Topics:

- Bias/Fairness
- Security discussion
- Subfields not covered
- Open directions in Deep Learning

CS 4644-DL / 7643-A
ZSOLT KIRA

Administrative

- Projects! Due 05/04

- CIOS is open! https://b.gatech.edu/cios
  - Please fill out and provide comments!

Bias & Fairness

ML and Fairness

- AI effects our lives in many ways
- Widespread algorithms with many small interactions
  - e.g. search, recommendations, social media
- Specialized algorithms with fewer but higher-stakes interactions
  - e.g. medicine, criminal justice, finance
- At this level of impact, algorithms can have unintended consequences
- Low classification error is not enough, need fairness

BUSINESS NEWS OCTOBER 10, 2018 / 3:12 AM / 6 MONTHS AGO

Amazon scraps secret AI recruiting tool that showed bias against women

Jeffrey Dastin 8 MIN READ

SAN FRANCISCO (Reuters) - Amazon.com Inc's (AMZN.O) machine-learning specialists uncovered a big problem: their new recruiting engine did not like women.

The team had been building computer programs since 2014 to review job applicants' resumes with the aim of mechanizing the search for top talent, five people familiar with the effort told Reuters.

Automation has been key to Amazon's e-commerce dominance, be it inside warehouses or driving pricing decisions. The company's experimental hiring tool used artificial intelligence to give job candidates scores ranging from one to five stars - much like

Slide By Aaron Roth

INDEPENDENT

GOOGLE'S ALGORITHM SHOWS PRESTIGIOUS JOB ADS TO MEN, BUT NOT TO WOMEN

REUTERS

Business Markets World Politics TV More

BUSINESS NEWS OCTOBER 9, 2018 / 8:12 PM / 5 MONTHS AGO

Amazon scraps secret AI recruiting tool that showed bias against women

Jeffrey Dastin

8 MIN READ

The New York Times

Facebook Engages in Housing Discrimination With Its Ad Practices, U.S. Says

By Katie Benner, Glenn Thrush and Mike Isaac

March 28, 2019

168

MIT Technology Review

Intelligent Machines

How to Fix Silicon Valley's Sexist Algorithms

Computers are inheriting gender bias implanted in language data sets—and not everyone thinks we should correct it.

PRO PUBLICA

Machine Bias

There's software used across the country to predict future criminals. And it's biased against blacks.

Slide By Aaron Roth

Machine Learning and Social Norms

- Sample norms: privacy, fairness, transparency, accountability…
- Possible approaches
  - "traditional": legal, regulatory, watchdog
  - *Embed* social norms in data, algorithms, models
- Case study: privacy-preserving machine learning
  - "single", strong, definition (differential privacy)
  - almost every ML algorithm has a private version
- Fair machine learning
  - not so much…
  - impossibility results

(Un)Fairness Where?

- Data (input)
  - e.g. more arrests where there are more police
  - Label should be "committed a crime", but is "convicted of a crime"
  - try to "correct" bias
- Models (output)
  - e.g. discriminatory treatment of subpopulations
  - build or "post-process" models with subpopulation guarantees
  - equality of false positive/negative rates; calibration
- Algorithms (process)
  - learning algorithm *generating* data through its decisions
  - e.g. don't learn outcomes of denied mortgages
  - lack of clear train/test division
  - design (sequential) *algorithms* that are fair

When the *training data* we collect does not contain representative samples of the true distribution.

Examples:

If we use data gathered from smart phones, we would likely be underestimating poorer and older populations.

ImageNet (a very popular image dataset) with 1.2million images. About 45% of these images were taken in the US and the majority of the rest in North America and Western Europe. Only about 1% and 2.1% of the images come from China and India respectely.

Often we are gathering data that contains (noisy) proxies of characteristics of interest. Some examples:

- Financial responsibility → Credit Score
- Crime Rate → Arrest Rate
- Intelligence → SAT Score

If these measurements are not measured equally across groups or places (or aren't relevant to the task at hand), this can be another source of bias.

Measurement Bias

Examples:

- If factory workers are monitored more often, more errors are spotted. This can result in a **feedback loop** to encourage more monitoring in the future.
  - Same principles at play with predictive policing. Minoritized communities were more heavily policed in the past, which causes more instances of documented crime, which then leads to more policing in the future.

- Women are more likely to be misdiagnosed (or not diagnosed) for conditions where self-reported pain is a symptom. In this case aspect of our data “diagnosed with X” is a biased proxy for “has condition X”.
- The feature we measure is a poor representation of the quality of interest (e.g., SAT score doesn’t actually measure intelligence)

What does it mean for a model to be fair or unfair? Can we come up with a numeric way of measuring fairness?

Lots of work in the field of ML and fairness is looking into mathematical definitions of fairness to help us spot when something might be unfair.

- There is not going to be one central definition of fairness, as each definition is a mathematical statement of which behaviors are/aren't allowed.
- Different definitions of fairness can be contradictory!

