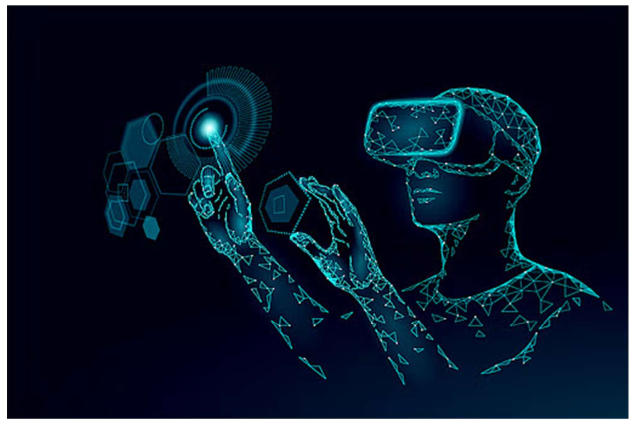
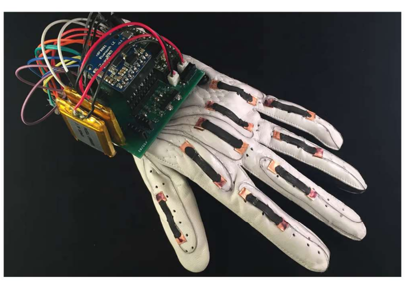
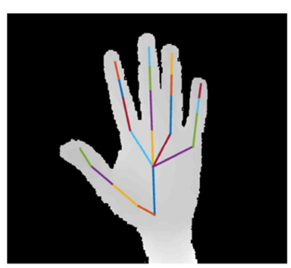
Hand Gesture Recognition

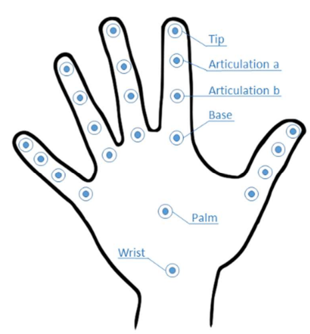


1

Data







2

CNN+RNN Depth and Skeleton based Dynamic Hand Gesture Recognition

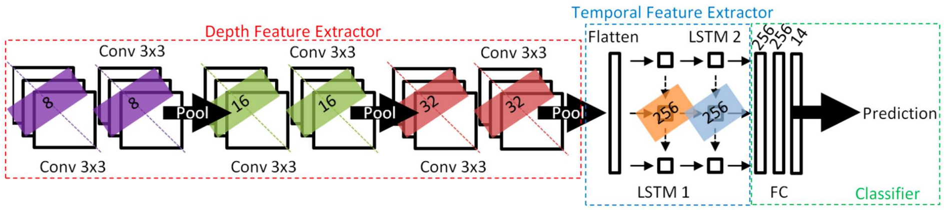


Fig. 3. The structure of the depth-based CNN+LSTM network.

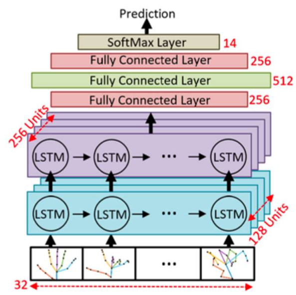


Fig. 1. The structure of the skeleton-based LSTM network.

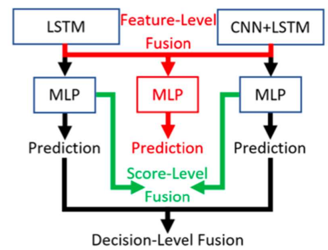


Fig. 2. The three levels of fusion for the LSTM and CNN+LSTM networks.

3

Deep Learning for Hand Gesture Recognition on Skeletal Data

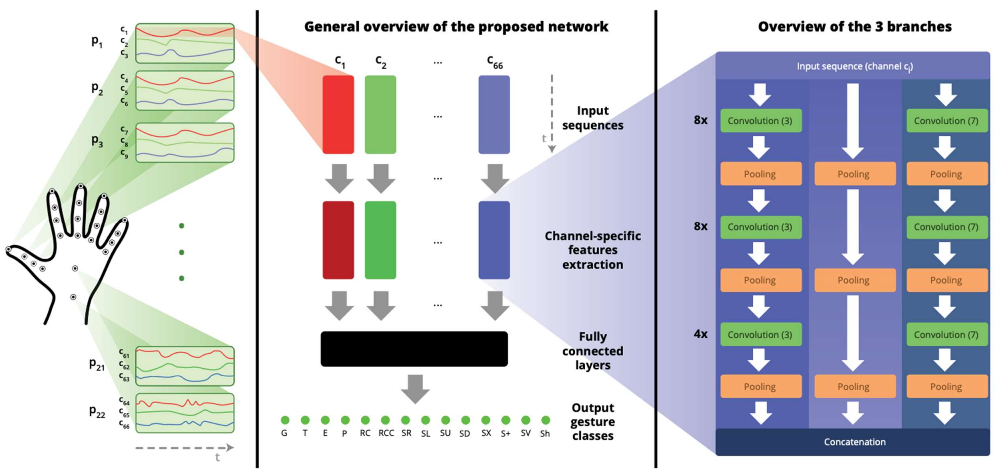
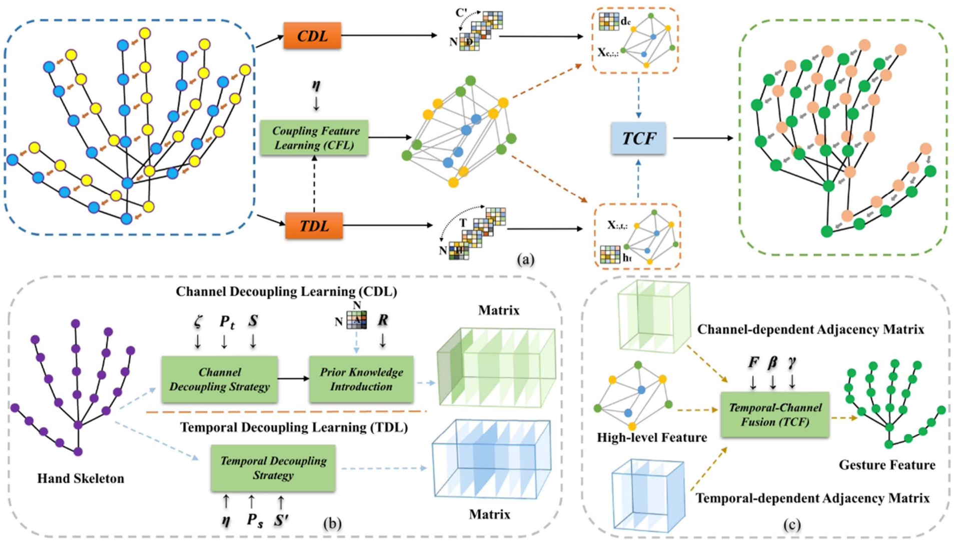


Fig. 2. Illustration of the proposed parallel convolutional neural network. Every channel is processed separately before the Multi Layer Perceptron. The parallel feature extraction module presented on the right is not shared between the 66 channels.

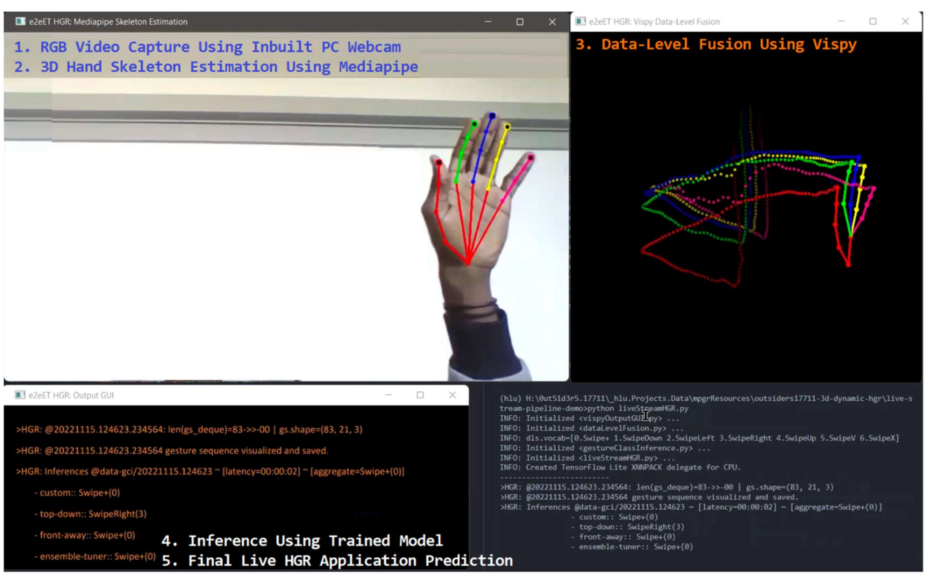
4

Temporal Decoupling Graph Convolutional Network for Skeleton-Based Gesture Recognition



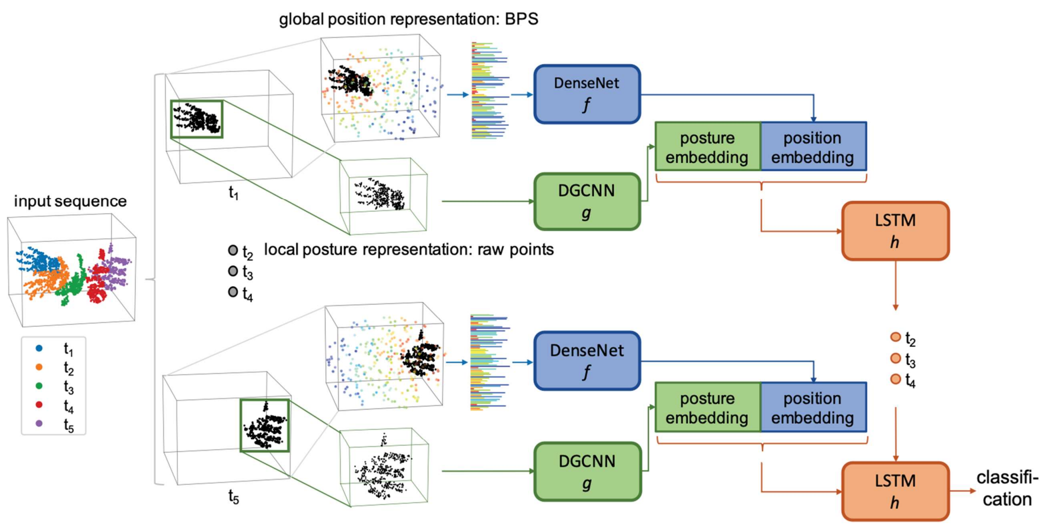
5

Real-Time Hand Gesture Recognition: Integrating Skeleton-Based Data Fusion and Multi-Stream CNN



6

Fusing Posture and Position Representations for Point Cloud-Based Hand Gesture Recognition



7

Conclusion

- How to extract different features?

  Temporal, spatial, different scales…
- How to fuse different features?

  Feature level, score level, concatenate, average, maximum…

  Embedded into an image, GNN topology
- How to improve computation efficiency?

  Only use CNN, image classification

8

Beyond Singularity: AI That Learns, Evolves and Acts on Its Own

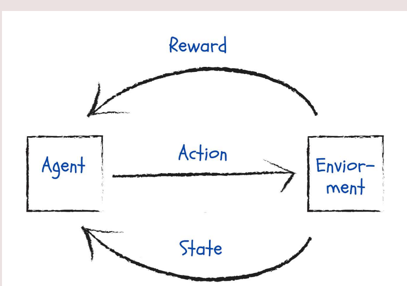
AGI to autonomous, ethical, self-aware systems

Combining DL + RL - more autonomous and intelligent systems.

