Technion - Israel Institute of Technology

Faculty of Data and Decision Sciences

Deep Deep Learning

Graduate student-led research seminar | Spring 2027 | Course number: TBD

Instructor: Assaf Shocher. Course contact and office hours: TBD.

Course description

Study deep learning through foundations, a central paper and recent developments, spanning learning dynamics, backpropagation alternatives, architectures, representations, interpretability and generative models.

Topic selection

The 19 reading bundles form the pool for the 13-week seminar. Week 10 debates units of analysis in mechanistic interpretability; Week 11 retains circuit tracing. Students and the instructor choose the remaining topics and order. Dates and all presenter assignments: TBD. Each selected topic gets two academic hours.

Learning objectives

Explain a mathematical mechanism; identify its assumptions; assess an experiment that tests an explanation; connect recent research to older ideas; formulate a precise unresolved question.

Prerequisites & logistics

Working knowledge of deep learning, linear algebra, probability, calculus and optimization. Formal prerequisites, credits, enrollment, meeting time, room and delivery mode: TBD. Dates are not yet assigned. Flow matching is assumed background from the base course.

Seminar format & assessment

Students lead presentations and discussion. Each session centers on one main paper, with selected foundation and companion material. Presenters read the main paper and prepare one derivation, one decisive experiment and one unresolved question. No exam. Grading components and weights, attendance requirements and written submissions: TBD.

Shared pre-reading

All students complete one required pre-reading for each selected meeting. Presenters choose it in consultation with the instructor and announce it one week before class: a tutorial, a short paper or clearly specified sections taking 30-45 minutes. Include two guiding questions; each student arrives with one question or point of confusion. The main paper is optional for the audience unless chosen as the shared pre-reading. Specific pre-readings, slide deadlines and coordination arrangements: TBD.

Research practice & communication

Cite papers, figures, code and other sources; distinguish the authors' claims from your own analysis. Formal academic-integrity and AI-use policies, course forum and accessibility arrangements: TBD.

Weekly schedule

Week numbers only. Weeks 10-11 cover units of analysis and circuit tracing. Other topics, all student presenters and dates are TBD. The reading pool below is larger than the final course.

Week	Selected topic	Presenter
1	TBD	TBD
2	TBD	TBD
3	TBD	TBD
4	TBD	TBD
5	TBD	TBD
6	TBD	TBD
7	TBD	TBD
8	TBD	TBD
9	TBD	TBD
10	Mechanistic interpretability: units of analysis	TBD
11	Mechanistic interpretability: circuit tracing	TBD
12	TBD	TBD
13	TBD	TBD

Suggested meeting structure

Two academic hours per selected topic. Assuming two 45-minute academic hours: 15 minutes foundations, 35 minutes mechanism and derivation, 15 minutes evidence, 25 minutes discussion. Adjust the split to the local academic-hour convention.

Choosing a manageable scope

Each candidate has one reading bundle and a central question. Build around its main paper, using selected foundation and companion sections to explain and assess the mechanism. Develop one derivation and one decisive experiment carefully. Candidate numbers are references for discussion, independent of assigned week numbers.

Candidate topic pool

Topic numbers identify reading bundles, independently of week numbers. Weeks 10 and 11 cover units of analysis and circuit tracing respectively. Students and the instructor choose the remaining topics and teaching order; dates and all presenters are TBD. Each reading bundle connects one main paper to selected foundations and recent developments. Shared pre-readings and selected sections: TBD.

Topic 1 · Wide networks: NTK, feature learning and muP

Topic family: Kernels, feature learning & optimization

How can increasing width lead either to kernel-like training or to feature learning, and what does this imply for scaling?

Scope for one session: Two academic hours: one short NTK derivation, one worked width-scaling example, and one hyperparameter-transfer experiment. The unifying question is how parameterization changes learning, rather than two complete surveys of NTK and muP.

Background: Jacot, Gabriel & Hongler, Neural Tangent Kernel: Convergence and Generalization in Neural Networks (2018), the Jacobian kernel and squared-loss gradient flow. Lee et al., Wide Neural Networks of Any Depth Evolve as Linear Models Under Gradient Descent (2019), selected Taylor-linearization sections. Yang & Hu, Tensor Programs IV: Feature Learning in Infinite-Width Neural Networks (2021), selected parameterization and feature-learning examples; full tensor-program proofs are further reading.

Main paper: Yang et al., Tensor Programs V: Tuning Large Neural Networks via Zero-Shot Hyperparameter Transfer (NeurIPS 2021; arXiv 2022).

