AI/ML Methods to Develop Superior Next Gen Autonomous Learning Robot Systems for Industrial Terrestrial and Space Applications*

Prof. Dr. V. David Sánchez A., Ph.D.
Brilliant Brains, Palo Alto, California
December 2024

Abstract

Optimization, planning and related processes host extremely relevant and challenging problems found in multiple scientific and industrial disciplines including robotics, VLSI among many others. They can be modeled in different ways with the generic goal in mind to reach the final state from a begin state in some optimal fashion, see Figure 1(a). I have launched for decades new programs and have conceived, developed and incorporated innovative AI/ML methods, algorithms to solve those problems in multiple hightech disciplines, among others, e.g., in automation, robotics, space, defense, computer vision, VLSI [6, 7, 8, 9]. Figure 1(b) and (c) show the deep impact high technology failures do cause to human life and the corporations' bottom line. One fatal accident of a Tesla model S autonomous vehicle after its collision with a Contra Costa County, California fire truck on February 18, 2023 and the chip division bug FDIV that cost Intel U.S.$ 475 million in 1994 are shown as examples. That restresses on the other hand the indispensable role of the verification of systems applying those new technologies in the framework of Research and Technology Development programs and projects throughout all industries, a part of my current activities, e.g. working with the design and verification of some of the most advanced system development programs at the NASA Jet Propulsion Laboratory (JPL) [4] and the U.S. Space Force (USSF) [5]. Figure 1(d) shows one of a multitude of industrial robotic applications in which a new generation of autonomous learning robots may have significant positive impact.

One area in which there has been steady progress w.r.t. the time and sample complexity of the learning systems applied, in particular for robotic applications, is in the provision of more powerful hardware to support industrial, physical AI-based systems, from humanoids to factories, that need to be accelerated across training, simulation and inference, see Figure 2. At the top row, several examples of such asset-, fleet-, factory-, warehouse-, network-, and infrastructure-scale robotic systems are shown. Well beyond those scenarios, the ecosystem of applications will extend among others to surgical rooms, data centers, traffic control systems, and entire smart cities operated by autonomous, interactive systems embodied by physical AI. Typical robotic embodiments that can perceive, reason, plan, act, and learn include manipulator arms, autonomous mobile robots (AMRs),

*This abstract has been granted permission for public release. The author is the youngest individual worldwide in history to be awarded the IEEE Fellow Prize ("Nobel" Prize in Engineering) [1] with mention "for leadership in neural and parallel computation, and pioneering contributions to autonomous space robots". He launched and led as Chief Scientist, EiC a scientific journal on AI/ML for 15 yrs. published by Elsevier Science [2]. As a decade civil servant at the German Aerospace Center DLR he launched with the Office of the German Federal Minister of Research and Technology in 1988 the First Federal Program for AI/ML Research and Technology Development in Germany [3], focused with his consortium including DLR and Siemens Corporate R&D in Munich, Germany on Learning Robotics and Automation. Most recently he helped NASA JPL design the next gen Mars autonomous helicopters [4] and the U.S. Space Force (USSF) design the next gen space defense systems against incoming hypersonic and ballistic missiles [5]. Before, he had led a highly classified DoD development of flying objects hosting neurochips with subcontractors including Lockheed Martin and SAIC.

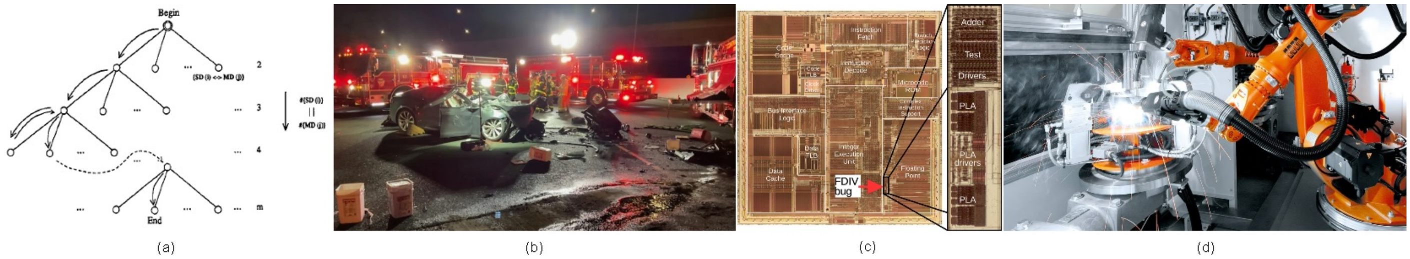

Figure 1: Sequential solution determination, optimization (a) Potential equivalent multi-paths between begin and end states [9] (b) Autonomous robotic vehicles' issues [TESLA] (c) VLSI electronic circuits' issues [INTEL] (d) Industrial robotic welding [KUKA]