ML and Fairness

- Fairness is morally and legally motivated
- Takes many forms
- Criminal justice: recidivism algorithms (COMPAS)
  - Predicting if a defendant should receive bail
  - Unbalanced false positive rates: more likely to wrongly deny a black person bail

Table 1: ProPublica Analysis of COMPAS Algorithm

	White	Black
Wrongly Labeled High-Risk	23.5%	44.9%
Wrongly Labeled Low-Risk	47.7%	28.0%

https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing

Why Fairness is Hard

- Suppose we are a bank trying to fairly decide who should get a loan
  - i.e. Who is most likely to pay us back?
- Suppose we have two groups, A and B (the sensitive attribute)
  - This is where discrimination could occur
- The simplest approach is to remove the sensitive attribute from the data, so that our classier doesn't know the sensitive attribute. Often called **"Fairness through unawareness"**

Table 2: To Loan or Not to Loan?

Age	Gender	Postal Code	Req Amt	A or B?	Pay
46	F	M5E	$300	A	1
24	M	M4C	$1000	B	1
33	M	M3H	$250	A	1
34	F	M9C	$2000	A	0
71	F	M3B	$200	A	0
28	M	M5W	$1500	B	0

Why Fairness is Hard

- However, if the sensitive attribute is correlated with the other attributes, this isn't good enough
- It is easy to predict race if you have lots of other information (e.g. home address, spending patterns)
- More advanced approaches are necessary

Table 3: To Loan or Not to Loan? (masked)

Age	Gender	Postal Code	Req Amt	A or B?	Pay
46	F	M5E	$300	?	1
24	M	M4C	$1000	?	1
33	M	M3H	$250	?	1
34	F	M9C	$2000	?	0
71	F	M3B	$200	?	0
28	M	M5W	$1500	?	0

Doesn't work in practice. This does not prevent historical or measurement bias. Protected attributes can be unintentionally inferred from other, related attributes (e.g., in some cities, zip code can be deeply correlated with race).

Definitions of Fairness – Group Fairness

- So we've built our classier . . . how do we know if we're being fair?
- One metric is demographic parity | requiring that the same percentage of A and B receive loans
  - What if 80% of A is likely to repay, but only 60% of B is?
  - Then demographic parity is too strong
- Could require equal false positive/negative rates
  - When we make an error, the direction of that error is equally likely for both groups

$$P(loan|no\ repay, A) = P(loan|no\ repay, B)$$
$$P(no\ loan|would\ repay, A) = P(no\ loan|would\ repay, B)$$

- These are definitions of group fairness
- Treat different groups equally"

Definitions of Fairness – Individual Fairness

- Also can talk about individual fairness | “Treat similar examples similarly"
- Learn fair representations
  - Useful for classification, not for (unfair) discrimination
  - Related to domain adaptation
  - Generative modelling/adversarial approaches

(a) Unfair representations

(b) Fair(er) representations

Figure 1: “The Variational Fair Autoencoder” (Louizos et al., 2016)

Conclusion

- This is an exciting field, quickly developing
- Central definitions still up in the air
- AI moves fast | lots of (currently unchecked) power
- Law/policy will one day catch up with technology
- Those who work with AI should be ready
  - **Think about implications of what you develop!**

Malwarebytes | Labs

AI, NEWS

Mythos: An AI tool too powerful for public release

by Pieter Arntz | April 20, 2026

The Risk Landscape

Alignment & Control

Reward hacking / specification gaming
Goal misgeneralization
Corrigibility: can we turn it off?
RLHF: aligning to preferences ≠ aligning to values

Cybersecurity

Adversarial attacks (fooling vision, NLP)
AI-powered phishing & social engineering
Autonomous vulnerability discovery
Model extraction & data poisoning

Mythos & Narrative

The Terminator vs. the paperclip maximizer
"AI consciousness" — hype or horizon?
Anthropomorphism: we say "it thinks"
Media narratives shape policy & funding

Societal & Systemic

Deepfakes & epistemic erosion
Concentration of power (who builds, who benefits?)
Labor displacement vs. augmentation
Bias amplification at scale

Separate into groups

Goal: Identify largest risks of increasing AI capabilities, and potential ways to mitigate them.

10 minutes discussion

Assign note-taker for debrief

Debrief

Parting Thoughts

Georgia Tech

Deep Learning Fundamentals

Linear classification
Loss functions
Optimization
Optimizers
Backpropagation
Computation Graph
Multi-layer Perceptrons

Neural Network Components and Architectures

Hardware & software
Convolutions
Convolution Neural Networks
Pooling
Activation functions
Batch normalization
Transfer learning
Data augmentation
Architecture design
RNN/LSTMs
Attention & Transformers

Applications & Learning Algorithms

Semantic & instance Segmentation
Reinforcement Learning
Large-language Models
Variational Autoencoders
Diffusion Models
Generative Adversarial Nets
Self-supervised Learning
Vision-Language Models
VLM for Robotics

We Learned a Lot!

