# Literature Review of In-Context Learning of Simple Function Classes

Yuxi Liu

2024-04-18

**Abstract**

This is a literature review on 223 papers that cited [Garg et al., 2022] as of 2024-04-18, categorized into 5 classes.

There is an appendix on how to do literature review with the help of large language models.

## 1 Overview

In [Garg et al., 2022], the authors studied how a GPT-2-like Transformer would perform in-context learning of random simple functions sampled from a class, such as random linear functions.

This paper is an exhaustive literature review of all 223 papers that has cited it at the time of writing (2024-04-18). We categorize them into the following classes.

1. **Bayes and generalization:** Many papers showed that the trained model acts like a Bayes optimal predictor with a prior fitted to the training set. However, a few papers showed that models can learn to generalize beyond the specific ICL tasks presented in the training set. They typically find that models, when trained on enough variety of ICL tasks, can generalize, and perform better on the test set than predictors that are Bayes-optimal on the training set.
2. **Mechanistic interpretation:** These papers investigate the internal mechanisms of trained models, particularly how they encode information and perform computations within their layers.

They often find that the trained model performs Bayes-optimal prediction.

3. **Theoretical foundations:** These papers contain formal proofs about the learning dynamics and generalization properties of ICL models, typically by heavy linear algebra.
4. **Empirical improvement:** These papers empirically explore how to improve ICL performance by varying dataset composition, model architecture, curriculum learning strategies, etc.
5. **Irrelevant:** These papers are ignored. They typically cite [Garg et al., 2022] in a perfunctory paragraph in the perfunctory literature review.

Most subsequent work follows [Garg et al., 2022] in training GPT-2-like transformers on ICL with linear regression tasks, so this is the default setting. We only note where they differed.

# 2 Previous work

This section reviews precedent works that are most relevant for [Garg et al., 2022] and its descendant works.

## 2.1 Empirical Findings

The widespread interest in ICL started with [Brown et al., 2020], which demonstrated the potential of ICL with GPT-3. Subsequent work, such as [Wei et al., 2022], has further improved ICL capabilities through instruction tuning, while [Kojima et al., 2022] introduced chain-of-thought (CoT) prompting, triggering widespread interest in prompt engineering.

[Min et al., 2022b] showed that the mechanism of ICL is nontrivial and unintuitive, by empirically investigating LLM with ICL. They found that it is unnecessary to have correct in-context examples, and that the examples are mainly informative for other reasons. This showed the necessity in deeply explaining simple toy models of ICL.

[Chan et al., 2022] showed that pretrained LLM are better able to do ICL if the pretraining dataset has certain properties, such as non-IID, burstiness, long-tailed distributions, and contextuality. This can be intuitively interpreted as saying that LLM pretrained on generating text that resembles the prompt format of ICL are better at ICL.

This motivates the toy model of [Garg et al., 2022], where the entire training dataset is purely ICL.

# 3 Garg et al

[Garg et al., 2022] conducted a comprehensive study on the ICL capabilities of transformers across various simple function classes. Their work aimed to understand how effectively transformers can learn and generalize from a few input-output examples provided as context.

The study employed a setup where a transformer would be trained on different function classes, including linear, sparse linear, two-layer neural networks, and decision trees. The training process involved presenting the model with prompts containing input-output pairs and optimizing it to minimize the prediction error on unseen inputs.

The key aspects of this work are

1. **Bayes optimal prediction:** For linear regression tasks, the trained transformers exhibited ICL behavior close to that of Bayesian predictors, which in this case means min-norm least squares regression.
2. **Generalization:** The trained transformers generalized to new inputs, even in out-of-distribution (OOD) scenarios such as skewed input covariance, or different orthants.
3. **Mechanistic interpretation:** For sparse linear regression, the trained transformers demonstrated performance comparable to iterative LASSO regression in a single pass. It was unclear how they performed this task, hinting at nontrivial mechanistic interpretation.
4. **Curriculum learning:** If the complexity of tasks gradually increased from easy to hard, training speed and performance greatly increased.

Each aspect of the work has been investigated further in subsequent work.