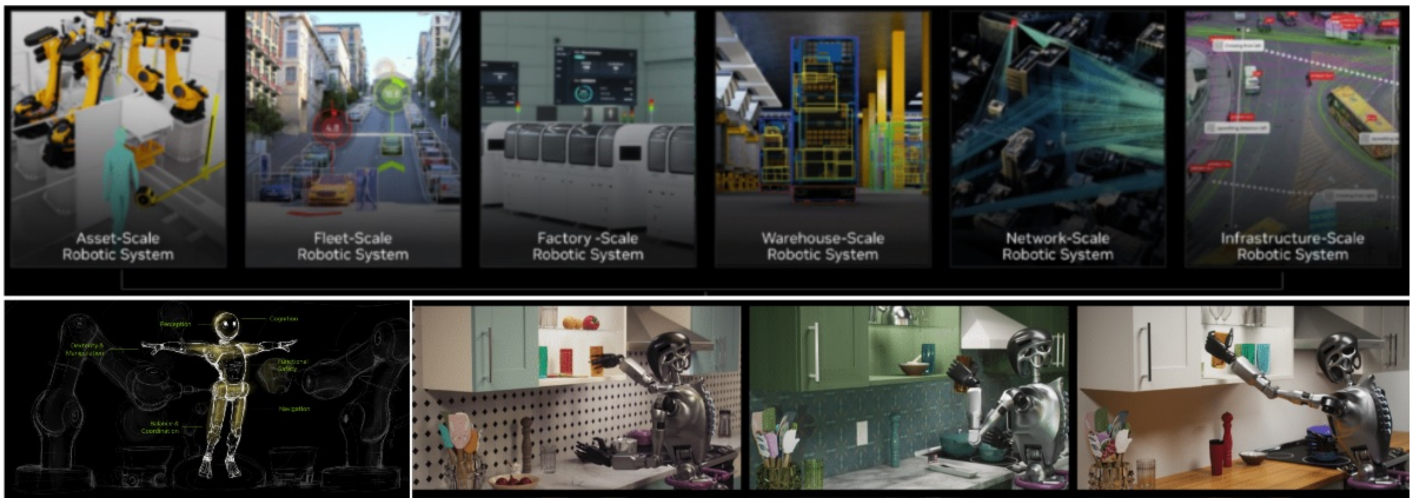

Figure 2: Applications of autonomous learning robotic systems [NVIDIA]

and humanoids. The bottom row of Figure 2 shows for the highly active of humanoid robot development to the left key areas of required improvement including dexterity and manipulation, balance and coordination, navigation, perception, cognition, and functional safety. On the right, a kitchen assistant is shown which needs to operate in different kitchen environments and autonomusly perform the specified tasking (actions) including, as shown, recognition and appropriate manipulation of kitchen objects.

For tasks involving sequential decision-making and dynamic environments needed in robotics, for a relatively long time, reinforcement learning (RL) [10] was the preferred choice next to supervised and unsupervised learning which could be applied in robotic recognition and classification subtasks when labeled data was either available or scarce, respectively. RL allows the interaction with the robot's environment and rewards successful behaviors to generate optimal actions for the task at hand. Actor-critic methods [11] aim at combining the strong points of actor-only and critic-only methods, e.g., when used for reinforcement learning. More complex state representations and the improvement of learning capabilities were enabled when deep neural networks (DNNs) were incorporated into RL leading to the origination of deep reinforcement learning (DRL) [12, 13].

Enabling artificial systems with fast learning capability has been a challenge for quite a while. Attempts to provide solutions led for example to imitation learning (IL) [14] which does not require careful, hand-crafted reward functions. Demonstrations (teaching) are typically provided in the form of state-action trajectories. The goal is learning behavior policy from these demonstrations. In more formal terms, given a finite dataset, in a practical setting: human demonstration data, $\xi^* = \{\tau_1 \cdots \tau_N\}$ of trajectories $\tau_i, \tau = (s, a), i = 1 \cdots N$ with s, a being the states and actions, respectively, the IL goal is to train a policy $\hat{\pi}(a|s, \theta)$ to reproduce an expert policy $\pi^*(a|s)$, typically provided by the aforementioned finite dataset. $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, p_0, \gamma)$ defines the asummed underlying Markov decision process (MDP) with $\mathcal{S}, \mathcal{A}$ being the state and action spaces, respectively. $\mathcal{T} : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$ is the state-transition dynamics and $\mathcal{R} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is the reward function, $p_0(s)$ is the initial state probability distribution, $\gamma \in [0, 1]$ is the discount factor to weight rewards. For a given timestep t and s_t, r_t the state and the reward following a policy from state s_t until the end of the episode at timestep T, the optimal expert policy is such that $\pi^* = \arg\max_{\pi \in \Pi} \mathbb{E}_{\tau \sim \pi}[R_0]$ and R_t is the discounted sum of rewards according to: $R_t = \sum_{k=0}^{T-t} \gamma^k \cdot r_{t+k+1}$.

Early attempts used Behavioral Cloning (BC) to supervised-learn policy mapping environment observations to optimal actions, susceptible to distribution shift [15]. The DAGGER (Dataset Aggregation) method [16] was introduced to iteratively train a stationary deterministic policy. It can provide a learning reduction with strong performance guarantees in both imitation learning and structured prediction. It addresses the issue of compounding errors in traditional BC approaches. Further improvements were provided among others by inverse reinforcement learning (IRL) [17] and inverse Q-learning (IQL) [18]. IRL extracts a reward function from observed optimal behavior. The issue of degeneracy, i.e., the existence of a large set of reward functions needs to be addressed. In one scenario, when the known policy is only given by a finite set of observed trajectories, a natural heuristics is proposed and the issue resolved using linear programming by choosing the reward function that maximally differentiates the observed policy from other, sub-optimal policies. IQL learns directly one single soft-Q function from expert data that implicitly represent both, reward and policy. It enables non-adversarial, dynamics-aware IL in offline and online settings. A wider exploration, range of actions in search of, while learning the optimal Q-function is promoted by the use of a soft-Q function. Softness is added via an entropy term to the reward calculation, controlled by a temperature parameter. The action probability is determined by softmaxing the Q-values which are assigned to state-action pairs and represent the quality, the expected reward for taking a particular action from the state at hand.

