AI Link Collection Report - 2025-07-08

Generated on July 08, 2025 at 01:09 PM

Total Items: 5

Report Statistics	
Total Links Processed	5
Generation Date	2025-07-08
Generation Time	13:09:04

■ Link Collection Details

1. The Era of Exploration

■ https://yidingjiang.github.io/blog/post/exploration/

■ Full Article Content:

The Era of Exploration

Key points:

- Large language models are the unintended byproduct of about three decades worth of freely accessible human text online.

- Ilya Sutskever compared this reservoir of information to fossil fuel, abundant but ultimately finite.

- Some studies suggest that, at current token■consumption rates, frontier labs could exhaust the highest■quality English web text well before the decade ends.

- Even if those projections prove overly pessimistic, one fact is clear: today’s models consume data far faster than humans can produce it.

- Large language models are the unintended byproduct of about three decades worth of freely accessible human text online.

- Ilya Sutskever compared this reservoir of information to fossil fuel, abundant but ultimately finite.

Some studies suggest that, at current token■consumption rates, frontier labs could exhaust the highest■quality English web text well before the decade ends. Even if those projections prove overly pessimistic, one fact is clear: today’s models consume data far faster than humans can produce it. David Silver and Richard Sutton call this coming phase the “Era of Experience,” where meaningful progress will depend on data that learning agents generate for themselves. In this post, I want to build on their statement further: the bottleneck is not having just any experience but collecting the right kind of experience that benefits learning. The next wave of AI progress will hinge less on stacking parameters and more on exploration, the process of acquiring new and informative experience. David Silver and Richard Sutton call this coming phase the “ Era of Experience,” where meaningful progress will depend on data that learning agents generate for themselves. In this post, I want to build on their statement further: the bottleneck is not having just experience but collecting the kind of experience that benefits learning. The next wave of AI progress will hinge less on stacking parameters and more on exploration, the process of acquiring new and informative experience. To talk about experience collection, we must also ask what it costs to collect them. Scaling is, in the end, a question of resources – compute cycles, synthetic■data generation, data curation pipelines, human oversight, any expenditure that creates learning signal. For simplicity, I’ll fold all of these costs into a single bookkeeping unit I call flops. Strictly speaking, a flop is one floating■point operation, but the term has become a lingua franca for “how much effort did this system consume?” I’m co■opting it here not for its engineering precision but because it gives us a common abstract currency. My discussion depends only on relative spend, not on the particular mix of silicon, data, or human time. Treat flops as shorthand for “whatever scarce resource constrains scale.”

Key points:

- To talk about experience collection, we must also ask what it costs to collect them.

- Scaling is, in the end, a question of resources – compute cycles, synthetic■data generation, data curation pipelines, human oversight, any expenditure that creates learning signal.

- For simplicity, I’ll fold all of these costs into a single bookkeeping unit I call.

- Strictly speaking, a flop is one floating■point operation, but the term has become a lingua franca for “how much effort did this system consume?” I’m co■opting it here not for its engineering precision but because it gives us a common abstract currency.

- My discussion depends only on relative spend, not on the particular mix of silicon, data, or human time.

- Treat flops as shorthand for “whatever scarce resource constrains scale.” In the sections that follow, I’ll lay out a handful of observations and connect ideas that usually appear in different contexts.

Exploration is most often used in the context of reinforcement learning (RL), but I will also use “exploration” in a broader sense – much wider than its usual role in RL – because every data-driven system has to decide which experiences to collect before it can learn from them. This usage of exploration is also inspired by my friend Minqi’s excellent article “General intelligence requires rethinking exploration.”

Key points:

- In the sections that follow, I’ll lay out a handful of observations and connect ideas that usually appear in different contexts.

- Exploration is most often used in the context of reinforcement learning (RL), but I will also use “exploration” in a broader sense – much wider than its usual role in RL – because every data-driven system has to decide which experiences to collect before it can learn from them.

- This usage of exploration is also inspired by my friend ’s excellent article “ General intelligence requires rethinking exploration The rest of the post is organized as the following: first, how pre■training inadvertently solved a part of the exploration problem, second, why better exploration translates into better generalization, and finally, where we should spend the next hundred thousand GPU■years.

- The rest of the post is organized as the following: first, how pre■training inadvertently solved a part of the exploration problem, second, why better exploration translates into better generalization, and finally, where we should spend the next hundred thousand GPU■years.

- Pretraining is exploration.

Key points:

- Pretraining is exploration The standard LLM pipeline is to first pretrain a large model on next-token prediction with a large amount of text and then finetune the model with RL to achieve some desired objectives.

- Without large-scale pretraining, the RL step would struggle to make any progress.

- This contrast suggests that pretraining has accomplished something that is difficult for tabula rasa RL (i.e., from scratch).

- The standard LLM pipeline is to first pretrain a large model on next-token prediction with a large amount of text and then finetune the model with RL to achieve some desired objectives.

- Without large-scale pretraining, the RL step would struggle to make any progress.

- This contrast suggests that pretraining has accomplished something that is difficult for tabula rasa RL (i.e., from scratch).

A seemingly contradictory and widely observed trend in recent research is that smaller models can demonstrate significantly improved reasoning abilities once distilled using the chain-of-thought generated by larger, more capable models. Some interpret this as evidence that large scale is not a prerequisite for effective reasoning. In my opinion, this conclusion is misguided. The question we should ask is: if model capacity is not the bottleneck for reasoning, why do small models need to distill from a larger model at all? A seemingly contradictory and widely observed trend in recent research is that smaller models can demonstrate significantly improved reasoning abilities once distilled using the chain-of-thought generated by larger, more capable models. Some interpret this as evidence that large scale is not a prerequisite for effective reasoning. In my opinion, this conclusion is misguided. The question we should ask is: if model capacity is not the bottleneck for reasoning, why do small models need to distill from a larger model at all? A compelling explanation for both observations is that the immense cost of pretraining is effectively paying a massive, upfront “exploration tax.” By themselves, models with no pretraining or smaller pretrained models have a much harder time reliably exploring the solution space and discovering good solutions on their own1. The pretraining stage pays this tax by spending vast amounts of compute on diverse data to learn a rich sampling distribution under which the correct continuations are likely. Distillation, in turn, is a mechanism for letting a smaller model inherit that payment, bootstrapping its exploration capabilities from the massive investment made in the larger model. A compelling explanation for both observations is that the immense cost of pretraining is effectively paying a massive, upfront “ exploration tax.” By themselves, models with no pretraining or smaller pretrained models have a much harder time reliably exploring the solution space and discovering good solutions on their own. The pretraining stage pays this tax by spending vast amounts of compute on diverse data to learn a rich sampling distribution under which the correct continuations are likely. Distillation, in turn, is a mechanism for letting a smaller model inherit that payment, bootstrapping its exploration capabilities from the massive investment made in the larger model. Why is this pre-paid exploration so important? In its most general form, the RL loop looks something like this:

Key points:

- Why is this pre-paid exploration so important? In its most general form, the RL loop looks something like this: • Exploration.

- The agent generates some randomized exploration trajectories.

- The agent generates some randomized exploration trajectories.

- The good trajectories are up-weighted and the bad ones are down-weighted.

Reinforce. The good trajectories are up-weighted and the bad ones are down-weighted. For this learning loop to be effective, the agent must be capable of generating at least a minimal number of “good” trajectories during its exploration phase. This concept is sometimes called coverage in RL. In LLMs, this exploration is typically achieved through sampling from the model’s autoregressive output distribution. Under this exploration scheme, the correct solutions need to already be likely in the naive sampling distribution. If a lower■capacity model rarely stumbles on a valid solution by random sampling, it would have nothing useful to reinforce. For this learning loop to be effective, the agent must be capable of generating at least a minimal number of “good” trajectories during its exploration phase. This concept is sometimes called coverage in RL. In LLMs, this exploration is

typically achieved through sampling from the model’s autoregressive output distribution. Under this exploration scheme, the correct solutions need to already be likely in the naive sampling distribution. If a lower■capacity model rarely stumbles on a valid solution by random sampling, it would have nothing useful to reinforce. Exploration without any prior information is a very difficult process. Even in the simplest tabular RL setting, where every situation (a state) and every action can be listed out in a table, theory says learning needs a lot of trials. A well known lower-bound on the sample complexity on the number of episodes for tabular RL is $\Omega(\frac{SAH^2}{\epsilon^2})$ (Dann & Brunskill, 2015), where S is the size of the state space, A is the size of the action space, H is the horizon, and ϵ is the “distance” to the best possible solution. This means that the minimum number of episodes grows linearly with the number of state-action pairs, and quadratically with the horizon. For LLMs, the state space now includes every possible text prefix, and the action space is any next token, both of which are very large. Without any prior information, RL in this setting would be practically impossible. Exploration without any prior information is a very difficult process. Even in the simplest tabular RL setting, where every situation (a state) and every action can be listed out in a table, theory says learning needs a lot of trials. A well known lower-bound on the sample complexity on the number of episodes for tabular RL is $\Omega(\frac{SAH^2}{\epsilon^2})$ (Dann & Brunskill, 2015), where S is the size of the state space, A is the size of the action space, H is the horizon, and ϵ is the “distance” to the best possible solution. This means that the minimum number of episodes grows linearly with the number of state-action pairs, and quadratically with the horizon. For LLMs, the state space now includes every possible text prefix, and the action space is any next token, both of which are very large. Without any prior information, RL in this setting would be practically impossible. Up to now, the hard work of exploration has been largely done by pretraining and learning a better prior from which to sample trajectories. However, this also means that the types of trajectories the model can sample naively are heavily constrained by the prior. To progress further, we must figure out how to move beyond the prior. Up to now, the hard work of exploration has been largely done by pretraining and learning a better prior from which to sample trajectories. However, this also means that the types of trajectories the model can sample naively are heavily constrained by the prior. To progress further, we must figure out how to move beyond the prior.
Exploration helps generalization

Key points:

- Exploration helps generalization Historically, RL research has focused on solving a single environment at a time, like Atari or MuJoCo.