# 4 Subsequent work

## 4.1 Optimality and generalization

Many subsequent papers investigated the trained models, and typically found a three-step process: When the model has low capacity (shallow and narrow), the trained model would learn to perform one-step gradient descent. When the model has high capacity, the trained model would learn to perform Bayes-optimal prediction, often rivaling the best available algorithm.

Several studies interpret ICL through the lens of Bayesian inference. [Min et al., 2022a] propose a noisy channel model for text classification with LLMs, where the probability of an input given a label is proportional to the product of the likelihood and prior.

[Xie et al., 2022] view ICL as implicit Bayesian inference, where the model integrates over possible concepts to generate outputs conditioned on the prompt.

These Bayesian interpretations suggest that LLMs may be performing a form of probabilistic reasoning, integrating prior knowledge with the information provided in the context to make predictions. This aligns with the observation that LLMs trained on diverse and realistic data exhibit better ICL capabilities, as such data provides a richer prior for the model to leverage.

Many empirical studies agree with [Garg et al., 2022] that the trained model acts like a Bayes-optimal ICL with a prior distribution fitted to the training dataset.

However, several studies showed that under certain conditions, the trained model can generalize beyond the training dataset.

[Yadlowsky et al., 2023] was the earliest subsequent work that tried mixtures of different function classes. They found that models trained on a balanced mixture had better generalization OOD, a finding confirmed subsequently.

[Raventós et al., 2024] show that with sufficient task diversity in the training set, ICL models can achieve near-optimal performance on unseen tasks, even surpassing Bayesian predictors based on the convex hull of the training data.

Similarly, [Panwar et al., 2023] and [Panwar et al., 2024] demonstrate that when exposed to a sufficiently diverse set of function classes during training, ICL models can generalize to entirely new classes, outperforming Bayesian predictors limited to the training set.

# 5 Mechanistic interpretation

A central question in ICL research is how LLMs encode information and perform computations within their layers to achieve their impressive performance. Several studies have investigated this aspect, revealing intriguing insights into the inner workings of ICL models.

[Guo et al., 2023] investigated ICL for featurized linear regression, where the input undergoes a fixed non-linear transformation. They theoretically and empirically demonstrate that Transformers can effectively learn this task, with lower layers encoding the features and upper layers performing ridge regression.

[Pathak et al., 2023] trained models on ICL with mixture noisy linear models, achieving performance nearly on par with optimal oracle algorithm, despite having no access to the oracle information.

[Ahuja et al., 2023] trained models on ICL with various linear inverse problems, obtaining models that resemble penalty-based and Bayesian approaches. They also show successful handling of mixed problem types, echoing findings from [Yadlowsky et al., 2023].

## 5.1 Gradient descent

A large cluster of subsequent work focused on interpreting the trained model as running varieties of gradient descent, or objections to this interpretation. Most of the supporting work are detailed in the section on theory.

The two papers that started this line of work are [Akyürek et al., 2023, Von Oswald et al., 2023], which constructed transformers implementing gradient descent and ridge regression, and found that trained models do agree with these algorithms. Low-capacity models would learn gradient descent, and high-capacity, ridge regression. [Von Oswald et al., 2023] also found that a learned model would in the early layers encode incoming tokens into a format amenable to GD, then performs GD in the later layers of the Transformer.

[Cheng et al., 2024] showed transformers learn *functional* gradient descent, enabling them to learn non-linear functions.

[Fu et al., 2023] objected that transformers learn a higher-order optimization method, i.e. Iterative Newton's Method.

[Ding et al., 2024] showed that T5-like transformers outperform GPT-like autoregressive transformers, with the former converging to optimal solutions while the latter converges to gradient descent.

[Shen et al., 2024] found that, while transformers have the capacity to simulate gradient descent for ICL, real-world models like GPT-3 exhibit different behavior on ICL tasks compared to models trained specifically for ICL, bringing doubt to the practical relevance of the toy model.

[Mahdavi et al., 2024] revisit the equivalence between ICL and gradient descent, showing that strong assumptions like feature independence are needed for exact equivalence and that under weaker assumptions, the process resembles preconditioned gradient descent.

### 5.2 Other mechanisms

Other than gradient descent and linear algebraic algorithms, there were a few other proposed mechanisms. [Ren and Liu, 2023] proposed contrastive learning, [Han et al., 2023] kernel regression, [Reddy, 2023] induction heads, [Abernethy et al., 2023] sequence segmentation.

