INFORMATION THEORY AND ITS USES IN COMPUTER SCIENCE

Final Project: Topic Ideas

November 3, 2026

Information theory offers a common language for questions about secrecy, randomness, computation, coding, and learning. The topics below are invitations to explore those connections. Each heading describes a subarea, with notable results and questions that could motivate a project. Most headings are broader than a single project; choose one construction, theorem, model comparison, or small family of examples to study closely.

Open questions provide motivation. A successful project can explain a known result and its proof idea, compare two approaches, or analyze a restricted case. In the list, *Open question* marks a specific unresolved target; *Direction* suggests a line of investigation without asserting that every instance is unsolved; and *Entry point* highlights interesting known results. The readings are suggestions, rather than a requirement to read every paper in full. Open-problem formulations and status notes were checked against the linked literature during November 2–3, 2026.

Contents

A Randomness and cryptographic foundations **3**
1 Two-source extractors with negligible error . . . 3
2 Privacy amplification with an active adversary . . . 3
3 Computational entropy and leakage . . . 4
4 Metacomplexity: how hard is it to recognize a simple object? . . . 4
5 One-way functions versus key agreement . . . 5

B Secret sharing and secure computation **6**
6 Secret sharing: access structures and share size . . . 6
7 Can we recognize a cheap secret-sharing problem? . . . 7
8 Private simultaneous messages (PSM) . . . 7
9 Conditional disclosure of secrets (CDS) . . . 8
10 Randomized encodings: reveal an answer through a simpler computation . . . 8
11 Which functions admit MPC without an honest majority? . . . 9
12 Secure sampling: which correlations can parties create privately? . . . 9
13 Communication and randomness in secure computation . . . 10
14 Oblivious transfer from noisy channels . . . 10

C Private access and hidden networks **11**
15 Private information retrieval and locally decodable codes . . . 11
16 Oblivious RAM and the price of hiding an access history . . . 11
17 Information-theoretic topology-hiding broadcast . . . 12

D Communication and computational complexity **12**
18 The log-rank conjecture . . . 12
19 Interactive compression: how much of a conversation is new? . . . 13
20 Parallel repetition beyond the basic theorem . . . 13
21 Understanding logspace derandomization . . . 14
22 Streaming algorithms against adaptive adversaries . . . 14
23 Converting one correlation into another without communication . . . 15

E Compression algorithms and distributed sources **16**
24 Lempel–Ziv: universality and robustness . . . 16
25 Reversible preprocessing and the Burrows–Wheeler transform . . . 17
26 Grammar compression: how far is greedy from optimal? . . . 18
27 Distributed compression and computation . . . 18

F Sparse recovery and group testing **19**
28 Compressed sensing: geometry, graphs, and sparse recovery . . . 19
29 Sparse Walsh–Hadamard transforms . . . 20
30 Group testing: finding a sparse set using pooled tests . . . 21

G Codes, noise, and computation **22**
31 List decoding and cryptographic proof systems . . . 22
32 Quantum error-correcting codes: sparse checks and local testability . . . 23
33 Error correction with a tiny streaming decoder . . . 24
34 Tree codes: error correction before the message is known . . . 24
35 Robust Gray codes: smooth representations that survive noise . . . 25
36 The binary deletion channel . . . 25
37 Zero-error capacity and graph powers . . . 26
38 Non-malleable codes: protecting meaning against tampering . . . 26
39 Efficient wiretap coding against an active adversary . . . 26
40 Computing on encoded data with little download . . . 27

H Entropy, learning, and privacy **27**
41 Entropy inequalities as a language for combinatorial lower bounds . . . 27
42 Compression, information, and generalization . . . 28
43 Differential privacy, fingerprinting, and computational hardness . . . 28

A. Randomness and cryptographic foundations

1. Two-source extractors with negligible error

An extractor turns imperfect randomness into nearly uniform bits. A single arbitrary weak source cannot be purified deterministically, but two independent sources can. The challenge is to give one efficiently computable function that works for *every* pair of sources with enough min-entropy. This is also a combinatorial construction problem: a one-bit extractor can be viewed as a coloring that is approximately balanced on every sufficiently large rectangle. Li's 2023 construction reaches entropy $O(\log n)$ for constant error. Negligible error is a different requirement: one error function $\varepsilon(n) = n^{-\omega(1)}$ must eventually beat every inverse polynomial.

- **Direction.** How low can the min-entropy be with negligible error? A concrete benchmark is a uniform deterministic polynomial-time $\mathrm{Ext} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ that works for every independent pair with $H_\infty(X), H_\infty(Y) \geq \alpha n$, for a fixed $0 < \alpha < 0.1$, and error $n^{-\omega(1)}$. Determine the current threshold and seek lower entropy, ideally sublinear or polylogarithmic.
- **Entry point.** Li's extractor construction also yields explicit Ramsey graphs whose largest clique or independent set has only polylogarithmic size. This turns the purification of weak randomness into a concrete combinatorial object previously supplied by the probabilistic method.

Reading: Li, *Two Source Extractors for Asymptotically Optimal Entropy, and (Many) More*, Theorem 7.6 and Section 8. *Status note:* A public Goodman–Doron announcement claims low-error extraction for $k = o(n)$, but does not specify its error bound. Check its full parameters before treating the $0.1n$ benchmark as open.

2. Privacy amplification with an active adversary

Alice and Bob share a weak secret about which an eavesdropper has partial information. Universal hashing can turn it into a key when the public seed is honestly transmitted. An active adversary can instead change the seed or other public messages. Non-malleable extractors address this problem by making the key look random even in the presence of an output computed using a tampered seed. This is a useful example of how a small change in the adversary's powers changes the required randomness object.

- **Entry point.** Universal hashing extracts nearly all of a weak secret's usable min-entropy even when the seed is public. Li's non-malleable extractors yield two-round active protocols with entropy loss $O(\log\log n + s)$ and communication $O(\log n + s)$ for security parameter $s \leq \alpha k$, for an absolute $\alpha > 0$. Two-round protocols secure against quantum side information are also known.
- **Direction.** Compare the entropy loss, number of rounds, and security parameter of active privacy-amplification protocols. Which guarantees already have optimal classical constructions, and what losses occur in known quantum-secure protocols? For repeated use, specify the number of sessions, adaptive tampering, and what earlier executions expose about the secret.

Reading: Li's extractor paper, Theorem 7.19; Boddu–Jain–Kapshikar, *Quantum secure non-malleable-extractors*, for the quantum comparison.

3. Computational entropy and leakage

A pseudorandom generator can stretch a short seed into a string that has little actual entropy but looks uniform to every efficient observer. Computational entropy formalizes this gap between what exists and what can be detected or predicted. Different definitions quantify over distinguishers and high-entropy witness distributions in different orders. These distinctions matter when a secret is used repeatedly or auxiliary information leaks: familiar Shannon entropy manipulations do not automatically transfer to computational entropy.

- **Entry point.** The Håstad–Impagliazzo–Levin–Luby theorem constructs pseudorandom generators from arbitrary one-way functions. Computational entropy is a key bridge in the proof: a source can have more apparent randomness than an efficient procedure can actually access.
- **Direction.** Compare known leakage chain rules for metric, HILL, and relaxed conditional entropy. Which hypotheses permit an ℓ-bit entropy loss, and what happens to advantage and computational cost? Under one-way permutations, the naive general rule fails for standard conditional HILL entropy. How does a simulator for one test differ from one simulator that fools an entire class? A 2026 revision strengthens leakage simulation and makes the order of quantifiers explicit.

Reading: Håstad–Impagliazzo–Levin–Luby, *A Pseudorandom Generator from Any One-Way Function*; Fuller–Reyzin, *Computational Entropy and Information Leakage*; Krenn–Pietrzak–Wadia's conditional-HILL counterexample; Hu–Vadhan, *Generalized and Unified Equivalences between Hardness and Pseudoentropy*, Sections 1.1–1.3 (July 2026 revision).

4. Metacomplexity: how hard is it to recognize a simple object?

Kolmogorov complexity measures the length of a shortest description of a string; time-bounded versions also limit how long the description may run. The minimum circuit size problem (MCSP) asks an analogous question for a Boolean function given by its full truth table. These problems connect compression to cryptography: suitable average-case hardness of time-bounded Kolmogorov complexity characterizes the existence of one-way functions. The precise time bound, approximation gap, and input distribution are essential to that connection.

- **Open question.** Is ordinary MCSP NP-hard under deterministic polynomial-time many-one reductions? The input is the full truth table and a circuit-size threshold; running time is measured in that input length.
- **Entry point.** Liu and Pass characterize one-way functions by mild average-case hardness of time-bounded Kolmogorov complexity on uniformly random strings, for a suitable polynomial time bound. Thus a problem about recognizing short descriptions captures the existence of a foundational cryptographic primitive.

