

Stratified Topographic Similarity

Jean-Baptiste Sevestre*
ENS-PSL, EHESS, CNRS

Matthieu Geist
Earth Species Project

Emmanuel Dupoux
ENS-PSL, EHESS, CNRS

Abstract

Topographic Similarity (TopSim) is the most widely used metric for compositionality in emergent communication research. However, when evaluated on a class of languages where compositional structure is fully known (simple natural-language word concatenations), TopSim, along with other well-known metrics, under-reports compositionality. We investigate and trace TopSim’s failure to a distributional imbalance in pairwise object distances that grows with the input space size, and propose stratified-TopSim, a minimal fix that remains stable across settings and reliably evaluates languages of varying compositional structure, on both controlled benchmarks and emergent languages.

1 Introduction

A central question in emergent communication (EC) is whether neural agents trained to coordinate develop compositional codes, that is, codes in which the meaning of a message is a systematic function of its parts (Lazaridou et al., 2018; Kottur et al., 2017). One metric dominates this evaluation: Topographic Similarity (Brighton and Kirby, 2006). It remains the de facto standard in the recent literature (Sevestre and Dupoux, 2025; Elberg et al., 2025; Kouwenhoven et al., 2025), in various forms, where it is generally reported as a headline measure of compositional structure rather than examined. Conclusions about when and why compositional structure emerges therefore rest on the metric being reliable across the settings in which it is applied.

The reliability of this metric has been questioned before. Conklin and Smith (2023) argue that emergent languages are in fact compositional but exhibit natural-language-like variation that existing metrics misread as non-compositionality.

More recently, Boldt and Mortensen (2026) find TopSim a weaker predictor than a morpheme-level alternative of how well agents built from induced morphemes can use an emergent language, and Gilberti et al. (2026) show that its edit-distance formulation fails to recognize composition that modifies or reorders characters. All three respond to these shortcomings by proposing new or modified measures. We take a complementary route: we identify a specific cause of TopSim’s failure and correct TopSim with a minimal change that leaves its definition and usage intact.

We build a benchmark of trivially compositional natural-language codes across three configurations (English and Mandarin adjectival constructions, and an English subject-verb-object construction) and find that all three standard metrics substantially under-report compositionality. We trace TopSim’s failure to a distributional imbalance in pairwise object distances that grows with the input space size, and propose stratified TopSim (strat-TopSim), a minimal fix that addresses this imbalance. Strat-TopSim remains stable across a wide range of input space sizes where classical TopSim degrades sharply, and reliably orders languages of varying compositional structure. On genuinely emergent languages obtained by training neural agents, classical TopSim spuriously declines as the input space grows, while strat-TopSim remains stable.

2 Background

Input space, objects and messages . In emergent communication, neural agents are trained to coordinate over a discrete channel, producing an emergent language mapping objects to messages (Lewis, 1969; Lazaridou et al., 2018). Following standard practice (Kottur et al., 2017; Chaabouni et al., 2020), the input the Speaker observes is called an *object*, written o : objects are tuples of N discrete attributes, each taking one of K values;

*Corresponding Author: jean-baptiste.sevestre@ens.psl.eu

the input space thus contains K^N distinct objects, and the referent vocabulary $N \cdot K$ distinct attribute values. A *message* $m(o)$ is the discrete sequence of symbols the Speaker emits over the channel after observing o , and that the Listener consumes to produce $\hat{o}$, its prediction of the attributes of the input object. Symbols are drawn from a channel vocabulary of size V and messages have a maximum length L ; both are fixed in advance and are unrelated to N and K . For instance, with $N=3$, $K=10$, $V=10$ and $L=7$, a Speaker may map the object $o = (2, 7, 0)$ to the message $m(o) = 4928022$. The natural-language codes of Section 3 instantiate the same object-message mapping with m predefined by construction rather than produced by training.

Compositionality metrics. Topographic Similarity (TopSim) (Brighton and Kirby, 2006) is the Spearman correlation between pairwise object distances (Hamming) and pairwise message distances (Levenshtein); higher TopSim indicates that similar objects produce similar messages. Positional Disentanglement (PosDis) and Bag-of-symbols Disentanglement (BosDis) (Chaabouni et al., 2020) are mutual-information-based scores measuring, respectively, how well each message position or each symbol count aligns with a single attribute. PosDis and BosDis both assume a fixed alignment between message structure and attributes, which holds only when attribute values are encoded by tokens of identical length and structure; TopSim makes no such assumption. Since enumerating all pairs is intractable, metrics are computed on sampled object pairs; we report mean and standard deviation over 5 independent samplings of $N_{\text{pairs}} = 1,000$ pairs each, unless stated otherwise, and use the same budget for every metric so that all are compared at equal sampling cost.

3 Compositionality Metrics on Trivially Compositional Language

Existing compositionality metrics in EC are typically validated on synthetic miniature languages or applied to emergent codes whose underlying structure is not directly accessible. To our knowledge, none has been systematically evaluated on languages whose compositional structure is fully known and uncontroversial: simple natural-language word concatenations such as *re-droundsmooth*.