## 6 Theory

Most theoretical studies based on [Garg et al., 2022] has the following kinds of contents:

1. **Existence by construction:** Write down the model parameters to perform an algorithm like gradient descent.
2. **Convergence proof:** The model actually converges to a global or local optimum.
3. **Mechanistic interpretation:** At the optimal point, the model performs some known algorithm like preconditioned gradient descent.

For theoretical tractability, most of them analyzed only a single linear self-attention block trained by gradient flow. Most of them verify their theorems experimentally. We note where they differed from this. While one might doubt the realism of this simplification, [Ahn et al., 2024b] empirically shows that such models trained for linear regression ICL reproduce most of the interesting phenomena exhibited by a standard decoder-only transformer. On the other hand, [Kim and Suzuki, 2024] highlight the role of feedforward layers in expanding the range of learnable functions to the Barron space, enabling greater flexibility in ICL.

[Bai et al., 2023] constructed linear transformers for various statistical algorithms, including least squares, ridge regression, LASSO, and convex risk minimization. They also proved guarantees for expressive power, prediction performance, and sample complexity.

[Ahn et al., 2024a] theoretically prove that linear transformers for ICL linear regression implement forms of preconditioned gradient descent, adapting to data distribution and variance.

[Lin and Lee, 2024] proved dual operating modes (learning and retrieval) in linear transformers, explaining phenomena like "early ascent" and robustness to biased labels.

[Wu et al., 2024] proved a statistical task complexity bound, showing that with only a few linearly independent linear regression tasks, the trained model would perform close to Bayes optimal.

[Zhang et al., 2024a] proved convergence to a global minimum despite non-convexity. At the optimum, the model is nearly Bayes optimal estimator. [Zhang et al., 2024b] extended the convergence proof to a linear self-attention block followed by a feedforward layer, and that, the feedforward layer strictly improves the global optimum. At the optimum, the model implements one-step gradient descent with learnable initialization. [Zhang et al., 2023] explains the curve shape of Figure 2 of [Garg et al., 2022].

[Vladymyrov et al., 2024] proved that, on noisy linear regression ICL tasks with unknown noise variance, linear transformers learn a gradient descent algorithm with noise-aware step-size adjustments and rescaling based on noise levels.

[Chen et al., 2024] analyzed the training dynamics of multi-head attention models for noisy multilinear regression, demonstrating convergence to a local minimum and identifying distinct phases in the learning process. Under another setting, [Huang et al., 2023] identified up to four distinct phases.

[Li et al., 2023a] analyzed the problem from the PAC learning perspective, proving that the trained transformers are stable ICL learners, with provable generalization bounds to unseen tasks.

# 7 Empirical advances

Following up on the curriculum design work in [Garg et al., 2022], [Bhasin et al., 2024] experimented with different curriculum strategies across various function classes and statistical distributions, confirming

its high efficiency.

Several papers tried different architectures than the autoregressive decoder-only transformer. [Grazzi et al., 2024] tried Mamba; [Yang et al., 2024, Gao et al., 2024] tried looped transformers.

[Li et al., 2023b] trained models to perform difficult ICL tasks, where the function class is 6-layered MLP. This required sophisticated CoT prompting.

There were several ablation studies. [Wibisono and Wang, 2023] shuffled input-output pairings in the prompt, and found the softmax layer was vital for "unshuffling" the data internally, acting as a mixture of experts. [Cui et al., 2024] found multi-headed attention superior to single-headed attention, particularly with noisy labels, local examples, and correlated features. [Xing et al., 2024] ablated components of the transformer, and found it important to include two attention layers with a look-ahead mask, positional encoding for connecting inputs and outputs, multi-head attention and high embedding dimensions.

# 8 Others

Some papers do not fit into a story, yet engages with [Garg et al., 2022], so we put all of them here.

[Garg, 2023] is the first author's PhD thesis, and chapter 2 reprints [Garg et al., 2022].

