

KSC3: From Distributional Structural Internalization to Transgenerational AI Evolution — The DSI and ES–DSI Mechanisms

Liu Ming

English Final V2.2 · August 2026

Abstract

KSC abstracts intelligent activity into three fundamental variables: Knowledge (K), Structure generation (S), and Constraint maintenance (C). Existing KSC work describes K as the knowledge and crystallized structures available to the current task, but a deeper question remains: for generative models, how do such structures enter K, and how do they subsequently influence S?

This paper proposes **Distributional Structural Internalization (DSI)** as a further interpretation of K. For structural knowledge, model capability is not well described as a binary state of “knowing” or “not knowing.” When facing a particular problem, multiple existing problem representations, relations, and solution structures may participate in subsequent generation with different initial tendencies. DSI focuses specifically on problem-solving structures that can transfer across surface-level instances: how the statistical distribution of such structures in the training environment is transformed into generative tendencies when a model encounters new but structurally related problems, so that the initial candidate-structure distribution of S already assigns different weights to different structures. DSI is therefore not a general renaming of the idea that “models learn the training distribution,” but an abstraction of the functional relationship by which **distributions of structural experience give rise to transferable initial structural tendencies**. This paper does not claim that the underlying training phenomena on which DSI depends are newly discovered. Prior work on distributional memorization, data-mixing ratios, procedural knowledge, and knowledge storage and extraction provides adjacent empirical foundations. The contribution here is to place these phenomena within KSC and provide a unified abstraction of the structural role of K.

Building on DSI, this paper further proposes the **ES–DSI mechanism of transgenerational structural evolution**. AI systems and humans continually generate new candidate structures, while engineering environments, real-world feedback, and human society impose External Selection (ES) on those structures. Selection does not merely keep or discard an individual result. Through adoption, replication, dissemination, and long-term persistence, it reweights the relative distribution of structures in future data and knowledge environments. Only selected structures that enter data accessible to later AI systems, are actually included in training, and receive effective training exposure have the opportunity to be internalized through DSI into new parameterized K. Consequently, successful structures that were low-probability outcomes in the S of an earlier generation and required extensive search to discover may, when propagation and effective-training-exposure con-

ditions are satisfied, become higher-probability initial candidates in later models.

The cycle can be summarized as:

$K_t \rightarrow (S_t, C_t) \rightarrow \text{New Structures} \rightarrow \text{ES} \rightarrow \text{Reweighted Structural Distribution} \rightarrow \text{Accessibility} \rightarrow \text{Effective Training Exposure} \rightarrow \text{DSI} \rightarrow K_{t+1} \rightarrow (S_{t+1}, C_{t+1})$

ES and DSI form the external and internal stages of this mechanism, respectively: **ES changes which structures are more prevalent in the world, while DSI changes which structures a model is more likely to generate when facing a problem.** Two transmission conditions connect the two: selected structures must enter data accessible to later AI systems and receive effective exposure in actual training. Software engineering is a candidate domain for observing ES–DSI because AI already participates in multiple forms of structural generation there, execution-based verification is relatively strong, human selection is frequent, and solution structures can be preserved in executable code.

Model trainers can choose which data enter a training set and adjust the mixing ratios of different data sources, but control over data selection is not equivalent to control over structure creation. Real-world production and knowledge ecosystems can continually generate, validate, and accumulate new structural experience with transfer value. Expert curation, controlled experiments, high-quality synthetic data, and private engineering environments can also produce effective accumulation. For general-purpose models, distributed external selection in public digital environments may provide scale, diversity, and long-term real-world feedback that are difficult for any single trainer to reproduce independently, but its contribution relative to other sources of structural experience remains an empirical question requiring comparative tests.

This paper further outlines observable, testable, and potentially quantifiable empirical paths, including the adoption and survival rates of candidate structures, propagation range, training exposure, changes in initial structural distributions, and task performance across model generations. It distinguishes between “the existence of differential retention” and “a positive contribution of such retention to capability accumulation.” ES does not itself guarantee that higher-quality structures will prevail. ES–DSI becomes a positive capability-accumulation mechanism only when real-world selection is sufficiently and consistently positively correlated with reusable structural quality, and when selected structures can enter environments accessible to later AI systems. This open ecological inheritance chain can carry both human-generated and AI-generated structures; only the component generated by AI or human–AI collaboration that produces an incremental effect in later AI systems constitutes the transgenerational AI self-feedback component examined in this paper. This paper does not claim that ES–DSI has been shown to be the dominant source of current AI capability growth. Instead, it proposes a testable long-term hypothesis: **as AI-generated structures account for a growing share of newly produced structures in the digital world, ES–DSI may become one of the important drivers of transgenerational AI capability growth.**

Keywords: KSC; Distributional Structural Internalization; DSI; External Selection; ES–DSI; transgenerational structural evolution; initial structural distribution; knowledge inheritance

1. Introduction

Generative AI is increasingly participating in the production of code, algorithms, software architectures, experimental designs, theoretical explanations, and other cognitive structures. The current generation of AI therefore no longer merely uses knowledge already present in human history; it is also beginning to participate in producing new structures that future intelligent systems may encounter and inherit.

This introduces a temporal dimension of AI capability growth that warrants separate study. Conventional analyses usually focus on model scale, training methods, inference compute, or model-internal self-improvement. Across model generations, however, another question arises: how do newly produced structures change the starting point from which future models solve problems?

Answering this question first requires a deeper understanding of K. For a generative model, what does it mean to “possess” a piece of structural knowledge? Does it merely mean that the model can state a principle when prompted, or does it also mean that, when facing a related problem, the model more naturally forms certain problem representations, relations, or solution structures?