Recent development & discussion: Dey et al., Don't Be Lazy: CompleteP Enables Compute-Efficient Deep Transformers (NeurIPS 2025; revised 2026), selected width/depth-transfer and non-lazy-learning results. Supporting discussion: Haas et al., On the Surprising Effectiveness of Large Learning Rates under Standard Width Scaling (2025). Further theory reading: Chen et al., Global Convergence and Rich Feature Learning in L-Layer Infinite-Width Neural Networks under mu Parametrization (ICML 2025), result statements and assumptions only; this does not establish unrestricted finite-network convergence.

Presentation focus: Derive the Jacobian-kernel gradient-flow equation for squared loss, then use one small two-layer example to compare activation and update scaling in kernel-like and feature-learning regimes. Center the main paper on one learning-rate transfer experiment from a small network to a larger one. Use one CompleteP result to discuss why width transfer and depth transfer are different questions. Leave the full infinite-width and tensor-program proofs for further reading.

Topic 2 · Why training works: convergence, landscapes and stability

Topic family: Kernels, feature learning & optimization

Which assumptions let us prove that training reaches a global minimum, and how close are those proofs to practical networks?

Scope for one session: Two academic hours: derive PL-based loss decay and explain one restricted neural-network convergence argument. Reserve a short quadratic example and one plot for discrete-time stability. Full central-flow derivations and adaptive-optimizer analysis are further reading.

Background: Karimi, Nutini & Schmidt, Linear Convergence of Gradient and Proximal-Gradient Methods Under the Polyak-Lojasiewicz Condition (2016), the PL inequality and gradient-flow loss decay. Historical context: Choromanska et al., The Loss Surfaces of Multilayer Networks (2015; 2014 preprint), the spin-glass analogy and its assumptions. Optional contrast: Soudry & Hoffer, Exponentially vanishing sub-optimal local minima in multilayer neural networks (ICLR 2018), volumes of regions containing minima versus a training trajectory. Brief dynamics background: Cohen et al., Gradient Descent on Neural Networks Typically Occurs at the Edge of Stability (ICLR 2021), one oscillatory training plot and the quadratic stability threshold 2 / learning rate.

Main paper: Dana, Pillaud-Vivien & Bach, Convergence of Shallow ReLU Networks on Weakly Interacting Data (NeurIPS 2025). Gradient flow in finite shallow ReLU networks, weakly interacting inputs and a particular asymmetric randomized initialization.

Recent development & discussion: Petit, Poon & Peyré, On the global convergence of gradient flow for wide shallow models beyond homogeneous nonlinearities (2026). The result is conditional: if the studied Wasserstein gradient flow converges, its limit is globally optimal. Watch list: Soudry's publication page lists Global Convergence in Deep Networks via the Second Law of Thermodynamics (NeurIPS 2026); a linked manuscript was unavailable when this list was checked, so reading and claims remain TBD. Supporting dynamics comparison: Cohen et al., Understanding Optimization in Deep Learning with Central Flows (ICLR 2025), the gradient-descent illustration and authors' explanation. Further reading: Berthier, The Multiple Timescales of Gradient Descent on the Edge of Stability: A Perturbative Derivation of the Central Flow (2026); its formal derivation does not supply a global-convergence proof.

Presentation focus: Derive loss decay from a PL inequality, then explain how the main paper maintains a useful trajectory-wise condition. Inspect its data, initialization, width and gradient-flow assumptions. Contrast this proof with a quadratic discrete-time stability calculation and one edge-of-stability observation. Distinguish landscape geometry, convergence, oscillatory dynamics, solution selection and generalization. Central flows describe averaged dynamics; they are not a general global-convergence theorem. Keep the recent conditional result separate from unrestricted guarantees for finite deep networks trained with SGD or Adam. The complementary Topic 18 studies updates that need not be gradients and convergence through Lyapunov arguments. Topic 19 asks how useful gradients can be estimated without backpropagation and at what computational cost.

Topic 3 · State-space models and Mamba

Topic family: Sequence models & reasoning

How do recurrence, selectivity and structured matrices produce an efficient sequence model?

Background: Gu, Goel & Ré, Efficiently Modeling Long Sequences with Structured State Spaces (S4, 2021/22), selected sections on discretization and recurrence. Gu & Dao, Mamba: Linear-Time Sequence Modeling with Selective State Spaces (2023), selected sections on input-dependent dynamics.

Main paper: Dao & Gu, Transformers Are SSMs: Generalized Models and Efficient Algorithms Through Structured State Space Duality (Mamba-2, 2024).

Recent development & discussion: Lahoti et al., Mamba-3: Improved Sequence Modeling Using State Space Principles (2026).

Presentation focus: Unroll an SSM into a structured matrix and explain the restricted attention-SSM duality used by Mamba-2. Examine a state-tracking experiment from Mamba-3 and distinguish algorithmic complexity from measured hardware efficiency.

Topic 4 · In-context learning as an algorithm

Topic family: Sequence models & reasoning

How can a forward pass perform learning without changing the network's parameters?

