ASME Journal of Computing and Information Science in Engineering
Online journal at:
https://asmedigitalcollection.asme.org/computingengineering

Advancements in Kinematic Synthesis: Data-Driven Methods for Mechanism Design

Anurag Purwar[1]
Computer-Aided Design and Innovation Lab,
Department of Mechanical Engineering,
Stony Brook University,
Stony Brook, NY 11794-2300
e-mail: anurag.purwar@stonybrook.edu

Qiaode Jeffrey Ge
Computational Design Kinematics Lab,
Department of Mechanical Engineering,
Stony Brook University,
Stony Brook, NY 11794-2300
e-mail: qiaode.ge@stonybrook.edu

This article presents an overview of recent advancements in the field of kinematic synthesis of mechanisms, with a particular emphasis on unified geometric and machine learning-driven approaches. Historically, mechanism design followed a sequential process. Type synthesis was performed first, followed by dimensional synthesis, often guided by designer intuition and analytical formulations. However, modern methods have increasingly embraced algebraic fitting techniques based on kinematic mapping, projective geometry, and singular value decomposition. These methods allow for simultaneous type and dimensional synthesis, offering a unified computational framework capable of handling revolute and prismatic joints, exact and approximate task specifications, and higher-order constraints. In parallel, data-driven methods, particularly deep learning techniques such as variational autoencoders (VAEs), conditional VAEs, convolutional neural networks (CNNs), reinforcement learning (RL), and transformers, are reshaping the synthesis landscape. Applications span planar, spherical, and spatial mechanisms, with the ability to explore diverse, defect-free designs in real-time. Emerging tools like MotionGen demonstrate the practical viability of these techniques, offering interactive, designer-centric workflows. This convergence of classical kinematics with artificial intelligence is transforming mechanism design into a data-rich, intelligence-augmented discipline. The article discusses open challenges in representation learning, dataset generation, generalization, and human-in-the-loop synthesis, and outlines a vision for the future where machine learning and algebraic synthesis come together to empower creativity, automation, and innovation in mechanism design. [DOI: 10.1115/1.4070204]

Keywords: kinematic synthesis, unified type and dimensional synthesis, algebraic fitting, kinematic mapping, planar quaternions, motion and path synthesis, planar, spherical, and spatial linkage mechanisms, machine learning, data-driven mechanism design, deep neural networks, variational autoencoder (VAE), conditional VAE (cVAE), CNN/FCNN encoders, graph neural networks (GNNs), reinforcement learning

1 Introduction

This article presents a historical and forward-looking exploration of the kinematic synthesis of linkage-based mechanisms. We do not aim to provide an exhaustive survey. Instead, our goal is to position recent advancements within a broader context and highlights emerging challenges and opportunities.

A mechanism is a collection of links connected with kinematic pairs (or joints), which can be either lower pair (with surface contact) or higher pair (with point or line contact). Although not always visible, mechanisms are widely embedded in engineered products and systems such as automobiles, robots, biomechanical devices, as well as production systems for these products. As such, the study of kinematic geometry, i.e., the geometry of motions of mechanisms, is an extensively researched and highly developed subject [1–6]. Since early days of Industrial Revolution, machine theorists and kinematicians have long sought to develop geometrical and analytical methods so that engineers could approach the machine design problem in a rational way. An account of this history is given by Moon [7], who was also instrumental in creating the Kinematic Models for Design Digital Library (KMODDL) at Cornell University [8].

Kinematic synthesis, in its broadest sense, consists of type, structural, and dimensional syntheses. *Type synthesis* is concerned with the decision regarding the *type* of mechanism to be employed, such as the choice between a cam mechanism and a linkage; *structural synthesis*, also known as *topological synthesis*, is concerned with the number of links and joints as well as their patterns of connections; *dimensional synthesis* is concerned with the determination of those dimensions that affect the kinematic behavior of a mechanism. This article deals only with kinematic synthesis of linkages with lower pairs, but many of the ideas presented are general enough to be applied to other types of mechanisms, such as gears and cams. In this context, the terms, type synthesis, structural synthesis, or topological synthesis, are used interchangeably.

It is widely accepted that linkage synthesis follows a two-step process: type synthesis followed by dimensional synthesis [9] (Fig. 1(*a*)). The study of type synthesis begins with mobility or

[1]Corresponding author.

Manuscript received July 25, 2025; final manuscript received October 7, 2025; published online November 27, 2025. Assoc. Editor: Jami Shah.

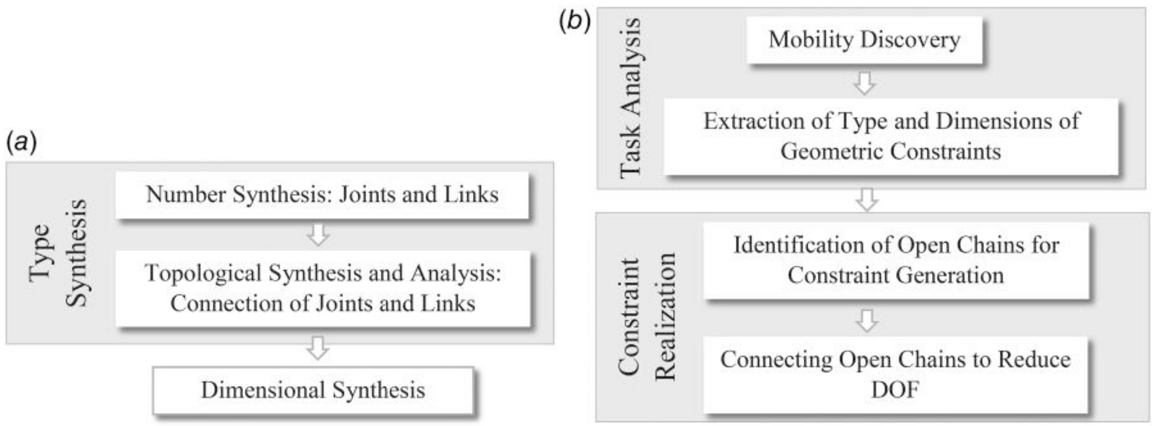

Fig. 1 Traditional versus task-driven approach: (*a*) traditional approach and (*b*) task-driven approach

movability analysis of planar mechanisms via the Grübler criterion [10] given by Grübler in 1883:

$$F = 3(n-1) - 2j \tag{1}$$

where F is the degree-of-freedom (DOF) of a mechanism, n is the number of links, and j is the number of single-DOF joints (R or P joints). It is remarkable that F depends only on the number of links and joints, and not on link dimensions, or other geometric, topological features. This forms the basis for *number synthesis*, i.e., the determination of the number of links and joints for a given value of F. The number synthesis gives rise to enumeration and classification of planar mechanisms [11].