Figure 3(a) shows the set-up of Adversarial Imitation Learning with an agent and a discriminator posed as a minimax game [19], in which the agent policy model generates actions interacting with an environment to attain the highest rewards using RL, while the discriminator, similar to in GANs, acts as a reward model that indicates how expert-like an action is. The approach also suffers from mode collapse and gradient penalization. Figure 3(b) shows the architecture used in BESO (BEhavior generation with ScOre-based Diffusion Policies) [20], an approach which applies score-based diffusion models (SDMs) to introduce a policy representation for goal-conditioned imitation learning (GCIL) behavior. In GCIL, the agent learns to perform actions from expert demonstrations based on a specified goal state. On the left, the action generation process is depicted. In the middle, the model architecture used [21] is shown that incorporates conditioning functions $c's$ according to: $D_\theta(a|s, g, \sigma_t) = c_{skip}(\sigma_t)a + c_{out}(\sigma_t) \cdot F_\theta(c_{in}(\sigma_t)a, s, g, c_{noise}(\sigma_t))$, includes two preconditoning layers and skip-connections. On the right, the inner model $F_\theta(a, s, g, \sigma_t)$ is shown as a transformer with causal masking. Combining BESO with classifier-free guidance (CFG) training of SDMs, it can learn a goal-dependent policy and a goal-independent policy simultaneously. Figure 3(c) shows on the left an outline of the IRL problem and its involved minimax game over policy π and rewards r. The expert behavior at the unique saddle point solution (π^*, r^*) is found using RL. On the right, the IQL problem as well as a red line representing the optimal policy manifold, the softmax action of Q, are shown. This time, the problem is posed over the policy π and the Q-function Q to find the solution (π^*, Q^*). The key insight is to allow the Q-function to concurrently represent the optimal behavior policy and the reward function. RL does not need to find the policy any more. Each potential Q-function can be mapped to a pair of discriminator and generator networks which leads to a simple non-adversarial algorithm used for imitation.

Diffusion probabilistic models (DPMs) [22], a type of generative models, create new data by diffusing an available input data distribution $q(\mathbf{x}^{(0)})$ with noise and learning how to dediffuse the noise from the noisy data. One main goal is to convert the data into an analytically tractable distribution $\pi(\mathbf{x})$. Adding and removing the noise, within the forward (inference) and reverse diffusion process respectively, happens progressively. The reversing process effectively reconstructs the original data distribution or generates similar new data with high quality. The associated algorithm makes use of a Markov chain [23] to gradually convert one distribution into another. The forward

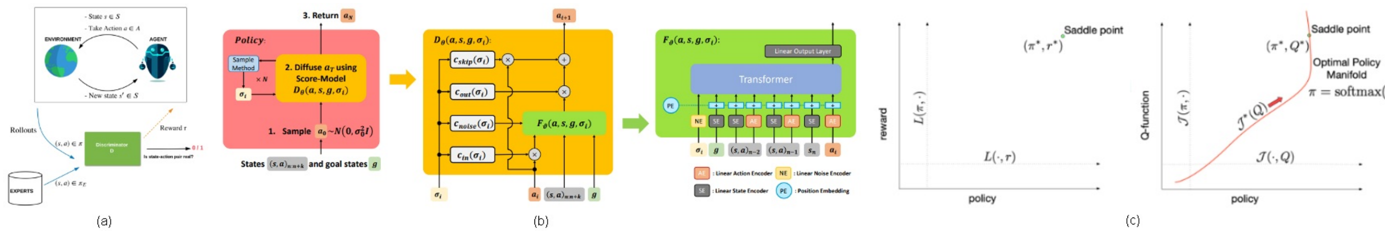

Figure 3: Imitation learning (IL) (a) Inverse Reinforcement Learning (IRL (b) BEhavior generation with ScOre-based Diffusion Policies (BESO) (c) Inverse Q-Learning (IQL)

trajectory repeatedly applies the Markov diffusion kernel T_π with β_t being the diffusion rate according to: $q(\mathbf{x}^{(t)}|\mathbf{x}^{(t-1)}) = T_\pi(\mathbf{x}^{(t)}|\mathbf{x}^{(t-1)}; \beta_t)$. Performing T steps of diffusion, the reversal trajectory describes the same trajectory, but in reverse, according to: $p(\mathbf{x}^{(0\cdots T)}) = p(\mathbf{x}^{(T)}) \prod_{t=1}^{T} p(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)})$. Learning involves estimating small perturbations to a diffusion process, i.e., training consists of finding the reverse Markov transitions which maximize the lower bound K on the log likelihood $L = \int d\mathbf{x}^{(0)} q(\mathbf{x}^{(0)}) \log p(\mathbf{x}^{(0)})$ according to $\hat{p}(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)}) = \arg\max_{p(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)})} K$. In a quasi-static process, the forward and reverse trajectories are identical and therefore: $L = K$. The reversal of the Markov diffusion chain which maps data to a noise distribution is estimated, the result being a model that can learn to fit any data distribution and remaining train-tractable.

The body of knowledge to solve related problems has emanated from nonequilibrium statistical physics including Langevin dynamics [24] which models the dynamics of molecular systems using Langevin equations that apply simplified models and account for omitted degrees of freedom using stochastic differential equations (SDEs). Langevin equations, reformulated as Fokker–Planck equations [25, 26] govern the probability distribution $p(\mathbf{x})$ of the random variable $\mathbf{x}$. Fokker–Planck equations are partial differential equations (PDEs) that describe the time evolution of the probability density function of the velocity of a particle under the influence of drag forces and random forces, as in Brownian motion. Further insight into nonequilibrium statistical physics, the Langevin and Fokker-Planck equations, their generalizations, solutions, and applications can be gained, e.g. in [27, 28]. To solve associated diffusion equations, we use stochastic differential equations (SDEs) which model stochastic processes according to: $\mathrm{d}\mathbf{x} = \mathbf{f}(\mathbf{x}, t)\ \mathrm{d}t + g(t)\ \mathrm{d}\mathbf{w}$ (forward Itô SDE) where $\mathbf{f}(\mathbf{x}, t)$ and $g(t)$ are the drift and diffusion coefficients, which represent the deterministic and stochastic influences, respectively, $\mathrm{d}t$ and $\mathrm{d}\mathbf{w}$ are the infinitesimal changes in time and noise (randomness), respectively. For the previously given forward SDE, the corresponding reverse SDE [29] is: $\mathrm{d}\mathbf{x} = [\mathbf{f}(\mathbf{x}, t) - g^2(t) \cdot \bigtriangledown_{\mathbf{x}} \log p_t(\mathbf{x})]\ \mathrm{d}t + g(t)\ \mathrm{d}\bar{\mathbf{w}}$. The score function of any given continuously differentiable probability density $p(\mathbf{x})$ is defined as $\bigtriangledown_{\mathbf{x}} \log p(\mathbf{x})$, needed to determine the reverse SDE as herewith shown. Figure 4(a) shows a solution of a 1-dimensional Fokker-Planck equation and two snapshots of its time evolution. Figure 4(b) shows three 2-dimensional clusters ($K = 3$). For Gaussian mixture models (GMMs), each cluster is centered at the means $\vec{\mu}_k$, has an ellipsoid around it of dimensions given by the covariance matrix Σ_k and a mixture component weight π_k. The a-posteriori probability can be estimated as: $p(\vec{x}) = \sum_{k=1}^{K} \pi_k \cdot \mathcal{N}(\vec{x}|\vec{\mu}_k, \Sigma_k)$ with $\mathcal{N}(\vec{x}|\vec{\mu}_k, \Sigma_k) = \frac{1}{\sqrt{(2\pi)^K |\Sigma_k|}} \exp\left(-\frac{1}{2}(\vec{x} - \vec{\mu}_k)^T \Sigma_k^{-1} (\vec{x} - \vec{\mu}_k)\right)$ and the constraint $\sum_{k=1}^{K} \pi_k = 1$. Figure 4(c) shows tractable forward and reverse diffusion processes in score-based generative modeling with SDEs, left and right, respectively. Perturbed data distributions evolve according to an SDE as the noise intensifies. With that SDE, data is mapped to the prior, a noise distribution, and estimating the score $\bigtriangledown_{\mathbf{x}} \log p_t(\mathbf{x})$, the SDE is reversed for generative modeling. The forward diffusion process consists of $\{\vec{x}(t)\}_{t=0}^{T}, t \in [0, T]$ with $\vec{x}(0) \sim p_0$ and $\vec{x}(T) \sim p_T$ given a dataset of samples. p_0, p_T are the data distribution and the prior distribution, respectively.