A further question follows if AI itself is already producing large numbers of new structures: what determines which of those structures persist in future digital environments? And how do differences among structures in the external world re-enter later models and alter subsequent structural generation?

This paper therefore centers on two connected questions:

How do existing structures in K influence the structural distribution when S generates its initial candidates?

and:

How can newly generated structures in the current generation pass through real-world selection and then become changes in the K of later generations?

The paper first addresses the former along the K axis of KSC, and then addresses the latter across time.

The paper makes three main contributions. First, it deepens the account of how K influences subsequent S through the concept of an **initial candidate-structure distribution**. Second, using transferable problem-solving structures as the unit of analysis, it connects real-world differential retention, training exposure, and the structural tendencies of later models. Third, it distinguishes general open-ecological inheritance from the self-feedback component arising specifically from AI-generated structures, and outlines empirical paths for testing this chain stage by stage.

2. The Unresolved Problem of K in KSC

2.1 K Is More Than a Collection of Facts

The foundational KSC work abstracts the functional roles in intelligent tasks into K, S, and C [1]:

- **K** provides knowledge and crystallized structures available to the current task;
- **S** forms new problem representations, variable relations, solution schemes, and search paths on the basis of existing K;
- **C** keeps the generative process aligned with goals, boundaries, premises, and acceptance constraints.

KSC2 further examines structural openness and structural autonomy along the S axis [2]. This paper instead returns to K: when K contains not only facts but also mature problem-solving structures, how do those structures influence new S?

Factual knowledge invites a binary intuition: “the model knows a fact, or it does not.” Yet much of the K that actually matters for complex-task capability does not take this form. Examples include ownership and lifecycle structures, divide-and-conquer, dynamic programming, transactions, state machines, proof by contradiction, and a wide range of software architectures, mathematical transformations, and experimental designs.

When facing a concrete problem, the key issue is often not whether the model can explain these terms, but rather:

whether relevant structures naturally enter the candidate set for the current problem, and how strongly they are prioritized relative to alternative structures.

2.2 How K Influences S

When a generative model faces an open-ended problem, multiple structures may be available for invocation or recombination. K therefore cannot be understood only as a static knowledge base; when S first generates candidate structures, K may also exert different prior influences on different paths.

The next section develops a way to describe this effect.

3. DSI: How Structural Distributions Are Internalized into K

3.1 From “Knowing” to Initial Structural Tendencies

This paper defines:

Distributional Structural Internalization (DSI): the process by which training transforms the statistical distribution of **problem-solving structures that can transfer across surface-level instances** in a data environment into initial structural tendencies when a model faces new but structurally related problems, causing different candidate problem representations, relations, and solution structures to carry different generative weights before explicit online search, tool feedback, or multi-round revision begins.

The “same structure” in this paper does not require identical surface expression or local implementation. If multiple instances, despite changes in objects, variable names, and local context, still share the decisive problem decomposition, relational organization, or solution procedure, and can be identified as functionally equivalent under prespecified structural criteria, they may be treated as belonging to the same structural category. Con-

crete studies can establish such equivalence through human annotation, program transformation, behavioral testing, or combinations of these methods.

For a current problem (x), suppose the structures that may participate in S are:

$Z_1, Z_2, \dots, Z_n$

A functional expression can be used to describe their differing tendencies:

$q_0(Z_i \mid x, K)$

Here, (q_0) denotes the initial candidate-structure distribution relative to subsequent explicit search and feedback-driven revision. It does not assume that the model explicitly stores a set of discrete structures and corresponding probabilities internally. The expression is intended only to describe an observable functional outcome:

under similar problem conditions, some structures are more likely than others to become the model’s first structural proposal or an early starting point for generation.

DSI is therefore not concerned with how many “pieces of knowledge” a model stores, nor does it simply rename the general fact that training distributions influence output distributions. It asks a narrower question: **can historical structural experience transfer across different surface forms and local contexts, and thereby reshape the initial candidate-structure distribution of later S ?**

3.2 Multiple Structures Participate in S with Different Weights

Consider a software-engineering problem:

An asynchronously loaded resource is shared by multiple objects. How should the relationships for creation, ownership, and release be designed?

Several candidate structures are possible in principle:

- **Z_1 : explicit ownership, with lifecycle managed by the owner;**
- **Z_2 : reference counting;**
- **Z_3 : ownership by a global singleton;**
- **Z_4 : each consumer creates and releases the resource independently;**
- **Z_5 : lifecycle maintained through temporary callbacks and local state.**

If the model lacks mature structural knowledge relevant to the problem, these candidate paths may be relatively diffuse. To illustrate DSI, consider a purely schematic distribution:

Candidate structure	Illustrative weight
Explicit ownership	0.25
Reference counting	0.20
Global singleton	0.20
Independent management	0.20
Temporary callbacks	0.15

These numbers are not measurements of any real model’s internal probabilities. They are used only to express a functional intuition: **when S first generates candidate structures, different structures need not occupy equal prior positions.**

Suppose that, in a later training environment, mature examples of explicit ownership and lifecycle management receive greater statistical weight. The illustrative distribution for similar problems might then become:

Candidate structure	Illustrative weight
Explicit ownership	0.60
Reference counting	0.20
Global singleton	0.10
Independent management	0.06
Temporary callbacks	0.04

The capability change illustrated here is not that the model has merely learned one more sentence. Rather:

before large-scale online search begins, more of the initial generative weight is already concentrated around mature structures.

Why the structural distribution in the training environment might change systematically is the subject of Section 4.

3.3 DSI Is Not Structural Determinism

DSI does not imply that existing K forces a model to use historical structures.

Suppose a genuinely novel problem appears—for example, resource-lifecycle rules are redefined so that mature structures such as ownership and reference counting no longer apply directly. Existing structures can then serve only as priors; they cannot replace new S.