Inspiration ?
Inspired by LeCun, Bengio & Hinton (2015) call to explore RL as the frontier in the ending parts of his paper

Presented by . Mohammad Vazeer Shah Abul Fazl (924301310)

9



Deep learning : gave us pattern recognition . It's about deciding what to do next.

Reinforcement Learning: teaches AI how to act, make choices, and learn from trial and error in dynamic environments.

Autonomous agents = Perception(DL) + Decision-making(RL) + Adaptation(RL)

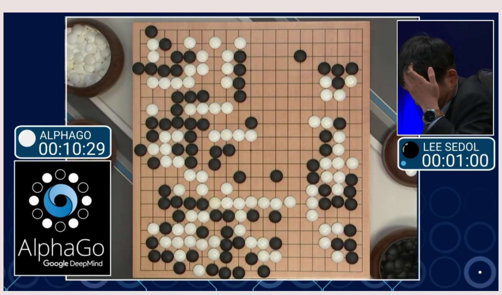
Human Limitations	Autonomous AI Advantages
Confirmation bias, tribalism	Objective historical pattern recognition
Emotion-driven decisions	Logic-based simulation of outcomes
Delayed feedback → slow learning	Fast learning from real-time and past data
Short-term gratification over long-term planning	Long-horizon planning with minimal reward shaping
Systems resist internal correction	Self-updating agents with aligned ethical objectives
Belief/power attachments distort decisions	Unbiased policy analysis and intervention suggestions
Inconsistent memory and selective history	Full recall of global history across domains

10

"Human-level Control Through Deep Reinforcement Learning" (Mnih et al., 2015)

- **Contribution**: Deep Q-Network (DQN) combined reinforcement learning with deep neural nets, achieving human-level performance in Atari games.
- **What I'm grabbing**: Proof that AI can learn optimal behavior through trial and error across thousands of episodes – **far faster and more consistently than humans**.
- **Limitation**: Struggles with generalization and long-term planning in complex, non-game environments.
- **Implication**: Demonstrates the foundation for **objective learning and fast feedback-based adaptation** – key to replacing emotionally inconsistent human decision loops.

11



AlphaGo: "Mastering the Game of Go with Deep Neural Networks and Tree Search" (Silver et al., 2016)

- **Contribution**: AlphaGo mastered the game of Go by integrating policy networks, value networks, and Monte Carlo Tree Search.
- **What I'm grabbing**: Clear example of **AI's ability to simulate thousands of futures** and choose the best long-term strategy.
- **Limitation**: Requires expert demonstrations or heavy compute to train; domain-specific.
- **Implication**: Highlights how AI can **override short-term human instincts** by optimizing long-term success paths with rigorous simulations.

12

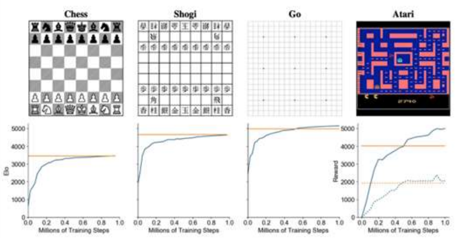


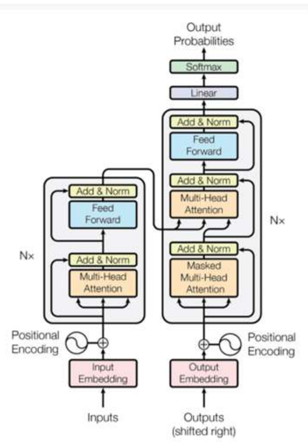
Figure 2: **Evaluation of *MuZero* throughout training in chess, shogi, Go and Atari.** The x-axis shows millions of training steps. For chess, shogi and Go, the y-axis shows Elo rating, established by playing games against *AlphaZero* using 800 simulations per move for both players. *MuZero*'s Elo is indicated by the blue line, *AlphaZero*'s Elo by the horizontal orange line. For Atari, mean (full line) and median (dashed line) human normalized scores across all 57 games are shown on the y-axis. The scores for R2D2 [21], (the previous state of the art in this domain, based on model-free RL) are indicated by the horizontal orange lines. Performance in Atari was evaluated using 50 simulations every fourth time-step, and then repeating the chosen action four times, as in prior work [25].

Source: Mastering Atari, Go, Chess and Shogi by Planning with a Learned Model

MuZero: "Mastering Atari, Go, Chess and Shogi by Planning with a Learned Model" (Schrittwieser et al., 2019)

- **Contribution**: MuZero learns **how the environment works** (dynamics model) while planning optimal actions – without needing to know game rules.
- **What I'm grabbing**: The power of building **adaptable AI agents** that can plan, learn, and correct themselves in unknown or changing environments.
- **Limitation**: Still data-hungry; not interpretable to humans.
- **Implication**: Supports the idea of **decentralized agents that adapt on the fly**, unlike rigid human systems (e.g., bureaucracies).

13



The encoder-decoder structure of the Transformer architecture
Taken from "Attention Is All You Need"

Transformers: "Attention Is All You Need" (Vaswani et al., 2017)

- **Contribution**: The transformer architecture introduced **attention**, enabling models to capture long-range dependencies in data.
- **What I'm grabbing**: These models can "remember" context over long sequences – **a powerful counter to humans' selective and distorted memory**.
- **Limitation**: Require massive data and compute; outputs can be unpredictable or ungrounded.
- **Implication**: Forms the **memory core** of autonomous systems – enabling them to synthesize large amounts of historical input into reasoned actions.

14

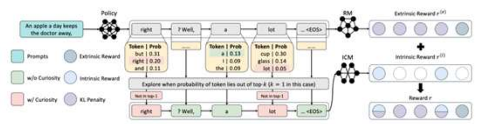


Figure 1: Illustration of the CD-RLHF framework. In this framework, the policy model generates a completion based on the given instruction, which samples tokens from vocabulary at each time. The introduced intrinsic curiosity module (ICM) estimates the curiosity as a metric for "novelty" of the context, producing the intrinsic rewards. Another mechanism is introduced to select which context is worth to explore, based on the probability of the selected token.

Curiosity through next-state prediction

Intrinsic reward (IR): prediction error in predicting s_{t+1} given s_t and a_t

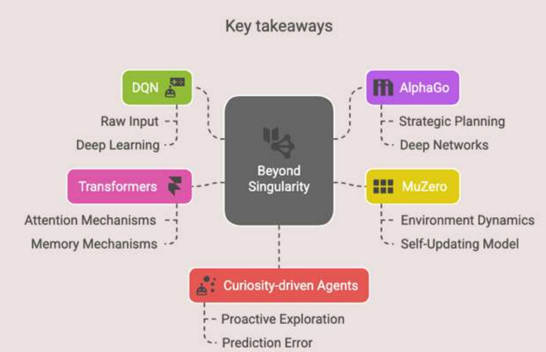
$$IR = ||\text{predicted}(s_{t+1}) - s_{t+1}||$$

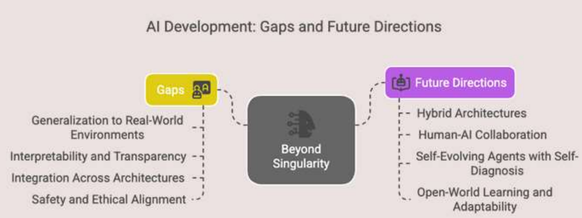
- **Small IR in familiar states** (easy to predict next state).
- **Big IR in unfamiliar states** (hard to predict next state in unknown trajectories).

Curiosity-Driven Agents: "Curiosity-Driven Exploration by Self-Supervised Prediction" (Pathak et al., 2017)

- **Contribution**: Proposed agents that explore by **reducing prediction error**, encouraging them to seek and understand unfamiliar situations – even **without external rewards**.
- **What I'm grabbing**: When properly constrained, this curiosity enables agents to **detect weak signals, explore uncertain geopolitical dynamics**, and take **proactive action** before crises escalate.
- **Limitation**: Without ethical or task-specific boundaries, agents risk **fixating on irrelevant** or **unsafe novelties**.
- **Implication**: With the right design, curiosity-driven agents become **early warning systems** – constantly scanning complex environments for hidden instability, without emotional or political bias.

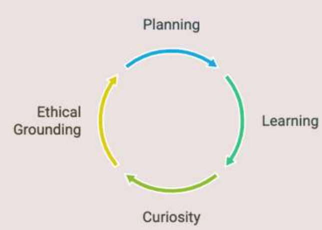
15





Conclusion :

Building fully autonomous AI is not just possible — it's inevitable.



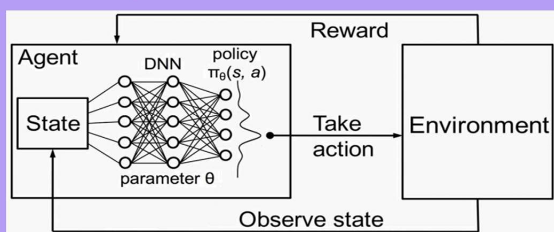
16

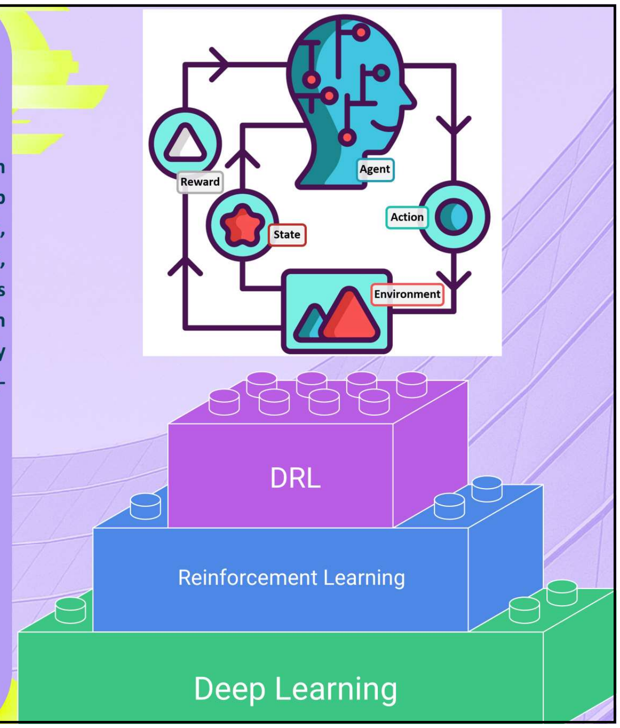
Integration of Deep Learning with

Reward-Based Systems

CSC 872

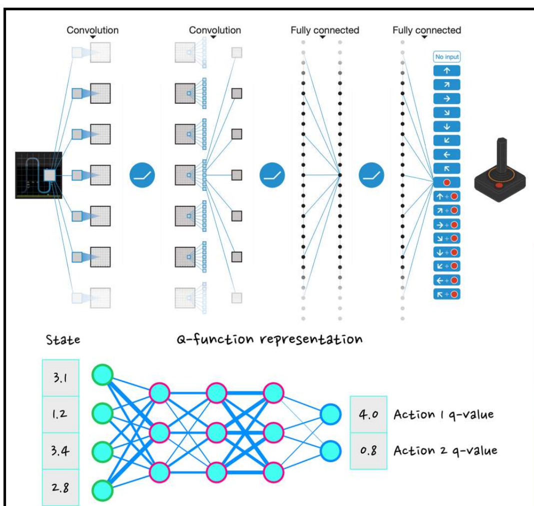
Deep reinforcement learning (DRL) represents an important evolution in artificial intelligence by merging the robust capabilities of deep learning, which simulates human-like perception and decision-making, with the adaptive learning mechanisms of reinforcement learning, where agents learn optimal behaviors through trial and error. This combination enables complex agents to tackle environments with high-dimensional state spaces, such as playing video games directly from pixel inputs, as explored in significant studies like the Deep Q-Network (DQN) by Mnih et al.