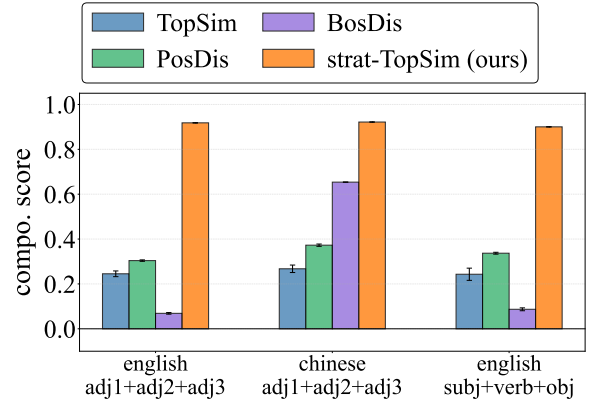


Figure 1: Compositionality scores on three trivially compositional natural-language configurations ($N=3$, $K=100$). Each object is encoded by concatenating one word per attribute: colors, shapes and textures for the two adjectival configurations, and subject, verb and grammatical object for the third.

Benchmark. We construct trivially compositional languages by mapping each attribute value to a real word and concatenating the resulting words. We consider three configurations, each with $N=3$ attributes, and $K=100$ values following Chaabouni et al. (2020), Auersperger and Pecina (2022), and Sevestre and Dupoux (2025): (i) three English adjectival attributes (colors, shapes, textures); (ii) the same in Mandarin Chinese; and (iii) an English subject-verb-object construction, whose third attribute is the grammatical object, not to be confused with the input object o . In every configuration the message carries one word per attribute: one adjective, or one subject, verb or grammatical object. Reading the three attributes of an object as adjective types, $o = (1, 2, 1)$ maps to $m(o) = \text{bluesquaresmooth}$; reading them as subject, verb and grammatical object, it maps to $m(o) = \text{youmovetable}$. All three are fully compositional by construction. The word lists are part of the released code.

Results. Figure 1 reports TopSim, PosDis, BosDis, and our proposed stratified TopSim (strat-TopSim, introduced in Section 4). Note that all four metrics are bounded above by 1, with 1 indicating ideal compositionality, so scores are directly comparable across methods. All three under-report compositionality: BosDis peaks at 0.65 on Mandarin Chinese and drops to 0.07 on the English adjectival code, while TopSim and PosDis are systematically low, with scores in $[0.22, 0.36]$ across the three configurations despite the languages be-

Distance class d	1	2	3
$N=3, K=6$	5.2%	31.5%	63.0%
$N=3, K=100$	0.03%	2.91%	97.06%

Table 1: Probability mass over Hamming distance classes. As K grows, the distribution concentrates sharply on $d=N$.

ing fully compositional. PosDis is subject to the alignment limitation noted in Section 2. We therefore set PosDis aside and focus on TopSim which makes no such alignment assumption. We report BosDis as defined by Chaabouni et al. (2020). In contrast, strat-TopSim assigns consistently high scores (0.90–0.92) across all three configurations, reflecting their actual compositional structure. We diagnose the source of TopSim’s failure and propose a correction in Section 4.

4 Stratified Topographic Similarity

A distributional imbalance. The instability of TopSim observed in Section 3 stems from a property of the input space itself. Consider two objects o, o' drawn from the input space of N attributes and K values. The Hamming distance $d_{\text{ham}}(o, o')$ between them follows a binomial distribution with N trials and success probability $1 - 1/K$, which concentrates sharply on its maximum N as K grows. Table 1 quantifies this: for $N=3, K=100$, over 97% of pairs fall into the single distance class $d=N$. This imbalance is a property of the input space, not of the sampling: it persists even when TopSim is computed exhaustively over all pairs.

When 97% of pairs share the same object-side distance, almost all the variation in object distance collapses into a single bin, while message-side distances spread freely. The correlation thus reflects mostly noise within that bin and is largely insensitive to the smaller distance classes, precisely where compositional structure is most informative. We make this argument formal for Pearson correlation in Appendix A; for Spearman, used by convention in EC, no closed-form expression exists, but the empirical results below confirm that the same pathology applies.

Stratified TopSim. To remove this imbalance, we modify the sampling scheme: instead of drawing pairs from the natural (binomial) distribution over Hamming distance classes, we stratify by distance class, ensuring that each class contributes equally to the correlation. Stratified TopSim (strat-

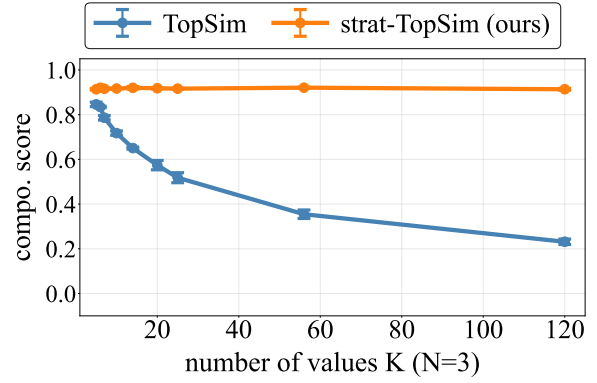


Figure 2: TopSim and strat-TopSim on a fully compositional natural language: the English adjectival configuration of Section 3 (colors, shapes, textures).