- This is equivalent to training and testing on the same datapoint.

- However, how well a given model does in the single environment does not say much about how well the model can handle truly novel situations.

- Machine learning is ultimately about generalization: for many hard problems, if we know them in advance, we can engineer a bespoke solution.

- What matters is succeeding on problems we haven’t seen or even anticipated.

- Historically, RL research has focused on solving a single environment at a time, like Atari or MuJoCo.

This is equivalent to training and testing on the same datapoint. However, how well a given model does in the single environment does not say much about how well the model can handle truly novel situations. Machine learning is ultimately about generalization: for many hard problems, if we know them in advance, we can engineer a bespoke solution. What matters is succeeding on problems we haven’t seen or even anticipated. The generalization performance of RL is critical for language models. During training, an LLM sees only a finite set of prompts, yet at deployment it must handle arbitrary user queries that could be very different from the ones seen during training. Notably, current LLMs excel at tasks with a verifiable reward (e.g., coding puzzles or formal proofs) because the correctness can be easily checked. The tougher question is to generalize these capabilities to fuzzier domains (e.g., generate a research report or write a novel) where feedback is sparse or ambiguous, and large-scale training and data collection are difficult. The generalization performance of RL is critical for language models. During training, an LLM sees only a finite set of prompts, yet at deployment it must handle arbitrary user queries that could be very different from the ones seen during training. Notably, current LLMs excel at tasks with a verifiable reward (e.g., coding puzzles or formal proofs) because the correctness can be easily checked. The tougher question is to generalize these capabilities to fuzzier domains (e.g., generate a research report or write a novel) where feedback is sparse or ambiguous, and large-scale training and data collection are difficult. What are some options we have for training a generalizable model? A recurring theme of deep learning is that data diversity drives robust generalization. Exploration directly controls the diversity of the data. In supervised learning, a labeled example reveals all its details in a single forward pass2 so the only way to increase data diversity is to collect more data. In RL, by contrast, each interaction only exposes a narrow slice of the environment. As a result, the agent must gather a sufficiently varied set of trajectories to build a representative picture. If the trajectories collected lack diversity (e.g., naive random sampling), the policy can overfit to that narrow slice and stumble even within the very same environment. What are some options we have for training a generalizable model? A recurring theme of deep learning is that data diversity drives robust generalization. Exploration directly controls the diversity of the data. In supervised learning, a labeled example reveals all its details in a single forward pass so the only way to increase data diversity is to collect more data. In RL, by contrast, each interaction only exposes a narrow slice of the environment. As a result, the agent must gather a sufficiently varied set of trajectories to build a representative picture. If the trajectories collected lack diversity (e.g., naive random sampling), the policy can overfit to that narrow slice and stumble even within the very same environment. This problem compounds when there are multiple environments. A popular RL generalization benchmark is Procgen, which is a collection of Atari-like games that have procedurally generated environments, so each game in principle contains “infinitely” many environments. The objective is to train on a fixed number of environments for a fixed number of steps and generalize to completely unseen environments3. This problem compounds when there are multiple environments. A popular RL generalization benchmark is Procgen, which is a collection of Atari-like games that have procedurally generated environments, so each game in principle contains “infinitely” many environments. The objective is to train on a fixed number of environments for a fixed number of steps and generalize to completely unseen environments Many existing approaches for this benchmark treat the problem as a representation learning problem and apply regularization techniques adapted from supervised learning (e.g., dropout or data augmentation). These help, but they overlook exploration, one of the most important structural components of RL. Since the agents collect their own data, they can improve generalization by changing exploration. In a previous work, my coauthors and I showed that pairing an existing RL algorithm with a stronger exploration strategy can double its generalization performance on Procgen without explicit regularization. In a more recent work, we found that better exploration also lets the model leverage more expressive model architectures and computational resources, and generalize better on Procgen as the result4. Many existing approaches for this benchmark treat the problem as a representation learning problem and apply regularization techniques adapted from supervised learning (e.g., dropout or data augmentation). These help, but they overlook exploration, one of the most important structural components of RL. Since the agents collect their own data, they can improve generalization by changing exploration. In a previous work, my coauthors and I showed that pairing an existing RL algorithm with a stronger exploration strategy can double its generalization performance on Procgen without explicit regularization. In a more recent work, we found that better exploration also lets the model leverage more expressive model architectures and computational resources, and generalize better on Procgen as the result While Procgen is certaintly not as difficult and complex as the problems LLMs are trained to solve today, the overall problem structure is essentially the same – the RL agent is trained on a finite set of problems and tested on new problems at test time without further training. The way we do exploration with LLMs today is fairly simple, typically limited to sampling from the model’s autoregressive distribution with tweaks to temperature or entropy bonus, so there is a large design space for potentially better exploration approaches. Admittedly, there have not been many successful examples in this direction. This could be because it is a very hard problem, it is not flop-efficient enough to be practical, or we just haven’t tried hard enough. However, if Procgen-style exploration gains do translate, we’re leaving efficiency – and perhaps entirely new capabilities – on the table. The next section discusses where we might look first. While Procgen is certaintly not as difficult and complex as the problems LLMs are trained to solve today, the overall problem structure is essentially the same – the RL agent is trained on a finite set of problems and tested on new problems at test time without further training. The way we do exploration with LLMs today is fairly simple, typically limited to sampling from the model’s

autoregressive distribution with tweaks to temperature or entropy bonus, so there is a large design space for potentially better exploration approaches. Admittedly, there have not been many successful examples in this direction. This could be because it is a very hard problem, it is not flop-efficient enough to be practical, or we just haven't tried hard enough. However, if Procgen-style exploration gains do translate, we're leaving efficiency – and perhaps entirely new capabilities – on the table. The next section discusses where we might look first. Two axes of scaling exploration

Two axes of scaling exploration Exploration, in the broad sense I'm using here, is deciding what data the learner will see. That decision happens on two distinct axes:

Key points:

- Exploration, in the broad sense I'm using here, is deciding what data the learner will see.
- That decision happens on two distinct axes: • World sampling – deciding where to learn.
- World here refers to a particular problem that needs to be solved.
- In supervised learning (or unsupervised pretraining), this axis covers data collection, synthetic generation and curation: gathering and filtering raw documents, images, or code, each of which corresponds to a "world".
- In RL, this corresponds to designing or generating environments, such as a single math puzzle or a coding problem.
- We can even arrange the worlds into curricula.

In both cases, world sampling is fundamentally about what "data points" the learner is allowed to see. This also decides the limit on all the information any agent can possibly learn. World sampling – deciding where to learn. World here refers to a particular problem that needs to be solved. In supervised learning (or unsupervised pretraining), this axis covers data collection, synthetic generation and curation: gathering and filtering raw documents, images, or code, each of which corresponds to a "world". In RL, this corresponds to designing or generating environments, such as a single math puzzle or a coding problem. We can even arrange the worlds into curricula. In both cases, world sampling is fundamentally about what "data points" the learner is allowed to see. This also decides the limit on all the information any agent can possibly learn. • Path sampling – deciding how to gather data inside a world. This step is unique to RL. Once a world is chosen, the agent still has to pick which trajectories to collect: random walks, curiosity■driven policies, tree search, tool-use, etc. Different path■sampling strategies can have different computation cost and produce very different training distributions even when the underlying world is identical. In short, path sampling is about what the learner "wants" to see. Path sampling – deciding how to gather data inside a world. This step is unique to RL. Once a world is chosen, the agent still has to pick which trajectories to collect: random walks, curiosity■driven policies, tree search, tool-use, etc. Different path■sampling strategies can have different computation cost and produce very different training distributions even when the underlying world is identical. In short, path sampling is about what the learner "wants" to see. In supervised learning or unsupervised pretraining, the second axis incurs a constant cost because a single forward (and backward) pass has access to all the information each data point contains (e.g., cross-entropy loss). Because there is no obvious way to "dig deeper" inside a single example (other than make the models larger), the exploration cost lives almost entirely on the first axis – world sampling. The flops can either go into acquire new worlds (e.g., new data points) or processing existing worlds (e.g., curation and synthetic data). In supervised learning or unsupervised pretraining, the second axis incurs a constant cost because a single forward (and backward) pass has access to all the information each data point contains (e.g., cross-entropy loss). Because there is no obvious way to "dig deeper" inside a single example (other than make the models larger), the exploration cost lives almost entirely on the first axis – world sampling. The flops can either go into acquire new worlds (e.g., new data points) or processing existing worlds (e.g., curation and synthetic data). In contrast, RL has much more flexibility in the second axis (in addition to the first axis). Because most random trajectories reveal little information about the ideal behavior, the information density (useful bits per flop) in RL is far lower than in supervised learning or pretraining. If we naïvely sample trajectory, we risk wasting flops on noise. So it’s imporant to be judicious about how we spend our flops5. There are also more options for spending flops to explore within each world. For example, we can either sample more trajectories from a single environment or we can spend more flops thinking about how to sample the next trajectory to discover high-value states and actions. In contrast, RL has much more flexibility in the second axis (in addition to the first axis). Because most random trajectories reveal little information about the ideal behavior, the information density (useful bits per flop) in RL is far lower than in supervised learning or pretraining.