Reading: Liu–Pass, *On One-way Functions and Kolmogorov Complexity*; the introduction to ECCC TR26-091 distinguishes ordinary MCSP from implicit variants.

5. One-way functions versus key agreement

One-way functions support pseudorandom generation and private-key encryption. Key agreement asks two parties with independent private coins to create a shared secret by talking over a public channel. Understanding whether one-wayness suffices exposes the boundary between private-key and public-key cryptography. Interactive Kolmogorov complexity gives a way to formulate this boundary as a comparison between closely related computational problems.

- **Open question.** Does the existence of classical one-way functions imply secure classical key agreement over an authenticated public channel, with polynomial-time honest parties and security against passive polynomial-time eavesdroppers?
- **Entry point.** Impagliazzo and Rudich give an oracle world with one-way permutations but no secure key agreement, ruling out broad black-box approaches. Ball–Liu–Mazor–Pass give a complementary characterization: for a suitable polynomial time bound, worst-case hardness of a promise problem about interactive Kolmogorov complexity is equivalent to key agreement. Changing the parameters gives an OWF characterization, recasting the implication as a reduction between related promise problems.
- **Direction.** What does the interactive description measure capture that an ordinary short description misses? Where would a reduction between the two promise problems have to use their different parameters?

Reading: Impagliazzo–Rudich, *Limits on the Provable Consequences of One-way Permutations*; Ball–Liu–Mazor–Pass, *Kolmogorov Comes to Cryptomania: On Interactive Kolmogorov Complexity and Key-Agreement*.

B. Secret sharing and secure computation

6. Secret sharing: access structures and share size

In threshold secret sharing, any sufficiently large coalition can recover the secret and every smaller coalition learns nothing. More generally, an access structure specifies an arbitrary monotone family of authorized coalitions. Every such family can be realized, but the shares may be enormous. Entropy inequalities give lower bounds on share size, while linear algebra supplies many constructions. Small, structured access structures can already leave a gap between the best scheme and the best lower bound. Here is a concrete example with only seven participants.

- **Open question.** Find a family of n-party access structures with polynomial-time decidable coalition membership for which every perfectly correct, perfectly private scheme for a one-bit secret has some share of superpolynomial bit length. Allow arbitrary nonlinear schemes and measure a share by the logarithm of its alphabet size.
- **Open question.** Place seven players on a circle and authorize a coalition exactly when it contains five consecutive players. Determine the infimum $c(5,7)$ of $\max_i \log|S_i|/\log|S|$ over perfectly correct, perfectly private schemes, allowing arbitrary finite secret alphabet S, share alphabets S_i, and nonlinear schemes. Known bounds are $3/2 \le c(5,7) \le 13/8$.
- **Entry point.** Shamir's polynomial construction realizes every threshold access structure with one field element per share and perfect privacy. For arbitrary access structures and one-bit secrets, recent worst-case constructions use $2^{n/2+O(1)}$ bits per share in linear schemes and $2^{0.494n+o(n)}$ in nonlinear schemes. Together with the matching worst-case linear lower bound, these constructions separate the two models.
- **Direction.** Recent exponential lower bounds apply to explicit access structures within a broad family capturing known general constructions. Which restriction prevents them from extending to arbitrary nonlinear sharing? Compare the CDS-gate formula model with unrestricted schemes.
- **Direction.** What happens when an adversary learns one chosen bit of every Shamir share? Sun–Wootters give new prime-field leakage-resilience results using polynomial intersection and list recovery. Compare this leakage model with learning a coalition of complete shares.

Reading: The introduction to Applebaum–Nir for general share complexity; Nir, *Resolving the Worst-Case Complexity of Linear Secret Sharing*, September revision; Herranz–Sáez, *(k, n)-Consecutive access structures*, Sections 4.3–6; Beimel–Nir, *The Power of Multislices in Monotone Computation*, for the restricted lower bounds; Sun–Wootters, *Improved Local Leakage Resilience of Shamir Secret Sharing and Worst-Case Optimal Polynomial Intersection* (October 2026 preprint).

7. Can we recognize a cheap secret-sharing problem?

Suppose an access structure has a very efficient secret-sharing scheme. Can an algorithm find it, or even recognize that an efficient scheme exists? This question combines cryptography with metacomplexity. The input representation matters enormously: a graph, monotone formula, monotone circuit, and full truth table present different computational tasks. Graph access structures give a particularly tangible setting in which graph covers help describe linear sharing complexity. In a bipartite graph, call a *layer* a subgraph that is a vertex-disjoint union of complete bipartite graphs. Equivalently, a layer has no induced four-vertex path, P_4.

- **Open question.** Is there a polynomial-time constant-factor approximation to the minimum number of layers whose union covers every edge of an explicitly given bipartite graph? Layers may overlap with one another, and each must use only edges of the original graph.
- **Entry point.** Recognizing cheap sharing can itself be hard: for monotone formulas, distinguishing one-bit shares from much larger necessary shares is coNP-hard. For bipartite graph access structures, combinatorial covers instead provide a structural description of constant-size linear sharing over $\mathbb{F}_2$.

Reading: Applebaum–Nir, *The Meta-Complexity of Secret Sharing*, Section 2.5, Question 2, and Section 6 of the revised version.

8. Private simultaneous messages (PSM)

In a private simultaneous-messages (PSM) protocol, senders use shared randomness hidden from a referee, send one message each, and let the referee recover a function of their inputs. Perfect privacy means that inputs with the same output induce identical transcript distributions. This is a minimal model of secure computation, closely related to decomposable randomized encodings. The shared seed is a resource: hiding it from the referee enables privacy, and charging for it creates a separate communication–randomness tradeoff.

- **Entry point.** Two senders can reveal XOR using one bit each and a shared hidden pad, while revealing nothing else to the referee. At the other extreme, Ball et al. prove that computing OR of many one-bit inputs privately requires superlinear total communication, using a connection to Ramsey theory and Higman's lemma.
- **Open question.** Find a uniformly polynomial-time computable family of total Boolean functions on two k-bit inputs requiring superpolynomial communication in classical two-sender PSM with perfect correctness and privacy. Shared hidden randomness is free; charge the logarithms of the two message-alphabet sizes.
- **Open question.** What is the exact total communication cost of OR with n senders holding one bit each, perfect correctness and privacy, and free hidden shared randomness? The known bounds are $\omega(n)$ and $O(n \log n)$. Is the latter optimal? Compare this with the weaker requirement that transcript supports, rather than distributions, agree.

Reading: Applebaum–Holenstein–Mishra–Shayevitz, *The Communication Complexity of Private Simultaneous Messages, Revisited*; Ball et al., *On the Complexity of Symmetric Boolean Functions in Multiparty Private Simultaneous Messages*.

9. Conditional disclosure of secrets (CDS)

Alice holds (x, s), Bob holds y, and they share randomness hidden from a referee. The referee already knows (x, y) and receives one message from each sender. It should recover the secret s when a public predicate $h(x, y)$ is true, while the joint messages reveal nothing about s when the predicate is false. Thus CDS hides an extra secret on rejecting inputs; PSM instead hides the inputs except for their function value. These related models connect communication complexity, secret sharing, and randomized encodings.

- **Entry point.** Equality over $\mathbb{F}_q$ has a two-field-element CDS: send $A = t + rx + s$ and $B = t + ry$ using independent uniform r, t. If $x = y$, their difference reveals s; otherwise the complete pair is uniform and hides the secret perfectly.
- **Open question.** Must CDS for inner product $h(x, y) = \sum_{i=1}^{k} x_i y_i \bmod 2$ require $k^{\Omega(1)}$ shared random bits when both correctness and privacy allow a sufficiently small constant error? Use a one-bit secret, measure privacy by total variation between transcripts for the two secret values on rejecting inputs, and charge $\log |\operatorname{supp} R|$. Communication and private randomness are unrestricted. Known robust lower bounds are only logarithmic.
- **Direction.** Compare communication cost with shared-randomness cost. Linear shared-randomness lower bounds for inner product are already known when either correctness or privacy is perfect. Which step loses this strength when both requirements allow error?

Reading: Applebaum et al., *Conditional Disclosure of Secrets: Amplification, Closure, Amortization, Lower-bounds, and Separations*; Hasegawa–Mataraarachchi, *New lower bounds for CDS and f-routing*, Section 5 of version 2.