TopSim) proceeds as follows. For each of N_{pairs} pairs, we (i) sample a first object o uniformly from the input space; (ii) sample a target distance d uniformly from $\{1, \dots, N\}$; and (iii) construct a second object o' by modifying exactly d attributes of o , with each new value drawn uniformly from the $K-1$ remaining values. We then compute object and message distances on these pairs and return their Spearman correlation (Algorithm 1).

Algorithm 1 Stratified TopSim

From object space O of N attributes, message m

Define number of pairs N_{pairs}

for $i = 1, \dots, N_{\text{pairs}}$ **do**

 Sample object $o_i \sim \text{Uniform}(O)$

 Sample distance $d_i \sim \text{Uniform}\{1, \dots, N\}$

 Build o'_i by modifying d_i attributes of o_i

end for

Compute $\delta_i^{\text{obj}} = d_{\text{ham}}(o_i, o'_i)$ for all i

Compute $\delta_i^{\text{msg}} = d_{\text{lev}}(m(o_i), m(o'_i))$ for all i

return Spearman($\delta^{\text{obj}}, \delta^{\text{msg}}$)

The fix is minimal: a few lines of code, no additional hyperparameters, and the same computational cost as classical TopSim. Strat-TopSim requires the ability to obtain a message for any object in the space, since it constructs a second object at a controlled Hamming distance and needs its message. Compositionality in EC is standardly evaluated on the emergent language as a whole, requiring access to the whole object space and to the messages it maps to. This is the assumption we work under throughout this paper. When this assumption does not hold, for instance when only a pre-existing, fixed set of object-message pairs is

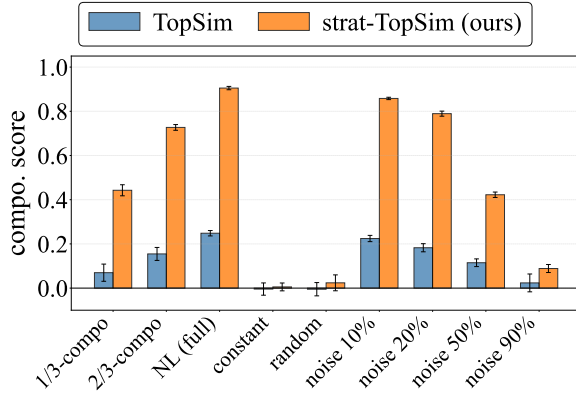


Figure 3: TopSim and strat-TopSim on constructed languages of varying compositionality.

available, as in a released train/test split or in observational data, we fall back on a weighted variant, given in Appendix B.

Strat-TopSim is stable across input space sizes. Figure 2 compares TopSim and strat-TopSim on a fully compositional natural-language code (the English adjectival configuration), varying K from 5 to 120 at fixed $N=3$. Classical TopSim degrades monotonically from 0.85 at $K=5$ down to 0.23 at $K=120$, despite the underlying language being identically compositional throughout. Strat-TopSim remains stable at 0.91 across the entire range. This confirms the diagnosis above: the failure of classical TopSim is a property of the distance distribution, not of the language being measured.

Strat-TopSim correlates with expected compositional score. A correction is only useful if it preserves the discriminative power of the metric. Figure 3 reports strat-TopSim and classical TopSim across nine languages spanning the compositionality spectrum, all at $N=3$ and $K=100$. We take as our reference language the fully compositional English adjectival code from Section 3 (colors, shapes, textures) and derive the other languages from it: 1/3-compo retains only one of the three attributes encoded compositionally and randomizes the other two; 2/3-compo retains two; and four noisy variants replace each token of the reference code with a random token with probability $p \in \{0.1, 0.2, 0.5, 0.9\}$. We also include two extreme baselines: random (each object emits a random message) and near-constant (all objects emit nearly the same message). Strat-TopSim assigns near-zero scores to the random and constant codes, uses the full $[0, 1]$ range on structured codes, and orders them in line with the

expected compositional structure, while classical TopSim gives a weaker signal with higher variance.

5 Case Study: Emergent Language

Sections 3 and 4 established the failure of classical TopSim on natural languages, where compositional structure is known by design. We now ask whether the same failure arises with emergent languages, where no ground-truth compositionality is directly available.

Out-of-distribution (OOD) test accuracy. In-distribution (IND) test accuracy is known to be uninformative about compositionality: emergent codes can reach near-ceiling IND accuracy without exhibiting compositional structure (Chaabouni et al., 2020; Auersperger and Pecina, 2022). Auersperger and Pecina (2022) show, however, that out-of-distribution (OOD) test accuracy (where the test set contains attribute combinations restricted to a small number of training contexts) is informative about compositionality: codes that generalize OOD must possess compositional structure, even if the converse does not hold: compositional structure does not guarantee OOD generalization. They never observe OOD success at low topographic similarity, but do observe compositional codes that nonetheless fail to generalize, so that OOD success is a sufficient but not a necessary signature of compositional structure. We therefore report OOD test accuracy alongside TopSim and strat-TopSim, treating it as a lower-bound indicator of compositionality that is independent of any internal metric.