Georgia Tech

Some existing works not covered…

Current / Past

- Graph neural networks
- Meta-learning
- AutoML
- 3D perception & reconstruction / NeRFs
  - Neural Radiance Fields
- AI for Tabular data, time-series, etc.
- Beyond supervised learning: Semi-supervised, domain adaptation, zero/one/few-shot learning
- Embodied AI & Embodied question answering
- Adversarial Learning
- Continual/lifelong learning without forgetting
- World modeling, learning intuitive/physics models
- Reasoning, Planning, Search
- Neural Theorem Proving, induction & synthesis
- AI for science
- MLSys and MLOps
- Evaluation…
- Alignment
- Security
- ……

Self-Supervised Learning: Three Paradigms

Contrastive Learning:

SimCLR (Chen et al., 2020): Augment, encode, contrast

MoCo v3 (2021): Momentum encoder + ViT

CLIP (Radford et al., 2021): Vision-language contrastive

Non-Contrastive:

BYOL (2020): Asymmetric architecture, no negatives

Barlow Twins (2021): Cross-correlation objective

VICReg (2022): Variance-invariance-covariance

Masked Prediction:

MAE (He et al., 2022): Mask 75%, reconstruct pixels

BEiT (2022): Discrete visual tokens

JEPA (Joint Embedding Predictive Architecture):

I-JEPA (2023): Predict abstract reps, not pixels

V-JEPA (2024): Video spatio-temporal masking

DINOv2 (Meta, 2024): Universal visual features, key vision foundation model

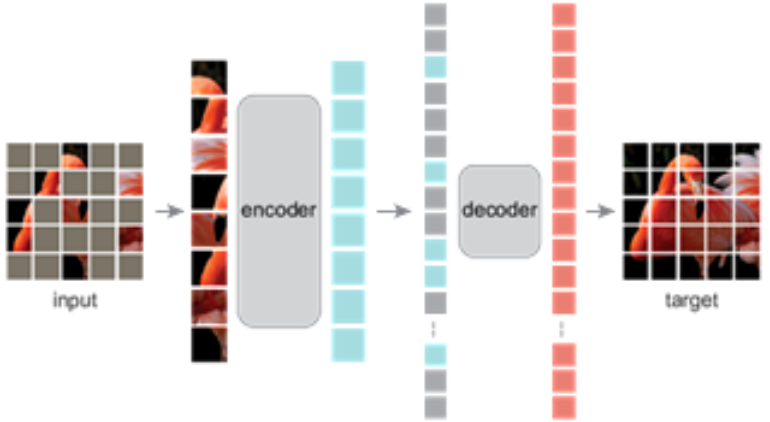


Figure 1. **Our MAE architecture**. During pre-training, a large random subset of image patches (*e.g.*, 75%) is masked out. The encoder is applied to the small subset of *visible patches*. Mask tokens are introduced *after* the encoder, and the full set of encoded patches and mask tokens is processed by a small decoder that reconstructs the original image in pixels. After pre-training, the decoder is discarded and the encoder is applied to uncorrupted

He et al., "Masked Autoencoders Are Scalable Vision Learners" (CVPR 2022) | Oquab et al., "DINOv2" (TMLR 2024) | Assran et al., "I-JEPA" (CVPR 2023)

Object Detection: Key Architectures

Key Architectures

Two-Stage Detectors:

R-CNN -> Fast R-CNN -> Faster R-CNN (Ren et al., 2015)

Region Proposal Network + classifier head

Cascade R-CNN (Cai & Vasconcelos, 2018)

Transformer-Based (DETR Family):

DETR (Carion et al., 2020): Set prediction with transformers

RT-DETR (Zhao et al., 2023): Real-time, ~54.8 mAP

Co-DETR (Zong et al., 2023): 66.0 mAP on COCO

Single-Stage (YOLO Family):

YOLOv9 (2024): PGI + GELAN architecture

YOLOv10 (2024): NMS-free dual assignments, 53.4 mAP

YOLO11 (Ultralytics, 2024): Latest practical release

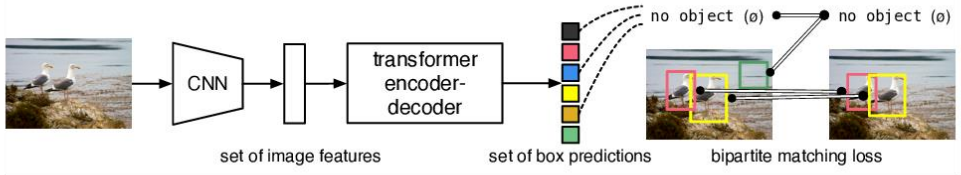


Fig. 1: DETR directly predicts (in parallel) the final set of detections by combining a common CNN with a transformer architecture. During training, bipartite matching uniquely assigns predictions with ground truth boxes. Prediction with no match should yield a "`no object`" (∅) class prediction.