Connections between DPMs and denoising score matching with Langevin dynamics were introduced in denoising diffusion probabilistic models (DDPMs) [31]. The associated algorithm was trained on a weighted variational bound accordingly. These models allow for progressive lossy decompression generalizing autoregressive decoding. To sample from the corresponding distribution,

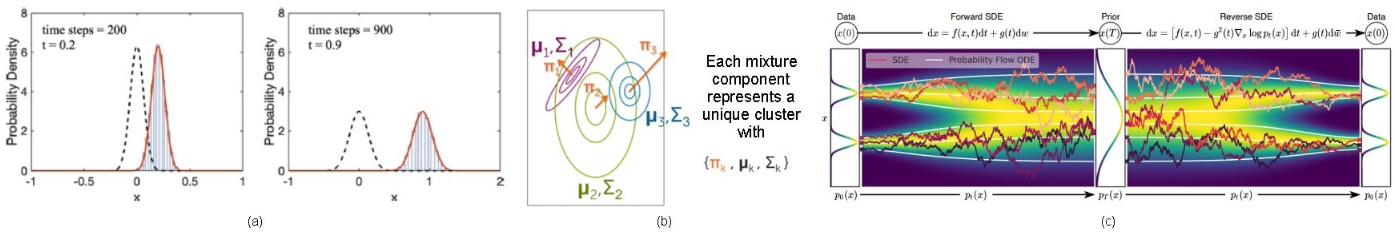

Figure 4: Gaussian probability distributions (a) Fokker-Planck equation: time evolution of probability density function (pdf) (b) Mixture of Gaussians: Three clusters in two dimensions (c) Forward and reverse diffusion processes and corresponding SDEs [30]

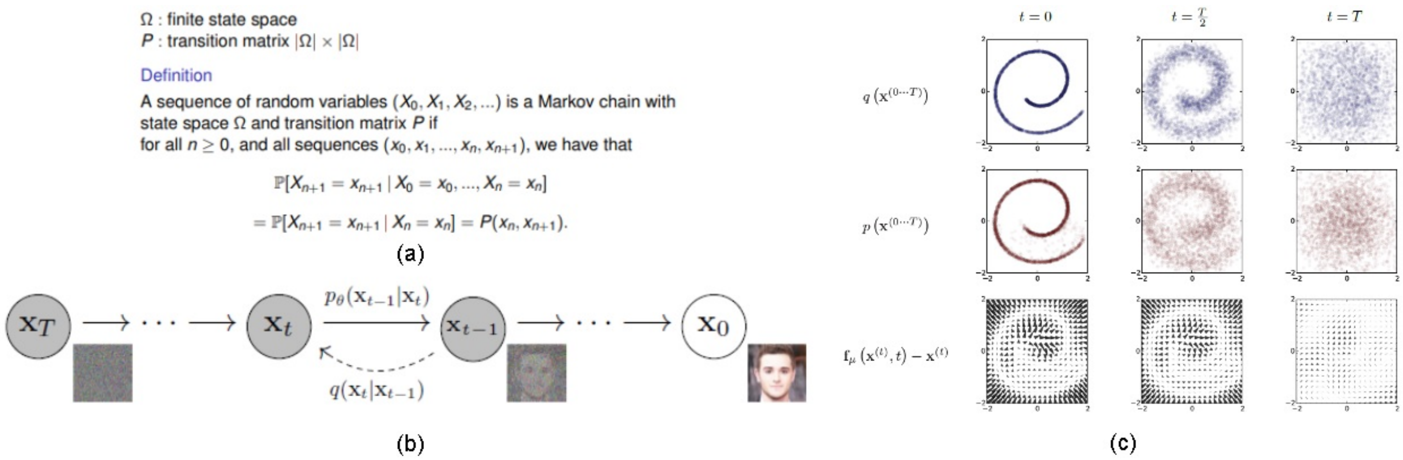

Figure 5: Markov chains and diffusion probabilistic models (a) Markov chain formal definition [MIT] (b) DDPM reverse trajectory [31] (c) DPM on 2-dim Swiss roll data [22]