The model still requires:

Existing K $\rightarrow$ S Search $\rightarrow$ New Structure

DSI therefore changes the starting point of search and the relative probability of candidate structures; it does not eliminate the generation of new structures.

3.4 DSI and Existing Empirical Research

This paper does not claim that the underlying phenomena on which DSI depends are newly discovered.

Wang et al. introduced **distributional memorization** to measure correlations between a model’s output probability distribution and frequency distributions in pretraining data, observing varying degrees of distributional memorization across multiple tasks [5]. In controlled data-mixing experiments, Gu et al. found that knowledge acquisition can vary nonlinearly with the mixing ratio of knowledge-intensive data and with model scale, even exhibiting phase-transition-like behavior [6]. In controlled experiments using biographical data, Allen-Zhu and Li found that reliable knowledge extraction is strongly

related to diversity of expression in the training data; mere occurrence during training is not sufficient to guarantee stable later extraction [7]. Merullo et al. further found, in their study of factual relations, a significant association between subject–object co-occurrence frequency in pretraining data and the formation of linear internal representations, providing more direct adjacent evidence that training distributions can affect internal knowledge representations [8].

Ruis et al. directly examined how procedural knowledge in pretraining data affects reasoning. They found that, across different problems within the same mathematical task, highly influential pretraining documents often contain reusable solution procedures rather than simply reproducing specific answers or intermediate steps [9]. This result is closer than factual co-occurrence evidence to the cross-instance structural transfer emphasized by DSI, but it still does not directly manipulate distributions of structural experience or measure probabilities over initial candidate structures.

Taken together, these results indicate that:

- the distribution of training data and model output probabilities can have measurable relationships;
- the relative weight of data within the training mixture can affect knowledge acquisition;
- the form and diversity of knowledge exposure can affect whether knowledge becomes reliably extractable from parameters;
- pretraining-data frequency can have measurable associations with the formation of some internal knowledge representations;
- procedural knowledge in pretraining data can influence a model’s reasoning strategy across different problems.

These studies do not directly establish the problem-solving structural tendencies defined in this paper. DSI is a structural extrapolation from these localized findings: **for problem-solving structures that can transfer across surface-level instances, the distribution of structural experience in the training environment can be mapped into initial structural tendencies when a model faces new but structurally related problems.** If an observed change remains confined to training instances, fixed wording, or local answer frequencies and does not transfer to structurally isomorphic new instances, it is insufficient evidence for DSI in the sense proposed here.

This abstraction does not require all structural knowledge to be determined by simple frequency. Data quality, context, diversity, training stage, model capacity, and optimization dynamics may all affect the final outcome of internalization.

3.5 How DSI Deepens the Interpretation of K

K in KSC can therefore be interpreted more fully:

For parameterized K formed through the internalization of historical structural experience, its role is expressed not only in what the model can state, but also in which existing structures have higher generative tendencies when S first generates candidate structures.

DSI thus clarifies that K is not merely callable content. For parameterized K, it can also

include prior effects of historical structural experience on the initial candidate-structure distribution of S . Here, “tendency” does not refer to every behavioral preference of a model. Architectural inductive biases, system prompts, decoding strategies, or other training objectives can also alter generation tendencies, but they are not therefore automatically part of the structural K discussed here. For structural knowledge, what matters is not only whether a concept can be restated, but also the relative weight its corresponding structure receives when the model faces a new, structurally related problem.

4. External Selection: How the Real World Reweights Structural Distributions

4.1 Generated Structures Do Not Automatically Become Future K

AI can generate large numbers of candidate structures—code, designs, fixes, algorithms, explanations, and more—at low cost. But generation alone does not guarantee high-quality knowledge growth. Candidate structures include valid and invalid results, stable and brittle results, and results with high or low reuse value.

Therefore:

Generated Structure $\neq$ Future K

Selection must occur between present generation and future K .

4.2 Definition of External Selection

This paper defines **External Selection (ES)** as:

the **differential retention** imposed on different candidate structures, after they are generated, by model-external processes such as engineering execution, real-world feedback, human judgment, user adoption, peer scrutiny, and long-term practice.

ES does not require a unified central selector or a single fitness function. As long as different structures acquire different probabilities of survival, reuse, and propagation through real-world processes, external selection exists in the sense used here. Real-world performance can be an important source of this differentiation, but it is not the only one: visibility, network effects, market preferences, and institutional environments may also influence which structures are retained. Thus, **ES describes differential retention itself; it does not presuppose that the direction of selection is necessarily aligned with structural quality.**

4.3 The Key Effect of ES Is Reweighting Structural Distributions

If selected structures continue to be adopted, copied, and disseminated, their probability of future occurrence rises; failed or low-value structures are more likely to fall out of propagation chains.

The long-term effect of ES is therefore not simple “keep/delete” filtering, but:

continuous reweighting of the relative distribution of structures in the real-world knowledge environment.

This can be represented conceptually as:

$$P_{\text{world}}(Z_i) \rightarrow P'_{\text{world}}(Z_i)$$

This does not assume that the real world contains a single directly measurable probability distribution. It indicates only that the relative opportunities for different structures to be observed, copied, and incorporated into future data environments can change.

This is the key point that allows ES to connect with DSI.

5. ES–DSI: A Mechanism of Transgenerational Structural Evolution in AI

5.1 Real-World Selection; Model Internalization

DSI explains how external structural experience that enters training can become model K. ES explains why the structural distribution of the external world can change through long-term selection.