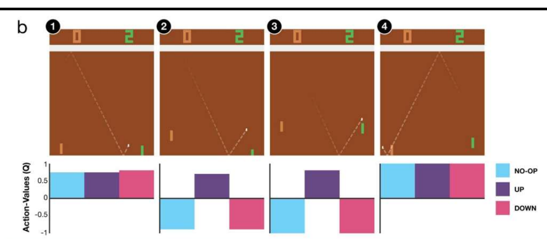


Presented by: Anurag Nepal

17



$$L_i(\theta_i) = \mathbb{E}_{(s,a,r,s')\sim \mathrm{U}(D)}\left[\left(r+\gamma \max_{a'} Q(s',a';\theta_i^-) - Q(s,a;\theta_i)\right)^2\right]$$



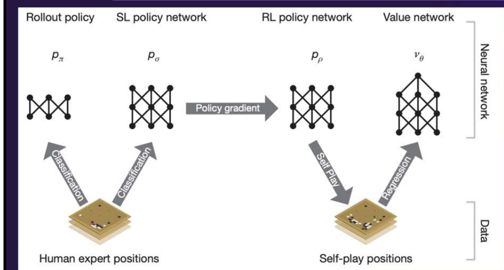
Pros	VS	Cons
High-dimensional inputs		Training instability
General-purpose agent		High sample complexity
Human-level performance		Long-term planning issues

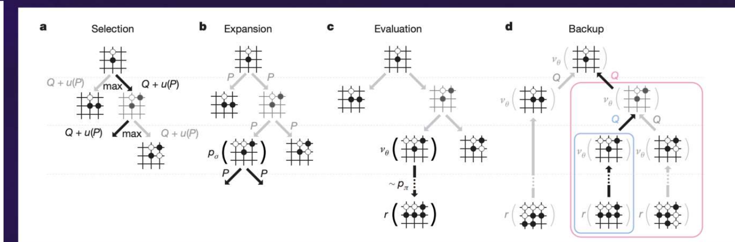
To stabilize the often-unstable training of deep neural networks in RL, two key techniques were introduced: experience replay, which stores experiences (state, action, reward, next state) and samples them randomly to break data correlations, and a separate target network, with periodically updated weights from the main Q-network, providing more stable targets for Q-value updates.

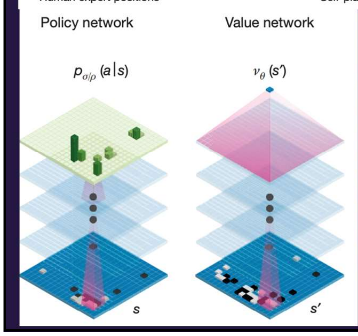
DQN - Human-level Control(2015)

18

AlphaGo - Mastering Go(2016)



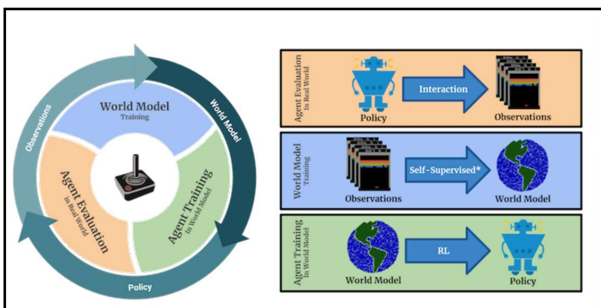




1 Each simulation traverses the tree by selecting the edge with maximum action value Q, plus a bonus u(P) that depends on a stored prior probability P for that edge. b, The leaf node may be expanded; the new node is processed once by the policy network pσ and the output probabilities are stored as prior probabilities P for each action. c, At the end of a simulation, the leaf node is evaluated in two ways: using the value network vθ; and by running a rollout to the end of the game with the fast rollout policy pπ, then computing the winner with function r. d, Action values Q are updated to track the mean value of all evaluations r(·) and vθ(·) in the subtree below that action.

Pros	VS	Cons
Complex game mastery		Human data reliance
Strategic planning		Computational demands
Board evaluation		Generalization challenges
		Sample efficiency

19



1) The agent starts interacting with the real environment following the latest policy (initialized to random).
2) The collected observations will be used to train (update) the current world model.
3) The agent updates the policy by acting inside the world model. The new policy will be evaluated to measure the performance of the agent as well as collecting more data (back to 1).

Pros	VS	Cons
Sample efficiency		Computational demand
Model-based planning		Model accuracy
Outperforms baselines		Stochastic environments

SimPLe - Model-Based Learning (2024)

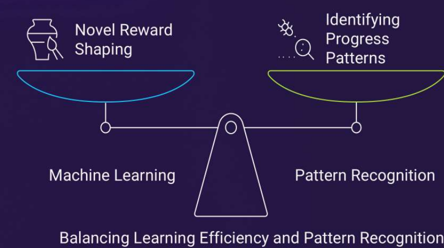
20

SORS - Self-Supervised Reward Shaping(2021)

This framework addresses the significant challenge of learning in environments with sparse rewards, where feedback is infrequent, typically only upon goal achievement. Such sparsity impedes learning and exploration. While manual dense reward design can help, it requires domain expertise and risks unintended reward exploitation. SORS aims to provide a denser reward signal automatically.

Pros	Cons
Improved performance	Sparse reward dependency
Sample efficiency	Computational overhead
Successful policies	Algorithm compatibility
Self-supervised learning	

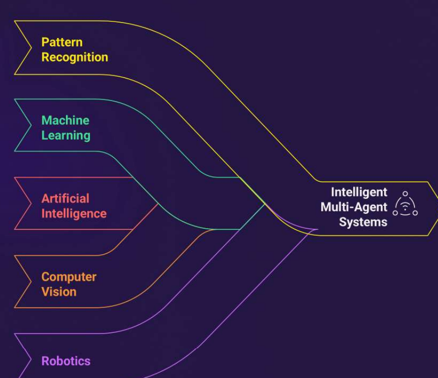
The basic idea of SORS is to alternate between updating the agent's policy using a standard RL algorithm and inferring a dense reward function from past experiences. Sparse rewards received from the environment rank the agent's trajectories. A classification-based reward inference algorithm then learns a dense reward function assigning higher rewards to trajectories that achieved higher sparse returns. This learned dense reward function further trains the agent's policy, accelerating learning.



21

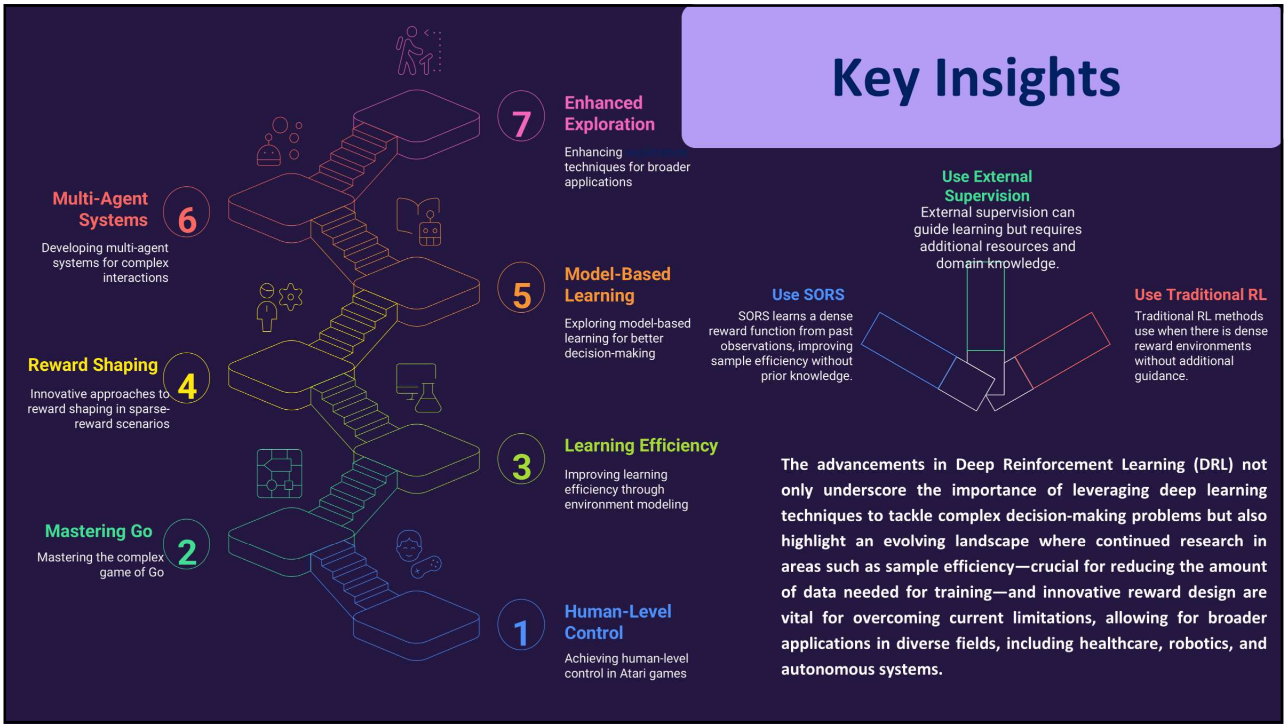
Multi-Agent Reinforcement Learning (MARL) focuses on the framework where multiple agents learn and interact within a shared environment, enabling the exploration of complex dynamics that arise from cooperative, competitive, or adversarial interactions among agents.

One significant challenge in MARL is non-stationarity, as the actions of one agent can affect the observations and rewards experienced by others, making it difficult for agents to learn stable policies when their environment is constantly changing due to the actions of other agents.



Multi-Agent Reinforcement Learning (MARL)(2021)

22



Key Insights
7 Enhanced Exploration
Enhancing techniques for broader applications
6 Multi-Agent Systems
Developing multi-agent systems for complex interactions
5 Model-Based Learning
Exploring model-based learning for better decision-making
4 Reward Shaping
Innovative approaches to reward shaping in sparse-reward scenarios
3 Learning Efficiency
Improving learning efficiency through environment modeling
2 Mastering Go
Mastering the complex game of Go
1 Human-Level Control
Achieving human-level control in Atari games
Use External Supervision
External supervision can guide learning but requires additional resources and domain knowledge.
Use SORS
SORS learns a dense reward function from past observations, improving sample efficiency without prior knowledge.
Use Traditional RL
Traditional RL methods use when there is dense reward environments without additional guidance.
The advancements in Deep Reinforcement Learning (DRL) not only underscore the importance of leveraging deep learning techniques to tackle complex decision-making problems but also highlight an evolving landscape where continued research in areas such as sample efficiency—crucial for reducing the amount of data needed for training—and innovative reward design are vital for overcoming current limitations, allowing for broader applications in diverse fields, including healthcare, robotics, and autonomous systems.