We streamline the training pipeline by viewing object detection as a direct set prediction problem. We adopt an encoder-decoder architecture based on transformers [47], a popular architecture for sequence prediction. The self-attention mechanisms of transformers, which explicitly model all pairwise interactions between elements in a sequence, make these architectures particularly suitable for specific constraints of set prediction such as removing duplicate predictions.

Our DEtection TRansformer (DETR, see Figure 1) predicts all objects at

Carion et al., "End-to-End Object Detection with Transformers" (ECCV 2020) | Ren et al., "Faster R-CNN" (NeurIPS 2015) | Wang et al., "YOLOv10" (2024)

Image Segmentation: Segment Anything and Beyond

SAM (Kirillov et al., Meta, 2023):
- Promptable segmentation foundation model
- Trained on SA-1B dataset (1B+ masks)
- Zero-shot transfer to arbitrary segmentation

SAM 2 (Ravi et al., Meta, 2024):
- Extended to video with streaming memory
- SA-V dataset: 51K videos, 643K masklets

Unified Architectures:
- Mask2Former (2022): Panoptic/instance/semantic
- OneFormer (2023): Task-conditioned queries

Open-Vocabulary:
- Grounded-SAM = Grounding DINO + SAM
- Florence-2 (Microsoft, 2024): Unified vision FM

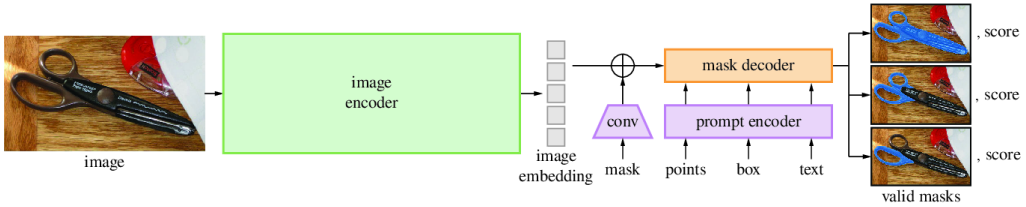


Figure 4: Segment Anything Model (SAM) overview. A heavyweight image encoder outputs an image embedding that can then be efficiently queried by a variety of input prompts to produce object masks at amortized real-time speed. For ambiguous prompts corresponding to more than one object, SAM can output multiple valid masks and associated confidence scores.

3. Segment Anything Model ... loss [15, 45, 64] over masks. To rank masks, the model pre-

Kirillov et al., "Segment Anything" (ICCV 2023) | Ravi et al., "SAM 2" (2024) | Liu et al., "Grounding DINO" (ECCV 2024)

Neural Scene Representations: NeRF to Gaussian Splatting

Point Cloud Processing:

PointNet (Qi et al., 2017): First direct point cloud net

PointNet++ (2017): Hierarchical set abstraction

Point Transformer v3 (2024): SOTA on ScanNet, nuSce

NeRF (Mildenhall et al., 2020):

MLP maps (x,y,z,dir) -> (color, density)

Instant-NGP (2022): Hash encoding, hours -> seconds

Zip-NeRF (2023): Anti-aliased + fast training

3D Gaussian Splatting (Kerbl et al., 2023):

Anisotropic 3D Gaussians + differentiable rasterizer

Real-time rendering (>100 FPS) at NeRF quality

Largely supplanted NeRF for real-time applications

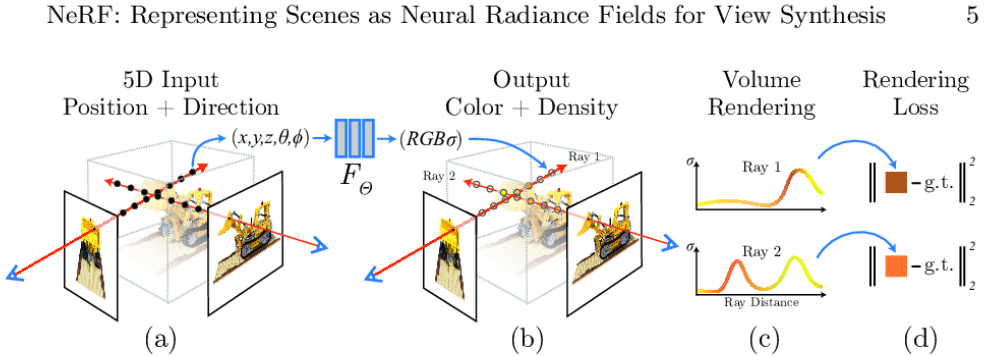


Fig. 2: An overview of our neural radiance field scene representation and differentiable rendering procedure. We synthesize images by sampling 5D coordinates