A necessary transmission condition connects them: **selected structures must enter data, training, or knowledge environments accessible to later AI systems.** The complete relationship is therefore:

ES: the real world reweights the structural distribution

↓

Accessibility: selected structures enter data environments accessible to later AI systems

↓

Effective Training Exposure: relevant structures are actually included in training and receive sufficient learning weight

↓

DSI: training internalizes that structural experience into initial structural tendencies

The core ES–DSI relationship can therefore be compressed as follows:

Real-world selection reweights what becomes more prevalent, but this creates only an opportunity for inheritance. Only structures that actually enter later AI learning environments can alter models through DSI. The K internalized by those models then influences what the next round of S is more likely to generate.

The complete KSC temporal cycle is:

$K_t \rightarrow (S_t, C_t) \rightarrow \text{New Structures} \rightarrow \text{ES} \rightarrow \text{Reweighted Structural Distribution} \rightarrow \text{Accessibility} \rightarrow \text{Effective Training Exposure} \rightarrow \text{DSI} \rightarrow K_{t+1} \rightarrow (S_{t+1}, C_{t+1})$

Here, (K_{t+1}) refers specifically to parameterized K changed through training. Retrieval, codebases, and tools can also turn external structures into K callable by later tasks, but they do not pass through DSI as defined in this paper.

The two central mechanisms remain:

- **ES: structures generated by an earlier generation $\rightarrow$ externally selected structural distribution;**

- **DSI: accessible distribution of structural experience → initial structural tendencies in the K of later models.**

“Accessibility” and “Effective Training Exposure” are not independent mechanisms proposed by this paper. They are two necessary transmission conditions determining whether the outcome of ES can enter DSI. This complete cycle is referred to here as the **ES–DSI mechanism of transgenerational structural evolution**.

5.2 General Ecological Inheritance and the AI Self-Feedback Component

ES–DSI does not apply only to AI-generated structures. Human-generated code, algorithms, theories, and engineering methods can likewise enter later models through real-world selection, propagation, and training exposure. The complete cycle is therefore, first of all, a mechanism of **structural inheritance in an open ecology**.

This paper further focuses on the incremental feedback introduced by AI:

AI or human–AI collaboration generates new structures → the real-world environment imposes ES → a subset enters accessible data and receives effective training exposure → later AI systems form new K through DSI.

Only this subset constitutes transgenerational AI self-feedback in the strict sense. Empirical studies should therefore record, as far as possible, the provenance of structures, the degree of AI participation, and their propagation paths, and compare against the counterfactual baseline: “without the AI-generated structure in question, would later models still have acquired the same structure?” If existing human knowledge cannot be distinguished from AI-added contributions, observed capability changes in later models can support only general ecological inheritance, not attribution to transgenerational AI feedback by itself.

5.3 How S in an Earlier Generation Can Become K in a Later Generation

A direct capability implication of ES–DSI is:

new structures that required costly search within the S of an earlier generation can become higher-probability initial candidates in the K of a later generation.

When later models face similar problems, they may therefore shift from “searching again” toward “recognizing a mature structure and adapting it locally.”

SCR provides a fuller account of the relationship among search space, coverage capability, and reliability [3]. In the present context, a more cautious formulation is: **when historically mature structures remain applicable to the current task, high-quality initial structural tendencies formed through DSI can reduce the expected amount of online search required for a system to reach mature candidate structures.** This does not mean that the objective solution space is removed, nor does it imply that every stronger tendency will reduce cost. Incorrect or outdated structural tendencies can likewise steer S toward low-value paths.

5.4 ES–DSI Is More Than Simple Knowledge Updating

Simple knowledge updating can be described as:

More Data → More Knowledge

ES–DSI instead emphasizes:

Generation → External Selection → Distribution Reweighting → Accessibility / Training Exposure → DSI → New K

Without quality-correlated external selection, training environments may accumulate erroneous, low-value, or accidental structures as readily as useful ones. In some recursive training settings, the unfiltered use of model-generated data can even cause quality degradation and model collapse [10]. Without propagation and training exposure, even high-value structures selected by the real world do not automatically enter later models. Without DSI or another stable inheritance mechanism, even structures encountered during later training may fail to become parameterized tendencies capable of changing the starting point of subsequent S.

ES, the transmission conditions, and DSI therefore answer three different questions:

Which structures are more likely to be retained and propagated in the real-world environment?

Which of those structures enter data accessible to later AI systems and receive effective training exposure?

and:

How does that structural experience actually change later AI?

5.5 Training-Data Selection and External Structural Selection

Model trainers can, in principle, choose which data enter a training set and adjust the mixing ratios of different data sources. But this control does not imply that equally valuable new structures have already been discovered and validated in the real world.

DSI does not depend on simple repetition of particular text or answers, but on the distribution of problem-solving structures that can transfer across surface-level instances. Structural generation on real problems, execution-based validation, elimination through failure, long-term use, and reuse across contexts can provide such experience. Trainers can also actively create new structural experience through expert curation, automated validation environments, and high-quality synthetic data, but whether these approaches can reproduce the scale, diversity, and long-term feedback of an open real-world ecology at low cost remains an empirical question.

Training-side data selection therefore primarily determines **which structural experiences a model encounters and internalizes**; external selection in real-world production and knowledge ecosystems helps determine **which new structures are generated, validated, copied, and accumulated**. These functions differ, but they are not mutually exclusive: trainers can actively organize structural generation and validation, while winning structures from the real-world ecology still have to pass through training-data selection.

For general-purpose models, distributed external selection in public code, papers, technical documentation, and other open digital environments may create a structural source with distinctive scale, diversity, and persistence. Private engineering environments, controlled experiments, and high-quality synthetic data can likewise accumulate structures, but their effects are usually confined initially to particular systems; only after authorization and entry into broader training or knowledge environments can they become trans-generational K shared by general-purpose models. How large the contribution of public digital environments is relative to these other sources is a comparative empirical question posed, rather than resolved, by this paper.