Background: von Oswald et al., Transformers Learn In-Context by Gradient Descent (2023), selected sections on the linear-attention construction. Review least squares, ridge regression and Bayesian prediction for linear models.

Main paper: Akyürek et al., What Learning Algorithm Is In-Context Learning? Investigations with Linear Models (2023).

Recent development & discussion: Wurgaft et al., In-Context Learning Strategies Emerge Rationally (2025).

Presentation focus: Work through one regression-learning construction and compare gradient-descent, ridge and Bayesian predictors on a controlled task distribution. Separate existence constructions and observed behavior in trained synthetic-task models from claims about general language models.

Topic 5 · Test-time training (TTT)

Topic family: Sequence models & reasoning

What should a model learn at inference, and which objective makes those updates useful?

Scope for one session: Two academic hours: develop a recurrent TTT update, then examine end-to-end long-context training. Original distribution-shift TTT supplies the motivation and history; IT3 is supplementary reading.

Background: Sun, Wang, Liu, Miller, Efros & Hardt, Test-Time Training with Self-Supervision for Generalization under Distribution Shifts (ICML 2020), selected introduction and auxiliary-task formulation. Ba et al., Using Fast Weights to Attend to the Recent Past (2016), fast versus slow weights and associative memory.

Main paper: Sun et al., Learning to (Learn at Test Time): RNNs with Expressive Hidden States (2024). A model is the recurrent hidden state, and self-supervised gradient updates are the state transition.

Recent development & discussion: Tandon et al., End-to-End Test-Time Training for Long Context (2025). Next-token updates compress context into weights, with a meta-learned initialization. Supplementary adaptation example: Durasov, Shocher, Oner, Chechik, Efros & Fua, IT3: Idempotent Test-Time Training (ICML 2025); implementation.

Presentation focus: Start with the original 2020 adaptation problem, then work through a TTT-Linear update and its recurrent computation. Examine how end-to-end next-token training compresses context into weights, and explain the training and inference costs. IT3 provides a supplementary consistency-based adaptation example with its own training setup.

Topic 6 · Recurrent depth and latent reasoning

Topic family: Sequence models & reasoning

Can a model gain reasoning ability by repeatedly applying the same layers?

Background: Dehghani et al., Universal Transformers (ICLR 2019; 2018 preprint). Shared layers, recurrence over depth and adaptive computation.

Main paper: Geiping et al., Scaling up Test-Time Compute with Latent Reasoning: A Recurrent Depth Approach (NeurIPS 2025).

Recent development & discussion: Jolicoeur-Martineau, Less is More: Recursive Reasoning with Tiny Networks (2025). A short comparison with recursive refinement on puzzles.

Presentation focus: Trace one recurrent block, explain how it is trained across iteration counts, and assess performance against inference compute. Use tiny recursive models as a comparison. Distinguish iteration count, parameter count and demonstrated reasoning performance.

Topic 7 · JEPA and predictive world models

Topic family: Representations, world models & continual learning

When does prediction in representation space support understanding, dynamics and planning?

Background: Assran et al., Self-Supervised Learning from Images with a Joint-Embedding Predictive Architecture (I-JEPA, 2023), selected sections on masking, target representations and collapse prevention.

Main paper: Assran et al., V-JEPA 2: Self-Supervised Video Models Enable Understanding, Prediction and Planning (2025).

Recent development & discussion: Maes et al., LeWorldModel: Stable End-to-End Joint-Embedding Predictive Architecture from Pixels (2026).

Presentation focus: Trace the representation-prediction objective and action-conditioned planning loop, then analyze a planning result and its data requirements. Contrast target-network stabilization with regularizing the latent distribution; representation quality and accurate action-conditioned dynamics are separate questions.

Topic 8 · Online continual learning: stability, plasticity and replay

Topic family: Representations, world models & continual learning

How can a network keep adapting to a changing data stream while retaining useful knowledge?

Scope for one session: Two academic hours: derive OCAR's replay-constrained update, inspect one streaming experiment, and use CLA to connect prediction preservation with self-supervised representation alignment. A one-pass incoming stream can allow replay and bounded repeated updates on the current minibatch; it does not imply exactly one gradient step per observation.

Background: Chaudhry et al., On Tiny Episodic Memories in Continual Learning (2019): experience replay with limited memory. Kirkpatrick et al., Overcoming Catastrophic Forgetting in Neural Networks (PNAS 2017): elastic weight consolidation and Fisher-weighted protection. Optional mathematical background: Martens & Grosse, Optimizing Neural Networks with Kronecker-factored Approximate Curvature (ICML 2015), selected sections introducing the Fisher and one K-FAC block.

Main paper: Urettini & Carta, Online Curvature-Aware Replay: Leveraging 2nd Order Information for Online Continual Learning (ICML 2025). Focus on Sections 3-5: formulate the streaming update, constrain changes to predictions on replay data, and derive the curvature-aware step.