Since the discovery of the spatial version of the Grübler criterion in 1917 [10], number synthesis has been extended to multi-DOF spatial and robotic mechanisms [6,12] including overconstrained systems [13,14]. To extend number synthesis one step closer to real-world applications, one has to distinguish rotational DOF from translational DOF. To this end, set-based methods for topological synthesis have been developed [15–18].

The field of kinematic synthesis and analysis has long benefited from authoritative reviews on analytical methods and digital computation [19–26], while recent machine learning (ML) surveys emphasize data-driven mappings [27]. This work connects these traditions by highlighting unified geometric formulations (Sec. 3) and learning-augmented pipelines (Sec.10), and by outlining open challenges and research opportunities. Rather than presenting a comprehensive survey, our goal in this article is to show how algebraic fitting/kinematic mapping and deep generative learning can be combined for simultaneous type and dimensional synthesis. We formalize hybrid pipelines in which learning imputes or filters task data, and unified geometry yields physically valid designs with guarantees. Finally, we offer a structured critique of current limitations—branch and circuit defects, generalization gaps, and dataset coverage/bias, and propose concrete mitigation strategies.

This article is divided into two parts: in the first part (Secs. 2–6), we review the latest advancements in modern algebraic approaches, and in the second part (Secs. 8–14), we present more recent deep generative neural network (NN) methods for kinematic synthesis of planar, spherical, and spatial linkage mechanisms. We finally present an ambitious five-year research roadmap for the kinematics community.

2 Data-Driven Methods for Mechanism Synthesis

The current state of art in kinematics and mechanism design is predicated on creating artificial boundaries between different types of mechanisms that apparently facilitate selection of the type of linkage mechanisms during the machine design process. However, this convenience comes at a great price of picking the wrong type at an early stage of the design process. Before dimensional synthesis, the designer must decide on the type of mechanism suitable for the motion requirement. While dimensional synthesis has been a subject amenable to mathematical treatments, identifying a mechanism type that matches with a given motion task is often driven by the designer's past kinematic experience. This two-step paradigm for mechanism design has been around for over a century. It is derived from the classical viewpoint that a mechanism is defined by links, joints, and the pattern of their interconnections. While this has led to effective means for mechanism classification and enumeration [6,11], it makes type synthesis a very challenging task, even for those who have been well trained in mechanism science. Erdman, Sandor, and Kota in their seminal Mechanism Design and Analysis text [3] summarize the importance and challenges of the type synthesis problem:

Type Synthesis strives to predict which combination of linkage topology and type of joints may be best suited to solve a particular task. Frequently, a novice designer may settle for a solution which merely satisfies the requirements, since there appears to be no method to find a *best* solution. Many experienced designers perform a rudimentary form of type synthesis, sometimes without being aware they are doing so. These experts have an innate *feel* for which type of linkage will work and which will not. This ability is developed only after designing linkages for many years and is difficult to pass on to young engineers. (Emphasis our's)

More significantly, the ability to decompose a design problem into type and dimensional syntheses is actually *data dependent*. We have recently shown that for motion generation problems, the type synthesis problem may not be solved without engaging in dimensional synthesis simultaneously. This necessitates a data-driven paradigm for simultaneous type and dimensional syntheses. The key is to establish a computational framework for mechanical design innovation by transforming the problem of *selecting* a mechanism type into that which is directly *computable* from the specified task. Here, the meaning of "data-driven" is two-fold: on one hand, the input data, i.e., the quantitative information about a specified task, drives the mechanism design process and will be used to determine both the type and dimensions of a desired mechanism; on the other hand, the data of previously designed and simulated mechanisms can be used to train ML models, which can infer new mechanisms when tasked with a new requirement.

2.1 Dimensional Synthesis for Motion Generation. As dimensional synthesis is highly conducive to mathematical formulations and digital computation, the volume of publications on the topic dwarfs those in type synthesis. A review of classical geometrical and analytical methods can be found in Ref. [10]. A comprehensive summary of the development of modern kinematics in the 40 years preceding 1992 can be found in Ref. [22], while somewhat more recent surveys can be found in Refs. [20,21,23–25].

The synthesis equations can be systematically derived with various mathematical formulation such as vector loop closure equations [3], quaternion and dual quaternion [28], and homogeneous matrices [29]. Su et al. [30,31] proposed an algebraic curve/surface formulation for spatial open chain synthesis that leads to a polynomial system. The techniques for finding solution to such a polynomial system in the context of mechanism design are well studied [32,33], including iterative optimization [34], exact analytical methods [35,36], and the continuation (homotopy) method [30,37]. In an editorial of *ASME Journal of Mechanisms and Robotics*, McCarthy gave a summary of the state of the art in dimensional synthesis [20]:

The goal of this survey is to show that in the past century our ability to analyze and design mechanisms and robotic systems of increasing complexity has depended on our ability to derive and solve the associated increasingly complex polynomial systems. From this, we can expect that advances in computer algebra and numerical continuation for the derivation and solution of the even more complex polynomial systems will advance research in mechanisms and robotics.

2.2 Combined Type and Dimensional Synthesis. Attempts have been made to address the problem of type synthesis together with that of dimensional synthesis, only for a very limited number of mechanism types [38–40]. They use the results of dimensional synthesis to evaluate mechanism types enumerated in type synthesis stage. Thus, their approaches do not reduce the complexity of the type synthesis problem. Researchers in artificial intelligence community have sought to develop *qualitative kinematics* [41] for bridging the gap between type and dimensional syntheses, but so far have achieved only very limited success.

