Inferring Graphics Programs from Images

Anonymous Author(s)
Affiliation
Address
`email`

Abstract

1 Introducing visual programs

In this work we consider programs that draw diagrams, similar to those found in papers.

We develop a hybrid architecture for inferring graphics programs. Our approach uses a deep network infer an execution trace from an image; this recovers primitive drawing operations like lines, circles, or arrows. For added robustness we use the deep network as a proposal distribution for a stochastic search over execution traces. Finally we use techniques in the program synthesis community to recover the program from its trace.

Each of these three components – the deep network, the stochastic search, the program synthesizer – confers its own advantages. From the deep network we get a very fast system that can recover plausible execution traces in about a minute. From the stochastic search we get added robustness; essentially the stochastic search can correct mistakes made by the deep network's proposals. From the program synthesizer we get abstraction: our system recovers coordinate transformations, for loops, and subroutines, which are useful for downstream tasks.

2 Neural architecture for inferring image parses

We developed a deep network architecture for efficiently inferring a execution trace, T, from an image, I. Our model constructs the trace one drawing command at a time. When predicting the next drawing command, the network takes as input the target image I as well as the rendered output of previous drawing commands. Intuitively, the network looks at the image it wants to explain, as well as what it has already drawn. It then decides either to stop drawing or proposes another drawing command to add to the execution trace; if it decides to continue drawing, the predicted primitive is rendered to its "canvas" and the process repeats.

Figure 2 illustrates this architecture. We first pass the target image and a rendering of the trace so far to a convolutional network. Given the features extracted by the convolutionional network, a multilayer perceptron then predicts a distribution over the next drawing command to add to the trace. We predict the drawing command token-by-token, and condition each token both on the image features and on the previously generated tokens. For example, the network first decides to emit the `CIRCLE` token conditioned on the image features, then it emits the x coordinate of the circle conditioned on the image features and the `CIRCLE` token, and finally it predicts the y coordinate of the circle conditioned on the image features, the `CIRCLE` token, and the x coordinate.

The distribution over the next drawing command factorizes:

$$\mathbb{P}_\theta[t_1 t_2 \cdots t_K | I, T] = \prod_{k=1}^{K} \mathbb{P}_\theta[t_k | f_\theta(I, \text{render}(T)), \{t_j\}_{j=1}^{k-1}] \tag{1}$$

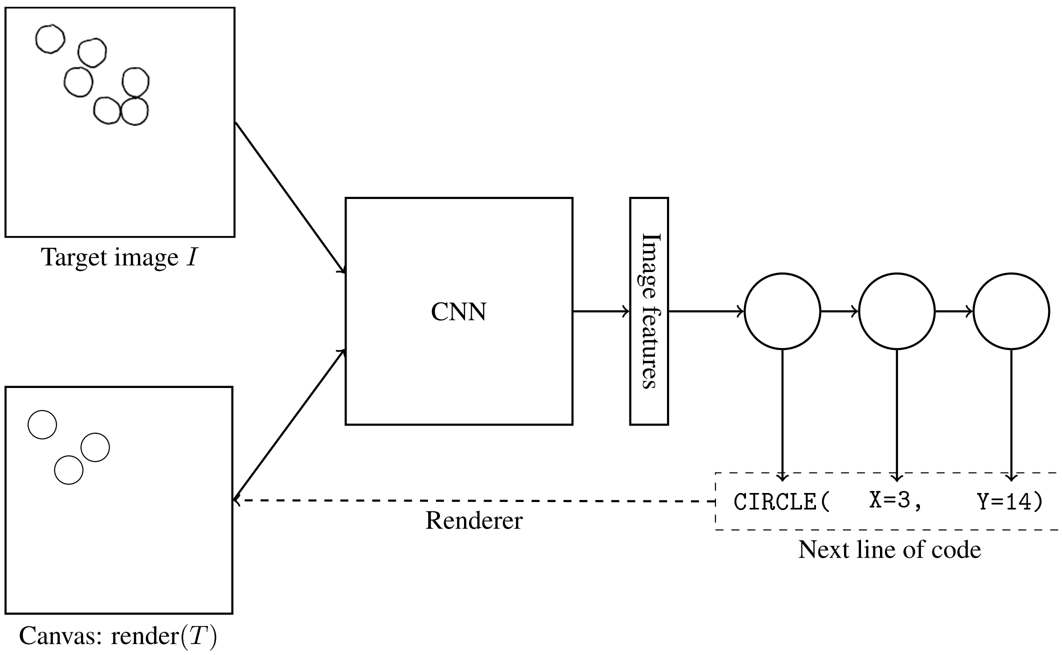
Submitted to 31st Conference on Neural Information Processing Systems (NIPS 2017). Do not distribute.

