

	<b>Category Shorthand</b>	<b>Category Description</b>	<b>Comment</b>	
	Learning to Synthesize	Using machine learning to (help) map a specifications to code		
	Generative Models	Using machine learning to generate natural code	Well covered by Allamanis et al survey	
	Understanding Code	Using machine learning to classify, label, or otherwise understand code	Well covered by Allamanis et al survey	
	Classical Synthesis	Mapping specifications to code using minimal if any learning		
	Graph Models	Machine learning on graphs		
	Program-like Neural Networks	Neural networks inspired by programs (e.g., Neural Turing Machines)	Not covered in this anthology	
<b>Comments</b>	<b>Category</b>	<b>Paper and link</b>	<b>Publication Venue</b>	<b>Authors</b>
The FlashFill paper.	Classical Synthesis	<a href="#">The Sketching Approach to Program Synthesis</a>	POPL 2011	Armando Solar-Lezama
	Classical Synthesis	<a href="#">Automating String Processing in Spreadsheets using Input-Output Examples</a>	CACM 2016	Sumit Gulwani
	Classical Synthesis	<a href="#">STOKE: A stochastic superoptimizer and program synthesizer</a>		Schkufza, Sharma, Aiken (and others)
First real ML approach to learning to map I/O to code?	Learning to Synthesize	<a href="#">A Machine Learning Framework for Programming by Example</a>	ICML 2013	Aditya Krishna Menon, Omer Tamuz, Sumit Gulwani, Butler Lampson, Adam Kalai
Combine trained neural network guidance with the special-purpose solutions from PL that work best.	Learning to Synthesize	<a href="#">DeepCoder</a>	ICLR 2017	Matej Balog, Alexander L. Gaunt, Marc Brockschmidt, Sebastian Nowozin, Daniel Tarlow
	Learning to Synthesize	<a href="#">Neuro Symbolic Program Synthesis</a>	ICLR 2017	Emilio Parisotto, Abdel-rahman Mohamed, Rishabh Singh, Lihong Li, Dengyong Zhou, Pushmeet Kohli
	Learning to Synthesize	<a href="#">RobustFill: Neural Program Learning under Noisy I/O</a>	ICML 2017	Jacob Devlin, Jonathan Uesato, Surya Bhupatiraju, Rishabh Singh, Abdel-rahman Mohamed, Pushmeet Kohli
Mostly a negative result about the usefulness of searching over program space using gradient descent.	Learning to Synthesize	<a href="#">TerpreT: A Probabilistic Programming Language for Program Induction</a>	ARXIV 2016	Alexander L. Gaunt, Marc Brockschmidt, Rishabh Singh, Nate Kushman, Pushmeet Kohli, Jonathan Taylor, Daniel Tarlow
A first attempt at applying ML to STOKE formulation.	Learning to Synthesize	<a href="#">Learning to Superoptimize Programs</a>	ICLR 2017	Rudy Bunel, Alban Desmaison, M. Pawan Kumar, Philip H.S. Torr, Pushmeet Kohli
	Learning to Synthesize	<a href="#">Neural Program Synthesis with Priority Queue Training</a>		Daniel A. Abolafia, Mohammad Norouzi, Jonathan Shen, Rui Zhao, Quoc V. Le
	Learning to Synthesize	<a href="#">DeepFix: Fixing Common C Language Errors by Deep Learning</a>	AAAI 2017	Rahul Gupta, Soham Pal, Aditya Kanade, Shirish Shevade
This entry and next are most complete source for "Generative Models" and "Understanding Code,"	Broad Survey	<a href="#">A Survey of Machine Learning for Big Code and Naturalness</a>		Miltiadis Allamanis, Earl T. Barr, Premkumar Devanbu, Charles Sutton
	Collected Papers	<a href="#">Automated Programming Papers</a>		
	Graph Models	<a href="#">See list at <a href="https://github.com/allamanis/graph-neural-network-papers">go/graph-neural-network-papers</a></a>		