If we naïvely sample trajectory, we risk wasting flops on noise. So it’s imporant to be judicious about how we spend our flops. There are also more options for spending flops to explore within each world. For example, we can either sample more trajectories from a single environment or we can spend more flops thinking about how to sample the next trajectory to discover high-value states and actions.

For most, if not all, machine learning problems, the high-level goal can be understood as maximizing information per flop. For that purpose, these two levers form a trade-off curve. If one spends too much resources on world sampling and not enough on path sampling, the agent may not extract any meaningful experience from the sampled worlds. Vice versa, if one spends too much resources on a small set of worlds, the agent could overfit to the training worlds and would not learn generalizable behavior that transfer across worlds. The ideal scenario happens somewhere in between where the resources are divided between sampling new worlds and running algorithms (i.e., better than random sampling) that can extract more information from a single world.

For most, if not all, machine learning problems, the high-level goal can be understood as maximizing information per flop. For that purpose, these two levers form a trade-off curve. If one spends too much resources on world sampling and not enough on path sampling, the agent may not extract any meaningful experience from the sampled worlds. Vice versa, if one spends too much resources on a small set of worlds, the agent could overfit to the training worlds and would not learn generalizable behavior that transfer across worlds. The ideal scenario happens somewhere in between where the resources are divided between sampling new worlds and running algorithms (i.e., better than random sampling) that can extract more information from a single world.

If you are familiar with scaling laws, what I just described sounds a lot like the Chinchilla scaling laws but the two axes correspond to the compute used for different types of sampling rather than parameters and data. At each performance level, one should to be able to trace out an isoperformance curve where the x-axis and y-axis are the compute put into interacting with any given environment and the compute given to the environments, whether it is for generating the environment or for running the environment (e.g., a generative verifier with CoT).

If you are familiar with scaling laws, what I just described sounds a lot like the Chinchilla scaling laws but the two axes correspond to the compute used for different types of sampling rather than parameters and data. At each performance level, one should to be able to trace out an isoperformance curve where the x-axis and y-axis are the compute put into interacting with any given environment and the compute given to the environments, whether it is for generating the environment or for running the environment (e.g., a generative verifier with CoT).

Of the two axes, path sampling is a relatively well-defined problem. A principled approach for doing exploration within an environment is to reduce the model’s uncertainty6. Many existing approaches for exploration have very strong sample complexity but they tend to be prohibitively expensive. Nonetheless, there is arguably a well-defined objective for path sampling and the main obstacle is to figure out a computationally efficient approximation. On the other hand, it is much less clear what the objective is for world sampling. One appealing idea is open-ended learning but even open-ended learning requires defining the universe of all environments (i.e., environment specs) or a subjective observer that judges whether an outcome is “interesting”.

Of the two axes, path sampling is a relatively well-defined problem. A principled approach for doing exploration within an environment is to reduce the model’s uncertainty. Many existing approaches for exploration have very strong sample complexity but they tend to be prohibitively expensive. Nonetheless, there is arguably a well-defined objective for path sampling and the main obstacle is to figure out a computationally efficient approximation. On the other hand, it is much less clear what the objective is for world sampling. One appealing idea is open-ended learning but even open-ended learning requires defining the universe of all environments (i.e., environment specs) or a subjective observer that judges whether an outcome is “interesting”.

What objective should world sampling optimize? The unfortunate reality (or fortunate, depending on your perspective) is that the space of environments is infinite, but our resources are finite. If we want to do something useful, then we must express some preference over the environments. I suspect that the problem of designing environments will eventually become similar to selecting pretraining data. It will be hard to say exactly why one environment will help another environment and we will need a lot of them. In other words, there may not be a single clean and nice objective for designing environment specs.

What objective should

world sampling optimize? The unfortunate reality (or fortunate, depending on your perspective) is that the space of environments is infinite, but our resources are finite. If we want to do something useful, then we must express some preference over the environments. I suspect that the problem of designing environments will eventually become similar to selecting pretraining data. It will be hard to say exactly why one environment will help another environment and we will need a lot of them. In other words, there may not be a single clean and nice objective for designing environment specs. The more likely scenario (probably already happening) is that everyone will start designing specs within their own expertise or domain of interest. When we have enough “human-approved” and “useful” specs, maybe we can try to learn some common principles and eventually automate the process by learning from them, much like how pretraining data is selected today. It would be inconvenient if we need the same amount of environments as pretraining data to achieve the same level of generality for decision making, but there are some preliminary evidence that this may not be the case. In a recent work, we found that a fairly small number of environments is enough to train an agent capable of general exploration and decision making in entirely out-of-distribution environments. Furthermore, using existing LLMs could also greatly accelerate the design process.

The more likely scenario (probably already happening) is that everyone will start designing specs within their own expertise or domain of interest. When we have enough “human-approved” and “useful” specs, maybe we can try to learn some common principles and eventually automate the process by learning from them, much like how pretraining data is selected today. It would be inconvenient if we need the same amount of environments as pretraining data to achieve the same level of generality for decision making, but there are some preliminary evidence that this may not be the case. In a recent work, we found that a fairly small number of environments is enough to train an agent capable of general exploration and decision making in entirely out-of-distribution environments. Furthermore, using existing LLMs could also greatly accelerate the design process.

Of course, these are all very high-level musings and how one exactly scales these two axes is much less obvious than scaling pretraining. However, if we were able to figure out a reliable way to introduce scale into world sampling, and a more intelligent way of path sampling, we should be able to see isoperformance curves that bends inwards towards the origin (maybe they won’t be as smooth). This style of scaling laws would teach us the best way to allocate computation resource between the environments and the agent.

Of course, these are all very high-level musings and how one exactly scales these two axes is much less obvious than scaling pretraining. However, if we were able to figure out a reliable way to introduce scale into world sampling, and a more intelligent way of path sampling, we should be able to see isoperformance curves that bends inwards towards the origin (maybe they won’t be as smooth). This style of scaling laws would teach us the best way to allocate computation resource between the environments and the agent.

Final thoughts

Key points:

- Final thoughts I could keep unfolding more tangents – better curiosity objectives, open-endedness, meta■exploration that learns how to explore – but I think it is more important to make the high-level point clear.
- I could keep unfolding more tangents – better curiosity objectives open-endedness meta■exploration that learns how to explore – but I think it is more important to make the high-level point clear.
- Existing paradigms of scaling have been incredibly effective but all paradigms will eventually saturate.
- The question is where to pour the next orders of magnitude of compute.
- I’ve argued that exploration – both world sampling and path sampling – offers a promising direction.
- We don’t yet know the right scaling laws, the right environment generators, or the right exploration objectives, but intuitively they should be possible.

The coming years will decide whether exploration can stretch our flops further on top of the existing paradigms. I think the bet is worth making. Existing paradigms of scaling have been incredibly effective but all paradigms will eventually saturate. The question is where to pour the next orders of magnitude of compute. I’ve argued that exploration – both world sampling and path sampling – offers a promising direction. We don’t yet know the right scaling laws, the right environment generators, or the right exploration objectives, but intuitively they should be possible. The coming years will decide whether exploration can stretch our flops further on top of the existing paradigms. I think the bet is worth making. Acknowledgement

Acknowledgement Many thanks to Allan Zhou, Sam Sokota, Minqi Jiang, Ellie Haber, Alex Robey, Swaminathan Gurumurthy, Kevin Li, Calvin Luo, Abitha Thankaraj, and Zico Kolter for their feedback and discussion on the draft. Many thanks to Allan Zhou, Sam Sokota, Minqi Jiang, Ellie Haber, Alex Robey, Swaminathan Gurumurthy, Kevin Li, Calvin Luo, Abitha Thankaraj, and Zico Kolter for their feedback and discussion on the draft. • A valid alternative possibility is that the RL optimization objective does not work well with smaller models, but this is likely not the case because before LLMs most successful applications of RL involved very small models. ■

A valid alternative possibility is that the RL optimization objective does not work well with smaller models, but this is likely not the case because before LLMs most successful applications of RL involved very small models. ■

A valid alternative possibility is that the RL optimization objective does not work well with smaller models, but this is likely not the case because before LLMs most successful applications of RL involved very small models. • This doesn’t mean the model can full exploit this information because the model can be limited in its computation power. It just means that if it wants to, that information is fully available. ■

This doesn’t mean the model can full exploit this information because the model can be limited in its computation power. It just means that if it wants to, that information is fully available. ■

Key points:

- This doesn’t mean the model can full exploit this information because the model can be limited in its computation power.
- It just means that if it wants to, that information is fully available.
- • For generalization to be tractable, we must assume that a “good enough” policy exists for all environments.
- This is analogous to assuming there is small or no label noise in supervised learning. ■

For generalization to be tractable, we must assume that a “good enough” policy exists for all environments.