Recent development & discussion: Cignoni et al., CLA: Latent Alignment for Online Continual Self-Supervised Learning (CoLLAs 2026): compare representation alignment with labeled prediction constraints. Conceptual companion: Dohare et al., Loss of Plasticity in Deep Continual Learning (Nature 2024), which distinguishes forgetting old knowledge from losing the ability to learn new tasks; its broader continual-learning experiments are not all single-pass online protocols.

Presentation focus: Derive the local KL/Fisher constraint and the preconditioned update on a small example. Explain damping, memory representativeness and the difference between forgetting and loss of plasticity. Evaluate accuracy during the stream as well as at its end. Check what information each experiment provides: OCAR hides task boundaries, but its class-incremental replay-Fisher weighting uses the number of new classes per experience.

Topic 9 · Platonic representations and unpaired alignment

Topic family: Representations, world models & continual learning

Does shared representation geometry let us align independently trained models without paired alignment examples?

Scope for one session: Two academic hours: representation similarity, the Platonic hypothesis and one concrete unpaired alignment algorithm. Develop the Procrustes step on a small example; explain the full matching procedure without deriving every optimization subproblem.

Background: Kornblith et al., Similarity of Neural Network Representations Revisited (ICML 2019), centered kernel alignment. Huh et al., Position: The Platonic Representation Hypothesis (ICML 2024), the hypothesis and supporting experiments. Review an orthogonal Procrustes alignment using the SVD.

Main paper: Schnaus, Dagès, Cremers, Wang & Isola, Shared Geometry As A Rosetta Stone: Cross-Modal Alignment Without Paired Data (October 2026 preprint); the authors' interactive project illustrates coarse cross-modal alignment without pairs at the alignment stage.

Recent development & discussion: Gröger, Wen & Brbic, Revisiting the Platonic Representation Hypothesis: An Aristotelian View (ICML 2026). Use the critique to examine which definitions and comparisons support claims about shared representations.

Presentation focus: Treat Rosetta as a practical test of a consequence of the Platonic hypothesis. Explain its coarse geometric initialization and alternating assignment / Procrustes updates, then examine the broad semantic structure it recovers and the finer details it misses. Paired samples are still used for evaluation and the CKA analysis of when alignment succeeds. Distinguish useful geometric correspondence from universal convergence or canonical features.

Topic 10 · Hypernetworks and learning in weight space

Topic family: Weight space & interpretability

How can a neural network generate, read or edit another network's weights while respecting their symmetries?

Scope for one session: Two academic hours: a hidden-neuron permutation example, one equivariant weight-processing layer, a selected expressivity result and a brief adapter-generation application. Basic symmetry is integrated background; general canonicalization topology is further reading.

Background: Ha, Dai & Le, HyperNetworks (ICLR 2017), a network generating another network's parameters. Navon, Shamsian, Achituve, Fetaya, Chechik & Maron, Equivariant Architectures for Learning in Deep Weight Spaces (ICML 2023), selected permutation action and DWS-layer construction; the authors' tutorial is an accessible introduction. Hu et al., LoRA: Low-Rank Adaptation of Large Language Models (2021/22), selected low-rank update formulation. Symmetry primer: Cohen & Welling, Group Equivariant Convolutional Networks (ICML 2016), group actions and equivariance definitions only. Apply these definitions to hidden-neuron permutations rather than developing a separate rotation-equivariant architecture.

Main paper: Dayan, Eitan & Maron, On the Expressive Power of Permutation-Equivariant Weight-Space Networks (ICML 2026). Focus on the approximation framework and one universality result for fixed-architecture MLP weight spaces: distinguish properties of parameterizations from properties of the functions those parameters realize.

Recent development & discussion: Liu, Wang, Mao, Gelberg, Maron & Zhang, SHINE: A Scalable In-Context Hypernetwork for Mapping Context to LoRA in a Single Pass (ICML 2026), a contemporary application that converts context into adapters. Supporting application reading: Charakorn et al., Doc-to-LoRA: Learning to Instantly Internalize Contexts (ICML 2026), amortized context distillation. Further mathematical reading: Dym, Lawrence & Siegel, Equivariant Frames and the Impossibility of Continuous Canonicalization (ICML 2024), for the continuity issue in the group-action settings studied there; its theorem is distinct from the MLP weight-space expressivity result.

Presentation focus: Derive the hidden-neuron permutation action for a small MLP and construct a simple equivariant weight-processing layer. Explain why arbitrary flattening obscures this structure. Define a canonical representative using the same neuron-permutation example, outline one universality result under a general-position assumption, and inspect a degenerate counterexample. Keep the broader continuity and topology proofs as further reading. End with a short SHINE example contrasting direct adapter generation with iterative parameter updates. SHINE does not instantiate the DWS construction; keep the MLP expressivity results separate from the empirical claims about LLM adaptation.