5.6 Relation to Existing Work on Generational Accumulation and Self-Consuming Training

Cook et al. proposed *Artificial Generational Intelligence*, studying how individual exploration, social learning, and cross-generational information transfer can produce cultural accumulation in reinforcement-learning environments, including both in-context and in-weights accumulation [4]. Their experiments also include explicit external-selection designs. Ren et al. analyzed generational changes in large language models through the framework of iterated learning, studying how small biases can be amplified through generational transmission and validating the effect across multiple LLMs [14]. These studies directly show that cross-generational information transfer and selection can systematically alter the behavioral distributions of later learners.

A second line of work closer to ES-DSI concerns self-consuming generative models. Ferbach et al. studied a cycle of “model-generated data $\rightarrow$ user preference-based filtering $\rightarrow$ retraining of later generations,” showing that data curation can produce implicit preference optimization in their theoretical setting [13]. Wei and Zhang further studied self-consuming loops with noisy or adversarial human filtering, showing that cross-generational selection can both move distributions toward particular preferences and produce systematic shifts when selection is manipulated [15].

This paper does not claim that it is the first to propose that “generated data can affect later models,” that “external selection can alter later training distributions,” or that “models can undergo generational evolution.” Compared with closed self-training or self-consuming loops, ES-DSI does not require earlier and later models to belong to the same training system; it does not require the selection process to be designed by model developers; and it does not require the direct purpose of real-world selection to be improvement of the next model generation. Its object of study is **how real-world production and knowledge ecosystems themselves can become a layer of structural selection and transmission across model generations**.

The research positioning of this paper does not depend on a broad priority claim over “generational influence” or “external selection.” Rather, it connects four elements within a single mechanism: transferable problem-solving structures as the unit of inheritance; real-world adoption, persistence, and propagation that need not serve later model training as the selection layer; transmission that can occur across organizations and training systems; and the initial candidate-structure distribution of later models as the inheritance outcome. The contribution of ES-DSI is this combined relationship and its decomposable

testing path, not the first discovery of any individual component phenomenon.

6. Software Engineering: A Priority Domain for Observing ES–DSI

Software engineering combines several conditions required for large-scale structural generation, real-world selection, structural propagation, and later training internalization, making it a priority candidate domain for observing ES–DSI.

6.1 AI Already Participates in Structural Generation

AI already participates in multiple software-development activities, including code generation, bug fixing, testing, migration, and engineering documentation. A large-scale repository study of self-disclosed generative-AI use in open-source projects has observed real-world applications across multiple development tasks and content types [16]. Because the provenance of AI-generated code is generally difficult to identify reliably, current evidence is better suited to supporting the claim that “AI has entered real engineering-structure production” than to estimating its overall share precisely. As AI participation increases, the proportion of new engineering structures produced by AI or human–AI collaboration may also grow.

This continually expands the pool of candidate structures entering ES.

6.2 Strong External Selection

Software structures can be subjected to real-world validation through compilation, automated tests, CI, benchmarks, performance tests, real user workloads, and long-term maintenance. The results of these validations can in turn affect the adoption, persistence, reuse, and propagation of different structures.

Many structures can therefore receive real-world feedback stronger than textual evaluation alone, producing observable patterns of differential retention.

6.3 Selection Outcomes Naturally Propagate

Truly effective engineering structures often do not remain isolated instances.

They may enter public or private repositories, libraries and frameworks, API and type designs, tests and specifications, issues and pull requests, technical articles and documentation, and later AI generation and maintenance processes.

The **real-world persistence** of these structures must be distinguished from their **accessibility to later models**. Public code, public documentation, or data licensed for training may change the training exposure of general-purpose models. Private code can be inherited by later systems only through enterprise-internal training, retrieval, tool use, or other authorized knowledge mechanisms. More specifically, only structures that are actually included in training and receive effective exposure can enter DSI. Structures accessed through retrieval, codebases, or tools can support other forms of K inheritance, but they do not belong to parameterized DSI as defined here.

6.4 Engineering Structures Can Be Preserved in Executable Form

Code and engineering structures have an additional property:

they do not merely describe solutions; they often directly preserve executable implementations of those solutions.

Once a mature ownership pattern, transaction pattern, retry structure, state machine, or testing pattern is crystallized into real code, later models that gain access to those instances through data or knowledge environments may encounter not merely an abstract principle, but many real structural instances embedded in different local contexts. Code does not necessarily preserve the full design rationale, failed alternatives, or applicability boundaries, but it usually preserves the executable implementation of a solution structure more directly than purely descriptive text does.

This fits precisely with the “distribution of structures,” rather than isolated knowledge statements, emphasized by DSI.

6.5 A Candidate Explanation for Growth in Programming Capability

ES–DSI therefore offers a candidate explanation for transgenerational growth in programming capability:

earlier AI systems and humans jointly produce more engineering structures; real engineering environments continually filter those structures; winning structures acquire greater propagation probability within the code ecosystem; and the subset that actually enters later training and receives effective exposure can then be internalized through DSI into parameterized K by later models.

Structures made callable through retrieval, codebases, or tools can also become K for later tasks, but they constitute a different inheritance path and are not included in the narrow definition of DSI used here.

This paper does not infer from this that current improvements in programming capability are mainly caused by ES–DSI. Model scale, architecture, training methods, context, and reasoning mechanisms may all be changing simultaneously. The significance of software engineering here is that it provides a real-world environment in which the relative contribution of ES–DSI can potentially be decomposed and tested.

7. Observability, Testability, and Possible Quantification

An important property of ES–DSI is that its key stages can, in principle, be turned into empirical objects of observation.