- This is analogous to assuming there is small or no label noise in supervised learning. ■.

For generalization to be tractable, we must assume that a “good enough” policy exists for all environments. This is analogous to assuming there is small or no label noise in supervised learning. • At the time of writing, I believe this sets a new state-of-the-art performance on the “25M easy” benchmark of ProcGen. ■

At the time of writing, I believe this sets a new state-of-the-art performance on the “25M easy” benchmark of ProcGen. ■

At the time of writing, I believe this sets a new state-of-the-art performance on the “25M easy” benchmark of ProcGen. • Interestingly, for many problems such as Atari, random sampling works reasonably well. I think that says much more about the environments than the exploration method itself. ■

Interestingly, for many problems such as Atari, random sampling works reasonably well. I think that says much more about the environments than the exploration method itself. ■

Key points:

- Interestingly, for many problems such as Atari, random sampling works reasonably well.

- I think that says much more about the environments than the exploration method itself.

- • There is a wide family of RL algorithms under the names of posterior sampling or information-directed sampling that try to direct the exploration to reduce the model’s uncertainty, but they are generally too expensive to be done exactly at the scale of LLMs.

- Various approximations exist but they have not been widely used for LLMs to the best of my knowledge. ■

There is a wide family of RL algorithms under the names of posterior sampling or information-directed sampling that try to direct the exploration to reduce the model’s uncertainty, but they are generally too expensive to be done exactly at the scale of LLMs.

- Various approximations exist but they have not been widely used for LLMs to the best of my knowledge. ■.

There is a wide family of RL algorithms under the names of posterior sampling information-directed sampling that try to direct the exploration to reduce the model’s uncertainty, but they are generally too expensive to be done exactly at the scale of LLMs. Various approximations exist but they have not been widely used for LLMs to the best of my knowledge.

■ **Shared by:** Sai
■ **Shared on:** Jul 07, 2025 at 11:21 AM
■ **Domain:** yidingjiang.github.io
■ **Length:** 6,325 words
■ **Processed:** Jul 08, 2025 at 01:08 PM

---

2. apple/DiffuCoder-7B-cpGRPO · Hugging Face

■ `https://huggingface.co/apple/DiffuCoder-7B-cpGRPO`

■ Website Information:

Website Description:

The Hugging Face page for DiffuCoder-7B-cpGRPO presents a refined variant of the DiffuCoder model, specifically designed for code generation tasks. This model leverages reinforcement learning through Coupled-GRPO, enhancing its performance on various benchmarks and reducing biases during decoding. The page provides detailed technical specifications, training methodologies, and practical usage examples, making it a valuable resource for developers and researchers in the field of machine learning and natural language processing.

In addition to the model description, users can access related resources, including a research paper that outlines the underlying principles of DiffuCoder and a GitHub repository for implementation. This comprehensive information caters to data scientists, AI practitioners, and software engineers seeking to integrate advanced code generation capabilities into their projects.

■ **Shared by:** Sai
■ **Shared on:** Jul 02, 2025 at 09:58 AM
■ **Domain:** huggingface.co
■ **Length:** 574 words

- **Processed:** Jul 08, 2025 at 01:07 PM

3. GitHub - apple/ml-diffucoder: DiffuCoder: Understanding and Improving Masked Diffusion Models for Code Generation

- `https://github.com/apple/ml-diffucoder`

■ Website Information:

Website Description:

The GitHub repository for DiffuCoder provides an open-source implementation of masked diffusion models specifically designed for code generation. This project is a companion to the research paper titled 'DiffuCoder: Understanding and Improving Masked Diffusion Models for Code Generation,' offering users access to the underlying code, documentation, and resources necessary for understanding and utilizing these advanced models in their own applications. The repository includes various files such as scripts for inference, setup instructions, and guidelines for contributing to the project.

Targeted primarily at researchers, developers, and practitioners in the fields of machine learning and software engineering, the repository serves as a collaborative platform for enhancing the capabilities of code generation through innovative diffusion techniques. Key features include a structured file organization for easy navigation, ongoing updates regarding model training, and community contributions that foster an environment of shared knowledge and improvement.

- **Shared by:** Sai
- **Shared on:** Jul 02, 2025 at 09:58 AM
- **Domain:** github.com
- **Length:** 2,884 words
- **Processed:** Jul 08, 2025 at 01:09 PM

4. The Chatbot is Dead. Long Live the Orchestrator

- `https://community.coda.io/t/the-chatbot-is-dead-long-live-the-orchestrator/56357`

■ Full Article Content:

The Chatbot is Dead. Long Live the Orchestrator

preload-content:

The Chatbot is Dead. Long Live the Orchestrator

The Chatbot is Dead. Long Live the Orchestrator Bill_French July 1, 2025, 3:20pm While the world was distracted by talking dolls, Grammarly quietly built an AI that could act. They didn’t use a better model. They used a better weapon: Coda. While the world was distracted by talking dolls, Grammarly quietly built an AI that could act. They didn’t use a better model. They used a better weapon: Coda. Your Prompts Are a Prayer to an Amnesiac God

Key points:

- Your Prompts Are a Prayer to an Amnesiac God Let’s be brutally honest.

- The entire AI industry is captivated by a lie.

- We’ve anointed “prompt engineering” as a mystical art, a high priesthood for coaxing wisdom from silicon gods.

- We are meticulously polishing the conversational skills of a machine with terminal amnesia.

- Let’s be brutally honest.

The entire AI industry is captivated by a lie. We’ve anointed “prompt engineering” as a mystical art, a high priesthood for coaxing wisdom from silicon gods. It’s a sham. We are meticulously polishing the conversational skills of a machine with terminal amnesia. We were promised an omniscient partner, an AI co-pilot. What we got was a brilliant intern with no long-term memory, an entity we must re-brief from scratch every five minutes. This isn’t productivity. It’s digital babysitting for a machine that can recite Shakespeare but struggles to remember your name or perform precise calculations. We were promised an omniscient partner, an AI co-pilot. What we got was a brilliant intern with no long-term memory, an entity we must re-brief from scratch every five minutes. This isn’t productivity. It’s digital babysitting for a machine that can recite Shakespeare but struggles to remember your name or perform precise calculations. Today’s large language models are amnesiacs by design. We celebrate their ballooning context windows—a million, two million tokens—as if it were memory. It’s not. It’s a bigger notepad. A volatile transcript that dissolves into the ether the moment you close the tab. This architecture condemns us to a state of digital shrapnel: your project plan lives in Slack, your research is scattered across browser tabs, your decisions are buried in email, and your draft is in a doc. We ask our AI to be intelligent, but we force it to operate blindfolded, guessing the shape of our work by touching one disconnected piece at a time. Today’s large language models are amnesiacs by design. We celebrate their ballooning context windows—a million, two million tokens—as if it were memory. It’s not. It’s a bigger notepad. A volatile transcript that dissolves into the ether the moment you close the tab. This architecture condemns us to a state of digital shrapnel: your project plan lives in Slack, your research is scattered across browser tabs, your decisions are buried in email, and your draft is in a doc. We ask our AI to be intelligent, but we force it to operate blindfolded, guessing the shape of our work by touching one disconnected piece at a time. This makes you the AI’s external hard drive. You are the connective tissue. You perform the soul-crushing labor of copying, pasting, and re-explaining context, bridging the gap with every single query. This isn’t a feature. It’s a catastrophic, unforgivable design flaw. This makes you the AI’s external hard drive. You are the connective tissue. You perform the soul-crushing labor of copying, pasting, and re-explaining context, bridging the gap with every single query. This isn’t a feature. It’s a catastrophic, unforgivable design flaw. Stop Describing. Start Commanding. Stop Describing. Start Commanding. For a moment, I thought the answer was contexting—the architectural discipline of curating a rich data environment for an AI agent. It was the right instinct, but the wrong verb. It was a step away from the vacant art of prompting, but it was still just talking at the machine. For a moment, I thought the answer was contexting—the architectural discipline of curating a rich data environment for an AI agent. It was the right instinct, but the wrong verb. It was a step away from the vacant art of prompting, but it was still just talking at the machine. You cannot build a skyscraper by describing it to a pile of bricks, no matter how eloquently. You need an architectural plan, a crane, and a crew. Prompting is the description. Coda is the crane and the blueprint. It transforms your context from a static pile of information into a dynamic set of executable instructions. You cannot build a skyscraper by describing it to a pile of bricks, no matter how eloquently. You need an architectural plan, a crane, and a crew. Prompting is the description. Coda is the crane and the blueprint. It transforms your context from a static pile of information into a dynamic set of executable instructions. You cannot build a skyscraper by describing it to a pile of bricks, no matter how eloquently. You need an architectural plan, a crane, and a crew. Prompting is the description. Coda is the crane and the blueprint. It transforms your context from a static pile of information into a dynamic set of executable instructions. The true frontier isn't what an AI knows. It's what it can do. This demands a new class of software: an Agentic Orchestration Framework. A system that doesn't just talk, but commands, coordinates, and executes. It directs multiple specialized agents—AI, automation, and human—across complex, multi-step workflows with unwavering precision. The true frontier isn't what an AI knows. It's what it can do. This demands a new class of software: an Agentic Orchestration Framework. A system that doesn't just talk, but commands, coordinates, and executes. It directs multiple specialized agents—AI, automation, and human—across complex, multi-step workflows with unwavering precision. The archetype for this framework has been hiding in plain sight: Coda. The archetype for this framework has been hiding in plain sight: Forget the "all-in-one doc" marketing. That was the Trojan horse. Coda is a workflow engine for manufacturing bespoke, agentic software without writing a line of code. Its architecture is the very blueprint for orchestration:

Key points:

- Forget the "all-in-one doc" marketing.
- That was the Trojan horse.
- Coda is a workflow engine for manufacturing bespoke, agentic software without writing a line of code.
- Its architecture is the very blueprint for orchestration: • Docs as the Command Center: The unified surface where human intent and AI execution converge.
- Docs as the Command Center: The unified surface where human intent and AI execution converge.
- • Structured Tables as the Memory: A shared, persistent "brain" that provides unwavering context, rendering the amnesiac chat log obsolete.

Structured Tables as the Memory: A shared, persistent "brain" that provides unwavering context, rendering the amnesiac chat log obsolete. • Packs as the Limbs: The API-driven connectors that give agents power over the real world—to manipulate Google Calendar, create Jira tickets, or rewrite Salesforce records. Packs as the Limbs: The API-driven connectors that give agents power over the real world—to manipulate Google Calendar, create Jira tickets, or rewrite Salesforce records. • Automations as the Nervous System: The rule-based engine that triggers actions and executes entire workflows with inhuman speed and reliability. Automations as the Nervous System: The rule-based engine that triggers actions and executes entire workflows with inhuman speed and reliability. CleanShot 2025-07-01 at 09.14.43@2x1274×842 43.6 KB

CleanShot 2025-07-01 at 09.14.43@2x 1274×842 43.6 KB This was never a document app. It's a factory for building intelligent actors. This was never a document app. It's a factory for building intelligent actors. The Grammarly Gambit: An Empire in Two Moves

Key points:

- The Grammarly Gambit: An Empire in Two Moves While the market obsessed over chatbot demos and press releases, Grammarly executed a two-step strategic coup to build the world's first true agentic productivity platform.
- Anyone who saw these as unrelated acquisitions wasn't just missing the story; they were illiterate in the language of power.
- While the market obsessed over chatbot demos and press releases, Grammarly executed a two-step strategic coup to build the world's first true agentic productivity platform.

- Anyone who saw these as unrelated acquisitions wasn’t just missing the story; they were illiterate in the language of power.

- Move 1 (Acquire the Brain): Seize the Orchestration Framework

In late 2024, Grammarly acquired Coda.

- They didn’t buy a popular doc app.

They bought the operating system for their future AI agents. This was the foundational act of war. As undeniable proof, Coda founder Shishir Mehrotra wasn’t just given a board seat; he was installed as Grammarly’s new CEO. He isn’t running a company; he is performing a hostile takeover of its DNA, injecting Coda’s agentic framework into a platform with 40 million daily active users. Move 1 (Acquire the Brain): Seize the Orchestration Framework

In late 2024, Grammarly acquired Coda. They didn’t buy a popular doc app. They bought the operating system for their future AI agents. This was the foundational act of war. As undeniable proof, Coda founder Shishir Mehrotra wasn’t just given a board seat; he was installed as Grammarly’s new CEO. He isn’t running a company; he is performing a hostile takeover of its DNA, injecting Coda’s agentic framework into a platform with 40 million daily active users. Move 2 (Conquer the Battlefield): Seize the Critical Interface

Months later, Grammarly acquired Superhuman. This was not about adding a slick email client. Superhuman is what Mehrotra calls the “perfect staging ground for orchestrating multiple AI agents simultaneously.” Email is the chaotic nexus where work, communication, and tasks collide. Grammarly didn’t buy Superhuman for its pathetic summarization features; they bought the most valuable turf in professional life to serve as the GUI for their Coda-powered agentic backend. Imagine it: A sales agent, a support agent, and a scheduling agent collaborating within a single email draft, orchestrated by the Coda engine, pulling live context from connected Packs, executing tasks across a dozen SaaS apps. That is the gambit. Move 2 (Conquer the Battlefield): Seize the Critical Interface

Months later, Grammarly acquired Superhuman. This was not about adding a slick email client. Superhuman is what Mehrotra calls the “perfect staging ground for orchestrating multiple AI agents simultaneously.” Email is the chaotic nexus where work, communication, and tasks collide. Grammarly didn’t buy Superhuman for its pathetic summarization features; they bought the most valuable turf in professional life to serve as the GUI for their Coda-powered agentic backend. Imagine it: A sales agent, a support agent, and a scheduling agent collaborating within a single email draft, orchestrated by the Coda engine, pulling live context from connected Packs, executing tasks across a dozen SaaS apps. That is the gambit. CleanShot 2025-07-01 at 09.17.51@2x1434×992 78 KB

CleanShot 2025-07-01 at 09.17.51@2x 1434×992 78 KB

Your AI Stack is a Museum Piece

Key points:

- Your AI Stack is a Museum Piece The chasm between the dying paradigm and the emerging one is not an increment; it is a cliff.

- One is a toy, the other is a weapon.

- One talks, the other acts.

- The stack Grammarly is building does not compete with the old one.

- It renders it irrelevant.

- The chasm between the dying paradigm and the emerging one is not an increment; it is a cliff.

One is a toy, the other is a weapon. One talks, the other acts. The stack Grammarly is building does not compete with the old one. It renders it irrelevant. The Moat Isn’t the Model; It’s the Machine

Key points:

- The Moat Isn’t the Model; It’s the Machine The winners of the AI war will not be the companies with the largest language model.
- That is a commodity race to the bottom.
- They will be the ones who own the orchestration framework that makes those models act.
- Building a better chatbot today is like perfecting the horse-drawn carriage in the age of the automobile.
- The real innovation was the assembly line and the highway system.
- The only defensible moat is the machine: the framework that enables action, the integrations that give it reach, the structured data that serves as its memory, and the user workflows that become its territory.

The winners of the AI war will not be the companies with the largest language model. That is a commodity race to the bottom. They will be the ones who own the orchestration framework that makes those models act. Building a better chatbot today is like perfecting the horse-drawn carriage in the age of the automobile. The real innovation was the assembly line and the highway system. The only defensible moat is the machine: the framework that enables action, the integrations that give it reach, the structured data that serves as its memory, and the user workflows that become its territory. Stop asking if your AI is smart. Start demanding that it act. Stop celebrating prompts. Start building engines of execution. Stop asking if your AI is smart. Start demanding that it act. Stop celebrating prompts. Start building engines of execution. The future of work isn’t a conversation. It’s a command. The companies that understand this are building empires. The rest are polishing tombstones. The future of work isn’t a conversation. It’s a command. The companies that understand this are building empires. The rest are polishing tombstones. ps. A warm welcome to Rahul and the SuperHuman team.

ps. A warm welcome to Rahul and the SuperHuman team. 15 Likes Exciting News: Grammarly to acquire Superhuman Christiaan_Huizer July 1, 2025, 4:13pm thx for sharing your ideas. interesting & promising.

what do you believe would be a next tool to acquire @Bill_French?

thx for sharing your ideas. interesting & promising.

what do you believe would be a next tool to acquire @Bill_French 2 Likes Melanie_Teh July 1, 2025, 4:26pm my $$ is on reclaim or motion… something along the lines of calendar/task management

my $$ is on reclaim or motion… something along the lines of calendar/task management edit: oh, cannot be reclaim as they got bought by dropbox

edit: oh, cannot be reclaim as they got bought by dropbox 1 Like Bill_French July 1, 2025, 4:30pm I’m uncertain, but I have to believe someone at Grammarly (who is apparently really dialed in to the way I think) is watching Pieces, Flowith, and Dia. I’m uncertain, but I have to believe someone at Grammarly (who is apparently really dialed in to the way I think) is watching Pieces Flowith 5 Likes Agile_Dynamics July 2, 2025, 11:21pm Shishir keeps talking about ‘surfaces’. Email is the most frequent surface for Grammarly usage (=>Superhuman)

Documents and sheets are another major surface (=>Coda)

Key points:

- Shishir keeps talking about ‘surfaces’.
- Email is the most frequent surface for Grammarly usage (=>Superhuman)

Documents and sheets are another major surface (=>Coda) So whats the next-biggest surface where we spend our time? So whats the next-biggest surface where we spend our time? How about messaging and chats and forums? How about messaging and chats and forums? So maybe Grammarly has it’s eye on one of those? Not the major players maybe, but a startup with oodles of innovation and AI expertise? So maybe Grammarly has it’s eye on one of those? Not the major players maybe, but a startup with oodles of innovation and AI expertise? Another key criteria for these acquisitions is a shared vision.