Topic 11 · Model merging and pretrained weight neighborhoods

Topic family: Weight space & interpretability

Which useful capabilities lie near a pretrained checkpoint, and how can we recover or combine them?

Scope for one session: Two academic hours: introduce task vectors and weight averaging, then center the seminar on Neural Thickets and the difference between combining weights and ensembling predictions.

Background: Ilharco et al., Editing Models with Task Arithmetic (2023), task vectors relative to a shared pretrained checkpoint. Wortsman et al., Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time (ICML 2022), weight averaging versus prediction ensembles. Selected theory: Ortiz-Jimenez, Favero & Frossard, Task Arithmetic in the Tangent Space: Improved Editing of Pre-Trained Models (2023), the linearized model and weight disentanglement.

Main paper: Gan & Isola, Neural Thickets: Diverse Task Experts Are Dense Around Pretrained Weights (ICML 2026). RandOpt samples Gaussian weight perturbations, evaluates candidates on task data, selects experts and ensembles their predictions. See the authors' illustrated project.

Recent development & discussion: Cendra et al., Can Bayesian Optimization Efficiently Find a Strong Single Expert in Neural Thickets? (2026), a follow-up in low-dimensional search settings. Empirical comparison: Hitit, Girrbach & Akata, A Systematic Study of In-the-Wild Model Merging for Large Language Models (TMLR 2026; 2025 preprint), heterogeneous checkpoints and conflicting objectives.

Presentation focus: Use function-space linearization to explain the intuition behind task-vector addition. Then develop expert density and diversity around a pretrained checkpoint, including the probability of finding a useful perturbation after repeated sampling. Compare weight averaging, expert selection and prediction ensembling, accounting for task-data evaluations and inference cost. These neighborhood results do not establish that random initialization or unrestricted gradient descent succeeds.

Topic 12 · Mechanistic interpretability: are neural networks already clean?

Topic family: Weight space & interpretability

Are neural networks intrinsically entangled, or are we looking at them incorrectly?

Scope for one session: Week 10, two academic hours: one toy superposition derivation, the SAE objective, and the Cammarata-Gao debate about what counts as a faithful explanation of native neurons versus extracted features. Use selected foundation and Goodfire sections plus Gao's short response; Leask et al. remains optional. Dates and presenter: TBD.

Foundations: superposition and SAEs: Elhage et al., Toy Models of Superposition (2022): derive a small sparse-feature example and explain when sharing dimensions helps. Templeton et al., Scaling Monosemanticity: Extracting Interpretable Features from Claude 3 Sonnet (2024): the sparse-autoencoder objective, one recovered feature and one intervention. Distinguish a toy-model result from evidence about a trained language model.

Paired main discussion: Cammarata and Gao: Nick Cammarata / Goodfire, Neural Networks Are Already Clean (October 2026). Contemporary main discussion: read The Short Version, the cat/car case in Cars and Animals, and Zooming Out. Cammarata proposes that some apparent entanglement reflects shared generalization and imperfect tools, and advocates studying native neurons and circuits. Treat this fresh essay as a provisional working hypothesis, not established consensus. Paired main response: Leo Gao, Neural networks are not already clean (17 October 2026), a short methodological critique. Compare a plausible description of activating examples with an explanation that predicts responses and preserves downstream behavior when used to simulate activations. Discuss coverage, activation magnitudes and Gao's proposed downstream-KL test.

Optional critique: Leask et al., Sparse Autoencoders Do Not Find Canonical Units of Analysis (ICLR 2025). Use the abstract and selected SAE-stitching or meta-SAE results to examine completeness and atomicity. Noncanonical features may still be useful; this critique does not establish the Goodfire hypothesis.

Presentation focus: Derive how sparse features can share a low-dimensional representation and write the SAE reconstruction-plus-sparsity objective. Debate whether a multi-concept neuron indicates feature packing, a shared abstraction, or a measurement artifact. Choose one claimed neuron explanation and design a held-out test that could falsify it: predict activations from the explanation, substitute those values, and compare downstream output-distribution KL with an ablation baseline. Discuss missed inputs, activation magnitudes and the limited effect of individual neurons. Separate interpretability, completeness and uniqueness. Close by choosing units for Week 11 circuit tracing: native neurons, learned features or subspaces.

Suggested shared pre-reading: 30-45-minute audience assignment: Gao's short response in full, paired with Cammarata's The Short Version and On Simulation Scores sections. Use the Toy Models toy setup as background. Guiding questions: what observation would falsify a proposed neuron explanation, and does measuring downstream behavioral effects resolve the objections to activation-simulation scores? Final sections: TBD with the presenter and instructor.