Over the past decade, we have presented methods for extracting and identifying hidden geometric constraints in the input motion data [42–58]. The identified geometric constraints, in parameterized form, are then mapped to the type and dimensions of a desired mechanism for the given task (Fig. 1(*b*)). This means that the type synthesis is carried out only for the task at hand and for the geometric constraints that are already extracted, rather than enumeration of mechanism types based solely on *innate feel*. This data- (or task-) driven approach significantly reduces the scope and complexity in type and dimensional synthesis.

However, the above-mentioned methods suffer from age-old problem of circuit-, branch-, and order-defect. Moreover, they do not account for imprecise user input and generate a limited number of solutions. To address these problems, in the last seven years (2018–2025), we have turned our attention to using NNs in an effort to learn effective input/output representations, optimal NN architectures, and mappings between kinematic tasks and mechanism types and dimensions [59–71].

3 Unified Geometric Methods for Mechanism Synthesis

The emergence of *unified geometric methods using algebraic fitting* in recent years has introduced a new paradigm for simultaneous type and dimensional synthesis for motion generation problems. These methods use algebraic formulations and numerical linear algebra (e.g., singular value decomposition (SVD)) to unify type and dimensional synthesis into a single computational framework. By casting the synthesis problem as one of the fitting algebraic curves or surfaces to task specifications in a higher-dimensional projective space, one can simultaneously solve for the optimal mechanism structure and dimensions. This section provides a review of the state-of-the-art in kinematic synthesis using algebraic fitting, summarizing theoretical foundations and computational methodologies from the recent literature.

3.1 Background and Problem Formulation. Early work on planar four-bar motion generation was epitomized by Burmester's classical solution, which showed that at most five discrete poses of a coupler can be exactly attained by a planar 4R linkage (precision-point synthesis). With more than five desired poses or continuous motion requirements, only approximate synthesis is possible. The approximate motion synthesis problem was facilitated by the concept of *kinematic mapping*, first applied to mechanism design by Ravani and Roth [72]. Kinematic mapping represents planar rigid-body displacements as points in a higher-dimensional projective space called *image space*. Specifically, a general planar displacement (rotation plus translation) can be mapped to a point on a three-dimensional manifold (often using four coordinates, sometimes called *planar quaternions* or Study parameters) that lies on a quadratic hypersurface (the Study quadric) in $\mathbb{P}^3$. In this image space, the constraints imposed by a single-degree-of-freedom mechanism correspond to algebraic surfaces, often called *constraint manifolds*. Remarkably, the motion of the coupler of a planar 1-DOF linkage can be described as the intersection curve of two such constraint manifolds in the image space. This viewpoint transforms the synthesis task into an algebraic geometry problem: given a set of target poses (points in image space), one seeks two algebraic surfaces whose intersection passes through those points. In other words, mechanism design becomes a problem of fitting algebraic surfaces to data points representing the desired motion.

Traditional approaches to mechanism synthesis often involved solving nonlinear equations derived from loop closure or employing graphical methods and heuristics for type selection. These methods were case-specific and did not easily generalize to mixed types of joints or to approximate problems. By contrast, the algebraic fitting approach leverages the powerful tools of linear algebra and projective geometry: it formulates a *single unified design equation* whose solutions encode all possible mechanism parameters that achieve the given task (within a tolerance). Pioneering researchers like Suh and Radcliffe, McCarthy and Soh, and Sandor and Erdman developed formulations for specific types (e.g., 4R linkages). However, only recently have truly unified formulations appeared that can seamlessly handle a combination of revolute (R) and prismatic (P) joints and both exact and approximate syntheses in one framework. The key problem formulation enabling this is to express all the kinematic constraints (for all dyad types) as instances of a single *general algebraic form*. In the planar case, this general form turns out to be a second-degree (quadric) surface equation in the image-space coordinates. The task of mechanism synthesis thus reduces to determining the coefficients of two such quadric surfaces that best "fit" the given set of image points representing the desired motion. This formulation naturally accommodates *approximate synthesis*: if an exact solution for all points does not exist (the system is overconstrained), one can instead solve for surfaces that minimize a fitting error, yielding a mechanism that approximates the desired motion optimally in a least-squares sense.

3.2 G-Manifold and Constraint Manifolds. In the image-space formulation of planar kinematics, the feasible motion of a dyad (a two-link serial chain forming half of a four-bar) is represented by an algebraic surface, known as a *constraint manifold*. Different types of dyads produce different families of surfaces (e.g., a revolute–revolute dyad constrains a coupler point to a circle, a prismatic–prismatic dyad constrains a line to remain oriented in a certain direction, etc.). Ge et al. [47] introduced the concept of a *generalized manifold (G-manifold)* as a unified representation for all these constraint surfaces. A G-manifold is defined by a homogeneous quadratic equation in the image-space coordinates or planar quaternions (Z_i; $i = 1 \ldots 4$) with *eight* linear coefficients ($q_1 \ldots q_8$):

$$q_1(Z_1^2+Z_2^2)+q_2(Z_1Z_3-Z_2Z_4)+q_3(Z_2Z_3+Z_1Z_4)+q_4(Z_1Z_3+Z_2Z_4)+q_5(Z_2Z_3-Z_1Z_4)+q_6Z_3Z_4+q_7(Z_3^2-Z_4^2)+q_8(Z_3^2+Z_4^2)=0 \quad (2)$$

Remarkably, by appropriate specialization of these coefficients, one recovers the specific constraint manifolds for each dyad type (RR, RP, PR, PP). For instance, if the leading coefficient q_1 is nonzero, the G-manifold describes a circular constraint corresponding to an RR dyad; otherwise, the equation specializes to include a prismatic joint in the corresponding dyads (RP/PR/PP), which are further differentiated on the basis of the pattern of zeros in the coefficients. In fact, an RR dyad's image-space constraint is a one-sheeted hyperboloid surface. Setting certain parameters to zero changes this surface: for example, a PR (revolute-prismatic) dyad yields a hyperbolic paraboloid constraint surface (a saddle shape) as a special case of the hyperboloid when the circle radius parameter is very large. The G-manifold formulation thus provides a single algebraic form in which all planar dyad constraints can be expressed.