10. Randomized encodings: reveal an answer through a simpler computation

A randomized encoding $\widehat{f}(x; R)$ lets a decoder recover $f(x)$, while its distribution can be simulated from $f(x)$ alone. The point is to make the encoder simpler or more local than the original computation; otherwise the encoding $f(x)$ itself is trivial. Decomposable encodings split the encoding into pieces depending on separate inputs and shared randomness, which is precisely the PSM constraint. Randomized encodings also organize the ideas behind garbled circuits and secure computation.

- **Entry point.** Perfect PSM protocols and perfect two-decomposable randomized encodings are exactly the same objects. A recent result goes further in a different model: statistically secure *additive* encodings exist for every finite function, with efficient constructions for NC^1 or NL. The evaluator learns from the sum of locally generated group elements, connecting randomized encodings to anonymous communication.
- **Direction.** Study a randomized encoding for a branching program or a garbled circuit. Which guarantees are perfect, statistical, or computational? Track encoder locality, encoding length, decoding time, and randomness, and explain which resource makes the result nontrivial.
- **Direction.** For additive encodings, what changes if correctness must be perfect, or privacy must survive the exposure of some parties' coins? Compare these requirements with ordinary statistical security.

Reading: Feige–Kilian–Naor, *A Minimal Model for Secure Computation*; Applebaum, *Garbled Circuits as Randomized Encodings of Functions: a Primer*; Bitansky et al., *Shuffling is Universal*, for additive encodings.

11. Which functions admit MPC without an honest majority?

General-purpose information-theoretic multiparty computation can exploit an honest majority, but losing that majority does not make every function impossible. Some outputs already permit all the information a protocol needs to disclose. The interesting question is to characterize the feasible functions. Start with finite deterministic functionalities, independent private coins, private channels, no correlated setup, and perfect security against passive corruptions: corrupted parties follow the protocol and pool their complete views.

- **Entry point.** Kushilevitz's matrix characterization identifies exactly which two-party common-output functions admit perfect private computation in this model. XOR is feasible and AND is not. The distinction comes from what the output already reveals, rather than simply from the size of the function.
- **Direction.** Recover the known characterization for Boolean common-output functions under a threshold adversary, then compare it with non-Boolean outputs or a chosen family of corrupt coalitions. Keep private-channel results distinct from authenticated-channel models where the adversary observes messages. What changes if computational assumptions or correlated setup are permitted?

Reading: Kushilevitz, *Privacy and Communication Complexity*; Chor–Kushilevitz, *A Zero-One Law for Boolean Privacy*.

12. Secure sampling: which correlations can parties create privately?

Even a protocol with no inputs can face a privacy barrier: the parties must create a joint distribution while each coalition's full view reveals no more than its own outputs. This separates ordinary correlation generation from secure sampling and connects MPC feasibility to common information.

- **Entry point.** A two-party distribution is securely samplable exactly when its dependence can be explained by a common variable determined by either output, with conditional independence given that variable. Recent work proves statistical sufficiency for four parties with one-bit outputs: checking every bipartition is enough.
- **Open question.** Does the bipartition criterion remain sufficient for five parties, each receiving one bit? Every two-group distribution obtained by partitioning the parties must be securely samplable. Require a finite protocol for every desired statistical correctness and privacy error $\varepsilon > 0$, independent private coins, private channels, no setup, and security against every passive corrupt coalition. Local computation and communication are unrestricted.

Reading: Anilkumar–Jain–Prabhakaran–Prabhakaran, *Towards Characterizing Secure Samplability*, Theorem 1 and Definition 3.

13. Communication and randomness in secure computation

Perfect privacy requires each party's complete view to be simulatable from its own input and prescribed output. The resource question is how much communication and randomness a protocol needs while satisfying that requirement. A concrete starting model has Alice holding x, Bob holding y, and Charlie receiving $f(x, y)$, with private pairwise channels, independent local coins, no setup, perfect correctness, and privacy against any single passive corruption.

- **Entry point.** Information-theoretic techniques give tight communication and randomness lower bounds for several three-party functions. Some admit communication-ideal protocols: one protocol simultaneously minimizes communication on every link and also uses the minimum randomness.
- **Direction.** For a chosen function, what is the minimum total communication of a perfectly private protocol? Arithmetic sum gives a concrete case to study. Fix the input-support or distribution convention, and distinguish transcript entropy from expected and worst-case bits. Can one protocol attain the separate lower bounds on all three links?
- **Direction.** Is there a protocol simultaneously attaining the minimum communication and randomness costs? For the cited paper, the randomness measure is $H(T_{12}, T_{23}, T_{31}, Z \mid X, Y)$, where T_{ij} are link transcripts and Z is Charlie's output. Compare one execution with many, retaining perfect privacy at each block length; asymptotic statistical security is a separate relaxation.

Reading: Data–Prabhakaran–Prabhakaran, *Communication and Randomness Lower Bounds for Secure Computation*. For the separate PSM model, see the papers linked in the PSM entry.

14. Oblivious transfer from noisy channels

Channel noise can be a cryptographic resource. In string oblivious transfer, Alice holds two independent uniform ℓ-bit strings and Bob holds an independent uniform choice bit. Bob should learn the chosen string while learning essentially nothing about the other, and Alice should learn essentially nothing about his choice. A noisy channel can supply the asymmetry of knowledge that ordinary noiseless communication cannot create.

- **Entry point.** Erasure channels give a particularly clear construction: Bob knows which symbols arrived, but Alice does not. For a binary symmetric channel, polarization and interactive communication can emulate useful erasure-like channels and improve known OT rates. Thus a channel-coding transform becomes a tool for secure computation.
- **Open question.** Determine the OT capacity of the binary symmetric channel with fixed crossover probability $0 < q < 1/2$. Maximize ℓ/n using n memoryless channel uses from Alice to Bob, unlimited interleaved noiseless public discussion, independent private coins, and no setup. Require decoding error and total-variation privacy errors to vanish against passive corruption of either party. There is no computational restriction. A more focused project can compare lower bounds at one value of q.

Reading: Suda–Watanabe, *An Improved Lower Bound on Oblivious Transfer Capacity Using Polarization and Interaction*, Sections II–VI.

C. Private access and hidden networks

15. Private information retrieval and locally decodable codes

Private information retrieval (PIR) hides which database entry a client reads. Multiple noncolluding servers permit information-theoretic privacy, and the queries can be interpreted as local decoding of an encoded database. This creates a bridge between cryptographic protocols and codes that recover one message bit by inspecting only a few codeword positions. Query count, alphabet size, noise tolerance, server count, and communication cost must all be tracked to make the bridge precise.

- **Open question.** How short can a binary three-query locally decodable code be? Fix positive constants for adversarial error tolerance and decoding advantage, require recovery of every message bit, and seek the optimal codeword length as a function of message length. A major gap remains.
- **Entry point.** Two noncolluding servers can return a requested bit using only two downloaded bits, with long queries. Perfect one-server PIR, in contrast, must download the whole database in the standard one-round model. Locally decodable codes turn coding constructions into private-retrieval protocols.

Reading: Chor et al., *Private Information Retrieval*; Basu et al., *Improved Lower Bounds for all Odd-Query Locally Decodable Codes* for the lower-bound frontier; Gupte–Ragavan, *Exponentially Fewer-Server PIR from Sparser S-Decoding Polynomials*, for recent server–communication tradeoffs. Its general constant-server statement is conditional, with unconditional cases for at most 15 servers. Janzer–Manohar's November 2026 linear three-query LCC result concerns correction of codeword coordinates and does not settle the general LDC question.

16. Oblivious RAM and the price of hiding an access history

Encrypting data hides its contents but can leave the addresses of reads and writes exposed. Oblivious RAM (ORAM) hides those access patterns while simulating ordinary memory. The information-transfer viewpoint proves strong lower bounds for online ORAM, where each request must be answered before the next arrives. If the whole request sequence is known in advance, some familiar arguments disappear, and unrestricted lower bounds encounter a striking connection with circuit complexity.

- **Open question.** Prove an $\Omega(\log N)$ amortized bandwidth lower bound for unrestricted offline ORAM with N logical words, equal logical and physical word sizes $\Theta(\log N)$, $O(1)$ private words, and statistical access-pattern privacy and negligible correctness error. An N-word database and N read/write requests, including write values, initially reside in external memory; there is no free private input tape or oracle. Output every requested read value. Allow arbitrary encodings; measure expected probes on worst-case inputs and charge every probe, including request reads, preprocessing, and output writes.
- **Entry point.** Larsen and Nielsen prove logarithmic bandwidth overhead for online ORAM under standard parameters, even with computational privacy and arbitrary encodings. For unrestricted offline ORAM, Boyle and Naor connect strong lower bounds to difficult lower bounds for sorting circuits.