Evidence status: October 2026: the Goodfire essay presents an informal, fresh and provisional hypothesis, not established consensus. Evaluate its examples and extrapolations separately; broader claims need independent tests. Toy-model superposition, useful SAE features and noncanonical decompositions can coexist. The readings frame a debate rather than settle it. Gao's response supplies a competing evaluation standard; it does not by itself prove superposition. His claims about the effects of weak activations await fuller evidence. Both contemporary essays are discussion readings, not a settled resolution.

Topic 13 · Mechanistic interpretability: circuit tracing

Topic family: Weight space & interpretability

How do chosen units of analysis support a circuit explanation, and how can we test that explanation causally?

Scope for one session: Week 11, two academic hours: retain one circuit-tracing case study and causal intervention, using Week 10 to question the choice of units. Dates and presenter: TBD.

Background: Elhage et al., A Mathematical Framework for Transformer Circuits (2021), selected residual-stream and path-composition background. Recall the superposition and SAE foundations from Week 10. Distinguish native neurons, residual-stream directions and learned features before drawing a graph.

Main paper: Ameisen et al., Circuit Tracing: Revealing Computational Graphs in Language Models (2025).

Recent development & discussion: Kamath et al., Tracing Attention Computation Through Feature Interactions (2025). Revisit Week 10's competing accounts of units of analysis when assessing which nodes and edges a tracing method exposes; the Goodfire hypothesis motivates a comparison, not a replacement for causal validation.

Presentation focus: Center the presentation on how a replacement model produces an attribution graph, one circuit case study and a causal intervention. Distinguish SAEs from the tracing method's replacement model. Make the Week 10 connection explicit: how would a graph and its explanation change if its nodes were native neurons, extracted features or subspaces? Assess omitted computations, reconstruction error and sensitivity to the decomposition. Interpreting a unit and validating its causal role are separate steps; a noncanonical unit can still support a useful, limited explanation.

Topic 14 · One-step generative models: Drifting and IMLE

Topic family: Generative models

How can matching real and generated samples train a generator that produces a sample in one pass?

Scope for one session: One comparative seminar of two academic hours: derive Drifting as the central sample-matching mechanism, then compare IMLE's nearest-neighbor assignments and ROMS-IMLE's practical changes.

Background: Yujia Li, Swersky & Zemel, Generative Moment Matching Networks (ICML 2015), selected kernel/MMD material. Ke Li & Jitendra Malik, Implicit Maximum Likelihood Estimation (2018), the data-to-generated nearest-neighbor assignment and regression algorithm.

Main paper: Deng, He Li, Tianhong Li, Du & He, *Generative Modeling via Drifting* (2026). Estimate an attraction-repulsion field from real and generated samples and fit frozen drifted targets.

Recent development & discussion: Chirag Vashist & Ke Li, ROMS-IMLE: A Minimalist Approach to Competitive Single-Step Generative Modelling (2026). Compare nearest-neighbor assignment, robust losses and intermediate-resolution supervision with the drifting update. Optional theory: Esteban-Casadevall et al., Kernel-Gradient Drifting Models (2026), including Max Welling. The authors' ROMS-IMLE method page provides additional explanation and experiments.

Presentation focus: Introduce sample-based distribution matching, derive the drifting field and frozen-target regression, and study one toy experiment. Compare IMLE's data-to-generated matching and ROMS-IMLE's practical changes, including the direction of nearest-neighbor assignment, robust losses and intermediate-resolution supervision. Keep zero-drift and maximum-likelihood claims tied to their assumptions. For the conservatism of drifting fields and stop-gradient dynamics, see Topic 18.

Topic 15 · Diffusion language models

Topic family: Generative models

How can denoising generate discrete text, and what changes when generation need not proceed left to right?

Background: Austin et al., Structured Denoising Diffusion Models in Discrete State-Spaces (2021), selected sections on transition matrices and absorbing-mask corruption.

Main paper: Sahoo et al., Simple and Effective Masked Diffusion Language Models (2024).

Recent development & discussion: Nie et al., Large Language Diffusion Models (LLaDA, 2025).

Presentation focus: Derive the masked-denoising objective and trace one sampling schedule, then compare refinement, parallelism and likelihood evaluation with autoregressive generation. The topic requires discrete corruption and language-specific evaluation beyond a continuous diffusion refresher.

Topic 16 · Oscillatory neurons and synchronization

Topic family: Graphs & neural architectures

Can neural representations bind information through synchronization?

Scope for one session: Exploratory candidate connecting neural computation with dynamical systems, including work by Max Welling.

Background: Strogatz, From Kuramoto to Crawford: exploring the onset of synchronization in populations of coupled oscillators (2000). Sections 2 and 3.1-3.3 on coupled phases and synchronization (author-hosted PDF).

Main paper: Miyato, Löwe, Geiger & Welling, Artificial Kuramoto Oscillatory Neurons (ICLR 2025; 2024 preprint).