Setup. We train Speaker and Listener agents in a Lewis reconstruction game (Lewis, 1969) on four input spaces of increasing size: $N=3$ with $K \in \{10, 20, 50, 100\}$. We follow the OOD-split procedure of Auersperger and Pecina (2022), adapted to $N=3$. To avoid confounds from architectural choices specific to their setup, we use a standard EC architecture: single-layer LSTM Speaker and Listener over a discrete channel, with the Speaker trained by REINFORCE and the Listener by cross-entropy. For each setting, we run 5 independent trainings and stop at epoch 30, when all runs reach $> 99\%$ IND test accuracy. We report mean and standard deviation across seeds for TopSim, strat-TopSim, and OOD accuracy. Full architecture, optimization, and OOD-split details are given in Appendix D.

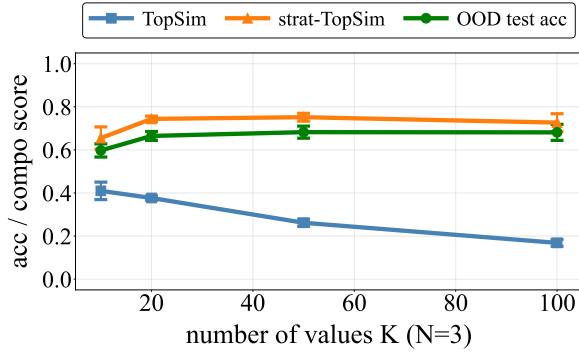


Figure 4: TopSim, strat-TopSim, and OOD test accuracy on emergent languages.

Results. Figure 4 reports the three quantities as a function of K . Classical TopSim drops monotonically from 0.42 at $K=10$ to 0.18 at $K=100$. In contrast, strat-TopSim is stable across $K = 20$ to 100 (0.76), and so is OOD accuracy (0.68); both showing a mild dip at $K=10$, which we tentatively attribute to the smaller input space favoring memorization-like solutions over fully compositional ones. Two observations follow. First, the decline of TopSim with K reproduces the pattern observed in our controlled benchmarks (Section 3), where compositional structure is known to be constant; the empirical decline is also consistent with the Pearson analysis of Appendix A. Second, the OOD-based indicator shows the same stability as strat-TopSim, in contrast to classical TopSim’s decline. Since OOD accuracy is computed from agent generalization rather than from the message strings themselves, the agreement is evidence that strat-TopSim’s stability reflects a property of the codes rather than an artifact of the metric. We note that strat-TopSim and OOD accuracy lie in similar numerical ranges across settings; these quantities measure different things and we make no claim of equivalence, leaving the relationship to future work.

6 Conclusion

Compositionality metrics for emergent communication are tools we use to interpret what neural agents communicate. We have shown that the three standard metrics (topographic similarity, PosDis, and BosDis) all substantially under-report compositionality in trivially compositional natural-language codes, and that the most widely used among them, TopSim, suffers from a distributional imbalance in pairwise object distances that grows with the input space size and increasingly mischaracter-

izes compositional languages as non-compositional. A simple correction, stratified TopSim, removes the imbalance and recovers reliable measurements, both on controlled benchmarks and on emergent languages, where classical TopSim spuriously declines as the input space grows while strat-TopSim remains stable. As EC research moves toward richer, larger input spaces, strat-TopSim provides a bias-corrected alternative that supports reliable empirical claims in regimes where classical TopSim fails, whenever referents are symbolic attribute tuples, with its weighted variant covering, given a sufficient number of available pairs, the case where only a fixed set of object-message pairs is available.

Limitations

Scope of validation. Our benchmark (Section 3) uses natural-language concatenations as a controlled proxy for compositional structure. While these languages are unambiguously compositional, they are also simpler than real natural language: words have fixed forms, no inflection, no agreement, and no recursive structure. Strat-TopSim is expected to behave well on such richer phenomena (since the underlying bias is purely sampling-driven), but we do not test this empirically. Extending the benchmark to morphologically richer codes is a natural next step, as are two further extensions: a construction with a distribution closer to that of natural language, and the deliberate introduction of artificial homonyms.

Symbolic input spaces only. Our analysis assumes objects are tuples of discrete attributes. The Hamming-distance distribution that drives the sampling bias does not transfer directly to continuous or perceptual inputs (e.g., visual EC, Chaabouni et al., 2022). Whether an analogous bias affects TopSim in those settings, and what form a correction should take, are open questions.

Relation to agentic language-model systems.

What our analysis requires is that referents be symbolic tuples, not that the agents be of any particular kind. TopSim and its variants could apply to language-model agents over structured referents, but would fail when referents do not decompose into a fixed set of attributes, as in agentic systems exchanging free-form natural language about unconstrained content, where Hamming distance between referents is undefined and TopSim is not the appropriate instrument. Extending distance-based

compositionality measurement to such settings is an open problem we do not address here.

Access to the object-message mapping. Stratified TopSim requires obtaining messages for the objects it constructs at controlled Hamming distances. Compositionality in EC is standardly evaluated on the emergent language as a whole, with access to the whole object space and to the messages it maps to, but that access is not available when only a pre-existing set of object-message pairs is available (e.g. train/test splits, observational data). For such settings, we propose a weighted variant (Appendix B).