the iterative evaluation, a noisy gradient ascent on the log-density $\log p(\mathbf{x})$, is performed applying annealed Langevin dynamics: $\mathbf{x}_t \leftarrow \mathbf{x}_{t-1} + \alpha_i \cdot s_\theta(\mathbf{x}_{t-1}, \sigma_i) + \sqrt{2\alpha_i} \cdot \mathbf{z}_t, 1 \leq i \leq L, 1 \leq t \leq T$. $\alpha_i = \epsilon \cdot \frac{\sigma_i^2}{\sigma_L^2} > 0$ is the step size, ϵ is the step size parameter, L is the number of steps, $\{\sigma_i\}_{i=1}^{L}$ is a sequence of noise scales, T is the number of iterations, diffusions, $\mathbf{z}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. $\mathbf{x}_0$ is an initial sample from any prior distribution $\pi(\mathbf{x})$. s_θ is the output of a joint neural network, called a noise conditional score network (NCSN), trained to approximate $\nabla_{\mathbf{x}} \log p(\mathbf{x})$, the score function. Figure 5(a) shows the formal definition of a Markov chain through the so called Markov property. Provided a graph of states and transitions with a probability assigned to each transition is given, according to the Markov property, a memoryless property of a stochastic process, only the present state influences the probability distribution of future states. In addition, being probabilities, they need to add up on a per state basis to 1. Figure 5(b) shows on the far left of the directed graph the input to the reverse trajectory of the used denoising diffusion probabilistic model (DDPM), visibly only noise, and on the far right the reconstructed original image. Figure 5(c) shows key modeling features of the diffusion probabilistic model (DPM) used on the 2-dim Swiss roll data. Time slices from the forward trajectory $q(\mathbf{x}^{(0 \cdots T)})$ are shown in the top row. The corresponding time slices from the trained reverse trajectory $p(\mathbf{x}^{(0 \cdots T)})$ are shown in the middle row. Finally, the bottom row shows the drift term $\mathbf{f}_\mu(\mathbf{x}^{(t)}, t) - \mathbf{x}^{(t)}$ for the reverse diffusion process of the same DPM when a radial basis function (RBF) network [32] is used to generate $\mathbf{f}_\mu(\mathbf{x}^{(t)}, t)$ and $\mathbf{f}_\Sigma(\mathbf{x}^{(t)}, t)$. These functions define the mean and covariance of the reverse Markov transitions for a Gaussian, respectively.

Figure 6(a) shows the density function $p(\mathbf{x})$ as contours and the score function $\nabla_{\mathbf{x}} \log p(\mathbf{x})$ as vector field of a mixture of two Gaussians. Once the score-based model $s_\theta(\underline{\mathbf{x}}) \approx \nabla_{\mathbf{x}} \log p(\mathbf{x})$ has been trained, Figure 6(b) shows how to use the Langevin dynamics to sample from the mixture of those two Gaussians. Figure 6(c) shows the annealed Langevin dynamics, a sequence of Langevin chains with gradually decreasing noise scales. Figure 6(d) is an example of the perturbation of data $\mathbf{x}$ with

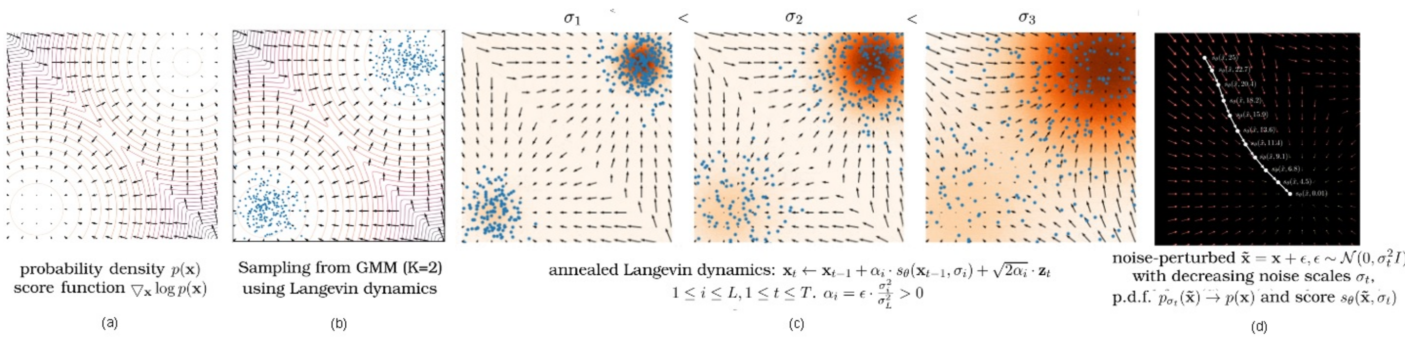

Figure 6: Annealed Langevin dynamics [CALTECH] (a) probability density and score function (b) sampling (c) sequence of Langevin chains with gradually decreasing noise scales (d) reaching a data point at score zero.

noise ϵ of a normal distribution $\mathcal{N}(0, \sigma_t^2 I)$ with decreasing noise scales σ_t, with p.d.f. $p_{\sigma_t}(\tilde{\mathbf{x}})$ and score function $s_\theta(\tilde{\mathbf{x}}, \sigma_t)$. It also shows the gradual approach towards a data point through an applied annealed Langevin dynamics sequence, iteration.

Variational auto-encoders (VAEs) [33] and generative adversarial networks (GANs) [34] are notable examples of generative modeling [35] methods and belong to the likelihood-based and implicit generative models [36], respectively. Other likelihood-based models include autoregressive models [37] that use past values to predict next values in a value sequence, normalizing flow models [38] that learn complex probability distributions by transforming a distribution into a more intricate one through a series of invertible transformations and provide accurate density estimation and sampling from the learned distribution, and Energy-Based Models (EBMs) [39] that encode dependencies between variables by associating a scalar energy, a measure of compatibility, to each configuration of the variables, and using a loss functional during learning to measure the quality of the available energy functions. The search, development, and use of unified frameworks to understand and compare learning methods comes handy. For example, in the case of Energy-based learning, both probabilistic and non-probabilistic learning can be viewed from that perspective and the absence of the normalization condition allows a much more flexible design of learning machines. So is the case too with score-based generative models, diffusion models in machine learning which, without knowing the full probability density function of the data distribution and making use of the stochastic differential equation (SDE) framework, learn the score function, i.e., the gradient of the log probability density function, recreate the training data and generate new data. They learn to navigate towards high probability areas of data space only using the estimated gradient.