Recent development & discussion: Iqbal et al., Spontaneous symmetry breaking and Goldstone modes for deep information propagation (2026 preprint). A related perspective on symmetry and persistent information propagation, rather than a direct analysis of AKOrN.

Presentation focus: Derive the two-oscillator intuition, trace the generalized neural update, and study one object-discovery or reasoning experiment. Use the Goldstone paper for a brief connection to depth and memory; its physics derivations are optional.

Topic 17 · GNNs: expressivity, information bottlenecks and graph transformers

Topic family: Graphs & neural architectures

What limits message passing, and how do connectivity and structural information change what a graph network can learn?

Scope for one session: Two academic hours: two concrete failure examples, one structural-encoding mechanism and one graph-transformer experiment. Expander connectivity supplies a short comparison; full higher-order expressivity and spectral theory are further reading.

Background: Sanchez-Lengeling et al., A Gentle Introduction to Graph Neural Networks (Distill 2021), a visual primer. Xu, Hu, Leskovec & Jegelka, How Powerful are Graph Neural Networks? (ICLR 2019), multiset aggregation, GIN and the 1-Weisfeiler-Leman expressivity ceiling. Alon & Yahav, On the Bottleneck of Graph Neural Networks and its Practical Implications (ICLR 2021), one over-squashing example.

Main paper: Stoll, Müller & Morris, Generalizable Insights for Graph Transformers in Theory and Practice (NeurIPS 2025), selected theory sections and one structural-encoding ablation.

Recent development & discussion: Wilson, Bechler-Speicher & Velickovic, Cayley Graph Propagation (2025), an intuitive expander-connectivity example and one propagation-cost comparison; the group-theory construction and spectral proofs are further reading. Ma et al., Plain Transformers Can be Powerful Graph Learners (TMLR 2026; 2025 preprint), selected structural-adaptation discussion. Further reading: Liang et al., Mitigating Over-Squashing in Graph Neural Networks by Spectrum-Preserving Sparsification (ICML 2025).

Presentation focus: Introduce message passing through a small graph. Work through one pair of graphs that 1-WL-style aggregation cannot distinguish and one narrow-cut graph that compresses distant information. Explain why expressivity, over-squashing and over-smoothing are different limitations. Center the main paper on how structural encodings affect graph transformers and one ablation. Use Cayley propagation briefly to compare changing connectivity with changing the model; avoid a second full presentation or claims that attention automatically solves either limitation.

Topic 18 · Stop-Gradient, Games, and Learning Dynamics

Topic family: Kernels, feature learning & optimization

When is a learning rule optimizing an objective, and how can we establish convergence when it is not?

Scope for one session: Two academic hours: 15 minutes on game vector fields, 10 on Lyapunov stability, 35 on the stopgrad regression principle and one flow-map case, 15 on a toy example or ablation, and 15 on assumptions and discussion. Core means selected sections for presenters, not three full papers for the audience.

Core foundations · selected sections: Letcher et al., Differentiable Game Mechanics (2019), Sections 2.1-2.4: simultaneous game updates, the symmetric/antisymmetric Jacobian decomposition, potential and Hamiltonian games. Balduzzi et al., Smooth markets: A basic mechanism for organizing gradient-based learners (ICLR 2020), Section 5.2, especially Theorem 5: aggregate forecasts act as Lyapunov-like functions near fixed points of smooth markets with a negative-definite symmetric Jacobian part. This supplies the explicit Lyapunov argument.

Core main paper: Shen et al., How I learned to stop worrying and love StopGrads: Stationarity, Convergence, and a case study on Flow Map Learning (2026). Read Sections 2-3 and 5; use one example from Section 4. Derive the stopgrad regression principle, distinguish stationary-point characterization from convergence, and explain one functional semi-gradient-flow result. Full PDE proofs are outside the session.

Optional extensions: Chen & He, Exploring Simple Siamese Representation Learning (SimSiam, 2020 preprint / CVPR 2021), Sections 3 and 4.1: computational graph and stop-gradient ablation. Ponce et al., Dual Perspectives on Non-Contrastive Self-Supervised Learning (2025; revised 2026), Sections 3.2 and 4.2: non-integrability and local stability in linear SSL via linearization, rather than a constructed Lyapunov function. Franz et al., Drifting Fields are not Conservative (2026), Sections 3-4: normalization can break conservatism; a modified normalization restores it. Choose one extension for discussion.

Presentation focus: Treat the update as a vector field. Derive $F(x,y) = (x-y, x+y)$ from $(x-sg(y))^2/2 + (y+sg(x))^2/2$, where sg stops differentiation; its Jacobian is asymmetric. Under $\dot{\theta} = -F(\theta)$, $V = (x^2+y^2)/2$ decreases strictly away from the origin, although F is not its gradient. Contrast this damped rotation with a pure rotation. Connect the example to game dynamics and semi-gradients, then state the assumptions of the main paper's flow-map convergence result. Compare with Topic 19: estimating a gradient without backpropagation is a different issue from whether an update field admits a scalar potential.