Using this unified view, one can interpret a planar four-bar linkage as the intersection of two G-manifolds (each corresponding to one of the two dyads that make up the four-bar). Traditionally, one might attempt to directly find two specific constraint manifolds that intersect at the task points; however, solving that problem directly is highly nonlinear. The breakthrough of algebraic fitting methods is to relax this initially: first, fit a more general surface (or family of surfaces) to the data, then later enforce that the surfaces correspond to real linkage constraints. In this approach, instead of immediately seeking two individual constraint manifolds, we consider a continuous one-parameter family of quadrics (a *pencil* of G-manifolds) that can interpolate the motion. By fitting this pencil to the task points, one essentially captures the "average" geometric constraints of the motion without yet deciding the joint types. The next step is then to pick out from this pencil the two specific members that correspond to actual dyad constraint surfaces (by imposing additional conditions on the coefficients). This two-step approach: fit a general form, then specialize, proved to be both computationally efficient and broadly applicable to different linkage types.

3.3 Planar Quaternions and Kinematic Mapping.

To apply the above ideas, one needs a concrete parameterization of planar displacements. A convenient choice, but not necessary, is the *planar quaternion*, also known as an image-space coordinate or Study parameter for $SE(2)$. One common definition is $Z = (Z_1, Z_2, Z_3, Z_4)$ with

$$\begin{aligned} Z_1 &= \frac{1}{2}\left(d_1 \sin\frac{\phi}{2} - d_2 \cos\frac{\phi}{2}\right) \\ Z_2 &= \frac{1}{2}\left(d_1 \cos\frac{\phi}{2} + d_2 \sin\frac{\phi}{2}\right) \\ Z_3 &= \sin\frac{\phi}{2} \\ Z_4 &= \cos\frac{\phi}{2} \end{aligned} \quad (3)$$

where (d_1, d_2) is the translation of the moving frame and ϕ is the rotation angle. Equation (3) defines a mapping from the Cartesian space parameters (d_1, d_2, ϕ) to a 3D projective quasi-elliptic space parameterized by the homogeneous coordinates of the point Z. This is called the kinematic mapping of planar displacements, and the corresponding 3D projective space is called the *Image Space* of planar displacement denoted by Σ. Thus, a planar displacement is represented by a point in Σ; a single-degree-of-freedom (DOF) motion is represented by a curve and a two DOF motion is represented by a surface in Σ. For details on kinematic mapping and the properties of the image space, see Refs. [2,72].

Using homogeneous coordinates for points and lines on the mechanism (e.g., a point on the coupler given by (x_1, x_2, x_3)), one can write linear transformation equations for how that point's coordinates transform from the moving frame to the fixed frame under displacement Z. Substituting these into the geometric constraint equations (e.g., the condition that a coupler point moves on a circle, or that a coupler line remains tangent to a fixed circle) yields quadratic equations in the Z_i as presented earlier in Eq. (2). The planar quaternion representation thus provides a unified coordinate system in which all these constraint conditions become algebraic equations of at most second degree. This is the key that allows forming a linear combination of basis equations to describe G-manifolds.

It is worth noting that the kinematic mapping approach is not limited to planar motions; an analogous approach exists for spherical mechanisms (mapping to $\mathbb{P}^3$ as well, but on a different quadric) and even for spatial mechanisms using Study's dual quaternions (mapping $SE(3)$ to points on a 7-dimensional Study quadric in $\mathbb{P}^7$). In this article, for brevity, we focus on planar mechanisms, but the success in the planar case has inspired similar developments in spherical and spatial mechanism synthesis [52,73–75] (e.g., using 10 homogeneous parameters for spherical RR dyads) and suggests a roadmap for future extension to spatial linkages (with significantly higher algebraic complexity).

3.4 Matrix-Based Approaches and Least-Squares Fitting.

Formulating the synthesis constraints in terms of image-space coordinates leads naturally to an over-determined linear system when more than the minimum number of task positions are specified. Suppose we have N desired planar poses (each pose provides specific values for (Z_1, Z_2, Z_3, Z_4) up to scale). We can substitute each pose's coordinates into the general G-manifold equation (the homogeneous quadratic form with eight coefficients $q_1, \ldots, q_8$). For pose i, this yields a linear equation in the q_j's. Stacking all N such equations for $i = 1 \ldots N$ gives a system $[A]\mathbf{q} = \mathbf{0}$, where $[A]$ is an $N \times 8$ design matrix dependent on the task poses. In general, N will be greater than 5 (for approximate synthesis problems), so no exact nonzero solution for $\mathbf{q}$ exists. Instead, one seeks a least-squares solution minimizing $[A]\mathbf{q}$ subject to some normalization (to avoid the trivial zero solution). This is exactly a problem of *total least-squares*, which is ideally solved by the SVD of $[A]$. SVD factors $[A]$ as $[U][S][V]^T$, where $[V]$ is 8×8 and its columns v_j are the right singular vectors of $[A]$. The singular vector corresponding to the smallest singular value gives the solution for $\mathbf{q}$ that minimizes the algebraic error $[A]\mathbf{q}$. In practice, when N is larger than 5, there will be a small set of singular values that are nearly zero; the subspace of $[V]$ corresponding to these smallest singular values represents (in an approximate sense) the family of quadric surfaces that come closest to containing all the task points.

Instead of selecting a single "best-fit" quadric, the algebraic fitting approach retains the null-space tied to the smallest singular values. This produces a low-dimensional family of solutions, known as a *pencil of G-manifolds*. For example, if N matches with the minimum required poses (say $N = 5$ poses for a four-bar, which is the exact synthesis scenario), then $[A]$ is 5×8 and has rank 5; three singular values will be zero (in exact arithmetic) and the null-space is three-dimensional. In that case, any $\mathbf{q}$ in this three-dimensional null-space defines a quadric that interpolates the five poses. A general linear combination $p = av_a + bv_b + cv_c$ (with v_a, v_b, v_c the basis of null-space) gives the coefficients of a candidate G-manifold. The true constraint surfaces we seek must satisfy additional nonlinear relationships (because not every quadric corresponds to a factorable pair of circles/lines) given by

$$\begin{aligned} q_1q_6 + q_2q_5 - q_3q_4 &= 0 \\ 2q_1q_7 - q_2q_4 - q_3q_5 &= 0 \end{aligned} \tag{4}$$