Examples of advances in realized autonomous and learning capabilities built in robotic systems follow. It is by no means an exhaustive choice. Multiple techniques my teams and I developed and demonstrated in space missions are now applied to massive terrestrial applications including to develop real-time visual-based navigation for autonomous vehicles [40], driving and flying, or to demonstrate via teleoperation [41] new skills to autonomous, learning robots via diffusion policies. Figure 7(a) to (d) show ARMAR-III humanoid robots designed and built at the Karlsruhe Institute of Technology (KIT), my alma mater, conceived as household assistants and Figure 7(e) and (f) show ARMAR VI conceived for maintenance and repair tasks in industrial environments. ARMAR-III can perform the following tasks: it can move independently in the kitchen, open the refrigerator, get out a bottle of apple juice, recognize and grab different objects without damaging them or letting them fall down, it can open, load, and empty the dishwasher. The robot hand has built-in sensors to determine the position of the robot fingers, pressure and tactile sensors provide contact and force information. With that information, it can determine whether it has something in its hand or it is touching and whether the object is deformable or not. It has 4 cameras as eyes, two close-up and two wide/long-shot cameras to determine the distance to obstacles and other objects it needs to manipulate. To avoid collisions while moving around, it generates an internal model of its environment. For a natural communication with humans, it understands human language. It is able to store what it learns and recall it when needed. It can learn through imitation by observing human demonstrations of new

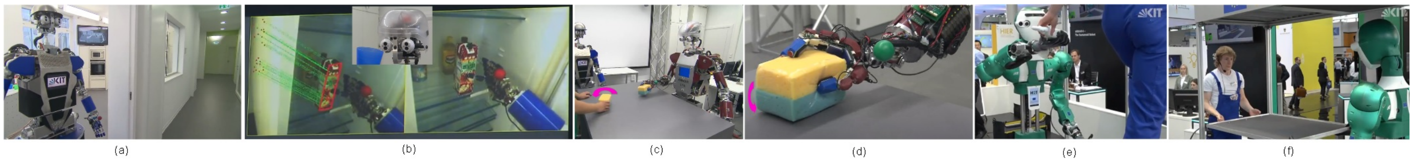

Figure 7: Imitation learning applied to humanoid robotics [KIT] (a) navigation (b) vision (c) human demonstration (d) robot imitation (e) receive spray bottle (f) help take cover down

skills. For example, from human demonstrations, it can learn, extract the form of the movement to clean up the table and the frequency of the movement. It can then reproduce the demonstrations' movement through its own movements and can adapt what it learned to clean up smaller or larger areas or to do it slower or faster. More recently, a memory system intended among others to build a bridge between sub-symbolic (sensorimotor) representation and symbolic (semantic) representation implemented in the robot software framework ArmarX has been described in [42].

Continuing advances in modern Machine Learning (ML) [43] and Deep Learning (DL) [44] have led to Generative AI (Gen AI) [45] and Foundation Models (FMs) [46] which notably include Large Language Models (LLMs) [47, Chapter 10]. Multiple Natural Language Processing (NLP) [48] tasks including summarization, question answering, sentiment, and machine translation can be posed as word prediction tasks, and as such, can be addressed using LLMs. whereas transformers have become the dominant architecture for state-of-the-art solutions to a variety of NLP tasks [49]. On the other hand, LLMs can also perform harmful tasks including bias, stereotypes, hallucinations, misinformation, privacy and copyright infringement, and propaganda. Those issues need to be addressed and solved when building real-world learning systems. If we go beyond language to behavior, i.e., develop artificial learning systems including intelligent, autonomous, learning robotic systems, then we need LBMs (Large Behavior Models) or the equivalent which learn from expert demonstrations diverse tasks including action patterns and contextual interactions. Rather than word prediction, emphasis is on actions, choices, and preferences and their derivations after training when exposed to unseen observations. They typically provide enhanced RL capabilities in corresponding applications, in particular with real-world feedback, but not only. LBMs have allowed, for example the Toyota Research Institute (TRI) and others, to be able to train learning robots with much less data and even to do it overnight per new skill as per their own reports [50]. Behavior cloning methods based on diffusion policies and human demonstrations have been the key for those developments. Diffusion policy is a novel way of generating robot behavior. As previously covered, the robot's visuomotor policy is essentially represented as a conditional denoising diffusion process. Several dextreous skills trained to TRI learning robots are shown in Figure 8. Multiple advanced applications of autonomous learning robots using beyond state-of-art learning methods including imitation and diffusion models are presented that represent a continuation of my work towards the factual intoduction of neurocomputers in industrial applications [51] and learning robots [52] to fulfill the inspiring dreams and bottom-line, factual needs of the industry and the marketplace. Associated success has originated partly by the incessant efforts towards increasing modeling dynamics [53] and the autonomy of advanced robotics systems, not only for space environments [54] but also for terrestrial applications [40, 41]. The integration of advanced AI/ML methods into real robotic systems is enabling them to become more human and obtain superhuman capabilities for terrestrial and space applications [55] and establishing a sophisticated automation and robotic infrastucture past AIOps for new manufacturing factories, but not only [56]. Multi-billion dollar investment opportunities and multi-trillion dollar market opportunities for autonomous learning robots are rapidly developing [57, 58] in diverse market segments including generalizable robots and robotaxi platforms redefining personal mobility.