Reading: Boyle–Naor, *Is There an Oblivious RAM Lower Bound?*, Definitions 2.1–2.3 and 2.8, and Section 3.3, for the offline input model; Larsen–Nielsen, *Yes, There Is an Oblivious RAM Lower Bound!*; Thießen–Vahrenhold, Section 1.2 and Table 2, distinguish the remaining offline lower-bound question.

17. Information-theoretic topology-hiding broadcast

Broadcasting a public value can reveal sensitive information about the network. Topology-hiding broadcast (THB) delivers the value while hiding the graph beyond the corrupted parties' neighbor lists and public parameters. Use synchronous private channels, independent local coins, no setup, and static passive corruptions. The vertex set is public, and connectivity below means vertex connectivity. Seek uniform protocols with computation and communication polynomial in the number of vertices.

- **Entry point.** Ball et al. obtain statistical information-theoretic protocols for 2-connected graph classes, but their complexity can depend polynomially on the number of graphs in the class. Banoun–Boyle–Cohen give perfect protocols for structured families, including wheels and friendship graphs. These results distinguish statistical feasibility, perfect security, and efficiency for a large graph class.
- **Open question.** Is there perfectly correct, perfectly topology-hiding broadcast tolerating one passive corruption for the class of all 2-connected graphs?
- **Open question.** Is there information-theoretically secure THB tolerating two passive corruptions for the class of all 3-connected graphs? Study both perfect security and negligible statistical error.

Reading: Ball et al., *Topology-Hiding Communication from Minimal Assumptions*, Theorem 1.4 and Section 5 for the statistical one-corruption construction and its cost; Banoun–Boyle–Cohen, *Information-Theoretic Topology-Hiding Broadcast: Wheels, Stars, Friendship, and Beyond*, Theorems 1–3 for perfect protocols on structured classes.

D. Communication and computational complexity

18. The log-rank conjecture

A deterministic communication protocol partitions its input matrix into monochromatic rectangles. Matrix rank provides a beautifully short lower bound on the number of communicated bits, but how far is it from a complete description? The log-rank conjecture asks whether rank captures deterministic communication complexity up to a polynomial in its logarithm. A project can approach this large question through special matrix families or through the ways in which randomized communication uses different measures.

- **Open question.** For every total Boolean function f, is $D(f) \leq \mathrm{poly}(\log(2 + \mathrm{rank}_{\mathbb{R}}\, M_f))$? Here D is worst-case deterministic two-party communication and M_f is the full Boolean matrix.
- **Entry point.** Matrix rank gives $D(f) \geq \log \mathrm{rank}_{\mathbb{R}}\, M_f$, while the known general upper bound is $O(\sqrt{\mathrm{rank}\, M_f})$. Explicit examples have $D(f) = \Omega((\log \mathrm{rank}\, M_f)^2)$, with the extraneous logarithmic losses removed by Pabbaraju's 2026 result. A large gap to the general upper bound remains.
- **Direction.** For XOR functions $F(x, y) = f(x \oplus y)$, how does low algebraic degree help? A November 2026 result proves log-rank for degree $O(\log \log s)$, where s is the Fourier sparsity of $(-1)^f$.

Reading: Song, *Alphabet-Preserving Lifting for the Log-Rank Conjecture*; Pabbaraju, *Optimal Unambiguous DNFs and Alon-Saks-Seymour*, Corollary 4.2; Zhang's degree-sensitive parity decision-tree bound, a November 2026 preprint.

19. Interactive compression: how much of a conversation is new?

The information revealed by a transcript can be much smaller than its length. Ordinary source coding suggests compressing to that information cost, but in an interactive protocol the next-message distribution depends on a private view. Information complexity equals an appropriate amortized communication rate over many copies, yet general single-copy compression can be much more expensive. This contrast makes the subject a useful test of how far the compression intuition extends beyond a single sender.

- **Entry point.** Braverman and Rao show that internal information complexity equals an appropriate amortized communication rate. Ganor, Kol, and Raz exhibit communication tasks with an exponential single-copy gap. Low information therefore has a precise operational meaning that can require amortization.
- **Direction.** Compare one-message, bounded-round, and amortized compression. Study an exponential separation between information and communication, and identify which restrictions allow substantially better guarantees. Distinguish error on each coordinate from error on the entire block, and expected from worst-case communication. Recent global-error amortization results show why these conventions affect the limiting rate.

Reading: Braverman–Rao, *Information Equals Amortized Communication*; Ganor–Kol–Raz, *Exponential Separation of Information and Communication*; Suruga's global-error amortization results (2026 preprint).

20. Parallel repetition beyond the basic theorem

Parallel repetition will be covered in class; projects can build on it through newer proofs, more players, entanglement, or different winning criteria. The central difficulty persists: independent questions do not force a strategy to act independently across coordinates. The useful information budget depends on the players' resources and on what counts as success.

- **Entry point.** Recent work gives an analytic approach through CSP inverse theorems, covering new multiplayer cases and reproving several known repetition theorems with weaker rates. A new preprint gives exponential repetition for classical multiplayer XOR games. For classical three-player games with GHZ-supported questions, a 2026 result obtains $\exp(-n^c)$ decay for some $c > 0$. Recent quantum results give exponential repetition and threshold bounds for finite two-player entangled games.
- **Direction.** How does the CSP inverse-theorem approach differ from an information-theoretic proof? Which assumptions on the question distribution permit stronger decay?
- **Direction.** What changes when players share entanglement, or when success means winning a fraction above the one-copy value? Track the gap, answer alphabet, threshold, and finite-dimensional strategy model.
- **Direction.** Understand the recent theorem identifying the communication threshold for repetition of finite total relations in terms of zero-error internal information complexity. Track its input-distribution, worst-case-communication, and global-success conventions.

Reading: Bhangale et al., *An Analytical Approach to Parallel Repetition via CSP Inverse Theorems*; Liu–Lovett–Mittal, *Improved Parallel Repetition for GHZ-Supported Games via Spreadness*; Jain, *Parallel Repetition for Entangled Games via Conditional Relative Entropy*; Braverman–Minzer, *Parallel Repetition for k-Player XOR Games*; Suruga, *The Exact Communication Threshold for Parallel Repetition*. The latter three are recent preprints.

21. Understanding logspace derandomization

A randomized logarithmic-space algorithm has a long random tape but very little memory. The equality $\mathsf{BPL} = \mathsf{L}$ says that every polynomial-time bounded-error computation in this model has a deterministic logarithmic-space simulation. Understanding the new proof is a substantial reading project: the interesting issue is how its estimates, recursion, and recomputation fit into the space budget.

- **Entry point.** The 2026 logspace derandomization theorem establishes $\mathsf{L} = \mathsf{RL} = \mathsf{BPL}$. Its acceptance-probability estimator achieves any fixed inverse-polynomial additive error using logarithmic space and polynomial time. Nisan's older generator gives a useful comparison: it replaces a long random tape with $O(\log^2 n)$ seed bits.
- **Direction.** What does the new proof estimate, and how does it evaluate those estimates without retaining a long computation history? Choose a central lemma and explain how it contributes to the complete deterministic simulation.
- **Direction.** How does language-class derandomization differ from constructing one ordinary pseudorandom generator that fools every branching program in a class? Compare the quantifiers and space costs in the new proof and the generator approach.

Reading: OpenAI, *Exact Derandomization of Logarithmic Space: L = RL = BPL*, Theorems 1.1–1.2 and the proof roadmap in Section 1.2; Vadhan's monograph for the classical generator background.

22. Streaming algorithms against adaptive adversaries

A streaming algorithm compresses a long sequence into a small state, often using a random sketch. Its standard guarantee typically fixes the stream independently of the random choices. An adversary that chooses future updates after observing outputs, or even the internal state, can exploit information about those choices. This ties the streaming lower-bound viewpoint to cryptographic notions of hidden randomness and adaptive interaction.

- **Entry point.** For second-moment estimation, ordinary randomized sketches use polylogarithmic memory. A white-box adversary that sees the entire state can force linear memory for constant-factor accuracy, even on insertion-only streams with output only at the end. It sees previous state and coins, and chooses the next update before fresh coins are drawn.
- **Direction.** Compare fixed streams, output-adaptive adversaries, and adversaries that see the full memory state. For one task, map the space and accuracy bounds, output schedule, and adversary's computational power. Investigate robustness against a computationally bounded adversary; in the full-state model, past random coins are exposed.

Reading: Efremenko–Kol–Saxena–Zhang, *White-Box Adversarial Streaming Lower Bounds beyond Two-Party Communication*, Theorem 1 and Section 1.2.