where $t_1 t_2 \cdots t_K$ are the tokens in the drawing command, I is the target image, T is an execution trace, θ are the parameters of the neural network, and $f_\theta(\cdot, \cdot)$ is the image feature extractor (convolutional network). The distribution over execution traces factorizes as:

$$\mathbb{P}_\theta[T|I] = \prod_{n=1}^{|T|} \mathbb{P}_\theta[T_n|I, T_{1:(n-1)}] \times \mathbb{P}_\theta[\texttt{STOP}|I, T] \tag{2}$$

where $|T|$ is the length of execution trace T, and the `STOP` token is emitted by the network to signal that the execution trace explains the image.

We train the network by sampling execution traces T and target images I for randomly generated scenes, and maximizing (2) wrt θ by gradient ascent. Despite the architecture being recurrent, training is fully supervised. In a sense, this model is like an autoregressive variant of AIR. When we have the generative model (the rendering function), we need not solve an unsupervised or reinforcement learning problem.



3 Generalizing to hand drawings

4 Neural networks for guiding SMC

Let $L(\cdot|\cdot) : \text{image}^2 \to \mathcal{R}$ be our likelihood function: it takes two images, an observed target image and a hypothesized program output, and gives the likelihood of the observed image conditioned on the program output. We want to sample from:

$$\mathbb{P}[p|x] \propto L(x|\text{render}(p))\mathbb{P}[p] \tag{3}$$

where $\mathbb{P}[p]$ is the prior probability of program p, and x is the observed image.

Let p be a program with L lines, which we will write as $p = (p_1, p_2, \cdots, p_L)$. Assume the prior factors into:

$$\mathbb{P}[p] \propto \prod_{l \leq L} \mathbb{P}[p_l] \tag{4}$$

Define the distribution $q_L(\cdot)$, which happens to be proportional to the above posterior:

$$q_L(p_1, p_2, \cdots, p_{L-1}, p_L) \propto q_{L-1}(p_1, p_2, \cdots, p_{L-1}) \times \frac{L(x|\text{render}(p_1, p_2, \cdots, p_{L-1}, p_L))}{L(x|\text{render}(p_1, p_2, \cdots, p_{L-1}))} \times \mathbb{P}[p_L] \tag{5}$$

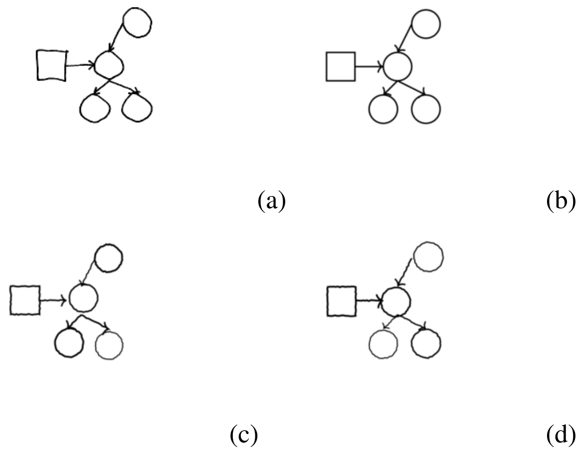


Figure 1: (a): a hand drawing. (b): Rendering of the parse our model infers for (a). We can generalize to hand drawings like these because we train the model on images corrupted by a noise process designed to resemble the kind of noise introduced by hand drawings - see (c) & (d) for noisy renderings of (b).

Now suppose we have some samples from $q_{L-1}(\cdot)$, and that we then sample a p_L from a distribution proportional to $\frac{L(x|\text{render}(p_1,p_2,\cdots,p_{L-1},p_L))}{L(x|\text{render}(p_1,p_2,\cdots,p_{L-1}))} \times \mathbb{P}[p_L]$. The resulting programs p are distributed according to q_L, and so are also distributed according to $\mathbb{P}[p|x]$.

