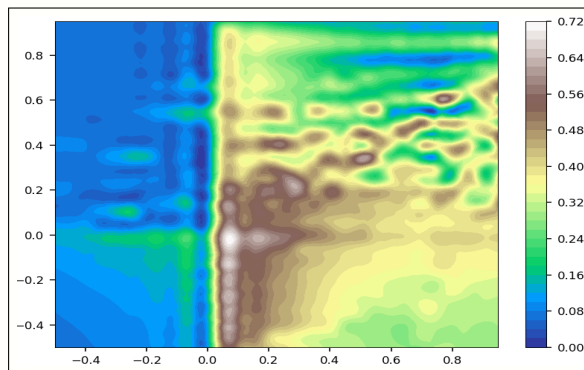
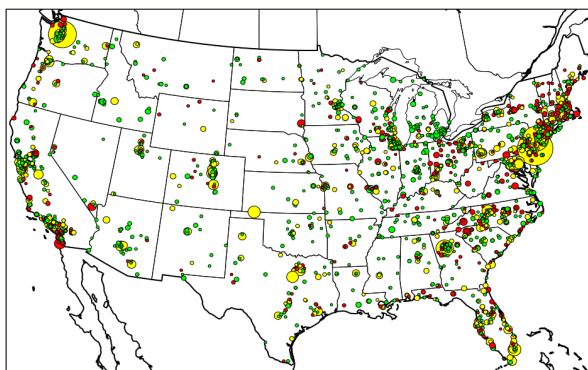
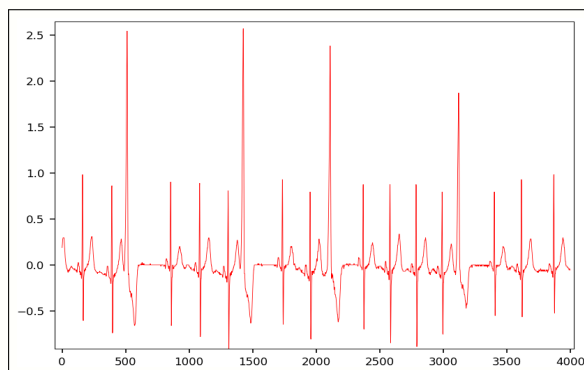
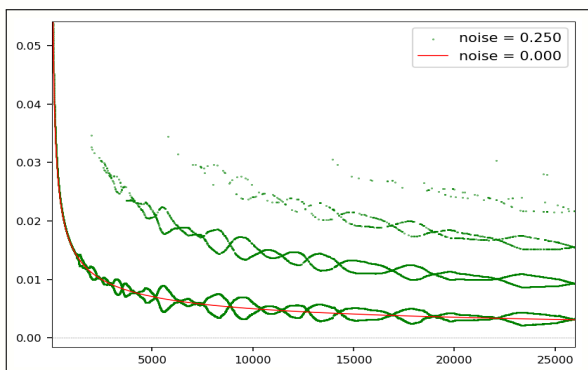
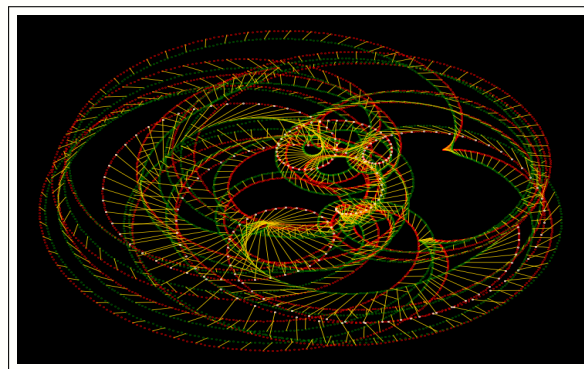
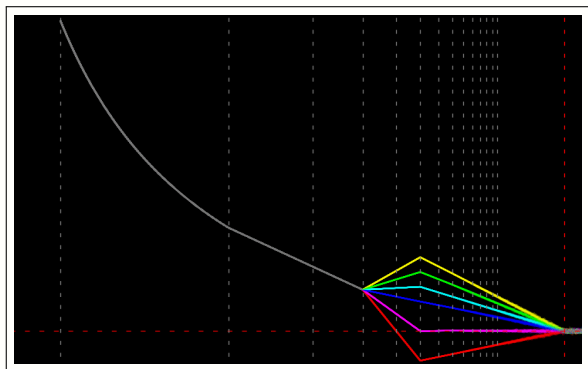
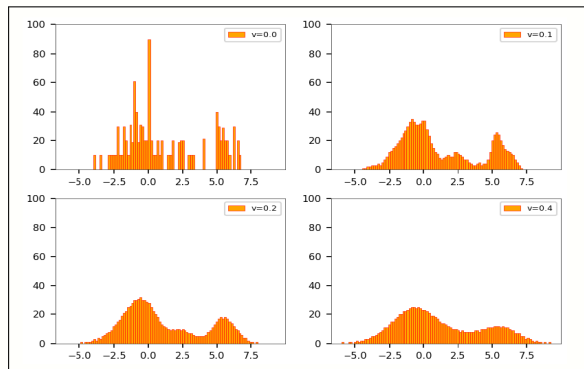
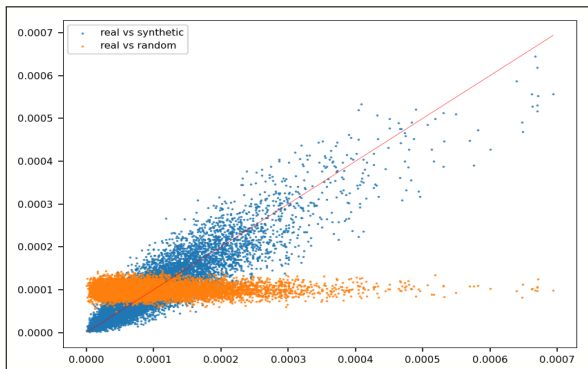