Mildenhall et al., "NeRF" (ECCV 2020) | Kerbl et al., "3D Gaussian Splatting" (SIGGRAPH 2023) | Wu et al., "Point Transformer V3" (CVPR 2024)

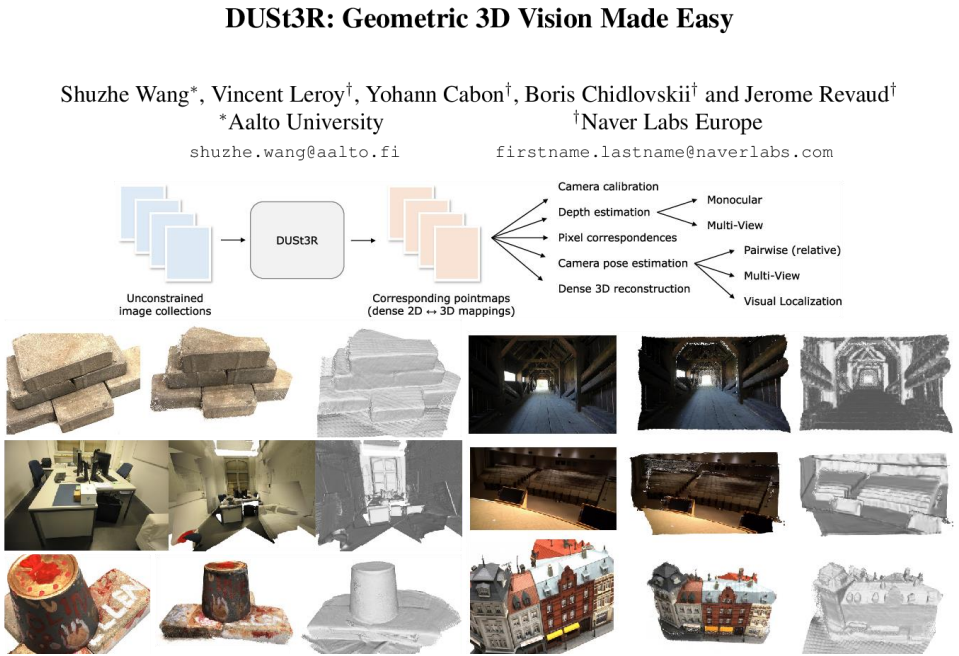
3D Foundation Models & Implicit Representations

3D Generation Foundation Models:
- Point-E (OpenAI, 2022): Text -> 3D point cloud
- Shap-E (OpenAI, 2023): Text/image -> implicit 3D
- LRM (Hong et al., 2023): Single image -> NeRF in ~5s

3D Reconstruction:
- DUSt3R (Naver, 2024): Dense 3D from uncalibrated pairs
- MASt3R (2024): + feature matching heads

Implicit Representations:
- DeepSDF (2019): Continuous signed distance function
- Occupancy Networks (2019): Resolution-independent
- NeuS/NeuS2: SDF from volume rendering

Trend: Feed-forward transformers replacing per-scene opt.



Wang et al., "DUSt3R: Geometric 3D Vision Made Easy" (CVPR 2024) | Hong et al., "LRM" (ICLR 2024)

Time Series: Transformers & The Linear Model Debate

Transformer-Based Methods:

Informer (Zhou et al., 2021, AAAI Best Paper): ProbSparse attention, O(L log L)

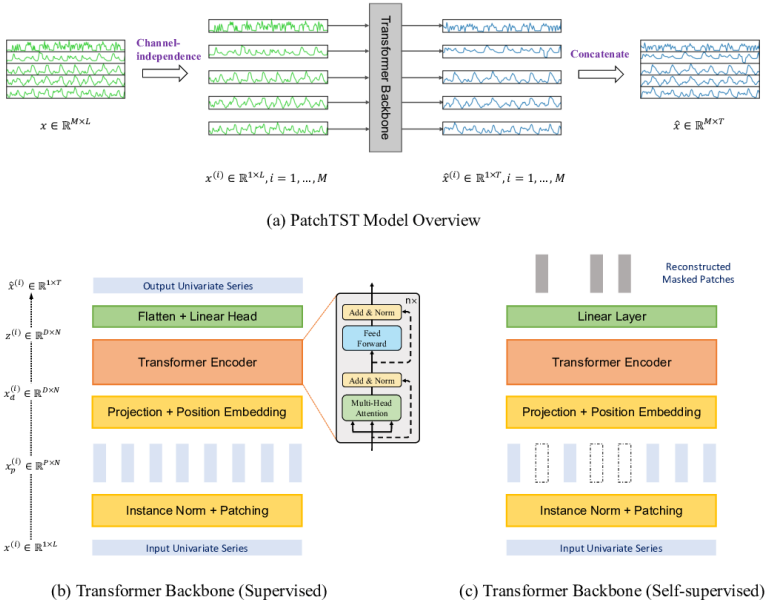
Autoformer (2021): Decomposition + auto-correlation

PatchTST (Nie et al., 2023): Patch-based tokens, channel-independent. Strong results

iTransformer (2024): Inverted - variates as tokens

"Are Linear Models Enough?" (Zeng et al., 2023): DLinear beat many transformers! Argued attention's permutation-invariance fails at temporal order.