Theoretical analysis restricted to Pearson. Our formal analysis of the sampling bias (Appendix A) is established for Pearson correlation, where closed-form expressions are available. For Spearman, used by convention in the EC literature and throughout this paper, no analogous closed form exists, and our claim that the same bias applies is supported empirically rather than analytically. A formal Spearman analysis is left to future work.

Scope of the emergent case study. Our emergent-language experiments use a single architecture and training setting. While this is sufficient to show that classical TopSim and strat-TopSim diverge systematically on emergent languages, a broader characterization across architectures, optimization methods, and channel configurations is left for future work.

Acknowledgments

We would like to thank Lin Yang and Jing Liu for reviewing the Chinese vocabulary used in our experiments. This work was granted access to the HPC resources of IDRIS under the allocation 2025-AD011015895R1 made by GENCI, and was supported in part by the Agence Nationale de la Recherche (ANR-17-EURE-0017 Frontcog, ANR-10-IDEX-0001-02 PSL*). ED in his EHESS role was funded by an ERC grant (InfantSimulator). Views and opinions expressed are those of the authors only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them.

References

- Michal Auersperger and Pavel Pecina. 2022. Defending compositionality in emergent languages. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Student Research Workshop*, page 285–291. Association for Computational Linguistics.
- Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization. (arXiv:1607.06450).
- Paul Bailey, Ahmad Emad, Ting Zhang, and Qing-shu Xie. 2018. Weighted and unweighted correlation methods for large-scale educational assessment: wCorr formulas. AIR-NAEP Working Paper 2018-01, American Institutes for Research, Washington, DC. NCES Data R Project Series #02.
- Brendon Boldt and David R. Mortensen. 2026. Communicating in emergent language with an induced morphological phrasebook. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, page 30101–30121. Association for Computational Linguistics.
- Henry Brighton and Simon Kirby. 2006. Understanding linguistic evolution by visualizing the emergence of topographic mappings. *Artificial life*, 12(2):229–242.
- Rahma Chaabouni, Eugene Kharitonov, Diane Bouchacourt, Emmanuel Dupoux, and Marco Baroni. 2020. Compositionality and generalization in emergent languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, page 4427–4442. Association for Computational Linguistics.
- Rahma Chaabouni, Florian Strub, Florent Alth  , Eugene Tarasov, Corentin Tallec, Elnaz Davoodi, Kory Wallace Mathewson, Olivier Tieleman, Angeliki Lazaridou, and Bilal Piot. 2022. Emergent communication at scale. In *International conference on learning representations*.
- Henry Conklin and Kenny Smith. 2023. Compositionality with variation reliably emerges in neural networks. In *The Eleventh International Conference on Learning Representations*. Poster. <https://iclr.cc/virtual/2023/poster/10921>.
- Rafael Elberg, Felipe del Rio, Mircea Petrache, and Denis Parra. 2025. A compressive-expressive communication framework for compositional representations. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.
- Miles Gilberti, Shane Storks, and Huteng Dai. 2026. Discovering properties of inflectional morphology in neural emergent communication. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, page 32360–32377. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Eugene Kharitonov, Rahma Chaabouni, Diane Bouchacourt, and Marco Baroni. 2019. Egg: a toolkit for research on emergence of language in games. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, page 55–60. Association for Computational Linguistics.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. (arXiv:1412.6980).

Satwik Kottur, José Moura, Stefan Lee, and Dhruv Batra. 2017. Natural language does not emerge ‘naturally’ in multi-agent dialog. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, page 2962–2967. Association for Computational Linguistics.

Tom Kouwenhoven, Max Peeperkorn, and Tessa Verhoeft. 2025. Searching for structure: Investigating emergent communication with large language models. In *Proceedings of the 31st International Conference on Computational Linguistics*, page 9977–9991. Association for Computational Linguistics.

Angeliki Lazaridou, Karl Moritz Hermann, Karl Tuyls, and Stephen Clark. 2018. Emergence of linguistic communication from referential games with symbolic and pixel input.

David Kellogg Lewis. 1969. *Convention: A Philosophical Study*. Wiley-Blackwell.

Mathieu Rita, Corentin Tallec, Paul Michel, Jean-Bastien Grill, Olivier Pietquin, Emmanuel Dupoux, and Florian Strub. 2022. Emergent communication: Generalization and overfitting in lewis games. *Advances in Neural Information Processing Systems*, 35:1389–1404.

Jean-Baptiste Sevestre and Emmanuel Dupoux. 2025. Frequency and compositionality in emergent communication. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, page 27262–27274. Association for Computational Linguistics.

Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. 1999. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12.

Ronald J. Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3):229–256.

A Pearson Analysis of the Sampling Bias

We make formal the intuition of Section 4 for Pearson correlation. We show that under a standard linear-regression assumption between object and message distances, Pearson correlation depends on K under uniform pair sampling, but becomes independent of K under stratified sampling.