How do we sample p_L from a distribution proportional to $\frac{L(x|\text{render}(p_1,p_2,\cdots,p_{L-1},p_L))}{L(x|\text{render}(p_1,p_2,\cdots,p_{L-1}))} \times \mathbb{P}[p_L]$? We have a neural network that takes as input the target image x and the program so far, and produces a distribution over next lines of code (p_L). We write $\text{NN}(p_L|p_1,\cdots,p_{L-1};x)$ for the distribution output by the neural network. So we can sample from NN and then weight the samples by:

$$w(p_L) = \frac{\mathbb{P}[p_L]}{\text{NN}(p_L|p_1,\cdots,p_{L-1};x)} \times \frac{L(x|\text{render}(p_1,p_2,\cdots,p_{L-1},p_L))}{L(x|\text{render}(p_1,p_2,\cdots,p_{L-1}))} \tag{6}$$

Then we can resample from these now weighted samples to get a new population of particles (here programs are particles), where each program now has L lines instead of $L-1$.

This procedure can be seen as a particle filter, where each successive latent variable is another line of code, and the emission probabilities are successive ratios of likelihoods under $L(\cdot|\cdot)$.

Comments for Dan. Right now I'm not actually sampling from the neural network - instead, I enumerate the top few hundred lines of code suggested by the network, and then weight them by their likelihoods. So actually the form of NN is:

$$\text{NN}(p_L|p_1,\cdots,p_{L-1};x) \propto \begin{cases} 1, & \text{if } p_L \in \text{top hundred neural network proposals} \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

Do you think this is a problem? The neural network puts almost all of its mass on a few guesses. In order to get the correct line of code I sometimes need to get something like the 50th top guess, so I don't want to literally just sample from the distribution suggested by the neural network.

5 Submission of papers to NIPS 2017

NIPS requires electronic submissions. The electronic submission site is

```
https://cmt.research.microsoft.com/NIPS2017/
```

Please read carefully the instructions below and follow them faithfully.

Algorithm 1 Neurally guided SMC

Input: Neural network NN, beam size N, maximum length L, target image x
Output: Samples of the program trace
Set $B_0 = \{$empty program$\}$
for $1 \leq l \leq L$ **do**
  for $1 \leq n \leq N$ **do**
    $p_n \sim \text{Uniform}(B_{l-1})$
    $p'_n \sim \text{NN}(\text{render}(p), x)$
    Define $r_n = p'_n \cdot p_n$
    Set $\tilde{w}(r_n) = \frac{L(x|r_n)}{L(x|p_n)} \times \frac{\mathbb{P}[p'_n]}{\mathbb{P}[p'_n = \text{NN}(\text{render}(p), x)]}$
  end for
  Define $w(p) = \frac{\tilde{w}(p)}{\sum_{p'} \tilde{w}(p')}$
  Set B_l to be N samples from r_n distributed according to $w(\cdot)$
end for
return $\{p : p \in B_{l \leq L}, p$ is finished$\}$

5.1 Style

Papers to be submitted to NIPS 2017 must be prepared according to the instructions presented here. Papers may only be up to eight pages long, including figures. This does not include acknowledgments and cited references which are allowed on subsequent pages. Papers that exceed these limits will not be reviewed, or in any other way considered for presentation at the conference.

The margins in 2017 are the same as since 2007, which allow for $\sim 15\%$ more words in the paper compared to earlier years.

Authors are required to use the NIPS LaTeX style files obtainable at the NIPS website as indicated below. Please make sure you use the current files and not previous versions. Tweaking the style files may be grounds for rejection.

5.2 Retrieval of style files

The style files for NIPS and other conference information are available on the World Wide Web at

`http://www.nips.cc/`

The file `nips_2017.pdf` contains these instructions and illustrates the various formatting requirements your NIPS paper must satisfy.

The only supported style file for NIPS 2017 is `nips_2017.sty`, rewritten for LaTeX 2_ε. **Previous style files for LaTeX 2.09, Microsoft Word, and RTF are no longer supported!**

The new LaTeX style file contains two optional arguments: `final`, which creates a camera-ready copy, and `nonatbib`, which will not load the `natbib` package for you in case of package clash.