These relationships appear as quadratic equations in the parameters a, b, c when we enforce that $\mathbf{q}$ yields two valid dyad constraints. In the example of $N = 5$, one ends up with two quadratic equations in three variables a, b, c. Solving these simultaneously typically yields up to 2 solution sets (each quadratic is a cone of solutions intersecting in generally 4 points), meaning up to 4 valid choices of $\mathbf{q}$ that correspond to real mechanisms. Each such $\mathbf{q}$ actually encodes two dyads (because $\mathbf{q}$ was for the combined G-manifold pencil that contains both dyad constraints), but in practice one must factor the resulting quadrics to identify the two individual constraint manifolds. We have shown that this factorization can be done by looking at the pattern of zero entries in $\mathbf{q}$: for example, an RR dyad is indicated if certain coefficient pairs are nonzero versus zero, a PR dyad if a different subset of coefficients vanish, etc. Thus, each solution for (a, b, c) yields a specific combination of dyad types. In the five-pose case, up to four dyad combinations may emerge; since a four-bar is composed of two dyads, these four dyads can be paired in different ways to form up to six distinct linkage solutions. This matches the classical Burmester theory result: at most six 4R linkages (or fewer if some are geometrically invalid) can interpolate five poses.

In summary, the matrix [A] encapsulates the task data, and its SVD provides a direct linear solution for potential mechanism constraints in the form of G-manifolds. Additional nonlinear constraints (often a small system of quadratic equations) are then solved to extract the actual mechanism designs. The use of SVD not only finds an optimal fit in the least-squares sense, but also handles rank-deficient cases robustly, allowing a unified treatment of exact synthesis (where some singular values are exactly zero) and approximate synthesis (where those values are small but nonzero). The outcome of this process is a set of possible dyads, from which full mechanisms are assembled. Notably, Deng and Purwar [76] formulated a similar matrix-based approach that also incorporates higher-order kinematic constraints into [A], such as prescribed coupler velocities and accelerations at certain poses. In their formulation, each pose constraint can contribute multiple linear equations (for position, velocity, etc.), increasing the rows of [A] accordingly. The SVD-based solution then yields mechanism parameters that best satisfy all specified position, velocity, and acceleration requirements in one step, demonstrating the flexibility of the linear algebra approach. Moreover, they have shown that kinematic mapping and use of quaternions is not strictly necessary for unified algebraic methods.

4 Unified Type and Dimensional Synthesis

One of the major advantages of algebraic fitting techniques is the ability to perform *type synthesis and dimensional synthesis simultaneously*. In conventional design, an engineer would first decide, for example, whether a mechanism should be a crank-rocker or a slider-crank (type selection) and then solve for the dimensions. In the unified framework, the type (i.e., which joints are R or P and in what connectivity) is an *output* of the computation rather than an input. The algorithms achieve this by implicitly considering all dyad types during the fitting process, and then deducing the joint types from the solution.

4.1 Dyad Classification and Joint Types. After solving the linear least-squares problem, we obtain one or more solution vectors $q = (q_1, , q_8)$ for the G-manifold coefficients. These coefficients determine the presence or absence of certain terms in the quadratic constraint equations. By examining which coefficients are essentially zero, one can identify the corresponding dyad type. For instance, for an RR dyad (two revolute joints), the general quadratic constraint has no special zero coefficients (all q_i may be nonzero in general). In contrast, a PR dyad (grounded prismatic, moving revolute) imposes $q_1 = q_2 = q_3 = 0$ in the solution vector [47]. An RP dyad (grounded revolute, moving prismatic) yields a different signature (in the aforementioned formulation, $q_1 = q_4 = q_5 = 0$). A PP dyad (two prismatic joints) would set even more coefficients to zero ($q_1 = q_2 = q_3 = q_4 = q_5 = 0$ in that formulation). These zero-pattern conditions correspond exactly to the additional homogeneous quadratic constraints (Eq. (4)) that are imposed in the second stage of the algorithm. In practice, the algorithm enumerates the possible real solutions for these quadratic conditions, each such solution yielding a specific set q_i and thus a specific dyad type and geometry. The solving stage is relatively low-dimensional. For instance, in the case of five poses, the problem reduces to solving two quadratic equations. Polynomial solvers or eigensolvers can then identify all viable dyad options. The type of each dyad emerges automatically from its coefficients, unifying what was historically a separate, combinatorial search problem. As a result, one run of the algorithm provides a menu of mechanism types: it might return, for example, that a slider-crank solution exists that approximately achieves the task, as well as several 4R crank-rocker solutions, etc., leaving the designer to choose which type best suits other criteria (like size, manufacturability, or kinematic properties).

4.2 Geometric and Kinematic Constraints in One Framework. A key feature of recent unified synthesis methods is the ability to handle *mixed constraint specifications simultaneously*. Traditional mechanism synthesis distinguished between different design requirements and problem types: motion generation (rigid-body pose interpolation) versus path generation (point tracing), pose versus velocity, acceleration specifications, exact versus approximate synthesis, etc., and often required separate formulations. Algebraic fitting frameworks have demonstrated the capability to incorporate *pose constraints and higher-order constraints together* by expanding the set of equations. For example, Deng and Purwar [76] showed that up to five mixed constraints (position, velocity, acceleration, or even a fixed pivot location) can be exactly satisfied by a single dyad, since a general dyad has five degrees-of-freedom in its design parameters. If more than five constraints in total are specified, the system becomes overconstrained, but the SVD approach still yields a least-squares optimal solution where not all constraints are met exactly. Deshpande and Purwar [77] presented a framework, which allows the user to designate certain poses or dimensions as critical (to be met exactly) while others can be approximated. The linear algebra formulation can accommodate this by splitting the design matrix into parts corresponding to exact versus approximate constraints and treating the exact ones with additional Lagrange multipliers or null-space adjustments. They specifically addressed such scenarios in what they termed the "extended Burmester problem," which involves not only arbitrary numbers of poses but also various geometric constraints like prescribed fixed pivot locations or coupler points lying on specified lines. They demonstrated that both linear (point/line constraints) and nonlinear constraints can be integrated by expressing the nonlinear ones as algebraic conditions and then either solving them exactly (if possible) or relaxing them into the least-squares fitting (if no exact solution exists). Sharma et al. [51] presented a generalized framework to solve mixed pose and point synthesis problems, known as the Alt-Burmester problems by unifying them into an approximate mixed synthesis framework. This is essentially a unification of *selective precision synthesis* (a concept studied by previous researchers for mixing exact and approximate positions): the ability to seamlessly combine all these requirements demonstrates the flexibility of the algebraic fitting approach.