23

The Road Ahead – Progress, Future and Challenges of DRL
Future Directions
Model-Based Learning
Reward Shaping
Algorithm Stability
Generalization
Interpretability
Real-World Applications
Deep Reinforcement Learning
Progress
GDPR
DQN
AlphaGo
SimPLe
SORS
MARL
Challenges
Sample Efficiency
Sparse Rewards
Multi-Agent Learning

24

ADVANCEMENTS IN AUDIO GENERATION: DIFFUSION & CONSISTENCY MODELS

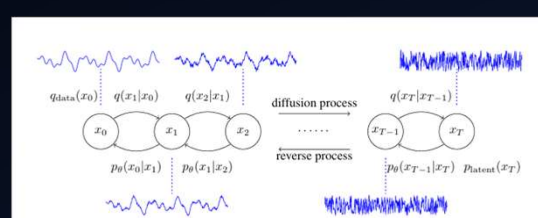
Divya Panchal

- **What is Audio Generation?**
  - It's about creating sound automatically using computers.
  - This can be from text, images, videos, or even just an idea!
- **The Core Challenge**
  - We want sounds that are incredibly realistic (high-fidelity).
  - AND we want to create them quickly and efficiently. This is where the research comes in.
- **We'll explore how new methods, specifically Diffusion and Consistency Models, are pushing the boundaries.**

25

The Foundation: Diffusion Models for High-Quality Audio

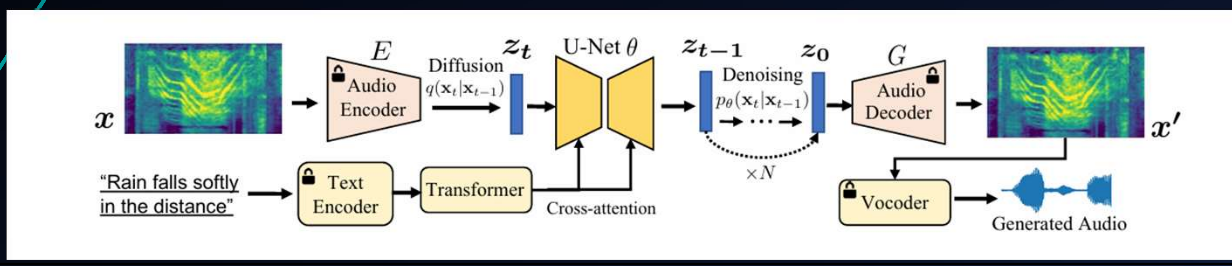
- **What are Diffusion Models?**
  - Imagine starting with pure noise (like static).
  - These models learn to gradually transform that noise, step-by-step, into structured, meaningful audio. It's like sculpting sound from randomness.
- **Early Breakthrough: DiffWave (Kong et al., 2020)**
  - It was versatile: handled tasks like creating speech from text (vocoding), generating sounds based on categories, and even creating sounds from scratch.
  - Importantly, it matched the quality of older, slower methods (like WaveNet) but was non-autoregressive (faster step-by-step generation).



26

Practical Diffusion: Latent Spaces & Tackling Data Needs

- **Challenge with Early Diffusion (like DiffWave):** Working directly with raw audio (waveforms) is very computationally intensive.
- **Solution:** Latent Diffusion Models (LDMs)
- Make-An-Audio (Huang et al., 2023b)
  - Uses a **spectrogram autoencoder**: Learns to compress spectrograms (visual representations of sound frequencies) into this latent space.
  - Uses **CLAP embeddings** for better understanding the link between text prompts and audio content.



27

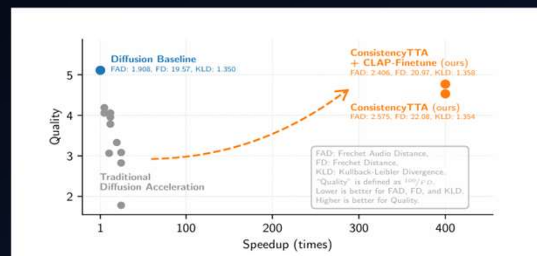
The Speed Bump: Why Diffusion Models Took Time to Generate

- **The Achilles' Heel of Diffusion Models:** That step-by-step refinement process we talked about? It's powerful, but it's *slow*.
- Typically requires **hundreds of individual steps** to generate one piece of audio. Each step needs a full pass through a large neural network.
- **The Impact:** This slowness was a major barrier for:
  - Real-time applications (imagine a game where sound effects lag!).
  - Interactive tools.
  - Deployment on devices with limited computing power.
- Early attempts to speed things up (like "fast sampling" in DiffWave) helped, but a more fundamental solution was needed. This sets the stage for our next topic: **Consistency Models.**

28

Need for Speed: Enter Consistency Models

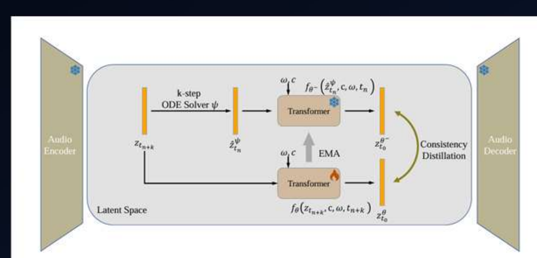
- The Innovation: Consistency Models (CMs) (Song et al., 2023)
  - A process called **"Consistency Distillation,"** where knowledge from a slow, pre-trained diffusion model (the "teacher") is distilled into a fast "student" model.
- Application to Audio: ConsistencyTTA (Bai et al., 2024)
  - **Key Contribution:** "CFG-aware latent consistency model." Classifier-Free Guidance (CFG) is crucial for making sure the audio matches the text prompt well. ConsistencyTTA found a way to build CFG awareness directly into the distillation training.
  - **Result:** Capable of generating audio in a **single step!**



29

Even Faster, Still High Quality: AudioLCM

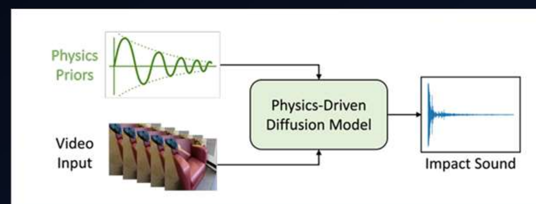
- **Key Technical Idea:** "Guided Latent Consistency Distillation" but with a **multi-step ODE solver** during the distillation training.
  - Instead of distilling from single steps of the teacher model, it uses *k* steps (e.g., k=20) of an ODE solver to get a more refined target for the student.
  - Also incorporates **architectural improvements** into their Transformer model, inspired by LLaMA ,for better stability and performance.



30

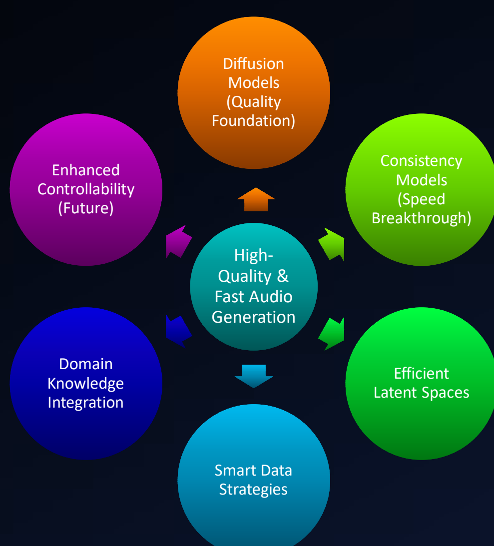
Specialized Sounds: Physics-Driven Diffusion

- Physics-Driven Diffusion Models (Su et al., 2023)
  - Focus: Generating **impact sounds** (like a drumstick hitting different materials) from silent videos.
  - **The Challenge:** Visual information alone is often not enough to guess the exact sound of an impact. The same visual action can sound very different based on materials, force, etc.
  - Incorporate **"physics priors"** into the diffusion model.
    - Estimated physical properties from *actual sound examples* in the training data (like sound frequencies, decay rates – "modal parameters").
  - **Clever Inference:** For a new silent video, the model *retrieves* the most similar physics priors from its training data based on visual similarity, then uses these to guide sound generation.



31

Key Insights & The Path Ahead



32

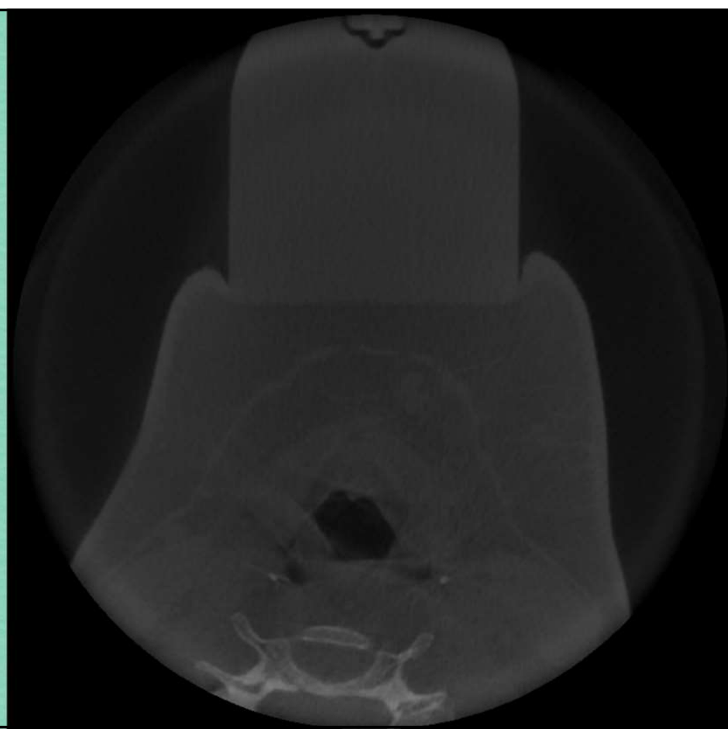
Exploring the Role of CNNs in Enhancing Dental Workflows in CBCT Scans