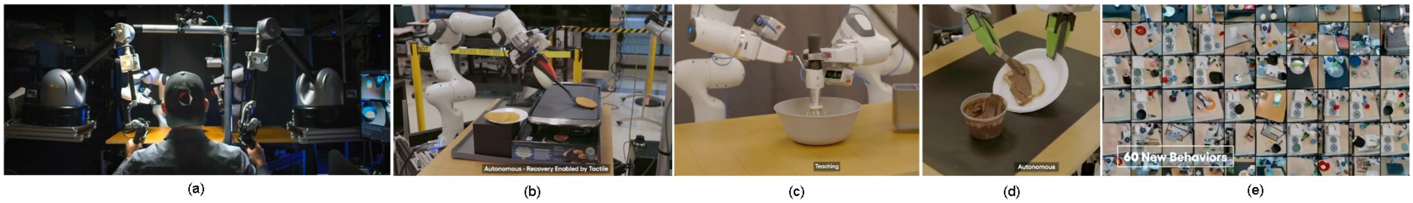

Figure 8: Dextreous skills learning via Large Behavioral Models (LBMs) [TRI] (a) skill demonstration through teleoperation (b) flip a pancake (c) use electric hand mixer (d) spread nutella on bread (e) so far (09/2024) 60 new behaviors

References

[1] V.D. Sánchez. *IEEE Fellow Award – "for leadership in neural and parallel computation, and pioneering contributions to autonomous space robots"*. 1995. https://profdrvdsaphd.lima-city.de/documents/IEEEFellow.pdf

[2] V.D. Sánchez. *Neurocomputing 50th volume anniversary*. Neurocomputing, 50, ix, 2003

[3] V.D. Sánchez. *Personal written communication with the Office of the German Federal Minister of Research and Technology BMFT – Start of the First German National AI/ML Research and Technology Development Program*. 1988.

[4] V.D. Sánchez. *Colonizing the Red Planet*. November 2022. https://profdrvdsaphd.lima-city.de/documents/MarsColonization.pdf

[5] V.D. Sánchez. *Advanced Hypersonic Technologies for Space and Terrestrial Applications (in German)*. August 2023. https://profdrvdsaphd.lima-city.de/documents/AdvancedHypersonic.pdf

[6] V.D. Sánchez. *Analysis and Implementation of Heuristical Algorithms for Two-Level Logic Minimization of Digital Circuits, 3 volumes (in German)*. Final Report, Karlsruhe Institute of Technology (KIT), February 1986.

[7] V.D. Sánchez and G. Hirzinger. *The State of the Art of Robot Learning Control Using Artificial Neural Networks*. In O. Khatib, J.J. Craig, and T. Lozano Perez (Eds.), The Robotics Review 2, Cambridge, Massachusetts, The MIT Press, 261–283, 1992.

[8] V.D. Sánchez et al. *Applications of Artificial Neural Networks in Automation, part 5: Neural Networks in the Robot Technology (in German)*. Automatisierungstechnische Praxis, 35 [5], 296-305, 1993.

[9] V.D. Sánchez. *Vision for Space Robots on a Parallel Distributed Architecture*. Integrated Computer-Aided Engineering ICAE, 1 [5], 431–452, 1994.

[10] R.S. Sutton and A.G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, 2nd Edition, 2018.

[11] V.R. Konda and J.N. Tsitsiklis. *On Actor-Critic Algorithms*. SIAM J. Control Optim, 42 [4], 1143–1166, 2003.

[12] V. Mnih et al. *Asynchronous Methods for Deep Reinforcement Learning*. In Proceedings of The 33rd International Conference on Machine Learning (ICML), New York, NY, 2016, 1928-1937.

[13] J. Schulman et al. *Proximal Policy Optimization Algorithms*. In Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS), Long Beach, CA, December 4–9, 2017.

[14] K. Arulkumaran and D.O. Lillrank. *A Pragmatic Look At Deep Imitation Learning*. In Proceedings of the 40th International Conference on Machine Learning (ICML 2023), Honolulu, HI, July 23–29, 2023, .

[15] D.A. Pomerleau. *ALVINN: An Autonomus Land Vehicle in a Neural Network*. In Proceedings of the 1st International Conference on Neural Information Processing Systems (NIPS), Denver, CO, December 1989, 305–313.

[16] S. Ross et al. *A reduction of imitation learning and structured prediction to no-regret onine learning*. In Proceedings of the 14th International Conference on Artificial Intelligence and Statistics, Ft. Lauderdale, FL, April 11 - 13, 2011, 627-635.

[17] A.Y. Ng and S.J. Russell. *Algorithms for Inverse Reinforcement Learning*. In Proceedings of the 7th International Conference on Machine Learning (ICML 2000), Stanford, CA, June 29–July 2, 2000, 663–670.

[18] D. Garg et al. *IQ-Learn: Inverse soft-Q Learning for Imitation*. In Proceedings of the 35th International Conference on Neural Information Processing Systems (NeurIPS), online, December 6–14, 2021.

[19] J. Ho and S. Ermon. *Generative Adversarial Imitation Learning*. In Proceedings of the Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, Dec 5–10, 2016, 4572–4580.

[20] M. Reuss et al. *Goal-Conditioned Imitation Learning using Score-based Diffusion Policies*. In Proceedings of Robotics: Science and Systems XIX (RSS2023), Daegu, Korea, July 10–14, 2023.

[21] T. Karras et al. *Elucidating the Design Space of Diffusion-Based Generative Models*. In Proceedings of the 36th Annual Conference on Neural Information Processing Systems (NeurIPS), New Orleans, LA, November 28 – December 9, 2022.

[22] J. Sohl-Dickstein et al. *Deep Unsupervised Learning using Nonequilibrium Thermodynamics*. In Proceedings of the 32nd International Conference on Machine Learning (ICML 2015), Lille, France, July 6–11, 2015, 2256–2265.

[23] A.A. Markov. *Extension of the law of large numbers to dependent quantities (in Russian)*. Izvestiia Fiz.-Matem. Obsch. Kazan Univ., (2nd Ser.), 15(1906), 135–156.

[24] P. Langevin. *Sur la théorie du mouvement brownien (in French)*. Comptes Rendus de l'Académie des Sciences, Paris, 146 (1908) 530–533.

[25] A.D. Fokker. *Die mittlere Energie rotierender elektrischer Dipole im Strahlungsfeld (in German)*. Annalen der Physik. 348 (4. Folge 43), 81–820, 1914.