[Bhattamishra et al., 2023] studied ICL with boolean function classes. They found that Transformers can nearly match the optimal learning algorithm for 'simpler' tasks, while their performance deteriorates on more 'complex' tasks. They also studied in-context curriculum, where simpler examples are presented earlier in the sequence. They also found that transformers can learn to implement two distinct algorithms to solve a single task, and can adaptively select the more sample-efficient algorithm depending on the sequence of in-context examples.

[Ahuja and Lopez-Paz, 2023] pointed out two different forms of OOD. If $x$ all appear in one orthant during training, but are unrestricted during testing, then we still have $Pr_{testing}(y|P) = Pr_{training}(y|P)$ for any prompt $P$ , since we still do OLS in both cases. If $x$ are noiseless during training, but are noisy during testing, then we do not have $Pr_{testing}(y|P) = Pr_{training}(y|P)$ for any prompt $P$, because we need to do ridge regression in testing.

[Sreenivasan, 2023] conducted theoretical and empirical studies on various ICL techniques, including variants of Chain of Thought (CoT) prompting, using [Garg et al., 2022]'s codebase and incorporating curriculum learning strategies.

# References


[Abernethy et al., 2023] Abernethy, J., Agarwal, A., Marinov, T. V., and Warmuth, M. K. (2023). A Mechanism for Sample-Efficient In-Context Learning for Sparse Retrieval Tasks.

[Ahn et al., 2024a] Ahn, K., Cheng, X., Daneshmand, H., and Sra, S. (2024a). Transformers learn to implement preconditioned gradient descent for in-context learning. *Advances in Neural Information Processing Systems*, 36.

[Ahn et al., 2024b] Ahn, K., Cheng, X., Song, M., Yun, C., Jadbabaie, A., and Sra, S. (2024b). Linear attention is (maybe) all you need (to understand transformer optimization).

[Ahuja and Lopez-Paz, 2023] Ahuja, K. and Lopez-Paz, D. (2023). A Closer Look at In-Context Learning under Distribution Shifts.

[Ahuja et al., 2023] Ahuja, K., Panwar, M., and Goyal, N. (2023). Transformers Can Learn To Solve Linear-Inverse Problems In-Context. In *NeurIPS 2023 Workshop on Deep Learning and Inverse Problems*.

[Akyürek et al., 2023] Akyürek, E., Schuurmans, D., Andreas, J., Ma, T., and Zhou, D. (2023). What learning algorithm is in-context learning? Investigations with linear models.

[Bai et al., 2023] Bai, Y., Chen, F., Wang, H., Xiong, C., and Mei, S. (2023). Transformers as Statisticians: Provable In-Context Learning with In-Context Algorithm Selection. *Advances in Neural Information Processing Systems*, 36:57125–57211.

[Bhasin et al., 2024] Bhasin, H., Ossowski, T., Zhong, Y., and Hu, J. (2024). How does Multi-Task Training Affect Transformer In-Context Capabilities? Investigations with Function Classes.

[Bhattamishra et al., 2023] Bhattamishra, S., Patel, A., Blunsom, P., and Kanade, V. (2023). Understanding In-Context Learning in Transformers and LLMs by Learning to Learn Discrete Functions.

[Brown et al., 2020] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language Models are Few-Shot Learners.

[Chan et al., 2022] Chan, S., Santoro, A., Lampinen, A., Wang, J., Singh, A., Richemond, P., McClelland, J., and Hill, F. (2022). Data distributional properties drive emergent in-context learning in transformers. *Advances in Neural Information Processing Systems*, 35:18878–18891.

[Chen et al., 2024] Chen, S., Sheen, H., Wang, T., and Yang, Z. (2024). Training Dynamics of Multi-Head Softmax Attention for In-Context Learning: Emergence, Convergence, and Optimality.

[Cheng et al., 2024] Cheng, X., Chen, Y., and Sra, S. (2024). Transformers Implement Functional Gradient Descent to Learn Non-Linear Functions In Context.

[Cui et al., 2024] Cui, Y., Ren, J., He, P., Tang, J., and Xing, Y. (2024). Superiority of Multi-Head Attention in In-Context Linear Regression.

[Ding et al., 2024] Ding, N., Levinboim, T., Wu, J., Goodman, S., and Soricut, R. (2024). CausalLM is not optimal for in-context learning.