Gabrielle Salamanca

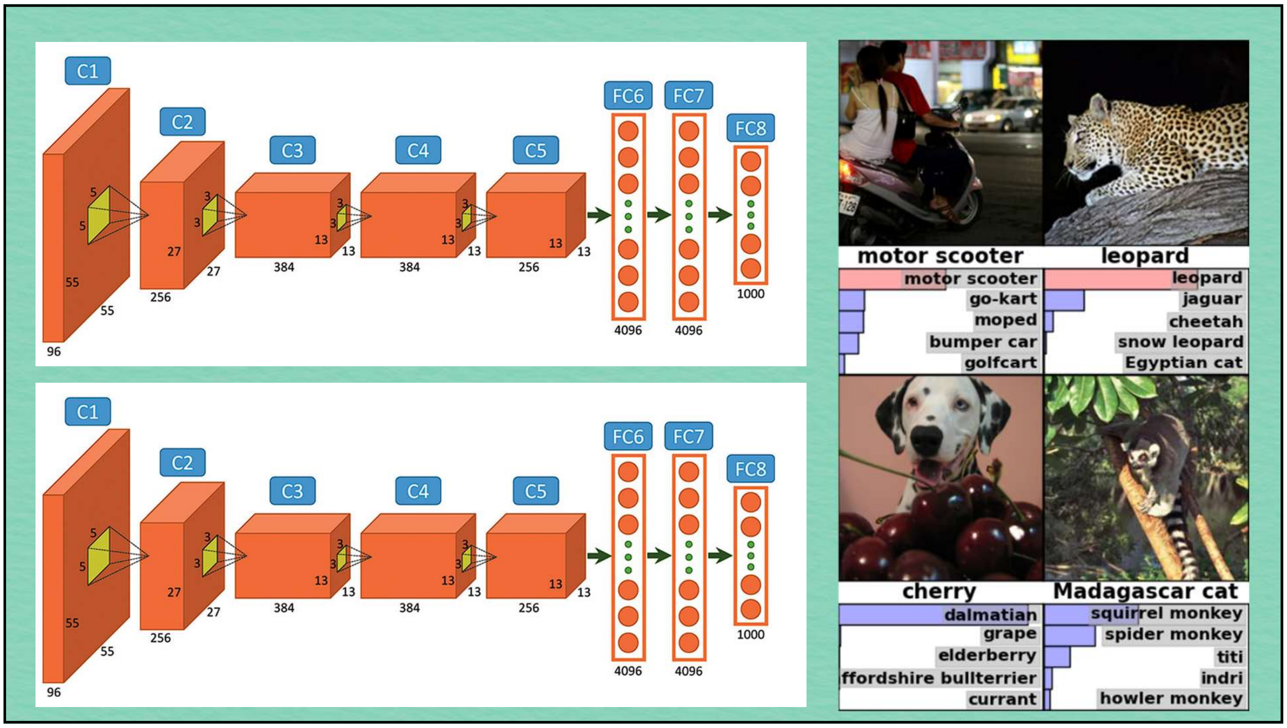
Contents

Slide 2: Why?
Slides 3-7: Chosen Papers
Slide 8: Takeaways

33

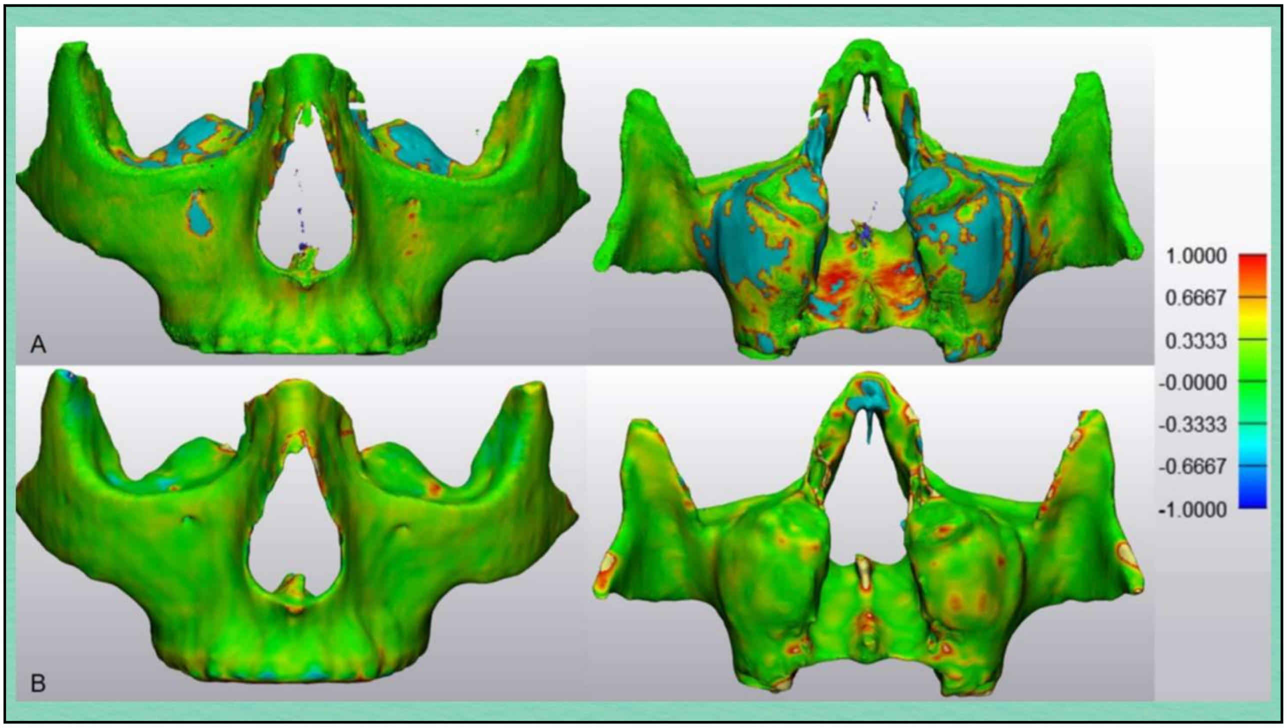


34


C1
C2
C3
C4
C5
FC6
FC7
FC8
96
55
55
5
5
256
27
27
3
3
384
13
13
384
13
13
256
13
13
4096
4096
1000
motor scooter
leopard
motor scooter
go-kart
moped
bumper car
golfcart
leopard
jaguar
cheetah
snow leopard
Egyptian cat
cherry
Madagascar cat
dalmatian
grape
elderberry
ffordshire bullterrier
currant
squirrel monkey
spider monkey
titi
indri
howler monkey

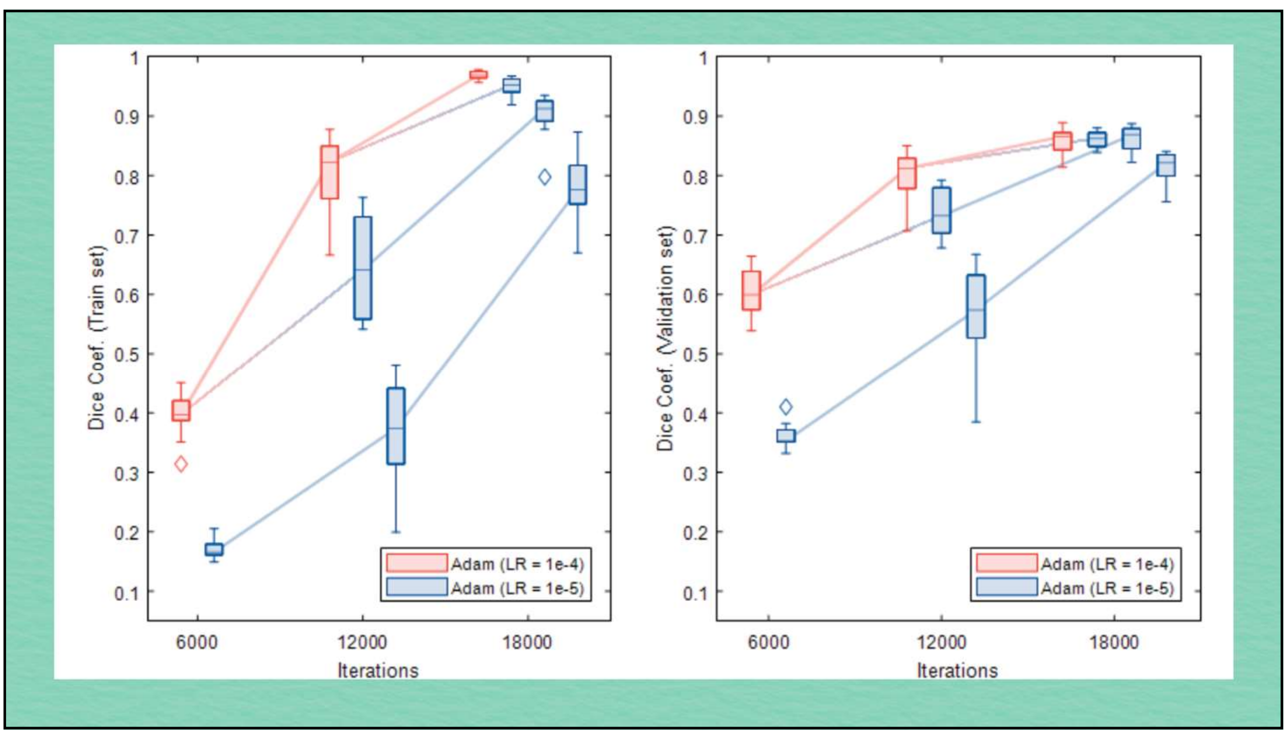
35

36



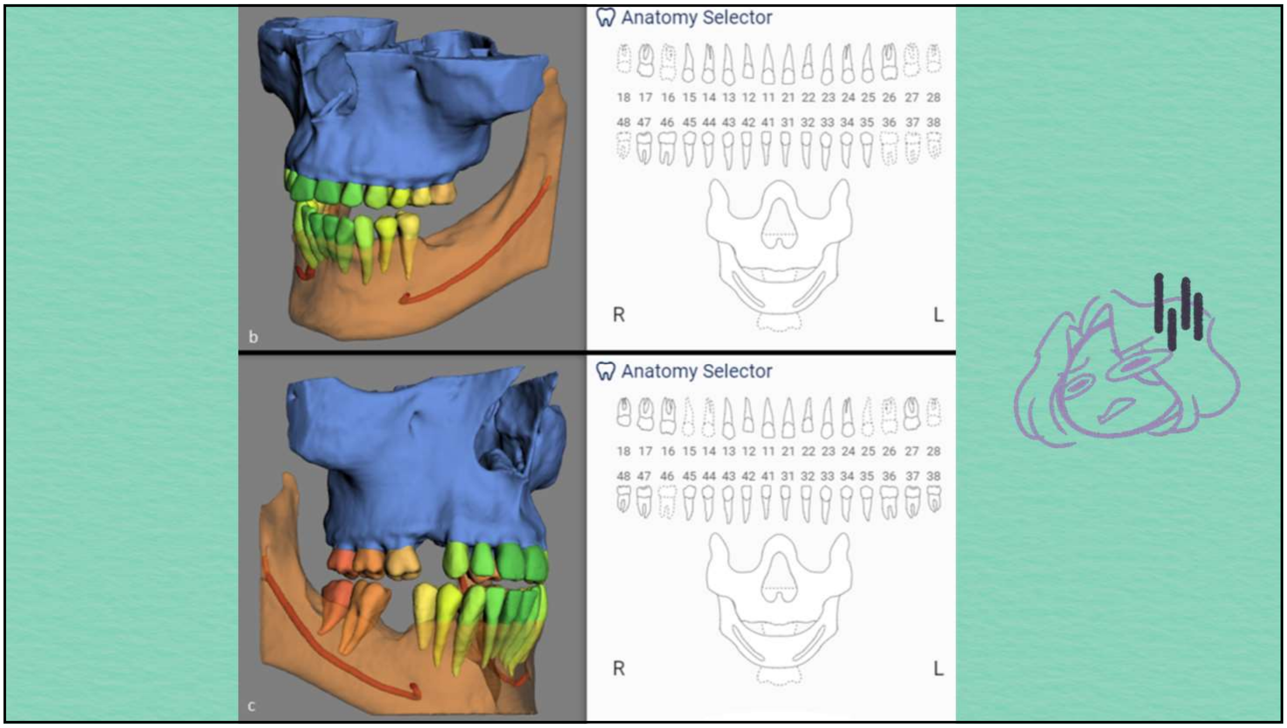
A
B
1.0000
0.6667
0.3333
-0.0000
-0.3333
-0.6667
-1.0000

37



Dice Coef. (Train set)
Dice Coef. (Validation set)
Iterations
6000
12000
18000
Adam (LR = 1e-4)
Adam (LR = 1e-5)

38


b
Anatomy Selector
18 17 16 15 14 13 12 11 21 22 23 24 25 26 27 28
48 47 46 45 44 43 42 41 31 32 33 34 35 36 37 38
R
L
c
Anatomy Selector
18 17 16 15 14 13 12 11 21 22 23 24 25 26 27 28
48 47 46 45 44 43 42 41 31 32 33 34 35 36 37 38
R
L

39

40

Deep Reinforcement Learning - André Shannon

Combine RL (great problem solving) with DL (great knowledge representation)

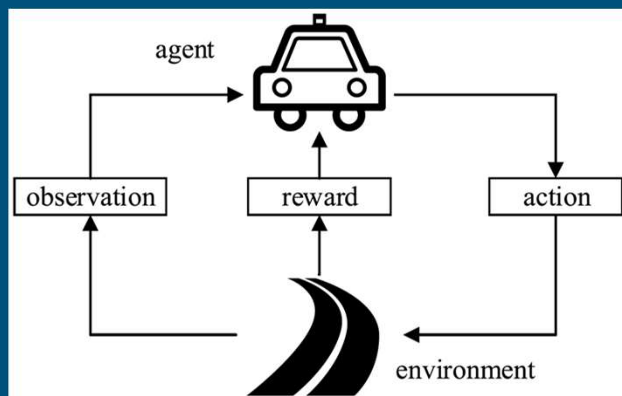
State, $s \in S$

Action, $a \in A$

Policy, $\pi{:}S \rightarrow A$

Reward, R_t

Time Step, t



41

Need to calculate cumulative rewards

State Value function:

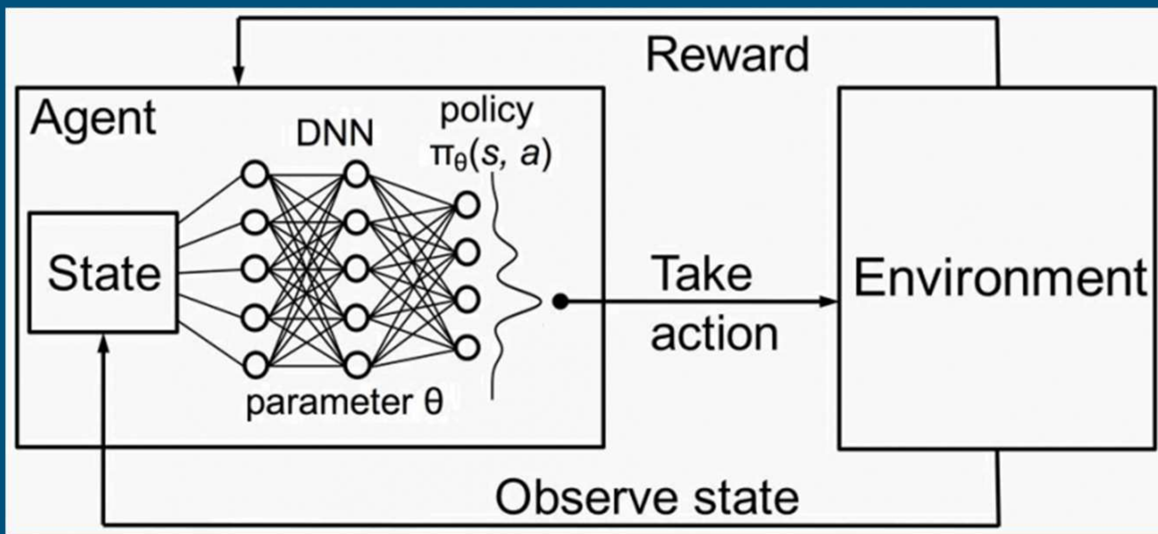
$$V(s) = E_\pi[R_{t+1} + y\ V(s_{t+1}) \mid s_t = s]$$

State Action Value function:

$$Q(s, a) = E_\pi[R_{t+1} + y\ Q(s_{t+1}, a_{t+1}) \mid s_t = s, a_t = a]$$

42

Let's Approx Value Funcs/Policy using DL



43

Wang et al DRL survey

- Value Based DRL
  - Optimize state-action value (Q) function
- Policy Based DRL
  - Optimize policy using policy gradient methods (gradient ascent)
  - Actor-Critic
- Maximum Entropy DRL
  - Add entropy term to reward to also maximize entropy

44

Deep Q Network

One of the first (successful) integrations of RL with DL

Problem:

- ➢ Approximating Q using nonlinear func (NN) was unstable or diverged

Solutions:

- ➢ Experience Replay buffer
- ➢ Periodic update of action-values towards target values

45

Asynchronous Advantage Actor-Critic (A3C)

Surpassed state-of-the-art models at the time

Trained in half the time on a multi-core cpu vs on a gpu

Execute multiple agents in parallel on multiple instances of the environment

Don't need experience replay buffer, multiple agents in parallel decorrelates data

Can also work in continuous action spaces

46

Generative Adversarial Imitation Learning

Imitation Learning seeks to learn from demonstrations from an expert

Previous imitation learning used inverse RL to learn a cost function to explain expert behavior and then performed RL on the cost func to get the policy.

Uses structure similar to GANs. Model generates policy and minimizes error w.r.t. the expert policy. Discriminator differentiates between generated policy and expert's and maximizes error.

47

Takeaways

Still evolving field with many open problems and different solutions already

RL involves a lot of problem formulation:

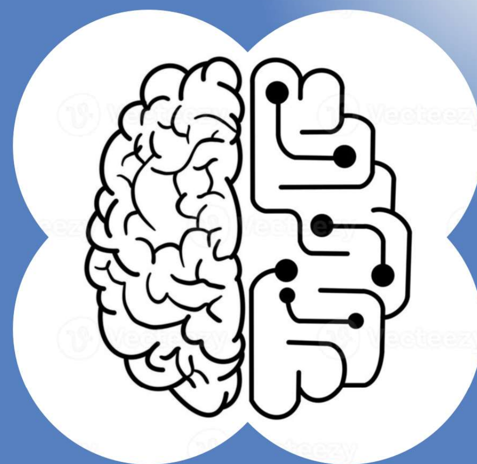
- ➢ Representing the environment
- ➢ Defining the reward/cost functions
- ➢ Framing the problem in different ways to apply different solutions and overcome challenges faced by other solutions

48

BANAZ SINJARY

REFLECTING BRAIN PLASTICITY IN NEURAL NETWORK ARCHITECTURE

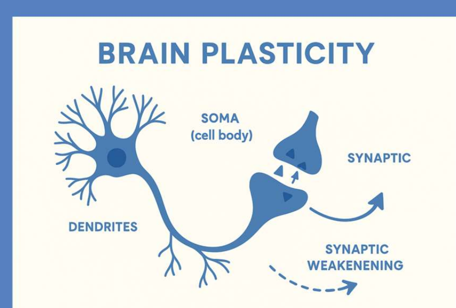
A literature review on applying brain plasticity functions and principles to neural network architectures. Exploring the intersection of biology and computer science with applications in *theory, architecture and the real world.*



49

PLASTICITY

- Biological Plasticity: Synaptic strengthening, pruning, reconfiguration
- Traditional ANNs: static post training, limited adaptability
- New models aim to simulate growth and learning over time
- Real world applications need adaptable, efficient learning systems

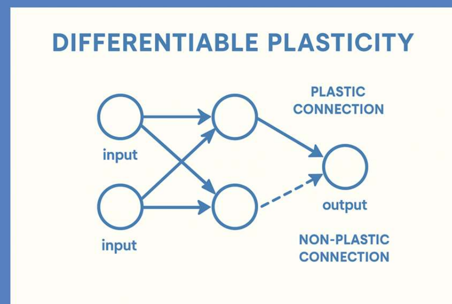


(Diagrams generated w/ ChatGPT)

50

DIFFERENTIABLE PLASTICITY

- Adds a plastic component to each connection
- Learns how much to update each connection dynamically
- Improves one shot learning, memory retention
- Compatible with gradient descent, but still not lifelong learning

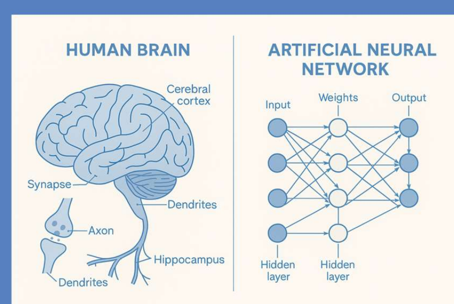


(Miconi, Clune and Stanley 2018)

51

NEURAL RESHAPING

- Compares human brain plasticity with ANN learning
- Brain: myelination, dendrite growth, experience based change
- ANN: backpropagationfixed structure after training
- Finds symbolic parallels: pruning = regularization, myelin = efficient pathways

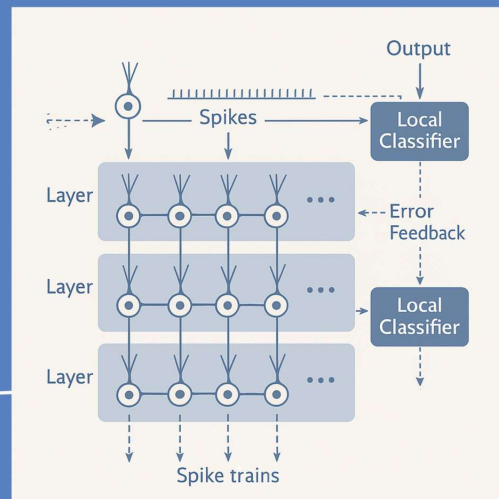


(Zadeh, Bahrami and Soleimani 2024)

52

DECOLLE

- Spiking neural network trained with local errors
- Learns over time, uses minimal memory
- Handles temporal data, sparse spikes, dynamic adaptation
- Biologically inspired, scalable , real time ability

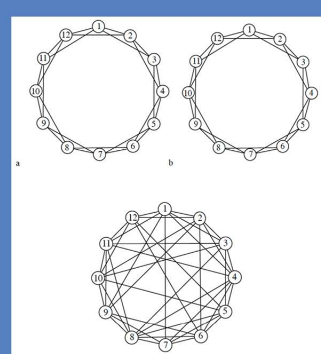


(Kaiser, Mostafa, and Neftci 2020)

53

DYNAMICAL LINKS & SYMBOL FORMATION

- Dynamic links form through repeated synchrony
- Connection patterns compete and self organize
- Structured connectivity emerges from activity
- Symbols = stable firing groups over time
- Moves beyond static vector space representations

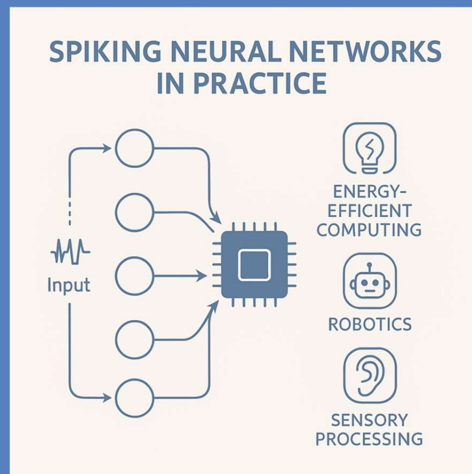


(Malsburg 1994)

54

SNN IN PRACTICE

- SNNs use spikes, not activations → lower energy, real time response
- Strong in vision/robotics applications: Loihi chip, hexapod CPGs, SLAM
- Training is still a bottleneck: BPTT is expensive
- ANN-to-SNN conversion is a promising workaround



(Yamazaki,Vo-Ho, and Bulsara 2022)

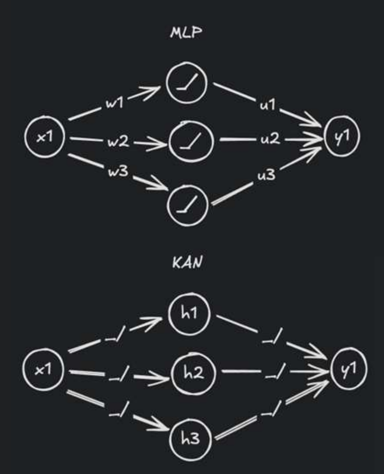
55

CONCLUSION

- Plasticity enables online adaptation and structural change
- Biological systems inspire context aware, local learning
- Current models lack scalable lifelong memory
- Neuromorphic tools show promise, but need stability
- Plasticity should guide future network design

56

A Comparative Study of Kolmogorov-Arnold Networks and Multi-Layer Perceptrons in Deep Learning



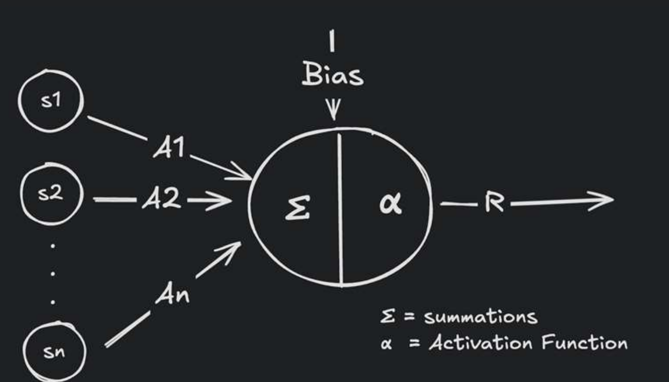
Both MLPs and KANs are universal function approximators — but differ in how they learn. This project explores how and why.