7.1 Observability of DSI

One can construct a set of problems (x) that share the same underlying structure while differing in surface form and local context, then observe whether different training distributions over structures produce systematic changes in the model’s initial generation probabilities (q_0) for different (Z_i). (q_0) can be estimated by repeatedly sampling a model’s first solution or early structural proposal before tool use, execution feedback, or multi-round

revision, and classifying those outputs according to prespecified structural-equivalence criteria.

For example, while controlling model scale and baseline capability, one could vary the occurrence rate, expressive diversity, contextual distribution, and training-mixture weight of a mature structure, and then test whether the model becomes more likely to autonomously form an equivalent structure on **structurally isomorphic new instances not seen during training**. Variable names, domain surface features, and local narratives can be intentionally changed to make simple wording memorization or answer reproduction less plausible as explanations.

DSI yields a directional prediction that carries a clear risk of failure: with model architecture, training budget, sample quality, and expressive diversity held comparable, increasing the effective training exposure of structure (Z) should increase the probability that the model first proposes (Z), or an equivalent structure, on unseen structurally isomorphic tasks. If this effect remains absent across multiple prespecified task families and structural categories, DSI would be rejected within that scope. If changes occur only for fixed expressions or previously seen instances, they would at most demonstrate general memorization or distribution fitting and would not independently support the narrower DSI claim proposed here.

7.2 Observability of ES

For AI-generated candidate structures, one can track:

- adoption rates;
- testing or verification rates;
- merge rates;
- long-term survival rates;
- numbers of replications and reuses;
- propagation multipliers into additional code, documentation, and data instances.

Two empirical propositions must be distinguished.

First, **does ES exist?** Do different candidate structures actually acquire systematically different probabilities of retention, reuse, and propagation? If such differences persist reliably, ES exists in the sense used in this paper.

Second, **does ES exert a positive selection effect on capability accumulation?** After controlling for confounders such as visibility, publication time, and project scale, do structures with stronger real-world validation and greater reusable value receive higher long-term propagation probabilities than random or failed structures? If this relationship is absent or reversed, ES may still exist, but the positive capability-accumulation interpretation that “ES–DSI preferentially converts successful real-world structures into later capability” would be weakened.

ES–DSI is therefore not inherently equivalent to capability progress. Its direction depends on what real-world selection is actually reweighting.

7.3 Cross-Generational Testing of ES–DSI

A more complete test must connect four levels:

Structural Quality / Real-World Selection

↓
Changes in Structural Distribution within Future Accessible Data / Knowledge Environments

↓
Actual Training Exposure of Later Models

↓
Changes in the Initial Structural Distribution of Later Models

For example, one could track a set of structures that emerged at identifiable times and have clear engineering-validation histories, then compare their initial appearance, engineering adoption and propagation, degree of entry into public or authorized knowledge environments, actual exposure in later training corpora, the natural generation probability of related structures across model generations, and changes in performance on related tasks.

Such a study could not by itself prove the source of all model capability growth, but it could test the ES–DSI chain stage by stage. For closed models whose training data are not disclosed, “actual training exposure” is usually difficult to establish directly. The preferred subjects for testing the full chain are therefore open model lineages with traceable data versions and training records. Closed models are better suited to providing temporal-correlation evidence and should not, by themselves, bear the burden of causal identification.

7.4 Potential Quantitative Indicators

Future research could attempt to record and compare:

- the number or growth rate of AI-generated candidate structures;
- the proportion of candidate structures adopted, verified, and retained over the long term;
- the extent to which winning structures are copied, reused, and propagated;
- exposure of related structures in later training;
- changes in the initial structural distribution when later models face related problems;
- changes in performance on related tasks across model generations;
- the provenance shares of AI-generated, human–AI-generated, and independently human-generated structures, together with their respective propagation paths.

This paper does not assume a simple functional relationship among these indicators. Their significance is that **ES–DSI need not remain at the level of a macro narrative that “AI will evolve”; it can be decomposed into a sequence of stages that may eventually be measured, compared, and intervened upon.**

8. Relationship to Other AI Capability-Growth Mechanisms

8.1 Scaling

Classical scaling-law work studies empirical power-law relationships among language-model loss, model scale, training-data scale, and training compute [11].

ES–DSI does not conflict with scaling. Scaling can improve a current model’s generative and learning capabilities; ES–DSI describes how structures produced by current AI

systems can pass through real-world selection and enter the K of future models.

The two may even reinforce each other:

Stronger Models → More High-Quality Structural Generation → Faster ES-DSI Cycle → Richer Future K

8.2 Self-Training, Self-Reward, and Self-Consuming Loops

In *Self-Rewarding Language Models*, models generate rewards for their own outputs through LLM-as-a-Judge and are then iteratively trained through DPO [12]. Meanwhile, work on self-consuming generative models has directly studied recursive loops in which data generated by earlier models enters later training; such data can also be filtered by users, human preferences, or external rules [13,15].

The distinction between ES-DSI and this work does not lie in whether the selector is external to the model, because external selection can itself be part of a deliberately designed training loop. A more stable boundary is that **self-training, self-reward, and self-consuming loops usually study iterative updates within a clearly defined training system, whereas ES-DSI does not require a unified training loop**. Earlier and later models may come from different organizations and training pipelines, and real-world selection need not be performed with the goal of improving the next model generation. ES-DSI is therefore not a particular algorithm, but a mechanism describing how open real-world environments can mediate structural inheritance across model generations.

8.3 Autonomous Recursive Self-Improvement

ES-DSI does not require AI systems to autonomously modify their own parameters, train their own successors, or independently set long-term goals.