Setup. Let $X = d_{\text{ham}}(o, o') \in \{0, 1, \dots, N\}$ denote the Hamming distance between two sampled objects and $Y = d_{\text{lev}}(m(o), m(o'))$ the Levenshtein distance between their messages. We assume the standard linear-regression relationship

$$Y = aX + b + \varepsilon, \quad (1)$$

where a, b are constants determined by the language (in particular, $a > 0$ measures how much message distance grows per attribute change in a compositional code) and ε is a zero-mean noise term with variance σ_ε^2 , independent of X . Under (1),

$$\text{Cov}(X, Y) = a \cdot \text{Var}(X), \quad (2)$$

$$\text{Var}(Y) = a^2 \cdot \text{Var}(X) + \sigma_\varepsilon^2, \quad (3)$$

so that Pearson correlation reduces to

$$\rho(X, Y) = \frac{a\sqrt{\text{Var}(X)}}{\sqrt{a^2\text{Var}(X) + \sigma_\varepsilon^2}}. \quad (4)$$

The value of ρ thus depends on the language (through a and σ_ε^2) and on $\text{Var}(X)$, the variance of the object-distance distribution induced by the sampling scheme.

Uniform sampling. Under uniform pair sampling, $X \sim \text{Binomial}(N, p)$ with $p = 1 - 1/K$, since each of the N attributes independently differs with probability $1 - 1/K$, its variance is

$$\text{Var}(X) = Np(1 - p) = \frac{N(K - 1)}{K^2}. \quad (5)$$

Substituting (5) into (4),

$$\rho_{\text{unif}}(X, Y) = \frac{a\sqrt{N(K - 1)/K^2}}{\sqrt{a^2 \cdot N(K - 1)/K^2 + \sigma_\varepsilon^2}}. \quad (6)$$

This quantity depends explicitly on K . As K grows, $\text{Var}(X) \rightarrow 0$, and $\rho_{\text{unif}} \rightarrow 0$ even though the underlying language (encoded in a and σ_ε^2) is unchanged. Classical TopSim thus degrades with K as a matter of arithmetic, not as a reflection of language structure.

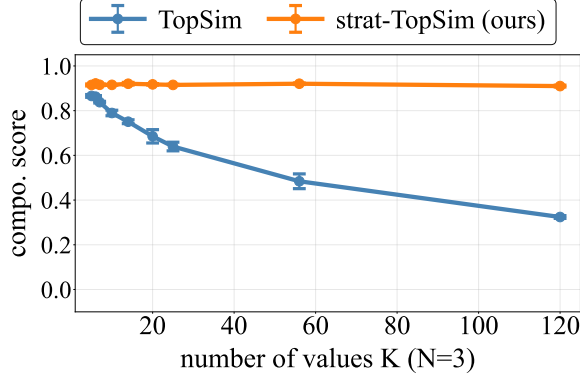


Figure 5: Same experiment as Figure 2 but using Pearson correlation, on the English adjectival configuration of Section 3. The K -dependence of classical TopSim and the stability of strat-TopSim hold under both Pearson and Spearman.

Stratified sampling. Under stratified sampling (Algorithm 1), $X \sim \text{Uniform}\{1, \dots, N\}$ by construction, independently of K . Its variance is

$$\text{Var}(X) = \frac{N^2 - 1}{12}. \quad (7)$$

Substituting (7) into (4),

$$\rho_{\text{strat}}(X, Y) = \frac{a\sqrt{(N^2 - 1)/12}}{\sqrt{a^2(N^2 - 1)/12 + \sigma_\epsilon^2}}. \quad (8)$$

This quantity depends only on a , σ_ϵ^2 and N (all properties of the language and its underlying structure) and is fully independent of K . Strat-TopSim therefore measures a property of the language itself rather than an artifact of the input space size.

Empirical confirmation across correlation choices. The analytical result above is established for Pearson. Figure 5 reports the same experiment as Figure 2 (vocabulary scan on a fully compositional natural-language code, $N=3$) using Pearson correlation rather than Spearman. The pattern is identical: classical TopSim with Pearson degrades from 0.81 to 0.33 as K grows, while strat-TopSim remains stable at 0.91. This empirically confirms both (i) the analytical Pearson prediction and (ii) that the same pathology affects Spearman in practice.

Note on Spearman. Spearman correlation operates on ranks of X and Y . Under binomial X , the rank distribution depends on K in a non-trivial way that admits no analogous closed-form analysis. Our use of Pearson above is therefore a matter of tractability rather than preference: Spearman

captures monotonic rather than linear relationships, which is what TopSim is after, and we retain it throughout the paper.

B Weighted TopSim

When messages cannot be obtained for arbitrary objects, for instance when only a pre-existing set of object-message pairs is available (as in fixed train/test splits), stratified sampling at controlled Hamming distances is not feasible. We describe here a weighted variant of TopSim that addresses the same sampling bias in this setting.