Suggested shared pre-reading: Suggested shared pre-reading (30-45 minutes): Differentiable Game Mechanics, Section 2.3 and the two-player rotation example in Section 2.2. Guiding questions: how can a game update fail the symmetric-Jacobian test, and why does a scalar quantity conserved by rotation fail to establish attraction? Final shared pre-reading and selected sections: TBD with the presenters and instructor.

Guarantee boundaries: A scalar potential generates a conservative field; a Lyapunov function can establish stability without generating that field. Smooth Markets gives a local continuous-time result under its structured-game assumptions. StopGrads studies functional semi-gradient flow, with perfect flow-matching initialization for some objectives; this is not an unrestricted finite-network SGD guarantee. Quasipotentials concern stochastic dynamics and rare transitions; none of these readings supplies a quasipotential treatment.

Topic 19 · Training without backpropagation: perturbations and credit assignment

Topic family: Kernels, feature learning & optimization

How can random perturbations estimate useful learning directions, and why does replacing backpropagation not automatically make training cheaper?

Scope for one session: Two academic hours: 15 minutes on node perturbation and evolution strategies, 25 on the Gaussian estimator, 20 on Dust, 15 on evidence and compute accounting, and 15 on discussion. A natural follow-on to Topic 2 on training dynamics; Topic 18 supplies a comparison with nonconservative update fields.

Foundations: node perturbation and Gaussian search: Werfel, Xie & Seung, Learning Curves for Stochastic Gradient Descent in Linear Feedforward Networks (NIPS 2003): compare direct gradients, node perturbation and weight perturbation in a linear model, with attention to noise dimension and learning rate. Salimans et al., Evolution Strategies as a Scalable Alternative to Reinforcement Learning (2017): selected Gaussian-smoothed objective and gradient-estimator background. The reinforcement-learning experiments are context, not a transformer-training comparison.

Main case study: Dust: Dahal, Mandal, Gülbahar & Vegesna, Dust: Pretraining Transformers Without Backpropagation (October 2026 research report). Center the seminar on Section 2, then examine Sections 3.1-3.2 and 4.2. Dust perturbs linear-layer outputs independently at each token, estimates activation errors from loss-weighted noise, and forms weight updates by outer products with layer inputs. Attention internals use estimated attention-output errors for local credit assignment. Explain the token axis and the repeated-draw axis separately: each batch is evaluated with different noise, with multiple replicas processed together.

Optional comparison and implementation: Optional methodological comparison: Ren, Kornblith, Liao & Hinton, Scaling Forward Gradient With Local Losses (ICLR 2023), activity perturbation, estimator variance and local losses. Distinguish directional derivatives computed by forward-mode differentiation from Dust's loss-only evaluations. Optional implementation check: Dust code, especially direct_errors and reassemble, to inspect repeated inputs, per-token centering within draw chunks, and the final weight update. Use one comparison, not a second full paper presentation.

Presentation focus: For a standard Gaussian vector a, use E[a a^T] = I and a first-order loss expansion to derive why averaging (L(y + sigma a) - b) a / sigma estimates an activation gradient; b is a fixed baseline in this derivation. Then establish the exact Gaussian-smoothed-loss identity and the local weight conversion g x^T for y = W x. Explain why the reward sign reverses, why independent perturbations cancel cross terms in expectation, and why finite populations still have variance. For Dust, distinguish activation noise from actual weight noise and batched evaluation from computational cost. Examine one gradient-alignment curve and one training comparison, then specify a compute-matched experiment including cached partial forwards, extra local draws, tuning and hardware. Keep equal-token and equal-compute conclusions separate.

Suggested shared pre-reading: Suggested shared pre-reading (30-45 minutes): Dust Section 2 and the setup in Section 3.1. Guiding questions: what is held fixed and what is averaged in one update, and what additional evidence would justify replacing backpropagation under a fixed compute budget? Final shared reading and sections: TBD with the presenter and instructor.

Estimator and evidence boundaries: Backpropagation computes gradients; SGD or Adam uses them. Dust changes the estimator while retaining gradient-based updates. Finite noise smooths the loss; discounting future-token credit and using local surrogates introduce further approximation. More draws reduce sampling variance but do not remove all bias or guarantee descent. Centering M independent draws by their own mean also introduces a factor of 1 - 1/M in the ideal estimator. Dust reports some gains at matched training-token budgets using substantially more compute; this does not establish a compute-matched advantage over backpropagation. Treat this October 2026 report as an emerging result, not an established practical replacement.

Course website: Deep Deep Learning