- It was highlighted during the Coda deal.
- It’s highlighted again by Shishir and Rahul during the Superhuman deal.
- Another key criteria for these acquisitions is a shared vision.
- It was highlighted during the Coda deal.

It’s highlighted again by Shishir and Rahul during the Superhuman deal. So the next target will have a messaging surface, an AI focus, and a clear vision that matches that of Shashir, Rahul, and their teams. So the next target will have a messaging surface, an AI focus, and a clear vision that matches that of Shashir, Rahul, and their teams. I dont know the marketplace well enough to identify candidates, but perhaps someone reading this does? I dont know the marketplace well enough to identify candidates, but perhaps someone reading this does? Just a thought. Just a thought. 2 Likes Nina_Ledid July 3, 2025, 9:03am As always, I’ve greatly enjoyed reading your perspective, @Bill_French, thanks for sharing! As always, I’ve greatly enjoyed reading your perspective, @Bill_French, thanks for sharing! I do remember from past conversations your concerns about Coda Pack’s limited support for common integration patterns (eg handling incoming webhooks, running persistent background services, or responding to external events)

I do remember from past conversations your concerns about Coda Pack’s limited support for common integration patterns (eg handling incoming webhooks, running persistent background services, or responding to external events) In your post above, you envision Coda as the blueprint for a new kind of orchestration framework. In your post above, you envision Coda as the blueprint for a new kind of orchestration framework. Do you think what once seemed like a limitation (Packs’ closed and controlled architecture) is now less of a drawback? Do you think what once seemed like a limitation (Packs’ closed and controlled architecture) is now less of a drawback? Or do you even see it potentially becoming a competitive advantage as the focus shifts from open interoperability to orchestrated execution? Or do you even see it potentially becoming a competitive advantage as the focus shifts from open interoperability to orchestrated execution? Thanks,

Nina

Thanks,

- Likes Stefan_Stoyanov July 5, 2025, 8:17pm The reason I’m not yet bullish on Coda’s future is due to one key problem that LLMs still don’t solve particularly well: data collection. Specifically, I’ve been thinking a lot about the following limitations of Coda, as I understand its vision going forward:

Key points:

- The reason I’m not yet bullish on Coda’s future is due to one key problem that LLMs still don’t solve particularly well: data collection.

- Specifically, I’ve been thinking a lot about the following limitations of Coda, as I understand its vision going forward: • Collecting data by typing into Coda’s quadrangle notes (cells) is too slow and impractical.

- Instead, data should come from voice/video meetings and recordings, mobile/desktop screen activity, linked/shared content sources, mobile phone conversations, and visual/audible/touch perception input devices.

- Pushing data from hardware devices in real time becomes even more critical than pulling it from structured database, because if data isn’t captured quickly enough, the use of the database can become unreliable and irrelevant even within a minute.

- Collecting data by typing into Coda’s quadrangle notes (cells) is too slow and impractical.

- Instead, data should come from voice/video meetings and recordings, mobile/desktop screen activity, linked/shared content sources, mobile phone conversations, and visual/audible/touch perception input devices.

Pushing data from hardware devices in real time becomes even more critical than pulling it from structured database, because if data isn’t captured quickly enough, the use of the database can become unreliable and irrelevant even within a minute. • Email is no longer a practical communication channel - it belongs to the past. As someone who has long supported email, it’s hard for me to admit this, but emails are now like the written contracts of a previous era. Today, very little needs to be formally written when people prefer adaptable, relevant content in audio/video formats. Email is no longer a practical communication channel - it belongs to the past. As someone who has long supported email, it’s hard for me to admit this, but emails are now like the written contracts of a previous era. Today, very little needs to be formally written when people prefer adaptable, relevant content in audio/video formats. • The joint venture company’s language support is essentially limited to English at the moment. In an era of LLMs—where tokenization of language is foundational to progress—this is concerning. Many LLMs still perform poorly in languages other than English. What does it mean to build a great app for the UK but a terrible one for France, simply because UX in the UK relies on quick chat-based communication, while in France it might depend on elaborate instructions—or worse, traditional filters and search? The joint venture company’s language support is essentially limited to English at the moment. In an era of LLMs—where tokenization of language is foundational to progress—this is concerning. Many LLMs still perform poorly in languages other than English. What does it mean to build a great app for the UK but a terrible one for France, simply because UX in the UK relies on quick chat-based communication, while in France it might depend on elaborate instructions—or worse, traditional filters and search? My understanding of LLMs and Coda’s vision is limited, so I’d really appreciate the community’s thoughts on these points. My understanding of LLMs and Coda’s vision is limited, so I’d really appreciate the community’s thoughts on these points. 1 Like Christiaan_Huizer July 6, 2025, 11:53am HI @Stefan_Stoyanov

Key points:

- @Stefan_Stoyanov Thanks for sharing your thoughts, those are valid concerns.

- Just so you know, I don’t have access to any internal “cookbook” or strategic plans.

- These are simply my best guesses and interpretations based on what’s publicly available and general market trends.

- Thanks for sharing your thoughts, those are valid concerns.

- Just so you know, I don’t have access to any internal “cookbook” or strategic plans.

- These are simply my best guesses and interpretations based on what’s publicly available and general market trends.

Regarding your first question about data collection, there are already many AI-powered tools out there that are really good at capturing voice from meetings, whether you’re in the room or online. These tools can pull out and organize key information, remarks, and action items from conversations. When we think about the “surfaces” and “orchestration” Grammarly is building, meeting platforms are a big communication area. It makes sense that Grammarly might look to acquire a company in this field. They’d likely be looking for strong AI features, existing connections to other tools, potential for their “agents” to expand, and solid support for many languages. Companies like Sembly AI, Fireflies.ai, Otter.ai, or MeetGeek come to mind. Regarding your first question about data collection, there are already many AI-powered tools out there that are really good at capturing voice from meetings, whether you’re in the room or online. These tools can pull out and organize key information, remarks, and action items from conversations. When we think about the “surfaces” and “orchestration” Grammarly is building, meeting platforms are a big communication area. It makes sense that Grammarly might look to acquire a company in this field. They’d likely be looking for strong AI features, existing connections to other tools, potential for their “agents” to expand, and solid support for many languages. Companies like Sembly AI Fireflies.ai Otter.ai MeetGeek come to mind. As for your comment on email, that’s an an interesting thought. While other ways of communicating have certainly grown, email is still a core, if not the core, communication hub for many businesses. That’s actually why Grammarly bought Superhuman. It’s not just about email itself; it’s also about the related areas Superhuman is working on, like chats, calendars, and tasks (direct overlap with Coda). The whole idea is that all these interactions can be managed and brought together within Coda, building out an AI-powered productivity system that covers all the main places where work happens. As for your comment on email, that’s an an interesting thought. While other ways of communicating have certainly grown, email is still a core, if not core, communication hub for many businesses. That’s actually why Grammarly bought Superhuman. It’s not just about email itself; it’s also about the related areas Superhuman is working on, like chats, calendars, and tasks (direct overlap with Coda). The whole idea is that all these interactions can be managed and brought together within Coda, building out an AI-powered productivity system that covers all the main places where work happens. Finally, on your point about language support for LLMs, Superhuman already includes AI-driven translation right within its email platform. This feature allows for easy translation of messages, which is super important for international teams and global operations. This capability directly tackles language barriers in a crucial communication space, showing that Grammarly is focused on making its AI tools truly effective for a wide range of users worldwide. Finally, on your point about language support for LLMs, Superhuman already includes AI-driven translation right within its email platform. This feature allows for easy translation of messages, which is super important for international teams and global operations. This capability directly tackles language barriers in a crucial communication space, showing that Grammarly is focused on making its AI tools truly effective for a wide range of users worldwide. welcoming further feedback.

welcoming further feedback. Chers, christiaan

Key points:

- Chers, christiaan 3 Likes Bill_French July 7, 2025, 6:05pm Indeed.
- It’s why I use Pieces.
- It captures everything, and I can use it as a long-term memory or a funnel into Coda and other tools.
- Capturing contexts is probably where Grammarly is heading.
- It’s why I use Pieces.