23. Converting one correlation into another without communication

Alice observes X^m and Bob observes Y^m, where the coordinate pairs are independent samples from a fixed joint distribution. They separately produce outputs that should have a prescribed joint distribution. Which target correlations can they synthesize without talking? Data processing gives obstructions, but mutual information alone is far from a complete answer. Maximal correlation and hypercontractivity provide stronger tools, and the problem connects distributed randomness generation to Boolean functions and noise stability.

- **Entry point.** Maximal correlation cannot increase under local processing and tensorizes across independent samples, giving impossibility results for every block length at once. Hypercontractivity yields strictly stronger obstructions for some source–target pairs, revealing structure that a single correlation statistic misses. Finite-alphabet approximate simulation also has computable bounded-block and gap-decision results, so the existence of some computable sample bound is already known.
- **Direction.** For a fixed finite-alphabet source and target, how many samples are needed to achieve total-variation error ε? Compare explicit constructions, lower bounds, and the known general computable bounds; seek sharp dependence on error for one specified source–target pair. Gap-decision results do not automatically decide exact boundary feasibility. Keep shared randomness separate from private local coins.

Reading: Kamath–Anantharam, *On Non-Interactive Simulation of Joint Distributions*; De–Mossel–Neeman, *Non interactive simulation of correlated distributions is decidable*; Ghazi–Kamath–Raghavendra, *Dimension Reduction for Polynomials over Gaussian Space and Applications*.

E. Compression algorithms and distributed sources

24. Lempel–Ziv: universality and robustness

Lempel–Ziv algorithms compress without knowing the distribution that generated the data. LZ77 refers back to earlier substrings; LZ78 builds a dictionary of phrases. Their theory connects the entropy rate of a random source with the compressibility of an individual sequence. It also reveals surprising behavior that an asymptotic entropy theorem does not capture: dictionary boundaries, finite memory, and small edits can all matter.

- **Entry point.** Lempel–Ziv schemes attain the entropy rate almost surely for every stationary ergodic finite-alphabet source, without knowing its parameters. LZ78 also competes asymptotically with every fixed finite-state information-lossless compressor on individual sequences. Nevertheless, Lagarde–Perifel construct an infinite binary sequence whose LZ78 compression ratio tends to zero, while prepending one bit makes its lower limiting ratio positive.
- **Direction.** How do the stationary-source and individual-sequence optimality statements relate? For a fixed i.i.d. or finite-order Markov source class, what finite-block redundancy bounds follow, and what changes if the dictionary or window size is bounded?
- **Direction.** Why can a single initial bit change so many LZ78 phrase boundaries? Compare the catastrophe construction with LZ77 stability, or investigate resets that improve stability with a quantified compression penalty. Specify the parsing variant and charge encoded bits, including pointers and dictionary descriptions, rather than only counting phrases.
- **Direction.** Does LZ complexity obey an entropy-like chain rule? Merhav's blockwise LZ78 bounds require changes in compression-block lengths; a naive limiting identity fails. Explain those changes while tracking empirical entropy and boundary losses separately.

Reading: Ziv–Lempel, *Compression of individual sequences via variable-rate coding*; Lagarde–Perifel, *Lempel-Ziv: a "one-bit catastrophe" but not a tragedy*, Section 2 and the finite construction in Section 4; Akagi–Funakoshi–Inenaga, *Sensitivity of string compressors and repetitiveness measures*, Section 7; Merhav, *Lempel-Ziv Complexity, Empirical Entropies, and Chain Rules*, Theorems 1–3.

25. Reversible preprocessing and the Burrows–Wheeler transform

The Burrows–Wheeler transform rearranges a string so that a subsequent compressor can exploit its structure. It preserves symbol counts, so its benefit cannot be explained by zero-order empirical entropy alone. The usual BWT stores a primary index to support inversion; the bijective BWT (BBWT), built using the string's Lyndon factorization, is invertible without that index. These transforms connect practical compression to combinatorics on words and to the question of what makes a representation useful.

- **Entry point.** Bannai–I–Nakashima construct binary strings of length n requiring $\Omega(n/\log n)$ LZ77 phrases, whose BBWT images need only $O(\log n)$ phrases and grammar symbols. Thus reversible preprocessing can dramatically improve dictionary compression. A second result shows that rotations and repeated BBWT applications can connect any two binary strings with the same number of each symbol.
- **Open question.** Does that reachability statement hold for every finite alphabet? In particular, can every pair of ternary strings with identical symbol counts be connected by a finite sequence of cyclic rotations and BBWT applications? The binary case is proved; the general statement is Conjecture 1 of Badkobeh et al.
- **Direction.** Explain the Fibonacci-word construction behind the compression improvement, or study algorithms for finding useful rotations before applying BBWT. If several transformations are used as a compressor, their descriptions must also be included in the encoding cost: reachability alone does not give a short code.

Reading: Bannai–I–Nakashima, *On the Compressiveness of the Burrows-Wheeler Transform*, Theorem 16; Badkobeh–Bannai–I–Köppl, *Bijective BWT Based Compression Schemes* (2026 journal version), Sections 4–5, especially Conjecture 1 and Theorem 5.

26. Grammar compression: how far is greedy from optimal?

A grammar can describe a string by naming repeated fragments and reusing their names. A straight-line grammar is acyclic and generates exactly one string. Unlike an arbitrary program, its size and expansion are easy to check, but finding the smallest such description is computationally hard. Practical algorithms such as RePair repeatedly replace frequent adjacent pairs; the local choice need not lead to a globally small grammar. These representations also support computation on a string without first decompressing all of it.

- **Entry point.** If $g^*(x)$ is the minimum total right-hand-side length of a grammar generating a length-n string x, polynomial-time algorithms achieve size $O(g^*(x)\log(2+n/g^*(x)))$. Jeż's recompression method gives a particularly simple route through an LZ77 factorization. By contrast, RePair can be worse than optimal by a factor $\Omega(\log n/\log\log n)$, even on binary strings.
- **Open question.** Is there a constant C and a uniform polynomial-time algorithm that, for every binary string x, outputs a straight-line grammar of size at most $Cg^*(x)$? Size here counts symbols on production right-hand sides, rather than encoded bits or derivation steps. An October 2026 preprint establishes exact NP-hardness even for binary strings; exact hardness alone does not rule out this approximation.
- **Direction.** What feature of the RePair lower-bound construction defeats repeated most-frequent-pair replacement? Analyze a restricted string family, keeping alphabet size and tie-breaking explicit. Alternatively, study the space/query-time tradeoff for random access to a grammar-compressed string, with space measured against grammar size rather than uncompressed length.

Reading: Jeż, *A really simple approximation of smallest grammar*; Bannai et al., *The smallest grammar problem revisited*, Theorem 3.7; Casel et al., *On the Complexity of the Smallest Grammar Problem over Fixed Alphabets*, Section 6.2; Chen, *Exact Algorithms and Complexity of the Smallest Grammar Problem* (October 2026 preprint); *Random Access in Grammar-Compressed Strings: Optimal Trade-Offs in Almost All Parameter Regimes* (ICALP 2026).

27. Distributed compression and computation

Two correlated datasets may be observed by encoders that cannot communicate with one another. A joint decoder receives their compressed descriptions. Correlation still saves communication, even though neither encoder sees the other dataset. Changing the goal from recovering both datasets to recovering only a function of them can reward algebraic structure rather than ordinary compression. Allowing distortion creates another family of tradeoffs.

- **Entry point.** For i.i.d. pairs (X,Y), Slepian–Wolf coding attains total rate arbitrarily close to $H(X,Y)$ with separate encoders and vanishing block error. The individual lower bounds are $H(X \mid Y)$ and $H(Y \mid X)$. Thus the encoders can exploit correlation without observing one another's data.
- **Direction.** How much can be saved if the decoder needs only $X^n \oplus Y^n$? When do linear syndromes outperform first reconstructing both sources? Compare the Körner–Marton example with the Slepian–Wolf rate region.
- **Direction.** What changes when each source need only be reproduced within a prescribed average distortion? Fix the source law, distortion measures, and separate-encoder model before comparing achievable regions.

Reading: Slepian–Wolf, *Noiseless Coding of Correlated Information Sources*; Körner–Marton, *How to encode the modulo-two sum of binary sources*.

F. Sparse recovery and group testing

28. Compressed sensing: geometry, graphs, and sparse recovery

A vector with only k important coordinates can often be recovered from far fewer than N linear measurements. Compressed sensing asks which measurement maps make this possible, how quickly recovery can run, and what survives noise or an approximately sparse input. There are two complementary pictures: Euclidean geometry and convex optimization, or sparse bipartite graphs and combinatorial decoding. Their guarantees use different norms and should be compared with those norms explicit.