The idea is to re-weight existing pairs so that each Hamming distance class contributes equally to the correlation, regardless of how many pairs fall in it. Given N_{pairs} pairs of objects and messages, we count the number of pairs c_h in each Hamming distance class h , and assign each pair a weight $w_i = 1/c_{h_i}$, where h_i is the Hamming distance class of pair i . We then compute the weighted Spearman correlation between object distances and message distances. Unlike its unweighted counterpart, it has no single definition. We use the weighted Spearman formula derived from Bailey et al. (2018): we compute the weighted Spearman correlation as a weighted Pearson correlation on weighted ranks. For a distance vector δ with weights w , the weighted rank of element j is

$$r_j(\delta) = \sum_i w_i \mathbf{1}(\delta_i < \delta_j) + \frac{1}{2} \sum_i w_i \mathbf{1}(\delta_i = \delta_j), \quad (9)$$

that is, the total weight strictly below δ_j plus half the weight tied at δ_j (the weighted mid-rank). Object and message distances are ranked this way, and the weighted Pearson correlation of the two rank vectors is returned.

The full procedure is given in Algorithm 2.

Algorithm 2 Weighted TopSim

From N_{pairs} pairs $\{(o_i, o'_i, m(o_i), m(o'_i))\}$
 Compute $\delta_i^{\text{obj}} = d_{\text{ham}}(o_i, o'_i)$ for all i
 Compute $\delta_i^{\text{msg}} = d_{\text{lev}}(m(o_i), m(o'_i))$ for all i
for each unique Hamming distance h **do**
 $c_h \leftarrow |\{i : \delta_i^{\text{obj}} = h\}|$
 $w_i \leftarrow 1/c_h$ for all i with $\delta_i^{\text{obj}} = h$
end for
return weighted-Spearman($\delta^{\text{obj}}, \delta^{\text{msg}}, \mathbf{w}$)

Weighted TopSim does not require obtaining messages for new objects and can be applied to any set of object-message pairs. Its reliability, however, depends on every Hamming distance class being populated, and this is limited by the very imbalance the method corrects: as K grows, pairs at small Hamming distance become increasingly rare under uniform sampling, so a fixed number of sampled pairs eventually contains none of them. Re-weighting cannot recover a class that was never sampled (it can only balance the classes that happen to occur), so in that regime the metric silently measures a different quantity. The number of sampled pairs must therefore grow with K , whereas stratified TopSim reaches the same balance at a budget independent of K .

Empirical validation. Figure 6 reports a vocabulary scan analogous to Figure 2 on a fully compositional natural-language code ($N=3$), at two pair budgets. The two budgets concern weighted TopSim; classical TopSim is reported at $N_{\text{pairs}}=1,000$ throughout, as it does not benefit from a larger budget: drawing more pairs leaves the proportion of each Hamming distance class unchanged, since that proportion is a property of the input space rather than of the sample (Section 4), so the imbalance persists at any budget. Over a $25\times$ increase, its value moves by at most 0.016. At the budget used elsewhere in this paper ($N_{\text{pairs}}=1,000$), weighted TopSim matches strat-TopSim up to $K=25$ (0.91–0.92) and then degrades, falling to 0.85 at $K=56$ and 0.82 at $K=120$, with the standard deviation rising from below 0.01 to about 0.07. This degradation reflects the budget rather than the estimator: at $K=120$ most 1,000-pair samples contain no pair at Hamming distance 1 at all, so the re-weighting balances two classes instead of three. Raising the budget to $N_{\text{pairs}}=25,000$ repopulates that class and removes the effect: weighted TopSim is then flat at 0.913–0.918 across the entire range with standard deviation below 0.012, agreeing with strat-TopSim (0.914–0.921) to within 0.005, while classical TopSim collapses from 0.85 to 0.24. Weighted TopSim thus recovers the same measurement as strat-TopSim without requiring messages for new objects, at a sampling cost that grows with K rather than remaining fixed.

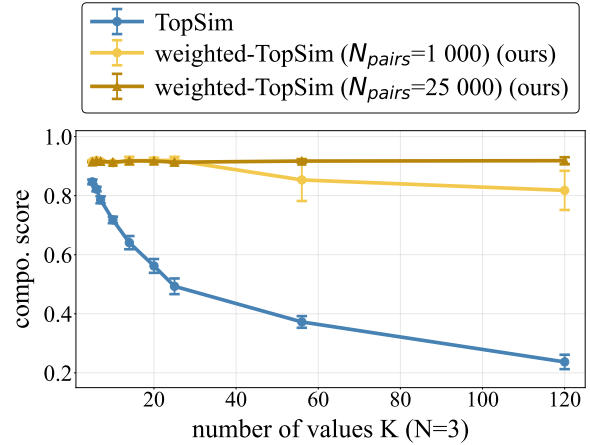


Figure 6: Vocabulary scan comparing classical TopSim and weighted TopSim on a fully compositional natural-language code ($N=3$; the English adjectival configuration of Section 3), at two pair budgets. At $N_{\text{pairs}}=1,000$ the low-distance classes become too sparsely sampled as K grows and weighted TopSim degrades; at $N_{\text{pairs}}=25,000$ it is stable across the full range.