[Fu et al., 2023] Fu, D., Chen, T.-Q., Jia, R., and Sharan, V. (2023). Transformers Learn Higher-Order Optimization Methods for In-Context Learning: A Study with Linear Models.

[Gao et al., 2024] Gao, Y., Zheng, C., Xie, E., Shi, H., Hu, T., Li, Y., Ng, M. K., Li, Z., and Liu, Z. (2024). On the Expressive Power of a Variant of the Looped Transformer.

[Garg, 2023] Garg, S. (2023). *Nature of Learning and Learning of Nature*. PhD thesis, Stanford University, United States – California.

[Garg et al., 2022] Garg, S., Tsipras, D., Liang, P. S., and Valiant, G. (2022). What can transformers learn in-context? a case study of simple function classes. *Advances in Neural Information Processing Systems*, 35:30583–30598.

[Grazzi et al., 2024] Grazzi, R., Siems, J., Schrodi, S., Brox, T., and Hutter, F. (2024). Is Mamba Capable of In-Context Learning?

[Guo et al., 2023] Guo, T., Hu, W., Mei, S., Wang, H., Xiong, C., Savarese, S., and Bai, Y. (2023). How Do Transformers Learn In-Context Beyond Simple Functions? A Case Study on Learning with Representations.

[Han et al., 2023] Han, C., Wang, Z., Zhao, H., and Ji, H. (2023). Explaining Emergent In-Context Learning as Kernel Regression.

[Huang et al., 2023] Huang, Y., Cheng, Y., and Liang, Y. (2023). In-Context Convergence of Transformers.

[Kim and Suzuki, 2024] Kim, J. and Suzuki, T. (2024). Transformers Learn Nonlinear Features In Context: Nonconvex Mean-field Dynamics on the Attention Landscape.

[Kojima et al., 2022] Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., and Iwasawa, Y. (2022). Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

[Li et al., 2023a] Li, Y., Ildiz, M. E., Papailiopoulos, D., and Oymak, S. (2023a). Transformers as Algorithms: Generalization and Stability in In-context Learning. In *Proceedings of the 40th International Conference on Machine Learning*, pages 19565–19594. PMLR.

[Li et al., 2023b] Li, Y., Sreenivasan, K., Giannou, A., Papailiopoulos, D., and Oymak, S. (2023b). Dissecting Chain-of-Thought: Compositionality through In-Context Filtering and Learning.

[Lin and Lee, 2024] Lin, Z. and Lee, K. (2024). Dual Operating Modes of In-Context Learning.

[Mahdavi et al., 2024] Mahdavi, S., Liao, R., and Thrampoulidis, C. (2024). Revisiting the Equivalence of In-Context Learning and Gradient Descent: The Impact of Data Distribution. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7410–7414, Seoul, Korea, Republic of. IEEE.

[Min et al., 2022a] Min, S., Lewis, M., Hajishirzi, H., and Zettlemoyer, L. (2022a). Noisy Channel Language Model Prompting for Few-Shot Text Classification.

[Min et al., 2022b] Min, S., Lyu, X., Holtzman, A., Artetxe, M., Lewis, M., Hajishirzi, H., and Zettlemoyer, L. (2022b). Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?

[Panwar et al., 2023] Panwar, M., Ahuja, K., and Goyal, N. (2023). Surprising Deviations from Bayesian View in In-Context Learning. In *I Can't Believe It's Not Better Workshop: Failure Modes in the Age of Foundation Models*.

[Panwar et al., 2024] Panwar, M., Ahuja, K., and Goyal, N. (2024). In-Context Learning through the Bayesian Prism.

[Pathak et al., 2023] Pathak, R., Sen, R., Kong, W., and Das, A. (2023). Transformers can optimally learn regression mixture models.

[Raventós et al., 2024] Raventós, A., Paul, M., Chen, F., and Ganguli, S. (2024). Pretraining task diversity and the emergence of non-Bayesian in-context learning for regression. *Advances in Neural Information Processing Systems*, 36.

[Reddy, 2023] Reddy, G. (2023). The mechanistic basis of data dependence and abrupt learning in an in-context classification task.

[Ren and Liu, 2023] Ren, R. and Liu, Y. (2023). In-context Learning with Transformer Is Really Equivalent to a Contrastive Learning Pattern.