The cycle can operate without a closed Recursive Self-Improvement system:

AI Generation → Real-World Selection → Change in Accessible Structural Distribution → Next-Generation Training → Change in K

This means that transgenerational accumulation of AI capability need not wait for the emergence of a complete autonomous self-improvement system.

8.4 Two Long-Term Research Hypotheses

As AI-generated structures account for a larger share of new structures in the digital world:

AI Generation → ES → More Propagating / Accessible Structures → DSI → Richer K → Stronger AI Generation

this feedback loop may become progressively stronger.

This paper therefore proposes a first long-term research hypothesis:

If the share of AI-generated structures becomes sufficiently large, if real-world selection is sufficiently positively biased toward reusable structural quality, and if winning structures can reliably enter environments accessible to later AI systems, ES-DSI may become one of the important drivers of transgenerational AI capability growth.

At present, this is a theoretical extrapolation. Its relative contribution must still be empirically decomposed from scaling, training algorithms, inference compute, and other

sources of capability growth. If real-world selection systematically favors low-quality structures, ES-DSI may instead amplify errors, biases, or local optima rather than produce capability growth.

The paper also proposes a second, comparative hypothesis:

For general-purpose models, distributed external selection in public digital environments may make a contribution to the scale, diversity, long-term real-world validation, and cross-organizational sharing of structural experience that is difficult for any single trainer to reproduce independently.

This hypothesis does not presuppose that public environments are superior to expert curation, private engineering data, controlled experiments, or high-quality synthetic data. The actual contribution of different sources to later K must be compared in terms of structural quality, accessibility, effective training exposure, and transfer across instances.

9. Scope and Boundaries of the Claims

First, **the paper’s originality claim is concentrated on connecting open real-world selection, training exposure, and the initial structural distribution of later models into a cross-model generational mechanism, with transferable problem-solving structures as the unit of analysis, rather than on the first observation of any individual component phenomenon.** Prior work on distributional memorization, procedural knowledge, knowledge acquisition, iterated learning, external selection, self-consuming loops, and cross-generational cultural accumulation all provides adjacent foundations.

Second, **DSI is not a synonymous reformulation of the claim that “all training distributions influence outputs.”** The focus is whether problem-solving structures that transfer across surface-level instances form observable initial structural tendencies. If a training intervention changes only fixed wording, seen instances, or local answer frequencies and does not transfer to structurally isomorphic new problems, it is insufficient to support DSI in the sense proposed here.

Third, **DSI does not imply that training frequency alone determines the initial structural distribution, nor does it classify every generative tendency of a model as part of K.** Data quality, context, diversity, model capacity, training order, and optimization processes may all affect the outcome. Architectural inductive biases, system prompts, and decoding strategies can also alter behavior. DSI is primarily intended to explain parameterized structural tendencies formed through the internalization of historical structural experience; externally callable K supplied through retrieval, codebases, tools, and similar mechanisms can arise through other processes.

Fourth, **ES itself guarantees neither global optimality nor capability growth.** Engineering environments, human societies, and markets can retain errors, biases, and local optima. The primary claim here is that differential retention can reweight the distribution of real-world structures. Such reweighting supports positive capability accumulation only when selection has a sufficiently stable positive relationship with reusable structural quality.

Fifth, **structures selected in the real world do not automatically enter later mod-**

els. Accessibility and effective training exposure are required between ES and DSI: public dissemination or authorization determines whether a structure has an opportunity to enter the data environment, while inclusion in the training set, sampling weight, and optimization dynamics determine whether it receives actual exposure. Private or inaccessible structures can persist in the real world without necessarily affecting the parameterized K of general-purpose models; structures accessed through retrieval or tools can influence other forms of K without thereby constituting DSI.

Sixth, **this paper does not claim that current AI capability growth is primarily caused by ES–DSI.** The proposition that “ES–DSI may become an important future driver” is a conditional and testable long-term research hypothesis.

Seventh, **this paper does not claim that public environments are the only source of high-value structural experience, nor does it deny that trainers can actively create new structural experience.** Expert curation, controlled experiments, automated validation environments, high-quality synthetic data, and private knowledge systems can all produce effective accumulation. The paper proposes only a comparative hypothesis: for general-purpose models and cross-organizational structural accumulation, distributed external selection in public digital environments may provide scale, diversity, and persistence that are difficult for any single trainer to reproduce independently. The relative contribution remains an empirical question.

Eighth, **general open-ecological inheritance is not equivalent to AI’s own trans-generational feedback.** Human-generated structures can likewise enter later models through ES and DSI. Only the identifiable incremental effect on later AI that arises from structures newly generated by AI or human–AI collaboration belongs to the transgenerational AI self-feedback component discussed here.

10. Conclusion

KSC3 begins by asking a deeper question along the K axis of KSC: for generative models, how does structural knowledge actually exert its effect?

DSI provides the following interpretation:

For parameterized K formed through the internalization of historical structural experience, its role is expressed not only in what the model can state, but also in which existing structures have higher generative tendencies when S first generates candidate structures.

Structural distributions in the training environment can be internalized into initial structural tendencies, giving different candidate structures different generative weights before explicit online search and feedback-driven revision begin.

Building on this interpretation, the paper proposes the ES–DSI mechanism of transgenerational structural evolution:

$K_t \rightarrow (S_t, C_t) \rightarrow \text{New Structures} \rightarrow \text{ES} \rightarrow \text{Reweighted Structural Distribution} \rightarrow \text{Accessibility} \rightarrow \text{Effective Training Exposure} \rightarrow \text{DSI} \rightarrow K_{t+1} \rightarrow (S_{t+1}, C_{t+1})$

Its core can be compressed into three statements:

ES reweights which structures become more prevalent in the world.