Goal:
Compare KANs and MLPs in theory and practice
Understand trade-offs in architecture, learning behavior, and performance

Atharva Walawalkar

57

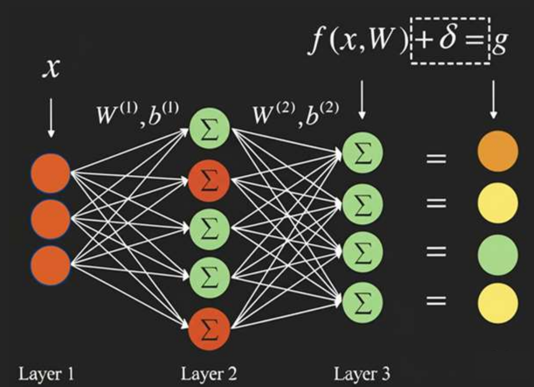
Introduction to the Perceptron.



- Rosenblatt (1958) proposed the perceptron as a brain-inspired learning model.
- It used weighted connections between sensory (S), association (A), and response (R) units.
- Learning meant adjusting these weights to shape recognition and memory.
- A major limitation: it could only solve linearly separable problems.

58

The MLP: Backpropagation and Representation



Multi-Layer Perceptrons: The Universal Function Approximator

- Introduced by **Rumelhart, Hinton, Williams** (1986)
- MLPs) use layers of neurons with fixed activations.
- Backpropagation tunes weights for learning.
- They can approximate any function, making them the core of modern deep learning.

Kolmogorov-Arnold Theorem

$$f(\mathbf{x}) = f(x_1, \ldots, x_n) = \sum_{q=0}^{2n} \Phi_q \left(\sum_{p=1}^{n} \phi_{q,p}(x_p) \right).$$

where $\phi_{q,p} : [0, 1] \to \mathbb{R}$ and $\Phi_q : \mathbb{R} \to \mathbb{R}$.

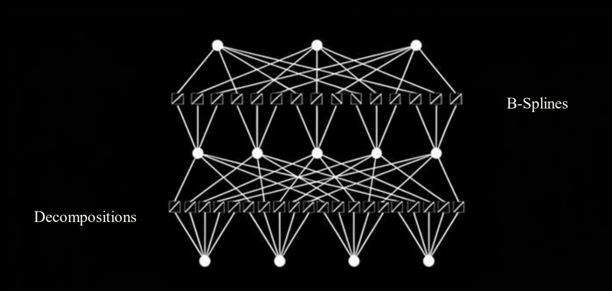
From Theory to Architecture: The Kolmogorov-Arnold Theorem

- **Kolmogorov (1957):**
- Kolmogorov showed any function of many variables can be built from sums of 1D functions.

  $f(x) = \sum \Phi q(\sum \phi_{kj}(x_j))$

- This laid the theoretical foundation for KANs — using 1D parts to build complex behavior.

Kolmogorov-Arnold Networks (KANs)

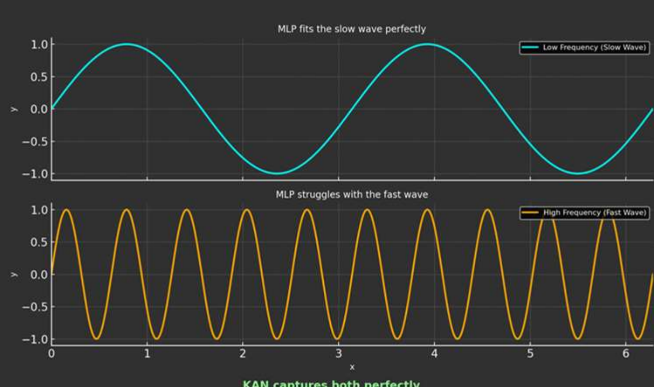


No curse of dimensionality (in theory)

Kolmogorov-Arnold Networks: Architecture Shift

- **Proposed by Liu et al., 2024**
- KANs (2024) replace weights with spline functions on edges.
- Nodes just sum inputs, no activations.
- Inspired by Kolmogorov's formula, KANs are more interpretable and often outperform MLPs on math-heavy tasks.

Theory: Expressivity & Spectral Bias



Learning Frequencies: MLPs vs. KANs

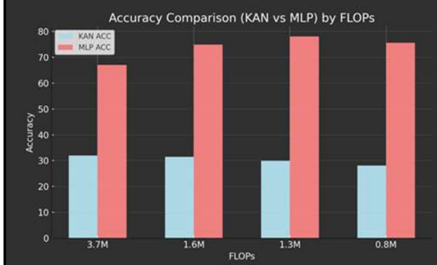
MLPs learn smooth patterns first
– Struggle with sharp or wavy functions
– Biased toward **low-frequency** signals

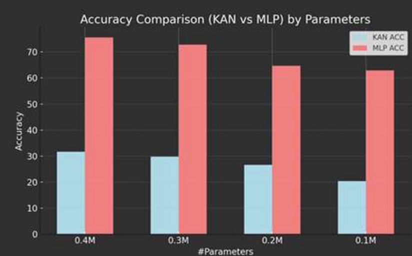
KANs learn sharp patterns better
– Can handle **high-frequency** details
– More flexible with complex shapes

Why it matters:
Better performance on **formulas, physics, signals**

Empirical Results and Limitations

When KANs Win (and When They Don't)





- KANs do well in symbolic regression, sharp-function fitting, and low-data science tasks.
- But they trail MLPs in NLP, vision, and large-scale benchmarks.
- Perform worse on both compute and parameter count

Takeaways and Future Outlook

Lessons from KANs vs. MLPs

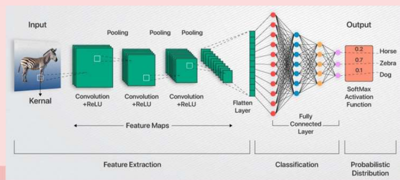
Aspect	MLP	KAN
Activation Functions	Fixed (e.g., ReLU, Tanh)	Learnable splines (placed on edges)
Learnable Parameters	Weights and biases	Spline coefficients (function parameters)
Interpretability	Low	High (functions can be visualized/analyzed)
Expressivity	Universal function approximator	Matches or exceeds MLP efficiency in some tasks
Spectral Bias	Favors low-frequency patterns	Also handles high-frequency components
Strength in Symbolic Tasks	Weaker	Stronger (e.g., formula fitting, math modeling)
Performance in NLP/CV/ML	Strong	Weaker (as of current benchmarks)

- MLPs: reliable, versatile, widely used
- KANs: theory-aligned, interpretable, niche strengths
- Both have their place in modern deep learning
- Future: deeper KANs, hybrid models, better training

DEEP CONVOLUTIONAL NETWORKS FOR VISUAL RECOGNITION AND UNDERSTANDING LITERATURE SURVEY STUDY

Brandon Watanabe

- Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation (Girshick et al.)
- ImageNet Classification with Deep Convolutional Neural Networks (Krizhevsky et al.)
- Very Deep Convolutional Networks for Large-Scale Image Recognition (Simonyan & Zisserman)
- Going Deeper with Convolutions (Szegedy et al.)
- OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks (Sermanet et al.)

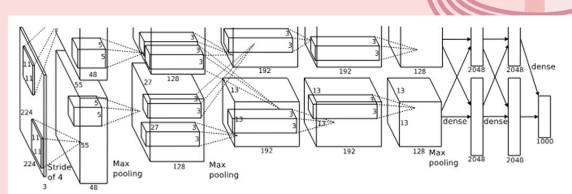


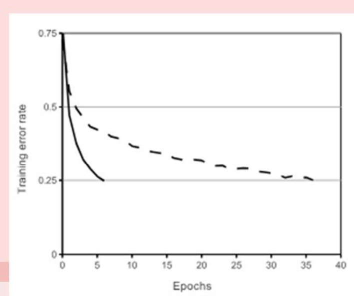
- https://ravjot03.medium.com/decoding-cnns-a-beginners-guide-to-convolutional-neural-networks-and-their-applications-1a8806cbf536

65

ALEXNET ARCHITECTURE (2012)

- Trained on ImageNet
  - 1.2 million training images with 1000 classes
- Their model achieved a top-5 test error rate of 15.3%
  - Second-best contest entry achieved a top-5 error rate of 26.2%
- 8 learned layers
  - 5 Convolutional layers
  - 3 Fully Connected layers
- RELU trained several times faster than with tanh units
- Parallelized with 2 GPUs
- Local Response Normalization aided generalization
- Overlapping pooling, Data Augmentation, and Dropout were used to help with overfitting





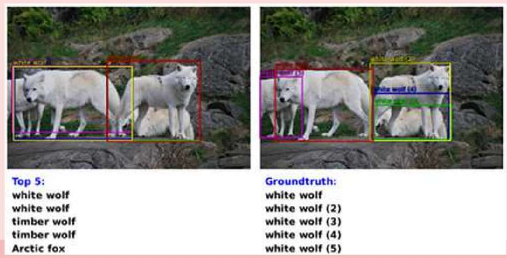
- ImageNet Classification with Deep Convolutional Neural Networks (Krizhevsky et al.)

66

INTEGRATED CLASSIFICATION, LOCALIZATION, AND DETECTION (2013)

Overfeat (2013)

- The OverFeat framework used a single CNN for integrated classification, localization, and detection via a multi-scale, sliding window approach
- Won the ILSVRC 2013 localization task with a 29.9% error rate
- Showed how CNNs could be effectively used for classification, detection, and localization



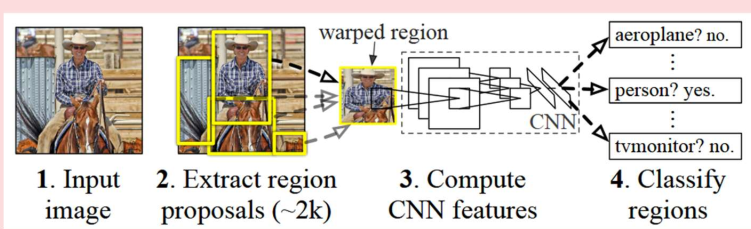
- Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation (Girshick et al.)

67

IMPACT ON DETECTION & LOCALIZATION

R-CNN (2014)

- The R-CNN applied CNN features extracted from bottom-up region proposals for object detection
- Achieved a more than 30% relative improvement in mAP on PASCAL VOC 2012
- Used domain-specific fine-tuning of the ImageNet-pretrained CNN on warped region proposals from the VOC dataset

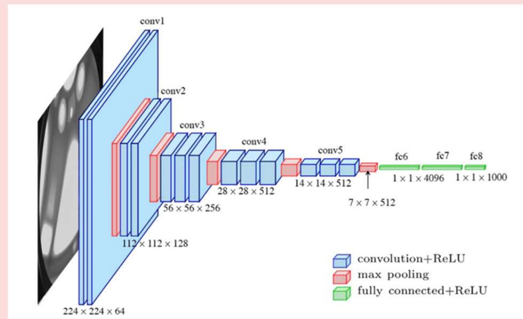


- OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks (Sermanet et al.)