Other: N-BEATS (2020), TimesNet (2023), TSMixer (2023)



(a) PatchTST Model Overview

(b) Transformer Backbone (Supervised)

(c) Transformer Backbone (Self-supervised)

Figure 1: PatchTST architecture. (a) Multivariate time series data is divided into different channels. They share the same Transformer backbone, but the forward processes are independent. (b) Each channel univariate series is passed through instance normalization operator and segmented into patches. These patches are used as Transformer input tokens. (c) Masked self-supervised representation learning with PatchTST where patches are randomly selected and set to zero. The model will reconstruct the masked patches.

Nie et al., "A Time Series is Worth 64 Words" (ICLR 2023) | Zeng et al., "Are Transformers Effective for Time Series Forecasting?" (AAAI 2023)

Time Series Foundation Models (2024-2026)

Major Foundation Models:

TimesFM (Google, 2024): 200M-param decoder-only, strong zero-shot forecasting

Chronos (Amazon, 2024): Tokenizes values into bins, fine-tunes T5 language models

Lag-Llama (2024): Probabilistic forecasting

Moirai (Salesforce, 2024): Any-variate attention

Multi-Task Models:

MOMENT (CMU, 2024): Classification, forecasting, anomaly detection, imputation

TimeGPT (Nixtla, 2023): First commercial FM API

Trend: Task-specific -> foundation models with zero-shot

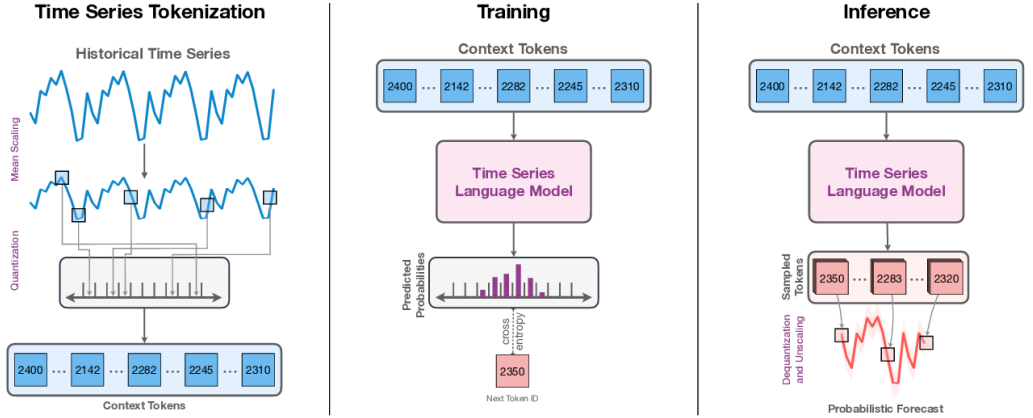


Figure 1: High-level depiction of CHRONOS. (**Left**) The input time series is scaled and quantized to obtain a sequence of tokens. (**Center**) The tokens are fed into a language model which may either be an encoder-decoder or a decoder-only model. The model is trained using the cross-entropy loss. (**Right**) During inference, we autoregressively sample tokens from the model and map them back to numerical values. Multiple trajectories are sampled to obtain a predictive distribution.

For the development of a useful general-purpose time series forecasting model, the scarcity of publicly available time series datasets, both in quantity and quality, is arguably more critical than the modeling framework. In addition to the comprehensive collection of public datasets we used to train CHRONOS, a central aspect of our approach is the integration of data augmentation strategies, including TSMixup and KernelSynth. TSMixup randomly samples a set of base time series from different training datasets, and

Ansari et al., "Chronos: Learning the Language of Time Series" (2024) | Das et al., "TimesFM" (ICML 2024)

Speech & Audio: From ASR to Universal Models

Recognition & Foundation Models

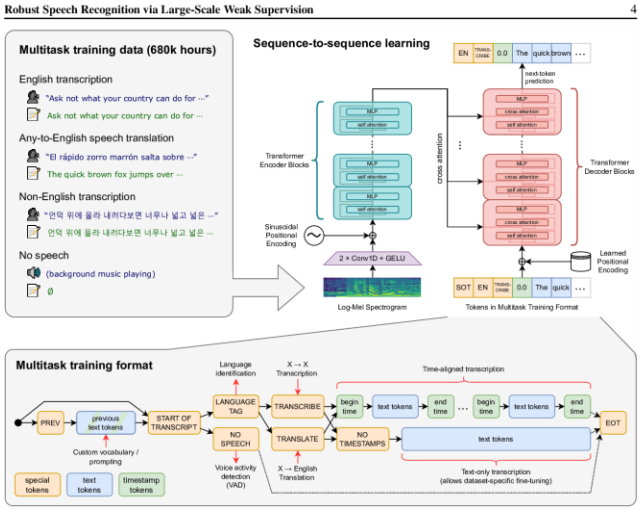
Whisper (Radford et al., 2022): 680K hrs, robust multilingual