5 Applications to Planar Four-Bar and Six-Bar Linkages

5.1 Case Studies: Four-Bar Linkages. The unified synthesis approach has been applied to a variety of planar four-bar design problems, both classical and novel. A typical example is the five-pose motion generation problem (the classic Burmester problem) and its extensions. Deshpande and Purwar [77] demonstrated that if no exact four-bar solution exists for five specified poses, their algorithm can relax one or more poses and compute an *optimal approximate solution* that minimizes the fitting error. They presented examples where even when a classical solution fails (due to an overconstrained task), the unified method yields a linkage that comes as close as possible to achieving the task, by effectively distributing the error among the poses. Furthermore, their method can incorporate additional requirements such as a fixed pivot needing to lie on a given line or at a specific location. This allows practical design considerations (like space constraints or pin joint locations) to be included in the synthesis automatically.

Another case study is the design of a stamping press linkage to move a tool through a sequence of required positions. In this problem, five critical poses of the stamping tool were specified (with orientation changes), and the algorithm produced several candidate four-bar mechanisms. Some of these were crank-rocker linkages and others slider-crank types, all satisfying the poses within a small error. The designer can then choose among the solutions based on criteria like Grashof condition (continuous rotation of input crank) or transmission angles. For example, one solution might be a crank-rocker that achieves the motion but with a change-point singularity near one pose, while another solution might be a slider-crank that produces a defect-free motion but requires a sliding joint. The unified approach lays all these options on the table in one computation.

Yet another example comes from biomechanics: Deng and Purwar [76] employed a related synthesis algorithm to generate a planar four-bar to guide an ankle through a gait cycle of many poses. They used 60 snapshots of a human ankle trajectory and selected five extreme poses for synthesis. The algorithm produced an RRPR linkage that best fit those poses, effectively designing a candidate ankle-assist device. An effective software user interface can assist in choosing which poses to treat as exact versus approximate to yield a good overall fit.

5.2 Six-Bar Linkages. The task-driven algebraic synthesis approach has also been extended to more complex linkages, such as six-bar mechanisms (forming either Watt or Stephenson-type six-bar). Zhao et al. [55] showed that the unified geometric algorithm could be applied to six-bar linkages as well. The authors introduced a classification of six-bar solutions into three categories: type I six-bars are those where the end-effector's motion is constrained by two dyads (meaning a four-bar subchain exists within the six-bar, effectively the end-effector is guided by both dyads like a regular four-bar and an additional link extends it to six-bar); type II are those where the end-effector is constrained by only one dyad (the other dyad is used elsewhere in the chain); and type III have the end-effector not directly constrained by any dyad. In practical terms, a type I six-bar can be designed by first finding a satisfactory four-bar for the task (two dyads constraining the coupler), then adding a triad to input/output or end-effector link. A type II or III might be needed if no four-bar can achieve the task (so one dyad is not enough to guide the motion, requiring a triad). Their algorithm begins by analyzing five task poses using the planar quaternion formulation to find all feasible dyads (RR, RP, PR, PP) that can reach those poses. If two or four feasible dyads are found, those can form one or multiple four-bar solutions (if four dyads exist, they can pair up in combinations to form up to six four-bars; if two exist, they form one four-bar). If no four-bar is possible or the solutions are unsatisfactory, the method proceeds to synthesize a six-bar. This is done by introducing a "triad" (3R chain) to constrain the end-effector through the five poses, then adding two dyads to reduce the system to 1-DOF. The classification (type I/II/III) determines how this process is done. The result is an algorithm capable of finding six-bar linkages with any combination of R and P joints that achieve five precision points.

An example provided in the literature is the design of a six-bar lifting mechanism for assisting an elderly or disabled person from sitting to standing. Five key body postures in the sit-to-stand motion were specified. The synthesis algorithm yielded a Stephenson-type six-bar linkage that guides a support linkage through these postures. The output mechanism was novel in geometry and achieved the motion smoothly. This demonstrates the power of the approach to handle complex multimode motions where simpler linkages fail. By automatically considering topological alternatives (using both revolute and prismatic joints), the method can uncover nonintuitive designs that a traditional approach might miss.

While the six-bar synthesis is more involved (the solution space is larger and the polynomial equations potentially higher degree), it fundamentally relies on the same principles: formulate image-space constraints (now including triad constraints for 3R chains), use SVD to fit a general solution, then impose quadratic conditions to get actual linkage parameters. The framework unifies the treatment of mixed R/P joints just as in the four-bar case. Angeles and colleagues [78–81] had earlier enumerated solutions for Burmester problems; the newer approaches build on that to allow approximate solutions and to combine dyads into higher-order linkages.

5.3 Software Implementation: MotionGen. To make these advanced synthesis techniques accessible to engineers and students, interactive software tools have been developed. One notable example is the *MotionGen* application (Purwar et al. [48]). MotionGen[2] implements the unified type-and-dimensional synthesis algorithm for planar four-bars and provides an intuitive graphical user interface for motion design; see Fig. 2 for an example mechanism generated for five positions. The user can input a set of desired poses by sketching or by entering coordinates; the app then computes all viable four-bar linkages that achieve (or approximately achieve) those poses. What makes MotionGen particularly powerful is the ability to enforce practical constraints through the interface: the user can specify that certain pivot points must lie on a vertical line (simulating, say, a sliding connection to ground), or that a particular link is fixed, etc., and the algorithm incorporates those as additional linear constraints in the $[A]$ matrix.

An example case from the MotionGen article is the design of a landing gear retraction mechanism for an aircraft. The required motion (folding the landing gear linkage into the wing) was specified by a few key positions. MotionGen found two families of solutions: a non-Grashof four-bar (triple-rocker) and a Grashof crank-rocker, both satisfying the poses. The non-Grashof solution had shorter links and avoided interference with the fuselage but could not rotate fully; the Grashof solution allowed full rotation but required more space. The app enabled the designer to weigh these trade-offs quickly. Another case involved reverse-engineering a mechanism from a picture: by tracing the motion of a film projector mechanism and specifying a line constraint, MotionGen could synthesize a four-bar linkage that closely reproduced the motion, effectively solving a design problem that traditionally would require significant insight.