- **Entry point.** With high probability, a suitably normalized Gaussian matrix with $O(k \log(eN/k))$ rows preserves the squared Euclidean norm of every k-sparse vector up to constant distortion. With a suitable distortion and order, ℓ_1 minimization gives stable recovery. Lossless expanders instead provide sparse measurement matrices with an ℓ_1 restricted-isometry guarantee and combinatorial recovery algorithms. Recent reductions even use sparse recovery to accelerate output-sparse and robust matrix multiplication.
- **Open question.** Can a uniform deterministic polynomial-time construction match the Gaussian row count? For every $1 \le k \le N/2$, construct a rational $m \times N$ matrix, with polynomial-bit entries and $m = O(k \log(eN/k))$, satisfying

  $$\tfrac{2}{3}\|x\|_2^2 \le \|Ax\|_2^2 \le \tfrac{4}{3}\|x\|_2^2 \quad \text{for every } k\text{-sparse } x \in \mathbb{R}^N.$$

  The matrix must work for all such vectors simultaneously.
- **Direction.** How do lossless expansion, isolated neighbors, and error-correcting-code syndromes lead to fast recovery? Compare measurement count, column sparsity, decoding time, and adversarial-noise guarantees. An ℓ_1 expander guarantee does not itself give Euclidean RIP.
- **Direction.** Reconstruct a recent sparse-matrix-multiplication reduction: which measurements can be computed without forming the full product, and what property of the recovery algorithm controls the runtime?

Reading: Berinde–Gilbert–Indyk–Karloff–Strauss, *Combining geometry and combinatorics: A unified approach to sparse signal recovery*; *Explicit Matrices with the Restricted Isometry Property: Breaking the Square-Root Bottleneck*, for the deterministic-construction frontier; Bennett et al., *Output-Sparse Matrix Multiplication Using Compressed Sensing*; Bringmann–Fischer–Nakos, *Robustifying Sparse Matrix Multiplication*.

29. Sparse Walsh–Hadamard transforms

The Walsh–Hadamard transform is Fourier analysis on $\mathbb{F}_2^d$, with $N = 2^d$ coordinates. If a signal has only a few significant Fourier coefficients, can we find them without reading all N input values? This connects sublinear algorithms, learning parities, Hadamard-code decoding, and compressed sensing. Query access to signal values is a stronger restriction than access to arbitrary linear measurements.

- **Entry point.** Cheraghchi–Indyk give a deterministic nonadaptive algorithm using $k^{1+\alpha}(\log N)^{O(1)}$ time for every fixed $\alpha > 0$. From query access to f, it outputs a k-sparse $\widetilde{f}$ with $\|\widehat{\widetilde{f}} - \widehat{f}\|_1 \leq C\|\widehat{f} - H_k(\widehat{f})\|_1$, where H_k keeps the k largest coefficients. The theorem assumes polynomial-in-log N bit precision. Linear lossless condensers are the key link to combinatorial sparse recovery.
- **Direction.** What would remove the k^α overhead while retaining deterministic, nonadaptive queries and the same approximation guarantee? Compare exact sparsity, approximate sparsity, noisy queries, and randomized support assumptions rather than transferring a bound between these models.
- **Direction.** A different project studies uniform Euclidean recovery from sampled Walsh rows. A September 2026 preprint, revised in October, claims a sufficient count $O(k \log(2k) \log(2eN/k))$ at fixed distortion in its subsampling regime, using Bernoulli sampling or sampling without replacement. It matches the known Walsh lower bound in its intermediate-sparsity range, so the older logarithmic gap should not be treated as untouched. Explain the relative-entropy proof method, and distinguish sample bounds from sublinear decoding time and a single matrix working for every signal from success for one fixed signal.

Reading: Cheraghchi–Indyk, *Nearly Optimal Deterministic Algorithm for Sparse Walsh-Hadamard Transform*; Haviv–Regev, *The Restricted Isometry Property of Subsampled Fourier Matrices*, for the earlier bound; Błasiok et al., *An Improved Lower Bound for Sparse Reconstruction from Subsampled Walsh Matrices*; Burstein–Iosevich–Krause, *Fourier Ratio and Restricted Isometry via Entropic Descent*, Theorems 1.4 and 1.6 and Section 5.4 (recent preprint).

30. Group testing: finding a sparse set using pooled tests

A group test reports whether a chosen pool contains at least one defective item. The aim is to identify a small defective set among N items using far fewer than N tests. Outcomes are Boolean ORs, giving a useful comparison with the linear measurements of compressed sensing. The answer depends on whether tests are adaptive, whether one design must handle every defective set, and how quickly the test outcomes can be decoded.

- **Entry point.** Price–Scarlett and Cheraghchi–Nakos give randomized, noiseless, nonadaptive schemes using $O(k \log N)$ tests and word-RAM decoding time. For each fixed set of k defectives, recovery succeeds with high probability over the design; in Price–Scarlett's basic theorem, failure is $O(k^{-c})$ for any fixed $c > 0$, so take $k \to \infty$. Wang–Gabrys–Guruswami sharpen leading constants in a uniform-random-k-subset model.
- **Open question.** Close the logarithmic gap for disjunct matrices. A binary matrix is k-disjunct if, for every column j and every set S of at most k other columns, there exists a row containing 1 in column j and 0 in every column of S. Such a matrix gives noiseless, nonadaptive, zero-error recovery of *every* set of at most k defectives. Let $t_k(N)$ be its minimum possible number of rows and set $R_k = \limsup_{N\to\infty}(\log_2 N)/t_k(N)$. Determine the order of R_k as $k \to \infty$: the known bounds are $\Omega(k^{-2}) \le R_k \le O((\log k)/k^2)$. This takes $N \to \infty$ first for each fixed k and imposes no construction or decoding-time bound on the extremal question.
- **Direction.** Compare adaptive splitting, randomized recovery for each fixed set, and zero-error recovery for every set. How do tests, construction, storage, and decoding time change? Code-based disjunct constructions give a concrete starting point; noisy tests or restricted pool sizes give further variants.

Reading: Price–Scarlett, *A Fast Binary Splitting Approach to Non-Adaptive Group Testing*, Theorem 1; Cheraghchi–Nakos, *Combinatorial Group Testing and Sparse Recovery Schemes with Near-Optimal Decoding Time*, Theorem 3.10; Wang–Gabrys–Guruswami, *Quickly-Decodable Group Testing with Fewer Tests*, Theorems 1–2; Idalino–Moura, *Cover-free families on hypergraphs and combinatorial group testing* (2026), Section 2.1, for the disjunct-matrix bounds.

G. Codes, noise, and computation

31. List decoding and cryptographic proof systems

List decoding replaces unique recovery by a short list of plausible messages. For polynomial codes, interpolation gives efficient algorithms beyond half the minimum distance. These ideas also underlie proximity testing and the soundness of succinct cryptographic proofs: a verifier checks a few positions and wants assurance that an oracle is close to a valid codeword. Combinatorial list size, efficient reconstruction, and soundness of a proximity test are related but distinct guarantees.

- **Entry point.** Guruswami–Sudan decodes Reed–Solomon codes up to error fraction $1-\sqrt{R}$. October 2026 work constructs ordinary Reed–Solomon codes over specially chosen polynomial-size extension fields with deterministic polynomial-time decoding to $1-R-\varepsilon$ and $O((1-R)/\varepsilon)$ list size. Other recent results treat arbitrary domains in sufficiently large characteristic and improve curve-decoding alphabet sizes. The field and domain hypotheses are essential.
- **Direction.** Follow one route from Reed–Solomon proximity testing to a cryptographic proof system. Which code property supplies soundness, and how do field size, characteristic, rate, and decoding radius affect the verifier's error and the proof length? Compare the new subspace-design, curve-decoding, and algorithmic approaches. Counterexamples over binary fields rule out transferring blanket beyond-Johnson guarantees to every evaluation domain.
- **Direction.** Which decoding algorithms run in $\mathrm{poly}(n, \log q)$, and which only in $q^{O(1)}$? The distinction matters for cryptographically large fields. Alternatively, study Goldreich–Levin decoding of the Hadamard code and its hardcore-bit application.

Reading: Vadhan's list-decoding chapter; Harsha–Kumar–Saptharishi's capacity-decoding exposition; Alrabiah–Goyal–Guruswami, *Explicit Reed-Solomon Subspace Designs: Optimal List and Line Decoding*; Goyal–Guruswami–Sun–Wootters, *Improved LCL Equivalences and Curve Decoding Alphabet Sizes*; Dao–Kominers–Thaler's efficient decoding/proof-size results, November 2026 revision; Dao et al.'s binary-field counterexamples. The recent results should be read with their precise field, domain, and resource hypotheses.