Wav2Vec 2.0 (2020): Self-supervised speech backbone

HuBERT (2021): Masked speech prediction

SeamlessM4T (Meta, 2023): Speech/text across ~100 langs

USM (Google, 2023): 2B params, 300+ languages



Synthesis & Generation

VALL-E (Microsoft, 2023): TTS as language modeling, 3-sec clone

F5-TTS (2024): Flow matching | Parler-TTS (2024): NL style

MusicGen (Meta, 2023): Transformer text -> music

AudioLDM 2 (2023): Latent diffusion text -> audio

Stable Audio (Stability AI, 2023): Variable-length music

Radford et al., "Robust Speech Recognition via Large-Scale Weak Supervision" (ICML 2023) | Wang et al., "VALL-E" (2023)

Tabular Deep Learning: Models & The Tree-vs-DL Debate

Key Architectures

TabNet (Arik & Pfister, 2021): Attention-based feature selection with sequential decision steps

FT-Transformer (Gorishniy et al., 2021): Standard Transformer on per-feature embeddings

SAINT (2021): Inter-sample + inter-feature attention

TabR (2023): Retrieval-augmented, looks up similar training examples at inference

TabPFN (2023): Single forward pass replaces training!

The Debate & Foundation Models

"Why do tree-based models still outperform DL on tabular data?" (Grinsztajn et al., 2022):

- DL struggles with uninformative features
- Irregular target functions favor trees
- Non-rotationally-invariant patterns

Foundation Models:

TabPFN v2 (2025): Competitive with tuned XGBoost

CARTE (2024): Pre-trained on 1000s of tables

Status: XGBoost/LightGBM still default for production

Grinsztajn et al., "Why do tree-based models still outperform deep learning on tabular data?" (NeurIPS 2022) | Gorishniy et al., "Revisiting Deep Learning Models for Tabular Data" (NeurIPS 2021)

Video: Understanding and Generation

Understanding & Classification

TimeSformer (Bertasius et al., 2021): Divided space-time attention for video classification

VideoMAE (Tong et al., 2022): Masked autoencoding for self-supervised video pre-training

InternVideo2 (Wang et al., 2024): Unified video FM

Video-LLaVA (2023): Video encoder + LLM

Key Challenges:

Temporal consistency over long sequences

Quadratic cost of spatiotemporal attention

Physics-aware generation & evaluation

Generation Models

Sora (OpenAI, 2024): Minute-long coherent video, diffusion transformers on spacetime patches

Runway Gen-3 Alpha (2024): Commercial high-fidelity

Kling (Kuaishou, 2024): Physically consistent clips

MovieGen (Meta, 2024): Long video + audio

Architecture Evolution:

Early: 3D CNNs (C3D, I3D, SlowFast)

Mid: Video Transformers (ViViT, TimeSformer)

Current: Diffusion Transformers (DiT) for gen, unified video-language models for understanding

Bertasius et al., "Is Space-Time Attention All You Need?" (ICML 2021) | Polyak et al., "Movie Gen" (Meta, 2024) | Brooks et al., "Video generation models as world simulators" (OpenAI, 2024)

Efficient Architectures & Inference Optimization

State Space Models (Alternative to Transformers):

Mamba (Gu & Dao, 2023): Selective SSM, linear-complexity sequence modeling

Mamba-2 (2024): Further efficiency gains

Key: O(n) vs O(n^2) for transformers

Hybrids: Jamba, Zamba combine SSM + attention

Flash Attention (Dao et al., 2022-23):

IO-aware tiling, near-linear attention memory

Now default in all major frameworks

Speculative Decoding (Leviathan et al., 2023):