At submission time, please omit the `final` option. This will anonymize your submission and add line numbers to aid review. Please do *not* refer to these line numbers in your paper as they will be removed during generation of camera-ready copies.

The file `nips_2017.tex` may be used as a "shell" for writing your paper. All you have to do is replace the author, title, abstract, and text of the paper with your own.

The formatting instructions contained in these style files are summarized in Sections 6, 7, and 8 below.

6 General formatting instructions

The text must be confined within a rectangle 5.5 inches (33 picas) wide and 9 inches (54 picas) long. The left margin is 1.5 inch (9 picas). Use 10 point type with a vertical spacing (leading) of 11 points.

Times New Roman is the preferred typeface throughout, and will be selected for you by default. Paragraphs are separated by ½ line space (5.5 points), with no indentation.

The paper title should be 17 point, initial caps/lower case, bold, centered between two horizontal rules. The top rule should be 4 points thick and the bottom rule should be 1 point thick. Allow ¼ inch space above and below the title to rules. All pages should start at 1 inch (6 picas) from the top of the page.

For the final version, authors' names are set in boldface, and each name is centered above the corresponding address. The lead author's name is to be listed first (left-most), and the co-authors' names (if different address) are set to follow. If there is only one co-author, list both author and co-author side by side.

Please pay special attention to the instructions in Section 8 regarding figures, tables, acknowledgments, and references.

7 Headings: first level

All headings should be lower case (except for first word and proper nouns), flush left, and bold.

First-level headings should be in 12-point type.

7.1 Headings: second level

Second-level headings should be in 10-point type.

7.1.1 Headings: third level

Third-level headings should be in 10-point type.

Paragraphs There is also a `\paragraph` command available, which sets the heading in bold, flush left, and inline with the text, with the heading followed by 1 em of space.

8 Citations, figures, tables, references

These instructions apply to everyone.

8.1 Citations within the text

The `natbib` package will be loaded for you by default. Citations may be author/year or numeric, as long as you maintain internal consistency. As to the format of the references themselves, any style is acceptable as long as it is used consistently.

The documentation for `natbib` may be found at

`http://mirrors.ctan.org/macros/latex/contrib/natbib/natnotes.pdf`

Of note is the command `\citet`, which produces citations appropriate for use in inline text. For example,

```
\citet{hasselmo} investigated\dots
```

produces

> Hasselmo, et al. (1995) investigated…

If you wish to load the `natbib` package with options, you may add the following before loading the `nips_2017` package:

```
\PassOptionsToPackage{options}{natbib}
```

If `natbib` clashes with another package you load, you can add the optional argument `nonatbib` when loading the style file:

```
\usepackage[nonatbib]{nips_2017}
```

As submission is double blind, refer to your own published work in the third person. That is, use "In the previous work of Jones et al. [4]," not "In our previous work [4]." If you cite your other papers that are not widely available (e.g., a journal paper under review), use anonymous author names in the citation, e.g., an author of the form "A. Anonymous."

8.2 Footnotes

Footnotes should be used sparingly. If you do require a footnote, indicate footnotes with a number[1] in the text. Place the footnotes at the bottom of the page on which they appear. Precede the footnote with a horizontal rule of 2 inches (12 picas).

Note that footnotes are properly typeset *after* punctuation marks.[2]

8.3 Figures

All artwork must be neat, clean, and legible. Lines should be dark enough for purposes of reproduction. The figure number and caption always appear after the figure. Place one line space before the figure caption and one line space after the figure. The figure caption should be lower case (except for first word and proper nouns); figures are numbered consecutively.

You may use color figures. However, it is best for the figure captions and the paper body to be legible if the paper is printed in either black/white or in color.

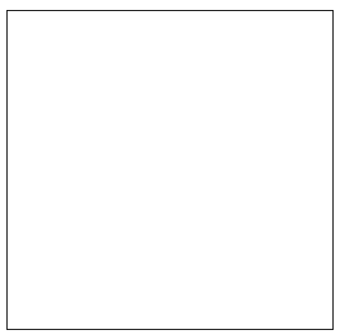


Figure 2: Sample figure caption.

8.4 Tables

All tables must be centered, neat, clean and legible. The table number and title always appear before the table. See Table 1.

Place one line space before the table title, one line space after the table title, and one line space after the table. The table title must be lower case (except for first word and proper nouns); tables are numbered consecutively.

