differentiable)
- For object detection on real-world dataset LM-O (with 12 classes)
Results: Faster

50x
Results: More Accurate (mAP)

+8.5%
Results: Requires Lesser Data

![Bar chart showing the number of images required by different methods: AutoSimulate, Bayesian Optimization, Random Search, and LTS. LTS requires the least data.]
Real World Object Detection

- We run each method for a 1,000 epochs
- Val. mAP: maximum validation mAP, Test mAP: test mAP.
- Images and Time: number of images generated and time spent to reach maximum validation mAP.

| Method                      | Val. mAP | Images | Time(s) | Test mAP |
|-----------------------------|----------|--------|---------|----------|
| REINFORCE (LTS)             | 40.2     | 86,150 | 114,360 | 37.2     |
| Bayesian Optimization       | 39.3     | 9,200  | 83,225  | 37.5     |
| Random Search               | 40.3     | 34,300 | 134,318 | 37.0     |
| AutoSimulate                | 37.1     | 8,950  | 23,193  | 36.1     |
| AutoSimulate(Approx Quad)   | 40.1     | 2,950  | 2,321   | 37.4     |
| AutoSimulate (Linear)       | **41.4** | 17,850 | 30,477  | **45.9** |
Results: Object Detections
Ablation: Generalization

• In this ablation, we study whether a simulator trained on a shallow network generalizes to a deeper network.

• Simulator was trained on Yolo and tested on Faster-rcnn.

| Method               | mAP (Faster-rcnn) | mAP (Yolo) |
|----------------------|-------------------|------------|
| REINFORCE (LTS)      | 51.8              | 37.2       |
| Bayesian Optimization| 46.0              | 37.5       |
| Random Search        | 50.3              | 36.8       |
| Ours                 | **55.1**          | **45.9**   |
Ablation: Effect of Freezing Layers

• It is a common practise to train on synthetic data with the initial layers of the network frozen and trained on real data.

• We see the relative performance of different simulator optimization methods under this scenario

| Method             | 0 frozen layers | 98 frozen layers | 104 frozen layers |
|--------------------|-----------------|------------------|-------------------|
|                    | mAP  Time(s) Images | mAP  Time(s) Images | mAP  Time(s) Images |
| REINFORCE (LTS)    | 37.2 114,360 86,150 | 33.0 114,360 86,150 | 31.9 145,193 104,600 |
| Bayesian Optimization | 37.5 83,225 9\,225 | 31.7 13,940 3,550 | 31.7 30,538 6,050 |
| Random Search      | 36.8 134,137 34,300 | 30.2 8,913 3,500 | 28.9 73,411 21,650 |
| Ours               | 45.9 30,477 17,850 | 37.1 2,321 2,950 | 35.8 958 1,000 |
Ablation: Effect of Network Size

• We first examine the effect of network depth on simulator training
• Our method is even faster in optimizing smaller networks.

| Method                  | Yolo-spp | Yolo-Tiny |
|-------------------------|----------|-----------|
|                         | mAP      | Time(s)   | Images | mAP      | Time(s)   | Images |
| REINFORCE (LTS)         | 37.2     | 114,360   | 86,150 | 24.7     | 3,475     | 11,550 |
| Bayesian Optimization   | 37.5     | 83,225    | 9,225  | 19.5     | 65,760    | 35,700 |
| Random Search           | 36.8     | 134,137   | 34,300 | 20.6     | 7,319     | 11,620 |
| Ours                    | **45.9** | **30,477**| 17,850 | **21.2** | **484**   | **280** |
Acknowledgements

- Vibhav Vineet (MSR Redmond)
- Tencent
- Ondrej Miksik (Microsoft AI, Zurich), Tomas Hodan (Czech Technical University Prague)
- Emanuel Shalev, Pedro Urbina (Microsoft Hololens Synthetics Team)
- Prof. M. Pawan Kumar (Oxford)
Thank You