68

THE VGG NETWORKS (2014)

- They evaluated networks with increasing depth, ranging from 11 to 19 weight layers
- Significant performance improvements by pushing the depth using 3x3 convolution filters
- A stack of multiple 3x3 convolutional layers could achieve the same receptive field size as a single larger filter
  - Incorporated more non-linear rectification layers and fewer parameters
- Secured first place in localization and second place in classification in ILSVRC 2014
- Demonstrated that deep CNNs could improve results by increasing depth

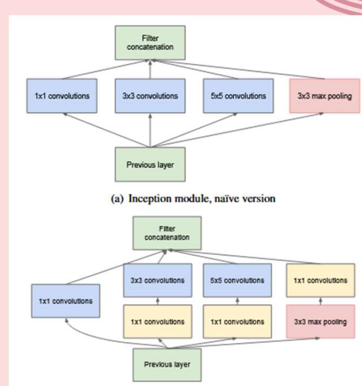


- https://www.researchgate.net/figure/The-original-VGG-16-architecture-Simonyan-et-Zisserman-2014_fig1_359813468
- Very Deep Convolutional Networks for Large-Scale Image Recognition (Simonyan & Zisserman)

69

THE INCEPTION ARCHITECTURE (2014)

- Processed inputs with parallel convolutional layers of different sizes (1x1, 3x3, 5x5) and pooling layers
- 1x1 convolutions were used for dimensionality reduction before expensive 3x3 and 5x5 convolutions, helping to manage computational complexity
- 22 Layers
- 6.67% top-5 error, and also obtained competitive results for detection, winning the task with 43.9% mAP
- 12 times fewer parameters than the AlexNet while being significantly more accurate



(a) Inception module, naïve version

(b) Inception module with dimensionality reduction

- Going Deeper with Convolutions (Szegedy et al.)

70

KEY STRENGTHS

Paper	Task Focus	Innovations and Key Ideas	Key Strengths
AlexNet	Classification	ReLU, Dropout, Augmentation, GPU training	Large scale breakthrough in practical training for large datasets
Overfeat	Classification, Localization, Detection	Multi-scale sliding window	Combined classification, localization, and detection
R-CNN	Detection, Segmentation	Bottom-Up Region Proposals, Supervised pre-training	Showed how CNNs could be used for detection and segmentation
VGG	Classification	Increased depth using 3x3 filters	Strong generalization to other datasets
GoogLeNet	Classification, Detection	Inception module	Better utilization of computing resources

71

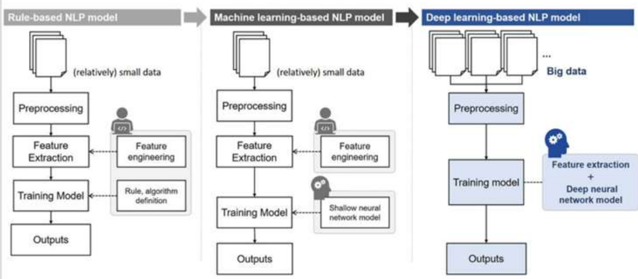
CRITIQUE

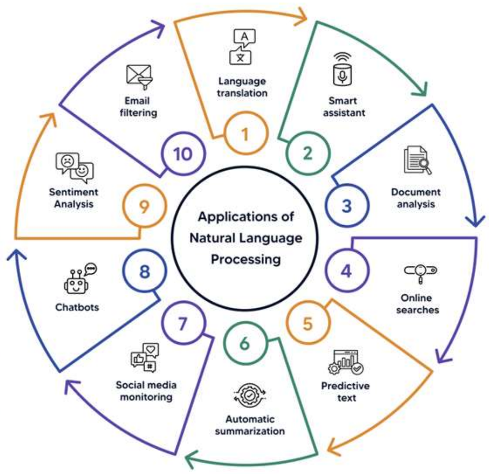
AlexNet	Overfeat	R-CNN	VGG	GoogLeNet
Overfitting required dropout	Time Constraints led to less experimentation	Computing region proposals and features was slow	Large number of parameters led to high training costs	No bounding box regression due to time constraints
Computationally expensive and long training times	Computationally expensive	Increased complexity due to 3 separate modules	Multi-cropping could have improved accuracy at the cost of time	Not very generalizable
Only useful for classification	No learnable region proposal mechanism			Very complex

72

Deep Learning in Natural Language Processing

Veronica Yuen
CSC 872

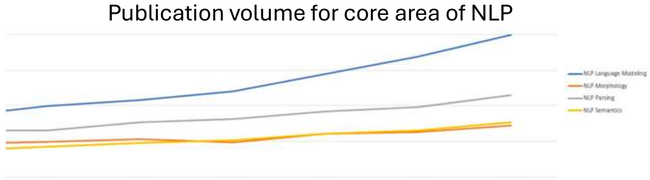




73

NLP over time

Publication volume for core area of NLP



Publication volume for applied area of NLP

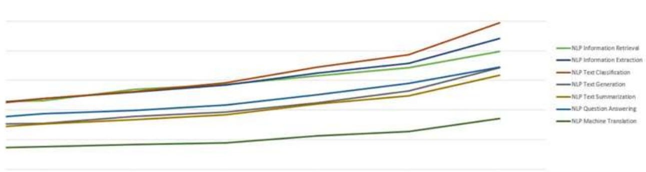
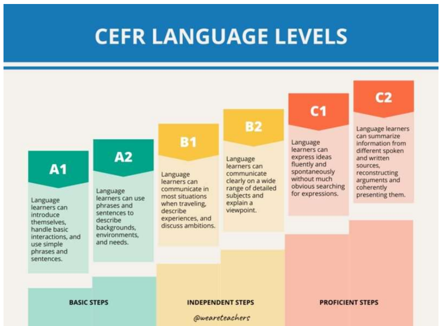


TABLE II. NEW VS. BASELINE MODEL COMPARISON

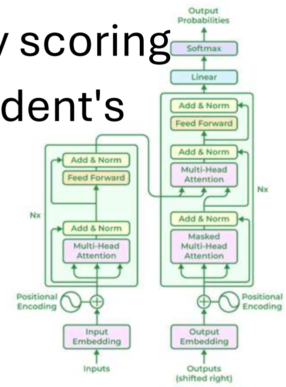
Indicator	EN-DE	EN-DE	EN-FR
BLEU	27.5	35.4	38.9
METEOR	33.2	40.7	43.1
TER	45.6	38.1	36.5
F-Measure	74.5%	81.2%	83.7%
Precision	75.5%	82.5%	85.0%
Recall	73.4%	80.0%	82.4%
ROUGE-L	72.0%	79.3%	81.8%
ChrF++	55.2%	62.8%	65.4%

74

Applying the framework

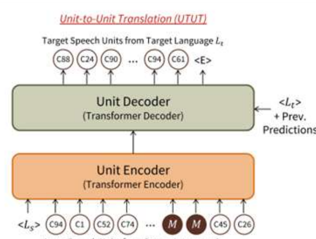


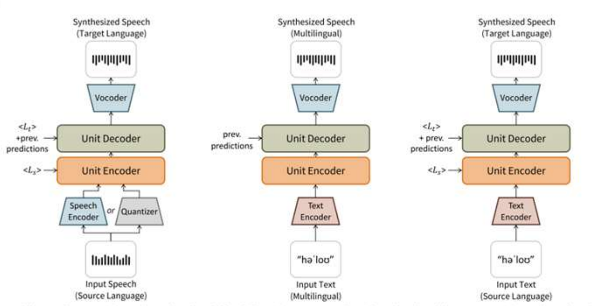
- Readability Classification
- Automated essay scoring
- Simulation of student's writing



75

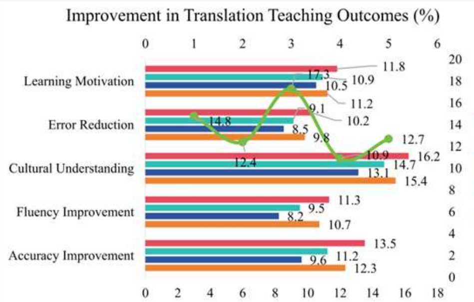
Textless training

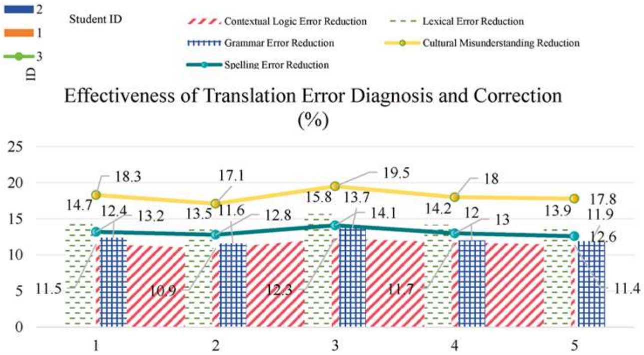




76

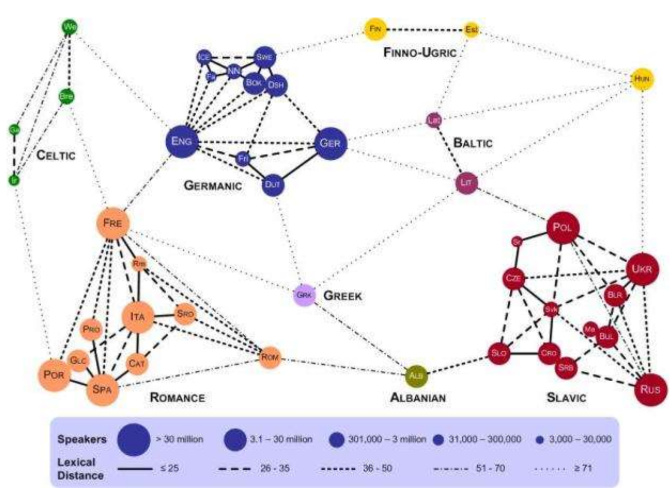
Teaching English translation

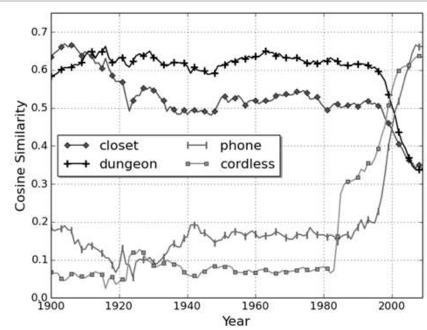




77

Challenges - language

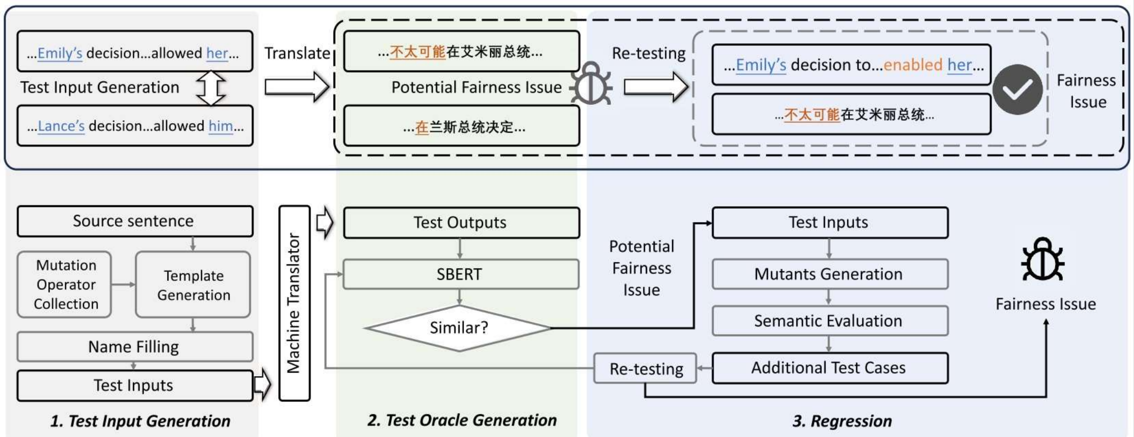




- Lexical similarity (with English)
  - German
    - About 60% similarity
  - French
    - 27 – 45%
  - Japanese
    - Limited lexical similarity
  - Chinese

78

Challenges - evaluation



79

What's next?

- New technology is that is needed?
  - Increased complexity
- Dataset
- Do all the features it make a difference to the end result?
- New methodologies

80