It captures everything, and I can use it as a long-term memory or a funnel into Coda and other tools. Capturing contexts is probably where Grammarly is heading. For approximately 2.4 billion people, it remains practical in most cases. It’s not going anywhere, at least not in your lifetime. If you see ‘email’ when you see Superhuman, you need to recalibrate. The AI engine in this tool was perfected in one of the harshest environments, but it is applicable in many operational domains. They didn’t buy Superhuman to support email better. For approximately 2.4 billion people, it remains practical in most cases. It’s not going anywhere, at least not in your lifetime. If you see ‘email’ when you see Superhuman, you need to recalibrate. The AI engine in this tool was perfected in one of the harshest environments, but it is applicable in many operational domains. They didn’t buy Superhuman to support email better. This is not Grammarly’s problem to solve. I think some of the higher-level executives and engineers have seen solutions that will address the multilingual challenges. This is not Grammarly’s problem to solve. I think some of the higher-level executives and engineers have seen solutions that will address the multilingual challenges. 1 Like Bill_French July 7, 2025, 6:11pm Nina, thank you for the thoughtful words and for engaging so directly with these architectural questions. Nina, thank you for the thoughtful words and for engaging so directly with these architectural questions. To your point: Packs, as currently conceived, remain constrained by several limitations I’ve outlined over time—chief among them, their closed nature and limited support for dynamic integration patterns. These constraints have historically hampered Coda’s ability to serve as a true orchestration hub, especially when it comes to handling real-world, real-time data flows. To your point: Packs, as currently conceived, remain constrained by several limitations I’ve outlined over time—chief among them, their closed nature and limited support for dynamic integration patterns. These constraints have historically hampered Coda’s ability to serve as a true orchestration hub, especially when it comes to handling real-world, real-time data flows. However, with Grammarly’s expanding reach across critical “surfaces”—and their likely trajectory toward integrating voice, vision, and other modalities—there’s a strong possibility that Packs will evolve. In this emerging context, Packs could become the ideal conduit for retrieving context-rich, multimodal data from the broader Grammarly ecosystem. Rather than being a bottleneck, Packs may soon serve as the connective tissue between Coda’s orchestration framework and the vast, real-time data streams generated by Grammarly’s “mothership.”

Key points:

- However, with Grammarly’s expanding reach across critical “surfaces”—and their likely trajectory toward integrating voice, vision, and other modalities—there’s a strong possibility that Packs will evolve.

- In this emerging context, Packs could become the ideal conduit for retrieving context-rich, multimodal data from the broader Grammarly ecosystem.

- Rather than being a bottleneck, Packs may soon serve as the connective tissue between Coda’s orchestration framework and the vast, real-time data streams generated by Grammarly’s “mothership.” The challenge, and the opportunity, is for Packs to move beyond their current boundaries—adapting to new integration standards and supporting the kind of pervasive, intelligent data exchange that true agentic orchestration demands.

- The challenge, and the opportunity, is for Packs to move beyond their current boundaries—adapting to new integration standards and supporting the kind of pervasive, intelligent data exchange that true agentic orchestration demands.

- 1 Like Bill_French July 7, 2025, 6:19pm Perhaps relevant - I can go from email to LLM to Coda without ever copying/pasting, ChatGPT, or any intermediate tool.

- Dia carries the weight of shortening the line from information to curation.

Perhaps relevant - I can go from email to LLM to Coda without ever copying/pasting, ChatGPT, or any intermediate tool. Dia carries the weight of shortening the line from information to curation. CleanShot 2025-07-07 at 12.15.20@2x1920×1588 146 KB

CleanShot 2025-07-07 at 12.15.20@2x 1920×1588 146 KB 2 Likes

Related topics

Related topics Replies Activity Coda AI and the User Experience October 8, 2023 Aurora - Easily Craft Prompts That SELL AI at Work Challenge August 7, 2023 Beyond The Prompt AI at Work Challenge August 16, 2023 Coda AI - It's not me its you October 6, 2023 Coda/Grammarly Self-Help Integration: Forget About It April 25, 2025:preload-content

- ■ **Shared by:** Amit Shah
- ■ **Shared on:** Jul 01, 2025 at 09:00 PM
- ■ **Domain:** community.coda.io
- ■ **Length:** 4,831 words
- ■ **Processed:** Jul 08, 2025 at 01:08 PM

5. Welcome to The Era of Evals

- ■ `https://www.linkedin.com/pulse/welcome-era-evals-brendan-foody-ezykc/`

■ Full Article Content:

Welcome to The Era of Evals

Key points:

- "urn:li:member:0" "true" "true" Reinforcement Learning (RL) is driving the most exciting advancements in AI.
- RL is becoming so effective that models will be able to saturate any evaluation.
- This means that the primary barrier to applying agents to the entire economy is building evals for everything.
- However, AI labs are facing a dire shortage of relevant evaluations.
- Academic evaluations that labs goal on don’t reflect what consumers and enterprises demand in the economy.
- Reinforcement Learning (RL) is driving the most exciting advancements in AI.

RL is becoming so effective that models will be able to saturate any evaluation. This means that the primary barrier to applying agents to the entire economy is building evals for everything. However, AI labs are facing a dire shortage of relevant evaluations. Academic evaluations that labs goal on don’t reflect what consumers and enterprises demand in the economy. Evals are the new PRD. Progress in accelerating knowledge work will converge on building environments and evaluations that map real workspaces and deliverables. This new RL-centric paradigm of human data is vastly more data efficient than pretraining, SFT, or RLHF. Most knowledge work includes recurring workflows as variable costs, but creating an environment or evaluation can transform that into a one-time fixed cost. Evals are the new PRD. Progress in accelerating knowledge work will converge on building environments and evaluations that map real workspaces and deliverables. This new RL-centric paradigm of human data is vastly more data efficient than pretraining, SFT, or RLHF. Most knowledge work includes recurring workflows as variable costs, but creating an environment or evaluation can transform that into a one-time fixed cost. Training on Verifiable Rewards

Key points:

- Training on Verifiable Rewards RL environments allow for rewarding outcomes and intermediate steps in an evaluation.

- Models take many attempts at a problem, using test-time compute to "think" before it answers.

- Human created autograders reward the attempts which were "good".

- Reinforcing on those "good" trajectories upweights the chains of thought that were used to get to the answer.

- This teaches models to think correctly about different types of problems as researchers iteratively hill climb evals.

- These environments can be thought of as existing on a spectrum of rigidity between two categories:.

Key points:

- RL environments allow for rewarding outcomes and intermediate steps in an evaluation.

- Models take many attempts at a problem, using test-time compute to "think" before it answers.

- Human created autograders reward the attempts which were "good".

- Reinforcing on those "good" trajectories upweights the chains of thought that were used to get to the answer.

- This teaches models to think correctly about different types of problems as researchers iteratively hill climb evals.

- These environments can be thought of as existing on a spectrum of rigidity between two categories: Objective domains: Games, like pac-man, chess, and Go, have clear states spaces, action spaces, and desired outcomes.

Math, code, and even some tasks in biology, can often be formulated with near game-like verifiability. This is where RL has achieved early massive success already, notably, AlphaProof, AlphaFold, and DeepSeek R1 and the many code generation models on the market today. Subjective domains: It’s more difficult to measure accuracy in many real world tasks such as generating investment memos, making legal briefs, providing therapy. This makes it difficult to verify that a model achieved desired outcomes. Additionally, experts often support multiple valid opinions about desired processes and outcomes. Rubric-based rewards serve as a way to learn from the messiness of expert human opinions. How to evaluate and train with rubrics as environments is an exciting area of research with roots laid as early as constitutional AI and RLAIF work from Anthropic.

Key points:

- • Objective domains: Games, like pac-man, chess, and Go, have clear states spaces, action spaces, and desired outcomes.

- Math, code, and even some tasks in biology, can often be formulated with near game-like verifiability.

- This is where RL has achieved early massive success already, notably, AlphaProof, AlphaFold, and DeepSeek R1 and the many code generation models on the market today.

- Objective domains: Games, like pac-man, chess, and Go, have clear states spaces, action spaces, and desired outcomes.

- Math, code, and even some tasks in biology, can often be formulated with near game-like verifiability.

- This is where RL has achieved early massive success already, notably, AlphaProof, AlphaFold, and DeepSeek R1 and the many code generation models on the market today.