32. Quantum error-correcting codes: sparse checks and local testability

Quantum error correction protects superpositions without measuring the encoded information itself. CSS codes use two classical parity-check matrices whose orthogonality makes their quantum checks commute. A stabilizer $[[n, k, d]]$ code stores k logical qubits in n physical qubits; d is the smallest weight of a nontrivial logical Pauli operator. Distance d permits ideal recovery from arbitrary errors on fewer than $d/2$ qubits. Quantum LDPC codes also bound each check's weight and each qubit's number of incident checks by constants; these conditions alone impose no geometric locality or decoding-time guarantee.

- **Entry point.** Tillich–Zémor's hypergraph product turns good classical LDPC codes into quantum codes of constant rate and distance $\Theta(\sqrt{n})$. Panteleev–Kalachev subsequently achieved constant rate and linear distance. Thus asymptotically good quantum LDPC codes already exist. A September 2026 preprint by Gay–Jeronimo claims the stronger combination of constant rate, relative distance, local-test soundness, check weight, and qubit incidence; its full proof is not audited here.
- **Direction.** Reconstruct the CSS construction. Why does $H_X H_Z^{\mathsf{T}} = 0$ make the checks commute, and why is the number of logical qubits $n - \operatorname{rank} H_X - \operatorname{rank} H_Z$? Explain what a syndrome measurement learns and why that need not reveal the encoded quantum information.
- **Direction.** Follow the hypergraph-product construction from a full-row-rank classical $[n_0, k_0, d_0]$ parity-check matrix. Derive the parameters $[[n_0^2 + (n_0 - k_0)^2, k_0^2, d_0]]$, and explain the square-root distance. A proof of this theorem is a focused project with linear-algebraic entry points; the modern linear-distance constructions offer further reading.
- **Direction.** What does local testability add to sparse checks? Formulate soundness using quantum Hamming distance to the codespace, rather than trace distance, and compare it with error correction and local decoding. An advanced project can explain the main statement and one ingredient of the new preprint, or why local testability alone is not a quantum PCP theorem.

Reading: Gottesman, *Stabilizer Codes and Quantum Error Correction*, Chapter 3, especially Section 3.3 (see the author's errata); Tillich–Zémor, *Quantum LDPC codes with positive rate and minimum distance proportional to $n^{1/2}$*, Theorem 1; Panteleev–Kalachev, *Asymptotically Good Quantum and Locally Testable Classical LDPC Codes*, Theorem 2; Gay–Jeronimo, *Asymptotically Good Quantum Locally Testable Codes*, Definition 1.2.1 and Theorem 2.4.3 (September 2026 preprint).

33. Error correction with a tiny streaming decoder

A receiver may have too little memory to store a long transmission. A stream-decodable binary code lets it scan the corrupted codeword once and write the recovered message, in order, to a write-only output tape. This computational restriction can force much longer codewords than ordinary error correction requires.

- **Entry point.** Such codes can correct any fixed error fraction below $1/4$. For an n-bit message, however, an s-bit decoder succeeding with probability at least $1 - 1/(2n^2)$ requires codeword length $\Omega(n^2/s)$. Recovering only one linear function of the message permits much shorter codes.
- **Open question.** Does the same length lower bound hold if success probability is only $2/3$? Require the guarantee for every message and every received word within a fixed positive fraction of adversarial bit flips. The decoder has private random coins and $s = (\log n)^{O(1)}$ bits of memory.
- **Open question.** Can constant-rate codes let such a decoder recover any selected message bit with high probability? The index is unknown to the encoder but may be known to the adversary, who may flip up to a fixed positive fraction of codeword bits.
- **Direction.** Compare one-pass streaming with recent low-space list decoders that can revisit the received word. Fathollahi–Ron-Zewi–Wootters obtain near-linear time and sublinear space at capacity over a constant alphabet; this access model does not resolve the streaming questions above.

Reading: Gupta–Guruswami–Singhal, *Tight Bounds for Stream Decodable Error-Correcting Codes*, Theorems 1–3 and Section 1.3; Fathollahi–Ron-Zewi–Wootters, *Time- and Space-Efficient List Decoding up to Capacity* (August 2026 preprint).

34. Tree codes: error correction before the message is known

An ordinary block code starts with a message that is already available. In an interactive conversation, future messages depend on replies that have not yet arrived. Tree codes encode each new symbol using only the symbols seen so far. Their distance requirement holds after every possible divergence of two conversations, throughout each subsequent interval starting at that divergence. This connects error correction, online computation, and reliable interactive communication.

- **Entry point.** Schulman proved that constant-distance tree codes with binary inputs and a constant output alphabet exist, enabling interactive communication with constant overhead against a constant fraction of adversarial errors. Explicit constructions remain harder. An October 2026 result improves the rate–distance tradeoff, while still using an alphabet that grows with depth for fixed positive distance. A recent information-theoretic barrier explains limitations of constructions satisfying an additional "immediacy" property.
- **Open question.** Is there a fixed finite alphabet Σ, a constant $\delta > 0$, and a uniform deterministic polynomial-time encoder for causal maps $E_n : \{0,1\}^n \to \Sigma^n$ for all depths n, such that whenever x, y first differ at position s,

$$\left|\{i \in [s,t] : E_n(x)_i \neq E_n(y)_i\}\right| \geq \delta(t - s + 1) \qquad (s \leq t \leq n)?$$

  Causality means output i depends only on n and the first i input bits. The target asks for efficient encoding, without also demanding efficient adversarial decoding or consistency between different depths.
- **Direction.** Compare existence, efficient encoding, and efficient decoding. What is immediacy, and why does its rate barrier leave the ordinary tree-code question open? A proof of the barrier or an explanation of the newest construction makes a focused reading project.

Reading: Cohen–Haeupler–Schulman, *Explicit Binary Tree Codes with Polylogarithmic Size Alphabet*; Cohen–Schulman–Srivastava, *The Rate-Immediacy Barrier in Explicit Tree Code Constructions* (CCC 2026); Yankovitz, *Better Binary Tree Codes From MDS Tree Codes*, Theorem 1.4 (October 2026 preprint).

35. Robust Gray codes: smooth representations that survive noise

A Gray code represents successive integers by bit strings differing in one bit. This makes small changes to a quantity easy to represent, but ordinary Gray decoding can turn a few corrupted bits into a large numerical error. Robust Gray codes combine this local smoothness with error correction: decoding a noisy representation should return an integer close to the original. Applications to private data structures give a connection between channel coding and differential privacy.

- **Entry point.** For every fixed crossover probability $0 < p < 1/2$ and rate gap $\varepsilon > 0$, efficient robust Gray codes approach rate $1 - h_2(p) - \varepsilon$. Under independent binary-symmetric-channel noise, the integer-estimation error has a tail exponentially decreasing in the error distance, plus a failure term vanishing with block length. Thus neighboring encodings can remain one bit apart while large numerical errors are unlikely.
- **Direction.** Explain why approximate integer recovery is compatible with the Gray property, and derive the capacity upper bound on the rate. Compare the construction with an ordinary Gray code followed by a block code: which requirement does that composition lose?
- **Direction.** Trace an application to private data structures or investigate a different noise model. Keep independent channel errors separate from a worst-case adversarial error budget, and state the desired numerical-error guarantee.

Reading: Con–Fathollahi–Gabrys–Wootters–Yaakobi, *Robust Gray Codes Approaching the Optimal Rate*, Corollary 1, Observation 1, and Sections 1.1–1.2.

36. The binary deletion channel

An erasure channel tells the receiver which positions disappeared. A deletion channel supplies only the surviving subsequence, so the receiver must also resolve alignment. Even independent deletions create a difficult channel optimization problem. Input processes with memory and descriptions in terms of runs can outperform the simplest guesses. This is a good setting for comparing an exact capacity theorem, an achievable rate, a converse bound, and a certified numerical approximation.

- **Open question.** Determine the exact capacity $C_{\mathrm{del}}(d)$ when each input bit is independently deleted with probability $d \in (0, 1)$. Measure rate per transmitted input bit and require decoding error tending to zero.
- **Entry point.** The binary deletion channel has positive capacity for every fixed deletion probability below one, even when almost all input bits disappear. Run-length descriptions and input processes with memory underlie useful achievability bounds despite the loss of alignment.