No-Blackbox, Secure, Efficient AI & LLM Solutions

Vincent Granville PhD, Chief AI Architect, [BondingAI.io](https://bondingai.io)

vincent@bondingai.io | linkedin.com/in/vincentg/

January 2026



Preface

Large language models and modern AI is often presented as technology that needs deep neural networks (DNNs) with billions of Blackbox parameters, expensive and time consuming training, along with GPU farms, yet prone to hallucinations. This book presents alternatives that rely on explainable AI, featuring new algorithms based on radically different technology with trustworthy, auditable, fast, accurate, secure, replicable Enterprise AI. Most of the material is proprietary and made from scratch, showcasing the culmination of decades of research away from standard models to establish a new framework in machine learning and AI technology.

I discuss an efficient DNN architecture based on a new type of universal functions in chapter 4, with DNN distillation and protection via watermarking in chapter 5. Then, in chapter 6, I discuss non-DNN alternatives that yield exact interpolation on the training set yet benefit from benign overfitting in any dimension. Accurate predictions are obtained with a simple closed-form expression, without gradient descent or other iterative optimization technique, essentially without training.

Case studies include 96% correct predictions for the next token on a Nvidia PDF repository, automated heart beat clustering and unusually high data compression rates (big data), anomaly detection and fraud litigation linked to large-scale cybersecurity breach (large Excel repository, automated SQL, time series and geospatial data) as well as predicting next sequence on real-world genome data with home-made LLM technology. Some datasets with 1000 dimensions are generated with the best and fastest tabular data synthesizer on the market, described in details in chapter 2 along with the best model evaluation metric. These cases correspond to different agents linked to the xLLM technology (extreme LLM) developed by the author.

I barely use Python libraries other than Numpy, staying away from TensorFlow, PyTorch or Keras. It gives you full control over the code. Also, I avoid mathematical and probabilistic models when not beneficial, making the content accessible to a larger audience not versed in statistical, probabilistic or mathematical jargon. While classic books on the subject include an introduction to calculus, algebra, probability and matrix theory, here it is replaced by outside-the-box problems with solutions. It includes quantum systems, quantum approximation, non-causal signal processing, convolution, automated curve fitting without iterative algorithm, and deep dive into one of the universal functions central to my DNN, a sister of the famous Riemann zeta function in number theory.

About the author

Vincent Granville is a well-known pioneering AI scientist and machine learning expert, Chief AI Architect at [BondingAI.io](https://bondingai.io), author and patents owner with some related to trustworthiness scores. Vincent worked with Visa (credit card fraud), Wells Fargo, eBay (Google keyword campaigns), NBC, Microsoft, and CNET. He is currently working on no-Blackbox, auditable, hallucination-free secure Enterprise AI requiring no GPU and offering relevancy and trustworthiness scores in prompt results. Also, Vincent recently developed new, explainable deep neural network models along with distillation-resistant watermarking technology for model and data protection, to detect unauthorized uses.