- Subjective domains: It’s more difficult to measure accuracy in many real world tasks such as generating investment memos, making legal briefs, providing therapy. This makes it difficult to verify that a model achieved desired outcomes. Additionally, experts often support multiple valid opinions about desired processes and outcomes. Rubric-based rewards serve as a way to learn from the messiness of expert human opinions. How to evaluate and train with rubrics as environments is an exciting area of research with roots laid as early as constitutional AI and RLAIF work from Anthropic. Subjective domains: It’s more difficult to measure accuracy in many real world tasks such as generating investment memos, making legal briefs, providing therapy. This makes it difficult to verify that a model achieved desired outcomes. Additionally, experts often support multiple valid opinions about desired processes and outcomes. Rubric-based rewards serve as a way to learn from the messiness of expert human opinions. How to evaluate and train with rubrics as environments is an exciting area of research with roots laid as early as constitutional AI and RLAIF work from Anthropic. Computer-use agents sit somewhere in the middle. For most of the tasks humans do on computers, goals start to become ambiguous and multi-faceted. Once defined, the actions and outcomes are programmatic and verifiable. These could include planning trips, responding to emails, shopping, or posting on social media. In all of these cases, containerized environments allow for horizontal scaling to learn online from thousands of interactions in parallel. Computer-use agents sit somewhere in the middle. For most of the tasks humans do on computers, goals start to become ambiguous and multi-faceted. Once defined, the actions and outcomes are programmatic and verifiable. These could include planning trips, responding to emails, shopping, or posting on social media. In all of these cases, containerized environments allow for horizontal scaling to learn online from thousands of interactions in parallel. Environments Create Experience

Key points:

- Environments Create Experience Eventually, our AI systems will learn automatically from signals in the real world like pupils’ test scores increasing, sales closing, maybe even bridges being built.

- However, intermediate rewards will always remain critical.

- Similar to how humans learn from other people, models will need guidance on which styles of teaching and sales techniques are most effective.

- Humans will remain an integral part of the environments models learn from.

- Eventually, our AI systems will learn automatically from signals in the real world like pupils’ test scores increasing, sales closing, maybe even bridges being built.

- However, intermediate rewards will always remain critical.

Similar to how humans learn from other people, models will need guidance on which styles of teaching and sales techniques are most effective. Humans will remain an integral part of the environments models learn from. We will never escape the era of data; it must follow us to the frontier. That frontier is human created environments that provide durable sources of experiential data. These environments can serve to train and evaluate models. We will never escape the era of data; it must follow us to the frontier. That frontier is human created environments that provide durable sources of experiential data. These environments can serve to train and evaluate models.

The Path Forward

Key points:

- The Path Forward Meeting today’s data demand requires rethinking the way we generate signal from human efforts.
- Creating evals and RL environments is the highest leverage and most durable use of people’s time.
- Mercor has helped pioneer environment generation using autograders and continues to push the boundaries of RL data with simulated workspaces, multi-turn support, and multi-modality.

- Meeting today’s data demand requires rethinking the way we generate signal from human efforts.
- Creating evals and RL environments is the highest leverage and most durable use of people’s time.
- Mercor has helped pioneer environment generation using autograders and continues to push the boundaries of RL data with simulated workspaces, multi-turn support, and multi-modality.

Knowledge work will quickly converge on building RL environments and evaluations for agents to learn from. As AI enters the workforce and operates over proprietary information and under unique professional contexts, these environments codify knowledge and goals for agents. Once individual steps of agentic workflows reach sufficient reliability, all that will be left will be RL training on the goals laid out by humankind. Knowledge work will quickly converge on building RL environments and evaluations for agents to learn from. As AI enters the workforce and operates over proprietary information and under unique professional contexts, these environments codify knowledge and goals for agents. Once individual steps of agentic workflows reach sufficient reliability, all that will be left will be RL training on the goals laid out by humankind.
"https://www.linkedin.com/feed/update/urn:li:ugcPost:7345546932519411712" "urn:li:ugcPost:7345546932519411712" "urn:li:ugcPost:7345546932519411712" "https://www.linkedin.com/signup/cold-join?session_redirect=%2Fpulse%2Fwelcome-era-evals-brendan-foody-ezykc%2F" "%reactionType% %selectionState%" Celebrate Support Insightful Comment "Link copied to clipboard." "Something went wrong while copying to clipboard." • Copy

- LinkedIn

LinkedIn • Facebook

Facebook • Twitter

Twitter "%numReactions% Reaction" "%numReactions% Reactions" "https://static.licdn.com/aero-v1/sc/h/cyfai5zw4nrqhyyhl0p7so58v" "https://static.licdn.com/aero-v1/sc/h/asiqslyf4ooq7ggllg4fyo4o2" "https://static.licdn.com/aero-v1/sc/h/22ifp2etz8kb9tgjqn65s9ics" "https://static.licdn.com/aero-v1/sc/h/a0e8rff6djeoq8iympcysuqfu" "https://static.licdn.com/aero-v1/sc/h/bn39hirwzjqj18ej1fkz55671" "https://static.licdn.com/aero-v1/sc/h/cryzkreqrh52ja5bc6njlrupa" "https://static.licdn.com/aero-v1/sc/h/2tzoeodxy0zug4455msr0oq0v" 81 Comments Daniel Fiott Chief Executive Officer @ Cloud Continuous

Chief Executive Officer @ Cloud Continuous • Report this comment

Report this comment Era of Evals has a nice ring to it. Love seeing this actually happening at enterprise scale. Era of Evals has a nice ring to it. Love seeing this actually happening at enterprise scale. 1 Reaction Kavyasree Kuruva Intern @GQT | Java Developer

Intern @GQT | Java Developer • Report this comment

Report this comment Thank you for the opportunity. I am a fresher. This is my email id kavyasreekoli@gmail.com

Key points:

- Thank you for the opportunity.
- This is my email id kavyasreekoli@gmail.com 1 Reaction • Report this comment

Report this comment Huge milestone, Brendan! Working with 6 out of the Magnificent 7 is no small feat.

- At agenQ, we’re big believers in how AI talent is reshaping the enterprise,Mercor’s momentum is proof.
- Excited to see what’s next! Huge milestone, Brendan! Working with 6 out of the Magnificent 7 is no small feat.
- At agenQ, we’re big believers in how AI talent is reshaping the enterprise,Mercor’s momentum is proof.

Excited to see what’s next! 1 Reaction 2 Reactions Satwik Behera Machine Learning Engineer | Data Scientist

Machine Learning Engineer | Data Scientist • Report this comment

Report this comment satwikbehera12@gmail.com

satwikbehera12@gmail.com 1 Reaction Archana B Research Scholar @ VNIT, Nagpur

|| Information Retrieval || Image Processing || Computer Vision || NLP

Research Scholar @ VNIT, Nagpur

|| Information Retrieval || Image Processing || Computer Vision || NLP • Report this comment

Report this comment archana.narayandas07@gmail.com

archana.narayandas07@gmail.com 1 Reaction See more comments To view or add a comment, sign in

To view or add a comment, sign in

More articles by Brendan Foody

More articles by Brendan Foody • The Secret Mercor Master Plan

Mar 4, 2025

The Secret Mercor Master Plan

Imagine a world where Jeff Bezos is a hedge fund investor, Howard Shultz is a salesman, and Reed Hastings is a teacher….

292

- Comments

The Secret Mercor Master Plan Mar 4, 2025

The Secret Mercor Master Plan

The Secret Mercor Master Plan Imagine a world where Jeff Bezos is a hedge fund investor, Howard Shultz is a salesman, and Reed Hastings is a teacher.…

Imagine a world where Jeff Bezos is a hedge fund investor, Howard Shultz is a salesman, and Reed Hastings is a teacher.… "%numReactions% Reaction" "%numReactions% Reactions" "https://static.licdn.com/aero-v1/sc/h/cyfai5zw4nrqhyyhl0p7so58v" "https://static.licdn.com/aero-v1/sc/h/asiqslyf4ooq7ggllg4fyo4o2" "https://static.licdn.com/aero-v1/sc/h/22ifp2etz8kb9tgjqn65s9ics" "https://static.licdn.com/aero-v1/sc/h/a0e8rff6djeoq8iympcysuqfu" "https://static.licdn.com/aero-v1/sc/h/bn39hirwzjqj18ej1fkz55671" "https://static.licdn.com/aero-v1/sc/h/cryzkreqrh52ja5bc6njlrupa" "https://static.licdn.com/aero-v1/sc/h/2tzoeodxy0zug4455msr0oq0v" 24 Comments

Sign in

Sign in Stay updated on your professional world

Stay updated on your professional world Sign in By clicking Continue to join or sign in, you agree to LinkedIn’s User Agreement, Privacy Policy, and Cookie Policy. By clicking Continue to join or sign in, you agree to LinkedIn’s User Agreement Privacy Policy Cookie Policy New to LinkedIn? Join now

New to LinkedIn? Join now

Explore topics

Explore topics • Sales

- Marketing

Marketing • IT Services

IT Services • Business Administration

Business Administration • HR Management

HR Management • Engineering

Engineering • Soft Skills

Soft Skills • See All

See All

■ **Shared by:** Vir Kashyap
■ **Shared on:** Jul 01, 2025 at 03:37 PM
■ **Domain:** www.linkedin.com
■ **Length:** 2,041 words
■ **Processed:** Jul 08, 2025 at 01:07 PM