Small draft model + large verifier = 2-3x speedup

NAS: Largely superseded by scaling laws; niche for edge

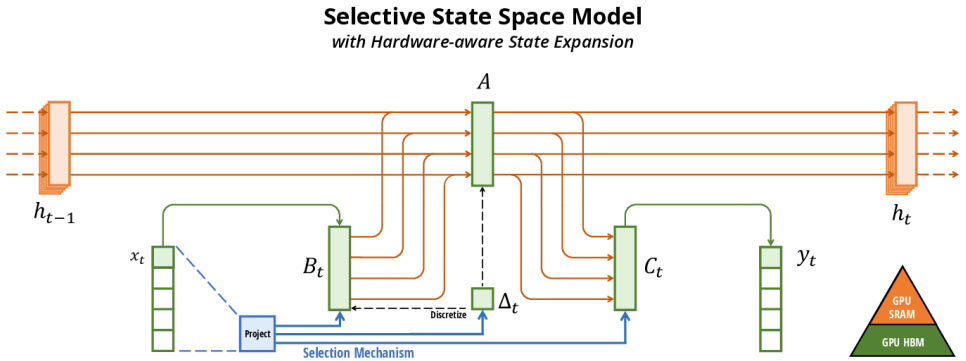


Figure 1: (**Overview**.) Structured SSMs independently map each channel (e.g. $D = 5$) of an input x to output y through a higher dimensional latent state h (e.g. $N = 4$). Prior SSMs avoid materializing this large effective state (DN, times batch size B and sequence length L) through clever alternate computation paths requiring time-invariance: the (Δ, $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$) parameters are constant across time. Our selection mechanism adds back input-dependent dynamics, which also requires a careful hardware-aware algorithm to only materialize the expanded states in more efficient levels of the GPU memory hierarchy.

Gu & Dao, "Mamba: Linear-Time Sequence Modeling with Selective State Spaces" (2023) | Dao, "FlashAttention-2" (ICLR 2024)

Scientific AI: Transforming Discovery

Protein Structure Prediction:

AlphaFold 2 (Jumper et al., 2021): Solved folding

Nobel Prize in Chemistry 2024

AlphaFold 3 (2024): Protein-ligand, DNA, RNA

ESM-2/ESMFold (2023): Protein language model

Weather & Climate:

GraphCast (DeepMind, 2023): Outperforms ECMWF

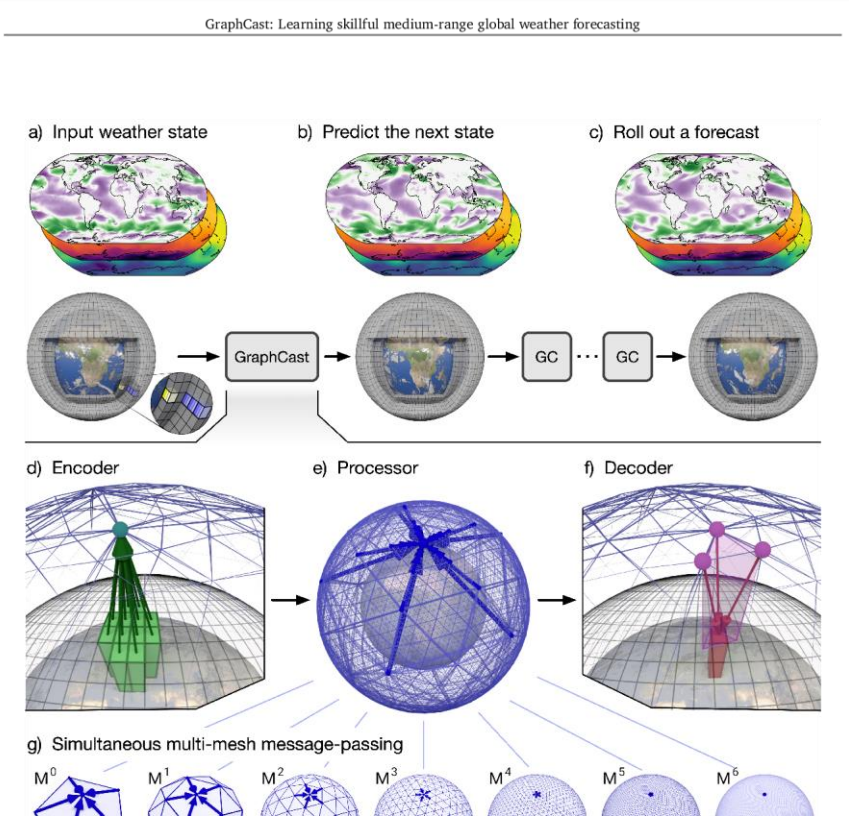
GenCast (2024): Probabilistic via diffusion

Materials: GNoME (2023): 2.2M new stable crystals

Mathematics:

AlphaProof (2024): IMO-level w/ Lean verification

AlphaGeometry (2024): Neuro-symbolic geometry



Lam et al., "GraphCast" (Science, 2023) | Jumper et al., "AlphaFold" (Nature, 2021) | Merchant et al., "GNoME" (Nature, 2023)

Things to Watch out For

- Research is cyclical
  - SVMs, boosting, probabilistic graphical models & Bayes Nets, Structural Learning, Sparse Coding, Deep Learning
  - Deep learning is unique in its depth and breadth, but...
  - Deep learning may be improved, reinvented, combined, overtaken
- Learn fundamentals for techniques across the field:
  - Know the span of ML techniques and choose the ones that fit your problem!
  - **Be responsible** in 1) how you use it, 2) promises you make and how you convey it
- Try to understand landscape of the field
  - Look out for what is coming up next, not where we are
- Have fun!

Thank you!