[26] M. Planck. *Über einen Satz der statistischen Dynamik und seine Erweiterung in der Quantentheorie (in German)*. Sitzungsberichte der Preussischen Akademie der Wissenschaften zu Berlin 24, 324–341, 1917.

[27] H. Risken. *The Fokker-Planck Equation: Methods of Solution and Applications*. Springer, 2nd Edition, 1996.

[28] S.F. Kwok. *Langevin and Fokker-Planck Equations and their Generalizations – Descriptions and Solutions*. World Scientific 2018.

[29] B.D.O. Anderson. *Reverse-time diffusion equation models*. Stochastic Processes and their Applications, 12 [3], May 1982, 313–326.

[30] Y. Song et al. *Score-based generative modeling through stochastic differential equations*. In Proceedings of the 9th International Conference on Learning Representations (ICLR 2021), online, May 3–9, 2021.

[31] J. Ho et al. *Denoising Diffusion Probabilistic Models*. In Proceedings of the Neural Information Processing Systems (NeurIPS 2020), online, Dec 6–12,, 2020, 6840–6851.

[32] V.D. Sánchez. *Special issue on RBF Networks, parts I-II*. Neurocomputing, 19 [1–3] and 20 [1–3], 1998.

[33] D.P. Kingma and M. Welling. *Auto-Encoding Variational Bayes*. In Proceedings of the 2nd International Conference on Learning Representations (ICLR 2014), Banff, Canada, April 14–16, 2014.

[34] I.J. Goodfellow et al. *Generative Adversarial Nets*. In Proceedings of the 28th International Conference on Neural Information Processing Systems (NIPS), Montreal, Canada, December 8–13, 2014, 2672–2680.

[35] G.M. Harshvardhan et al. *A comprehensive survey and analysis of generative models in machine learning*. Computer Science Review, 38, November 2020.

[36] S. Mohamed and B. Lakshminarayanan. *Learning in implicit generative models*. In Proceedings of the 5th International Conference on Learning Representations (ICLR 2017), Toulon, France, April 24–26, 2017.

[37] H. Larochelle and I. Murray. *The neural autoregressive distribution estimator*. In Proceedings of the 14th International Conference on Artificial Intelligence and Statistics, Fort Lauderdale, FL, April 11-13, 2011, 29–37.

[38] L. Dinh et al. *NICE: Non-linear independent components estimation*. In Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015), Workshop, San Diego, CA, May 7–9, 2015.

[39] Y. LeCun et al. *A Tutorial on Energy-Based Learning*. In G. Bakir et al (Eds.), Predicting Structured Data, Cambridge, Massachusetts, The MIT Press, 2007.

[40] V.D. Sánchez. *Autonomous Vehicles for Space Exploration and Terrestrial Applications*. 2008. http://profdrvdsaphd.lima-city.de/documents/Autonomous.pdf

[41] V.D. Sánchez. *Telerobotics for Space Colonization and Industrial Applications*. 2007. https://profdrvdsaphd.lima-city.de/documents/Telerobotics.pdf

[42] F. Peller-Konrad et al. *A memory system of a robot cognitive architecture and its implementation in ArmarX*. Robotics and Autonomous Systems, 164, June 2023, 1–20.

[43] V.D. Sánchez. *Modern Machine Learning Technology*. December 2017. https://profdrvdsaphd.lima-city.de/documents/ModernMachineLearningTechnology.pdf

[44] C.M. Bishop and H. Bishop. *Deep Learning – Foundations and Concepts –*. Springer, 2024.

[45] V.D. Sánchez. *Advanced Gen AI & LLM Foundations and Applications – Paving the way to a more powerful and diverse ML –*. December 2023. https://profdrvdsaphd.lima-city.de/documents/AdvancedGenAILLMs.pdf

[46] R. Bommasani et al. *On the Opportunities and Risks of Foundation Models*. Report, Stanford Institute for Human-Centered Artificial Intelligence (HAI), Center for Research on Foundation Models (CRFM), Stanford University, August 2021.

[47] D. Jurafsky and J. Martin. *Speech and Language Processing – An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition –*. Prentice Hall, 3rd Edition Draft, August 20, 2024.

[48] C. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. The MIT Press, 1999.

[49] L. Tunstall et al. *Natural Language Processing with Transformers – Building Language Applications with Hugging Face –*. O'Reilly, Revised Edition, 2022.

[50] Toyota Research Institute (TRI). *Robots can learn new actions faster thanks to AI techniques*. The Economist, Science & Technology, November 27, 2024.

[51] V.D. Sánchez. *Neurocomputers in Industrial Applications (in German)*. Technische Rundschau, 82 [25], 60-65, 1990.

[52] V.D. Sánchez et al. *Neural Nets in Robotics (in German)*. Informationstechnik it, 33 [6], 317-322, 1991.

[53] V.D. Sánchez. *Modeling Dynamics for Communications, Navigation, Guidance and Control Applications*. Invited presentation at Rockwell Collins, Advanced Technology Center (ATC), Cedar Rapids, IA, 2011.

[54] V.D. Sánchez. *Increasing the Autonomy of Space Robots*. Invited presentation at the NASA Ames Research Center, Moffett Field, CA, 2002.

[55] V.D. Sánchez. *Enabling Robots to become more human and obtain superhuman capabilities for terrestrial and space applications using advanced AI/ML*. November 2024. https://profdrvdsaphd.lima-city.de/documents/EnablingSuperhumanRobotsUsingAdvancedAIML.pdf

[56] V.D. Sánchez. *Advanced Automation for mission-critical Information Technology past AIOps*. November 2024. https://profdrvdsaphd.lima-city.de/documents/AdvancedAutomationMisCriITPastAIOps.pdf

[57] D. Ledenkof. *Why Robotics Represents A $24 Trillion Opportunity For Investors*. Forbes, September 6, 2024.

[58] ARK Investment Management, Inc. *Big Ideas 2024*. Annual Research Report, January 31, 2024.