MotionGen outputs not just the linkage dimensions but also an interactive animation. The user can toggle through the different solution linkages (often numbered up to six) and see the coupler move through the prescribed poses. This visual feedback is invaluable in practice, as it allows checking for branch defects or circuit

[2]http://www.motiongen.io

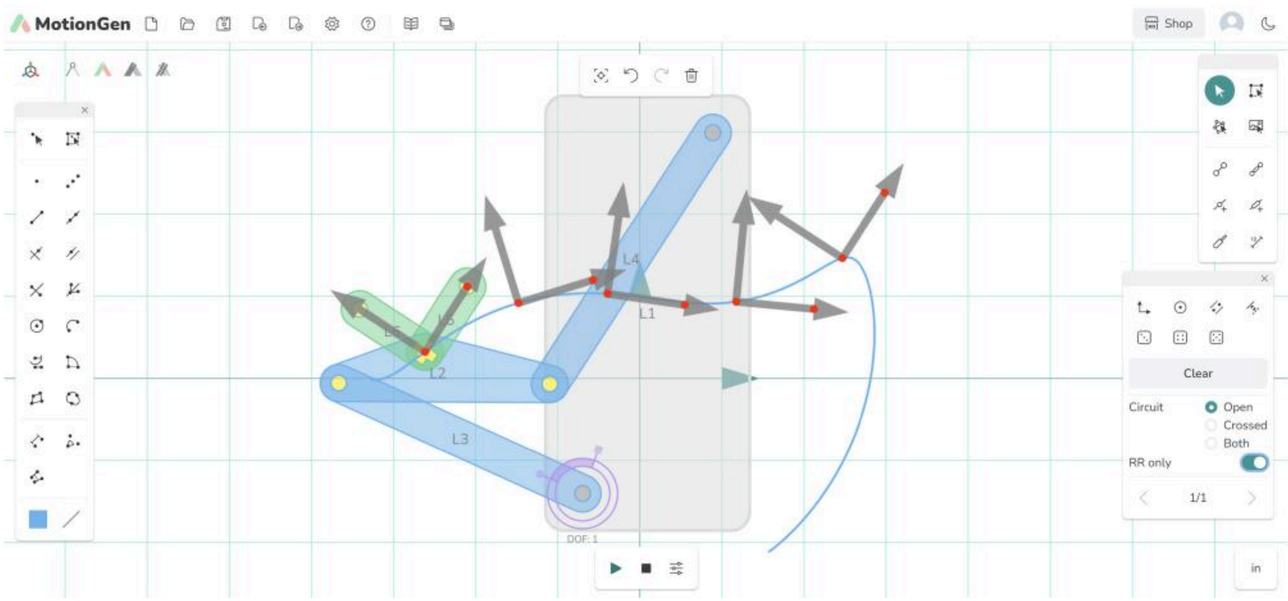

Fig. 2 An example four-bar linkage mechanism synthesized in MotionGen using algebraic fitting algorithm for five positions: users can input and tweak poses interactively and have the app compute type and dimensions

defects (situations where the linkage might have to flip to a different motion branch to reach all poses, which may be undesirable). In the past, designers would have to manually analyze the result or run separate simulations; with MotionGen, the analysis is built-in. The app also facilitates iterative design: if none of the solutions is satisfactory (perhaps one linkage is too large, another hits a singularity), the user can adjust the poses slightly or add a mild constraint (like a desired location for a pivot) and re-run the synthesis, instantly obtaining a new set of solutions. In educational settings like undergraduate Kinematic of Machinery class [82,83], this has been shown to greatly enhance understanding, as students can experiment with motion requirements and immediately see mechanism solutions.

Overall, MotionGen illustrates the practical impact of unified algebraic fitting methods reviewed earlier. What once was purely theoretical—fitting surfaces in a 3D projective space—has now been packaged into software that outputs real mechanism dimensions ready for prototyping. It serves as a bridge between advanced mathematical techniques and everyday engineering design.

6 Challenges and Open Problems With Unified Geometric Methods

Despite the significant progress in unifying mechanism synthesis, several challenges and open-ended issues remain:

1. *Computational Complexity and Robustness:* The core computations (SVD and solving small polynomial systems) are very fast for planar problems. However, as the number of constraints grows or when extending to more complex linkages, the size and degree of the polynomial systems can increase. One challenge is ensuring numerical stability when singular values are poorly separated. In such cases, deciding how many singular vectors to keep (i.e., the null-space dimension) is nontrivial. Methods to automatically detect the true rank deficiency of $[A]$ in the presence of noise are needed to distinguish an approximate solution from no solution. Another computational issue arises in the second-phase polynomial solving: although planar five-pose synthesis yields a quartic that is easily handled, if additional linear constraints are added (e.g., two specified pivots), one ends up solving a pair of quadratic and linear equations which is straightforward. But in more complicated scenarios, one might get higher-degree equations or need to resort to numerical root-finding. Ensuring the solver finds *all* solutions (including complex ones, to verify there are no missing real solutions) can be challenging. Leveraging tools from computational algebraic geometry (like Gröbner bases or resultants) is an area of ongoing development to improve robustness and reliability of the solution-finding stage.

2. *Solution Discrimination and Branch Defects:* When multiple mechanism solutions exist, choosing the "best" mechanism is not always straightforward. The algebraic fitting approach typically outputs all mathematically valid solutions, but some of these may be impractical. For instance, a solution might have two links that coincide (leading to a degenerate mechanism), or a linkage that functions but has a very poor transmission angle at some pose, or it might trace the desired coupler motion but only by switching circuit halfway (a branch defect). Automatically detecting and filtering out such cases is difficult to incorporate into the purely algebraic synthesis step. Researchers have addressed branch defect analysis as a separate step; for example, Schröcker et al. [84] showed how to identify branch transitions by examining the image-space curve's continuity. Integrating these checks into the design algorithm is an open problem. Ideally, the synthesis method could be augmented to include secondary objectives (maximize transmission angle, avoid singular positions, etc.) to preferentially return the most usable mechanism. This leans into multi-objective optimization, which in the context of algebraic fitting could be done by weighting certain error components or adding constraints (like ensuring a Grashof condition if a rotating crank is desired). Some initial work in this direction exists (using penalty functions for circuit limits, etc.), but a comprehensive solution is not yet established.