Note that publication-quality tables *do not contain vertical rules.* We strongly suggest the use of the `booktabs` package, which allows for typesetting high-quality, professional tables:

`https://www.ctan.org/pkg/booktabs`

This package was used to typeset Table 1.

[1]Sample of the first footnote.
[2]As in this example.

Table 1: Sample table title

Part		
Name	Description	Size (μm)
Dendrite	Input terminal	$\sim$100
Axon	Output terminal	$\sim$10
Soma	Cell body	up to 10^6

9 Final instructions

Do not change any aspects of the formatting parameters in the style files. In particular, do not modify the width or length of the rectangle the text should fit into, and do not change font sizes (except perhaps in the **References** section; see below). Please note that pages should be numbered.

10 Preparing PDF files

Please prepare submission files with paper size "US Letter," and not, for example, "A4."

Fonts were the main cause of problems in the past years. Your PDF file must only contain Type 1 or Embedded TrueType fonts. Here are a few instructions to achieve this.

- You should directly generate PDF files using `pdflatex`.
- You can check which fonts a PDF files uses. In Acrobat Reader, select the menu Files>Document Properties>Fonts and select Show All Fonts. You can also use the program `pdffonts` which comes with `xpdf` and is available out-of-the-box on most Linux machines.
- The IEEE has recommendations for generating PDF files whose fonts are also acceptable for NIPS. Please see `http://www.emfield.org/icuwb2010/downloads/IEEE-PDF-SpecV32.pdf`
- `xfig` "patterned" shapes are implemented with bitmap fonts. Use "solid" shapes instead.
- The `\bbold` package almost always uses bitmap fonts. You should use the equivalent AMS Fonts:

  ```
  \usepackage{amsfonts}
  ```

  followed by, e.g., `\mathbb{R}`, `\mathbb{N}`, or `\mathbb{C}` for $\mathbb{R}$, $\mathbb{N}$ or $\mathbb{C}$. You can also use the following workaround for reals, natural and complex:

  ```
  \newcommand{\RR}{I\!\!R} %real numbers
  \newcommand{\Nat}{I\!\!N} %natural numbers
  \newcommand{\CC}{I\!\!\!\!C} %complex numbers
  ```

  Note that `amsfonts` is automatically loaded by the `amssymb` package.

If your file contains type 3 fonts or non embedded TrueType fonts, we will ask you to fix it.

10.1 Margins in LaTeX

Most of the margin problems come from figures positioned by hand using `\special` or other commands. We suggest using the command `\includegraphics` from the `graphicx` package. Always specify the figure width as a multiple of the line width as in the example below:

```
\usepackage[pdftex]{graphicx} ...
\includegraphics[width=0.8\linewidth]{myfile.pdf}
```

See Section 4.4 in the graphics bundle documentation (`http://mirrors.ctan.org/macros/latex/required/graphics/grfguide.pdf`)

A number of width problems arise when LaTeX cannot properly hyphenate a line. Please give LaTeX hyphenation hints using the `\-` command when necessary.

Acknowledgments

Use unnumbered third level headings for the acknowledgments. All acknowledgments go at the end of the paper. Do not include acknowledgments in the anonymized submission, only in the final paper.

References

References follow the acknowledgments. Use unnumbered first-level heading for the references. Any choice of citation style is acceptable as long as you are consistent. It is permissible to reduce the font size to `small` (9 point) when listing the references. **Remember that you can go over 8 pages as long as the subsequent ones contain *only* cited references.**

[1] Alexander, J.A. & Mozer, M.C. (1995) Template-based algorithms for connectionist rule extraction. In G. Tesauro, D.S. Touretzky and T.K. Leen (eds.), *Advances in Neural Information Processing Systems 7*, pp. 609–616. Cambridge, MA: MIT Press.

[2] Bower, J.M. & Beeman, D. (1995) *The Book of GENESIS: Exploring Realistic Neural Models with the GEneral NEural SImulation System.* New York: TELOS/Springer–Verlag.

[3] Hasselmo, M.E., Schnell, E. & Barkai, E. (1995) Dynamics of learning and recall at excitatory recurrent synapses and cholinergic modulation in rat hippocampal region CA3. *Journal of Neuroscience* **15**(7):5249-5262.