Accessibility and effective training exposure determine which selected structures actually enter the training process that forms parameterized K in later AI systems.

DSI describes how that accessible structural experience forms initial structural tendencies in later AI systems.

Thus, new structures that were costly to generate within an earlier generation’s S can, after being selected by the real world, propagated, and actually introduced into environments accessible to later AI systems, be transformed across generations into later K, causing the next generation of S to begin from a different initial candidate-structure distribution. This chain can carry structures generated by humans or AI; the incremental component attributable to structures newly produced by AI or human–AI collaboration constitutes transgenerational AI self-feedback.

Trainers can determine which structural experiences a model encounters and can also actively create new structures through expert curation, controlled experiments, automated validation, and synthetic data. Real-world ecosystems provide another, distributed pathway for structural discovery, validation, and accumulation. Whether public digital environments make a contribution to general-purpose models that is difficult to reproduce independently in terms of scale, diversity, long-term real-world feedback, and cross-organizational sharing is a comparative hypothesis proposed by this paper, not an established conclusion.

This creates an open pathway for AI capability growth that differs from simple model scaling or model-internal self-training. The stages of ES–DSI—generation, selection, propagation, exposure, internalization, and capability change—can in principle be observed and intervened upon separately. Once structural categories, training conditions, and directional predictions are specified in advance, its local claims can be exposed to clear conditions of empirical failure and support a path toward further quantification.

References

- [1] Liu, M. **Knowledge, Structure, and Constraint: Human–AI Differences and Collaboration from the KSC Perspective.** Zenodo, 2026. DOI: 10.5281/zenodo.21834579.
- [2] Liu, M. **A Progressive Structural Framework for ASI: Structural Openness, AACC, and Structural Autonomy — Structural Differences Between AGI and ASI from the KSC Perspective.** Zenodo, 2026. DOI: 10.5281/zenodo.21960896.
- [3] Liu, M. **From the SCR Relation to Search-Space Governance: Foundational Mechanisms of Delivery Reliability in AI Software Engineering.** Zenodo, 2026. DOI: 10.5281/zenodo.21902420.
- [4] Cook, J., Lu, C., Hughes, E., Leibo, J. Z., & Foerster, J. **Artificial Generational Intelligence: Cultural Accumulation in Reinforcement Learning.** *Advances in Neural Information Processing Systems 37 (NeurIPS 2024)*, Main Conference Track, 2024. DOI: 10.52202/079017-1907.
- [5] Wang, X., Antoniadis, A., Elazar, Y., Amayuelas, A., Albalak, A., Zhang, K., & Wang, W. **Generalization v.s. Memorization: Tracing Language Models’ Capabilities Back to Pretraining Data.** *International Conference on Learning Representations (ICLR 2025)*,

2025.

[6] Gu, X., Lyu, K., Li, J., & Zhang, J. **Data Mixing Can Induce Phase Transitions in Knowledge Acquisition.** *Advances in Neural Information Processing Systems 38 (NeurIPS 2025)*, Main Conference Track, 2025. DOI: 10.52202/085713-5449.

[7] Allen-Zhu, Z., & Li, Y. **Physics of Language Models: Part 3.1, Knowledge Storage and Extraction.** *Proceedings of the 41st International Conference on Machine Learning (ICML 2024)*, PMLR 235:1067–1077, 2024.

[8] Merullo, J., Smith, N. A., Wiegrefe, S., & Elazar, Y. **On Linear Representations and Pretraining Data Frequency in Language Models.** *International Conference on Learning Representations (ICLR 2025)*, 2025.

[9] Ruis, L., Mozes, M., Bae, J., Kamalakara, S. R., Gnaneshwar, D., Locatelli, A., Kirk, R., Rocktäschel, T., Grefenstette, E., & Bartolo, M. **Procedural Knowledge in Pretraining Drives Reasoning in Large Language Models.** *International Conference on Learning Representations (ICLR 2025)*, 2025.

[10] Shumailov, I., Shumaylov, Z., Zhao, Y., Papernot, N., Anderson, R., & Gal, Y. **AI Models Collapse When Trained on Recursively Generated Data.** *Nature*, 631:755–759, 2024. DOI: 10.1038/s41586-024-07566-y.

[11] Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., & Amodei, D. **Scaling Laws for Neural Language Models.** arXiv:2001.08361, 2020.

[12] Yuan, W., Pang, R. Y., Cho, K., Li, X., Sukhbaatar, S., Xu, J., & Weston, J. E. **Self-Rewarding Language Models.** *Proceedings of the 41st International Conference on Machine Learning (ICML 2024)*, PMLR 235:57905–57923, 2024.

[13] Ferbach, D., Bertrand, Q., Bose, A. J., & Gidel, G. **Self-Consuming Generative Models with Curated Data Provably Optimize Human Preferences.** *Advances in Neural Information Processing Systems 37 (NeurIPS 2024)*, Main Conference Track, 2024. DOI: 10.52202/079017-3256.

[14] Ren, Y., Guo, S., Qiu, L., Wang, B., & Sutherland, D. J. **Bias Amplification in Language Model Evolution: An Iterated Learning Perspective.** *Advances in Neural Information Processing Systems 37 (NeurIPS 2024)*, Main Conference Track, 2024. DOI: 10.52202/079017-1220.

[15] Wei, X., & Zhang, X. **Self-Consuming Generative Models with Adversarially Curated Data.** *Proceedings of the 42nd International Conference on Machine Learning (ICML 2025)*, PMLR 267:66300–66321, 2025.

[16] Xiao, T., Fan, Y., Calefato, F., Treude, C., Kula, R. G., Hata, H., & Baltes, S. **Self-Admitted GenAI Usage in Open-Source Software.** arXiv:2507.10422, 2025.