Vincent is also a former post-doc at Cambridge University, and the National Institute of Statistical Sciences (NISS). He published in *IEEE Transactions on Pattern Analysis and Machine Intelligence* (500+ citations), *Journal of Number Theory*, and *Journal of the Royal Statistical Society* (Series B). He is the author of multiple books, available [here](#), including “Synthetic Data and Generative AI” (Elsevier, 2024). Vincent lives in Washington state, and enjoys doing research on stochastic processes, dynamical systems, experimental math and probabilistic number theory. See Vincent’s profile on LinkedIn, [here](#).

Contents

1	Innovative AI and Machine Learning Solutions	7
1.1	Generic, all-in-one curve fitting, regression and clustering	7
1.1.1	Solution, R-squared and backward compatibility	7
1.1.2	Model upgrades	8
1.1.3	Case studies	9
1.1.3.1	Logistic regression, two ways	9
1.1.3.2	Ellipsoid and hyperplane fitting	10
1.1.3.3	Curve fitting: 250 examples in one video	10
1.1.3.4	Confidence region for the fitted ellipse: application to meteorite shapes	11
1.1.4	Python code	12
1.1.4.1	Sampling points on an ellipse arc	12
1.1.4.2	Training set and ellipse parameters	13
1.1.4.3	Confidence regions versus curve fitting	13
1.2	Sampling Outside the Observation Range	13
1.2.1	Quantile convolution	13
1.2.2	Truncated Gaussian mixtures and bias detection	15
1.2.3	Case studies	16
1.2.3.1	Conclusion	16
1.2.4	Python code	17
1.3	Cybersecurity Use Case: Detecting Fraudulent Internet Traffic	20
1.3.1	Introduction: click fraud	20
1.3.2	Click data: general patterns	21
1.3.2.1	Time series analysis: random spikes	22
1.3.2.2	Geospatial data: problematic zipcodes	23
1.3.3	Pattern discovery: deeper dive	23
1.3.3.1	Anatomy of an hourly spike	23
1.3.3.2	Anatomy of a giant suspicious zipcode	24
1.3.3.3	Other findings	24
1.3.4	Addendum	25
1.3.4.1	Conclusion	25
1.3.4.2	Data and Python code	25
1.4	Synthesizing DNA sequences with LLM techniques	28
1.4.1	Project and solution	28
1.4.2	Python code	30
1.5	Parsing PDFs with tables and images for RAG/LLM	33
1.5.1	Hierarchical chunking with multi-index	34
1.5.2	Duplicate and overlapping chunks	35
1.5.3	PDF parser and data	35
1.6	Pattern detection and compression for electrocardiogram data	41
1.6.1	Compressing ECG signals	41
1.6.2	Pattern detection and heart beat classification	42
1.6.2.1	Heart beats clustering	43
1.6.3	Dataset and Python code	43
2	Fast, efficient NoGAN Tabular Data Synthesis	47
2.1	Fast, high-quality NoGAN synthesizer for tabular data	47
2.1.1	Project description	47
2.1.2	Solution	49

2.1.3	Python implementation	50
2.2	Hierarchical Bayesian NoGAN for data synthesis	56
2.2.1	Methodology	56
2.2.1.1	Base algorithm	56
2.2.1.2	Loss function	57
2.2.1.3	Hyperparameters and convergence	58
2.2.2	Case studies	59
2.2.2.1	Synthesizing the student dataset	60
2.2.2.2	Synthesizing the Telecom dataset	61
2.2.2.3	Other case studies	63
2.2.2.4	Auto-tuning the hyperparameters	63
2.2.2.5	Evaluation with multivariate ECDF and KS distance	65
2.2.3	Conclusion	65
2.2.4	Python implementation	66
2.3	Boosting Model Evaluation with Smart Adaptive Loss Functions	75
2.3.1	Project and solution	76
2.3.2	Python code	79
3	From a Deep Math Conjecture to Quantum Systems	84
3.1	Introduction	84
3.2	Logistic map and the digit sum function	85
3.2.1	Model comparison, with illustrations	85
3.2.2	Normality of special math constants	88
3.2.3	Applications and references	88
3.3	Re-balancing an uneven digit distribution	89
3.3.1	Digit-balancing transforms	89
3.3.2	Digit block balancing	91
3.4	Conclusion	95
3.5	Main Python code	96
4	A New Type of Deep Neural Network Architecture	99
4.1	From standard to non-standard deep neural networks	99
4.1.1	Architecture of a non-standard DNN	100
4.1.2	Gradient descent algorithm	100
4.1.3	Reparameterization and latent features	102
4.1.4	Distillation, cross-validation, and noise injection	103
4.1.5	Adaptive equalizer and other optimization techniques	103
4.2	Case studies	104
4.2.1	The workhorse of non-standard DNNs	105
4.2.2	A new type of universal approximation	105
4.2.3	Dealing with singularities and extreme events	106
4.2.4	Large language models	108
4.3	Model evaluation	109
4.3.1	Model with multidimensional input data	110
4.3.2	An interesting universal function	110
4.3.3	Example with singularities	110
4.4	Conclusions	111
4.5	Python code	111
5	Protecting AI Models and Data with Watermarking	118
5.1	Testing your model on synthetic data	118
5.1.1	Methodology	119
5.1.2	Results and conclusion	122
5.2	Impact of initial weights on convergence	123
5.3	Weights ghosting, blurring and model watermarking	125
5.3.1	Model sensitivity to weight ghosting and blurring	126
5.3.2	Model watermarking with Python code	127
5.3.3	References	133
6	Blackbox-free Alternative to Deep Neural Networks	134
6.1	The model	134