[Shen et al., 2024] Shen, L., Mishra, A., and Khashabi, D. (2024). Revisiting the Hypothesis: Do pretrained Transformers Learn In-Context by Gradient Descent?

[Sreenivasan, 2023] Sreenivasan, K. (2023). *Towards Understanding the Challenges in Scaling Frontier Machine Learning Models*. The University of Wisconsin-Madison.

[Vladymyrov et al., 2024] Vladymyrov, M., von Oswald, J., Sandler, M., and Ge, R. (2024). Linear Transformers are Versatile In-Context Learners.

[Von Oswald et al., 2023] Von Oswald, J., Niklasson, E., Randazzo, E., Sacramento, J., Mordvintsev, A., Zhmoginov, A., and Vladymyrov, M. (2023). Transformers learn in-context by gradient descent. In *International Conference on Machine Learning*, pages 35151–35174. PMLR.

[Wei et al., 2022] Wei, J., Bosma, M., Zhao, V. Y., Guu, K., Yu, A. W., Lester, B., Du, N., Dai, A. M., and Le, Q. V. (2022). Finetuned Language Models Are Zero-Shot Learners.

[Wibisono and Wang, 2023] Wibisono, K. C. and Wang, Y. (2023). On the Role of Unstructured Training Data in Transformers' In-

Context Learning Capabilities. In *NeurIPS 2023 Workshop on Mathematics of Modern Machine Learning*.

[Wu et al., 2024] Wu, J., Zou, D., Chen, Z., Braverman, V., Gu, Q., and Bartlett, P. L. (2024). How Many Pretraining Tasks Are Needed for In-Context Learning of Linear Regression?

[Xie et al., 2022] Xie, S. M., Raghunathan, A., Liang, P., and Ma, T. (2022). An Explanation of In-context Learning as Implicit Bayesian Inference.

[Xing et al., 2024] Xing, Y., Lin, X., Suh, N., Song, Q., and Cheng, G. (2024). Benefits of Transformer: In-Context Learning in Linear Regression Tasks with Unstructured Data.

[Yadlowsky et al., 2023] Yadlowsky, S., Doshi, L., and Tripuraneni, N. (2023). Pretraining Data Mixtures Enable Narrow Model Selection Capabilities in Transformer Models.

[Yang et al., 2024] Yang, L., Lee, K., Nowak, R., and Papailiopoulos, D. (2024). Looped Transformers are Better at Learning Learning Algorithms.

[Zhang et al., 2024a] Zhang, R., Frei, S., and Bartlett, P. L. (2024a). Trained Transformers Learn Linear Models In-Context. *Journal of Machine Learning Research*, 25(49):1–55.

[Zhang et al., 2024b] Zhang, R., Wu, J., and Bartlett, P. L. (2024b). In-Context Learning of a Linear Transformer Block: Benefits of the MLP Component and One-Step GD Initialization.

[Zhang et al., 2023] Zhang, Y., Zhang, F., Yang, Z., and Wang, Z. (2023). What and How does In-Context Learning Learn? Bayesian Model Averaging, Parameterization, and Generalization.

# A Tips for using LLM for literature review

You might need to correct for two biases of most LLM nowadays: the sensibility bias and the agreeing bias.

The sensibility bias is when it tends to find a moral to every episode, an uplifting message to every episode, a worthwhile data point to every paper. Some papers are just garbage, and some episodes are nonsense. This is bad because it means it tends to embellish the review with empty nonsense. I correct for this by

> You are a literature reviewer for a machine learning journal. You are precise, informative, and neutral in tone. Don't be optimistic nor pessimistic. Simply inform, and don't attempt to tell a satisfying story. Don't apologize. Don't be polite, courteous, impolite, or verbose. Just perform the task.

The agreeing bias is when it tends to find something to agree to in every opinion. By default, if I ask whether a paper is relevant to Garg et al, it always finds a way to say yes. I correct for this by

> Be discriminating: If the work is only tangentially related, such that the work is not relevant for subsequent work that builds upon Garg et al, then reply "NO RELATION".
>
> You know that you are too eager to see relations where there are not, and if you realize that there really is no relation, you will stop the reply with "NO RELATION", even when you are halfway through a reply.