Reading: Cheraghchi–Ribeiro, *An Overview of Capacity Results for Synchronization Channels*; *Binary Deletion Channel Capacity to Within One Hundredth of a Bit* is an optional recent computational reading.

37. Zero-error capacity and graph powers

Ordinary channel capacity permits a decoding error that tends to zero. Zero-error capacity asks for messages that are never confusable. A channel's confusability graph turns this into a problem about independent sets in strong graph powers: coding across several uses can outperform choosing a maximum independent set separately in each use. The five-cycle has capacity $\sqrt{5}$, while even the seven-cycle remains unresolved. Small graph examples make the gap between one-shot and asymptotic reasoning visible.

- **Open question.** Determine $\Theta(C_7)$, where $\Theta(G) = \sup_{m \geq 1} \alpha(G^{\boxtimes m})^{1/m}$, $\boxtimes$ denotes strong product, and α is independence number.
- **Entry point.** A single use of the five-cycle channel supports only two nonconfusable symbols, but two uses support five messages. Lovász's theta bound proves the asymptotic capacity is exactly $\sqrt{5}$. For C_7, recent recursive constructions improve the lower bound; Tandon's refinement gives $\Theta(C_7) \geq 3.25883262\ldots$, while the exact capacity remains open.

Reading: Lovász, *On the Shannon Capacity of a Graph*; Buys–Polak–Zuiddam, *Lean-verified lower bounds for the Shannon capacity of odd cycles*, for recent constructions; Tandon, *Strengthening Recursive Constructions for Zero-Error Shannon Capacity*, for a subsequent refinement.

38. Non-malleable codes: protecting meaning against tampering

An error-correcting code recovers the original message after a bounded number of changes. A non-malleable code has a different objective: after tampering, the decoded message should be the original message or have a distribution unrelated to it, with an explicit convention for "unchanged." This can make sense even when many coordinates are modified. The permitted tampering family is crucial; bitwise tampering and separate tampering of two large states lead to very different constructions.

- **Entry point.** Efficient non-malleable codes exist for independent bitwise tampering. Even the much richer two-split-state model admits constant-rate codes with exponentially small error: each half may undergo an arbitrary transformation, provided the transformations act separately.
- **Direction.** How are two-source non-malleable extractors converted to codes for split-state tampering? Compare rate, error, efficient encoding, and the difficulty of sampling an extractor's preimages.

Reading: Dziembowski–Pietrzak–Wichs, *Non-Malleable Codes*; Li's extractor paper, Section 7.3, for split-state constructions.

39. Efficient wiretap coding against an active adversary

Wiretap coding combines reliable transmission with information-theoretic secrecy. In the binary active Wiretap II model, Eve first chooses a set of rn transmitted coordinates to observe, then uses those observations to choose up to pn bit flips. The read locations are nonadaptive; the flips may occur anywhere. The encoder uses private randomness, with no preshared secret. Random coding achieves rate $1 - h_2(p) - r$, but supplying efficient algorithms remains a separate challenge.

- **Open question.** For fixed $0 < p < 1/2$ and $0 < r < 1$ with $h_2(p) + r < 1$, and any fixed $\eta > 0$, construct uniform explicit codes of rate at least $1 - h_2(p) - r - \eta$ with polynomial-time encoding and decoding, vanishing worst-message decoding error against every allowed attack, and negligible mutual-information leakage for every message distribution and read set.
- **Entry point.** Random pseudolinear codes with $O(n^2)$-bit descriptions attain the random-coding rate and support efficient encoding. Their extra structure reduces the randomness needed to describe a good code, while efficient decoding remains a separate obstacle.

Reading: Ruzomberka et al., *Derandomizing Codes for the Binary Adversarial Wiretap Channel of Type II*, Section II-D and conclusion.

40. Computing on encoded data with little download

Error-correcting codes spread data across servers so failures do not destroy it. A user may need only a function of the data, rather than all original symbols. Each server can compute locally on its stored symbol and return a short answer. Linear functions can permit substantial savings, but nonlinear functions behave differently. This connects coding, communication, secret sharing, and distributed computation. The basic objective here is exact function recovery with low download; privacy is an additional model choice.

- **Entry point.** Blackwell–Wootters prove a download lower bound already for a line $f(z) = a + bz$ evaluated at every nonzero point of an odd-characteristic field $\mathbb{F}_q$. In their model of nonadaptive local bit queries, recovering ab from every chosen set of $s \geq 3$ servers requires at least $2\log_2(q-1)-3$ downloaded bits, nearly the cost of retrieving both coefficients. A simple quadratic function can therefore defeat hoped-for bandwidth savings.
- **Direction.** Compare linear and quadratic function recovery. Which parts of the lower-bound proof use odd characteristic or correctness for every server set? Study a fixed evaluation domain, binary extension fields, or more message coefficients, keeping the query model explicit.
- **Direction.** What changes if the user permits approximate answers, interaction, or a small error probability? Distinguish downloaded bits from the number of contacted servers and from upload cost.

Reading: Blackwell–Wootters, *Limitations to Computing Quadratic Functions on Reed-Solomon Encoded Data* (ITCS 2026), Definitions 1–3 and Theorem 4.

H. Entropy, learning, and privacy

41. Entropy inequalities as a language for combinatorial lower bounds

Shearer's inequality bounds a set by the sizes of overlapping projections, turning entropy into a counting tool. More generally, linear inequalities between subset entropies constrain network coding and secret sharing. Nonnegativity of conditional mutual information supplies the Shannon inequalities, but four random variables already obey additional inequalities. The geometry of possible entropy vectors exposes limitations of otherwise natural linear-programming lower bounds.

- **Open question.** Characterize the closed region of entropy vectors $(H(X_S))_{\emptyset \neq S \subseteq [4]}$ arising from finite jointly distributed random variables, including limits of such vectors.
- **Open question.** Can a single application of the maximum-entropy extension method prove a linear entropy inequality that no finite iteration of the copy lemma can prove? This compares two particular proof methods; see Problem 7 of the reading.
- **Entry point.** Zhang and Yeung found a valid four-variable entropy inequality that cannot be derived from Shannon's basic inequalities. The copy lemma turns auxiliary random variables into a systematic source of such stronger inequalities, with applications to secret-sharing and network-coding bounds.

Reading: Csirmaz, *Exploring the entropic region*, a tutorial with further questions.

42. Compression, information, and generalization

Learning requires a rule fitted to observed data to remain accurate on new data. Compression arguments explain this by counting the possible summaries a learner retains, while information-based arguments charge the dependence between a dataset and its learned output. These perspectives need not give the same guarantee: an expectation bound can miss a rare memorization event, and a deterministic learner with continuous output may have an unhelpfully large information measure. Concrete examples help identify the useful notion of information for a given learning task.

- **Entry point.** For i.i.d. data and bounded loss, the magnitude of expected generalization error is $O(\sqrt{I(S;W)/n})$, where S is the sample and W the learned output. Conditional mutual information offers another route: condition on a pool of paired examples and measure how much the output reveals about the random bits selecting the training sample. This framework connects compression, VC dimension, and differential privacy.
- **Direction.** When does this conditional-information bound explain generalization that ordinary mutual information misses? Compare the two measures for a deterministic learner or one that memorizes on a rare event, keeping expected and high-probability guarantees distinct.

Reading: Xu–Raginsky, *Information-theoretic analysis of generalization capability of learning algorithms*; Steinke–Zakynthinou, *Reasoning About Generalization via Conditional Mutual Information*; Moran–Yehudayoff, *Sample Compression Schemes for VC Classes*.

43. Differential privacy, fingerprinting, and computational hardness

Differential privacy limits how much the output distribution changes when one person's record changes. It permits useful population statistics, but accuracy on too many queries can allow reconstruction or tracing of an individual. Fingerprinting codes turn that observation into lower bounds: accurate answers can act like a pirate codeword that points back to a database participant. The same area also contains separations between what is information-theoretically possible and what an efficient algorithm can release.

- **Entry point.** Fingerprinting codes yield nearly optimal sample lower bounds for approximate differential privacy when answering large families of counting queries. A code designed to trace a colluding pirate becomes a tool for showing that sufficiently accurate aggregate answers can identify a database participant.
- **Direction.** For one query family, compare sample complexity with computational complexity. How do pure versus approximate privacy, central versus local privacy, or adaptive versus fixed queries change the result?

Reading: Bun–Ullman–Vadhan, *Fingerprinting Codes and the Price of Approximate Differential Privacy*; Lyu–Talwar, *Fingerprinting Codes Meet Geometry: Improved Lower Bounds for Private Query Release and Adaptive Data Analysis* (STOC 2025), for sharper dependence on the approximate-privacy parameter.