6.1.1	Description and formulation	134
6.1.2	Model parameters	136
6.1.3	General architecture and components	136
6.1.3.1	Beyond simple prediction	137
6.1.3.2	Recalibrating predicted values	137
6.1.3.3	Non-linear piecewise interpolation and batch processing	138
6.1.3.4	Performance analysis and numerical stability	138
6.1.3.5	Cutoff threshold to speed up computations	138
6.1.3.6	Training set, cross-validation and model evaluation	139
6.1.3.7	Fixing hallucinations	139
6.1.3.8	Sensitivity to noise injection	140
6.1.3.9	Nodes distillation	140
6.1.3.10	Normalization with stable transforms	140
6.1.3.11	Dealing with a mix of categorical, numerical and text data	141
6.1.3.12	Replicability	141
6.1.3.13	High quality random numbers	141
6.1.3.14	Slicing the response	141
6.2	Case studies and results	142
6.2.1	Mixture with positive and negative weights	142
6.2.2	A spectacular example defying all statistical laws	144
6.2.3	Specialized language models	145
6.2.3.1	Finding related multitokens	146
6.2.3.2	Memory optimization and training set size	147
6.2.3.3	The function $f(x)$	148
6.2.3.4	Predicting the next token	149
6.2.3.5	Enhanced model	149
6.3	Addendum	150
6.3.1	Next steps and related models	150
6.3.2	Data and Python code	150
6.3.2.1	Numerical data	151
6.3.2.2	Text data for LLMs	156
7	Creative Problems in Signal Processing, Linear Algebra and Analysis	164
7.1	Non-causal discrete convolution with Gaussian kernel	164
7.1.1	Problem	165
7.1.2	Solution	166
7.1.3	Python code	166
7.2	Three beautiful Taylor series	169
7.2.1	Solution	169
7.2.2	Generalizations	170
7.3	Beautiful integrals computed via statistical techniques	170
7.3.1	Solution	171
7.3.2	Generalization	172
7.4	Random Fibonacci sequences and continued fractions	172
7.4.1	Solution	172
7.4.2	Python code	173
7.5	Hyperbolic functions in perspective geometry	174
7.5.1	Apparent versus real distance	175
7.5.2	Computer vision problems	176
7.6	Matrix operations: A^B , factoring A into prime matrices	177
7.6.1	Computing A^B when A and B are matrices	177
7.6.2	Factoring an integer matrix into prime matrices	177
7.7	Miscellaneous problems	180
8	Creative Problems in Number Theory	183
8.1	Synthetic primes, quantum states, and the Riemann Hypothesis	183
8.1.1	Definitions	183
8.1.2	Building a Beurling eta function by deletion	184
8.1.3	Building a Beurling eta function by swapping	184
8.1.4	Applications and Python code	185
8.2	Quantum derivatives, GenAI, and the Riemann Hypothesis	191

8.2.1	Cornerstone result to bypass the roadblocks	192
8.2.2	Quantum derivative of functions nowhere differentiable	193
8.2.3	Project and solution	194
8.2.4	Python code	198
8.3	Approximations to mathematical function	202
8.3.1	Finding the roots of ζ with fast-converging series	202
8.3.2	Approximation based on quantization	203
8.4	Simple normality test with application to PRNGs	204
8.4.1	High performance computing with Chebyshev polynomials	205
8.4.2	Application with test of randomness and Python code	205
8.4.3	Problem and solution	207
	Bibliography	209
	Index	212