C Robustness to the Number of Attributes

Section 4 reported the stability of strat-TopSim as the number of values per attribute K varies. We complement this here by varying the number of attributes N at fixed $K=20$. Figure 7 shows that classical TopSim plateaus around 0.60 across $N \in \{2, 3, 4, 5, 6\}$, while strat-TopSim consistently scores higher (0.85–0.95) and increases mildly with N (as more distance classes become available for stratification). Across all values of N , strat-TopSim more accurately reflects the fully compositional structure of the underlying natural-language.

D Case Study: Experimental Details

Game and agents. We use a Lewis reconstruction game (Lewis, 1969): the Speaker observes an object $o = (k_1, k_2, k_3)$ and emits a message; the Listener consumes the message and reconstructs the three attribute values. Both agents are single-layer LSTMs (Hochreiter and Schmidhuber, 1997) with layer normalization (Ba et al., 2016). The Speaker has a hidden size of 320 and an input embedding of dimension 20; the message is generated symbol by symbol from a vocabulary of size $V=10$, including an EOS token, with a maximum length of 10. The Listener has a hidden size of 150 and a token embedding of dimension 15. At test time, symbols are selected greedily (argmax).

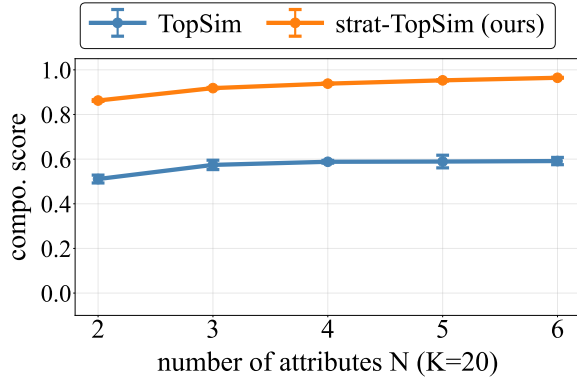


Figure 7: TopSim and strat-TopSim as a function of the number of attributes N , on a fully compositional natural-language code ($K=20$; the English adjectival configuration of Section 3, extended to N attributes). Strat-TopSim remains substantially above classical TopSim across the range.

Optimization. The communication channel is discrete, so we use a hybrid optimization scheme. The Speaker is trained with REINFORCE (Williams, 1992), with an entropy regularization coefficient of 1.0. Gradient variance is reduced via a batch-normalization baseline (Sutton et al., 1999). The Listener is trained with cross-entropy loss and an entropy regularization coefficient of 0.2. Both agents use Adam (Kingma and Ba, 2014) with a learning rate of 0.001. To avoid overfitting, we adopt the *early stopping listener* procedure of Rita et al. (2022), in which the Listener is periodically re-initialized and retrained from scratch until an early-stopping criterion is met on the IND-validation set. The criterion therefore applies per Listener retraining episode, not to the run as a whole: the run itself has a fixed budget of 30 epochs in every setting, so that runs at different K are trained for the same number of Speaker updates. Agents are trained with a batch size of 4096 and 64 gradient steps per epoch.

Input space and OOD split. We use $N=3$ attributes with $K \in \{10, 20, 50, 100\}$ values, and construct two parallel splits. Auersperger and Pecina (2022) introduce an OOD evaluation for emergent communication in a setting of $N=2$ attributes and $K=100$ values: they designate the value 0 as a held-out symbol, expose the model to objects that contain at least one 0 and whose remaining attribute value is at most 1, and evaluate on all remaining zero-containing tuples. We extend this construction to arbitrary N and K via the threshold T : among tuples containing at least one

zero, those whose remaining attribute values all lie in $\{0, \dots, T\}$ form the OOD-train subset, and all other zero-containing tuples form the OOD-test set. We set $T = 1, 2, 5, 10$ for $K = 10, 20, 50, 100$ respectively, keeping the OOD-train/OOD-test ratio approximately constant across settings (in the range 1–3%/97–99%). The IND split operates on the subspace of tuples with no zero (i.e. all attribute values in $\{1, \dots, K-1\}$). These are partitioned into IND-train, IND-validation, and IND-test sets with proportions 0.5/0.25/0.25. Agents are trained on the union of OOD-train and IND-train, so the model sees the held-out symbol only in the small set of low-value contexts defined by T . At evaluation, OOD test accuracy is computed on OOD-test, IND test accuracy on IND-test.

Training and evaluation. For each of the four settings, we run 5 independent trainings. We stop all runs at epoch 30, at which point every run reaches at least 99% IND test accuracy. TopSim and strat-TopSim are then computed by sampling object pairs and generating their associated messages greedily (argmax). OOD test accuracy is computed on the held-out object set described above. We report mean and standard deviation across the 5 seeds for each metric.

Code and reproducibility. For reproducibility, the code is available at <https://github.com/jbsevestre/stratified-topographic-similarity>. It was originally derived from the EGG library (Kharitonov et al., 2019). Since the EGG repository has been archived, we release the implementation standalone rather than contributing it upstream. The release contains the implementation of stratified and weighted TopSim, the agents and training loop used in this case study, the benchmark word lists of Section 3, and the configuration files and seeds needed to reproduce every figure and number reported in the paper.

Compute. Each training run takes approximately 2 hours on a single NVIDIA H100 GPU; the full experiment (4 settings $\times$ 5 seeds) totals around 40 GPU-hours.