3. *Higher-Order and Spatial Linkages:* The extension of algebraic fitting to *spatial mechanisms* remains a formidable challenge. Planar and spherical cases benefit from relatively low-dimensional image spaces ($\mathbb{P}^3$) and quadratic constraint forms. For general spatial linkages (like a six-degree-of-freedom manipulator or even a single-loop Bennett mechanism), the image space lives in $\mathbb{P}^7$ and the constraint manifolds are defined by Study quadric and other higher-degree surfaces. A direct analogous approach would require fitting two algebraic hypersurfaces in $\mathbb{P}^7$ for a general spatial four-bar (which is a much more complex computation). Some progress has been made for specific cases: for example, Robson et al. [85] used algebraic methods for a spatial TS (universal-spherical) chain synthesis, and there are recent works on using dual quaternions in least-squares fitting for spatial RR,

ES, and SE chains [73], where E stands for a planar joint. However, a unified "all types" spatial linkage synthesis method is still out of reach.

Even within planar mechanisms, handling *higher mobility* systems (e.g., two-degree-of-freedom linkages or adjustable mechanisms) is an interesting direction. The current frameworks focus on 1-DOF linkages guiding a rigid body through a series of poses. If one wanted to design, say, a mechanism with a changeable configuration that can follow two different paths (like an adjustable four-bar), the theory would need to incorporate multiple intersection curves in the image space or additional parameters for mechanism configuration. This could be viewed as synthesizing a mechanism with a parametric motion—an unsolved problem that might require merging synthesis with control or tuning theory.

7 From Algebraic Fitting to Learning-Augmented Synthesis

Algebraic fitting (Sec. 3) guarantees physically valid dyads via image-space constraints and SVD-based pencils, but is sensitive to noisy/under-specified tasks and rank decisions. Learning (discussed later in Sec. 10) is robust to noise and supports one-to-many mappings, but benefits from physics to avoid invalid designs. While algebraic fitting provides a direct computational solution, designers often have additional criteria (minimize size, avoid branch and circuit defects, use preferred link lengths, etc.) that are not part of the basic synthesis equations. Currently, a practical workflow might be: use the algebraic method to get candidate solutions, then run a numerical optimization (e.g., genetic algorithm or gradient-based) to fine-tune a solution for secondary objectives. An open question is how to integrate these steps. One could envision a hybrid algorithm that generates an initial population of solutions via algebraic fitting (guaranteeing good baseline performance) and then uses stochastic optimization to improve specific metrics. Alternatively, machine learning techniques could be applied: for example, training a model on many synthesis results to predict which design a user is likely to prefer, or to quickly approximate the mapping from task specifications to mechanism dimensions (bypassing some computation). These directions are speculative but represent the cutting edge of combining classical kinematics with modern AI.

8 Machine Learning-Driven Approaches in Mechanism Design

Mechanism synthesis has historically relied on analytical or optimization-based methods that require precise inputs (e.g., discrete precision points) and often yield a small number of highly sensitive solutions. Even slight perturbations in a prescribed motion or path can drastically alter the result or introduce defects (branch, circuit, order) in the mechanism. This sensitivity stems from the nonlinear nature of the design space and the limitations of interpolation-based formulations. In practice, however, design specifications are frequently *imprecise* or *uncertain*, and designers are more interested in exploring a *diverse set of plausible solutions* rather than a single exact solution. Traditional methods offer little guidance when an initial guess fails—the process can devolve into a tedious trial-and-error exercise of tweaking input specifications. The aforementioned geometric methods suffer from these problems as well.

Recent advances in deep learning provide a compelling alternative for such problems characterized by difficult analytical representation, noisy or sparse input data, and enormous solution spaces. *Deep neural networks* can serve as powerful function approximators that learn complex mappings from mechanism specifications to designs, without the need for hand-crafted features. Unlike classical neural networks that required manually engineered features (e.g., Fourier coefficients of a curve) to map non-Euclidean design data into a Euclidean space, modern deep learning can learn the feature representation directly from data. This ability is crucial in mechanism design, where both inputs (paths, motions) and outputs (mechanism topology and dimensions) may lie on non-Euclidean manifolds (e.g., space of coupler motions/curves or graph-based linkages). By leveraging architectures suited to the data type (convolutional networks for images embedding coupler curves, graph neural networks (GNNs) for graph-based mechanism topologies, etc.), deep learning methods can automatically extract salient features and embed mechanism design data into a latent Euclidean space for efficient learning.

In summary, the motivations for adopting data-driven, learning-based approaches in kinematic synthesis include: (1) handling imprecise or probabilistic inputs (such as a roughly sketched path) and providing intelligent feedback or corrections, (2) exploring *large design spaces* to find many diverse solutions (including variation in mechanism type and dimensions) rather than a single optimum, (3) leveraging knowledge from big data (large databases of known mechanisms) to guide synthesis toward feasible regions and avoid pitfalls like defect-laden designs, and (4) enabling a higher-level design process where the algorithm can suggest creative alternatives and the designer can steer or select among them. As discussed below, our recent research has made progress on some of these fronts by integrating deep learning models with mechanism design.

9 Representations of Mechanisms and Motion for Learning

A cornerstone of data-driven mechanism design is the choice of representation for mechanism motion (e.g., coupler paths) and mechanism structure, since this is what neural networks will consume or produce. Early works on applying neural networks to mechanism synthesis introduced engineered descriptors of the coupler curve. A classic approach is to represent a closed coupler path as a Fourier descriptor: treating the (x, y) coupler coordinates as a complex function $z(t) = x(t) + iy(t)$, one can express it as a Fourier series $z(t) = \sum_{m=-\infty}^{\infty} a_m e^{2\pi imt}$, with coefficients a_m that capture shape characteristics. For a discrete sequence of N points, the descriptors can be computed as

$$a_m = \frac{1}{N}\sum_{k=0}^{N-1} z_k e^{-2\pi im\,(k/N)}$$

where $z_k = x_k + iy_k$ are the sample points. Truncating to a finite number of harmonics m yields an efficient, reduced representation of the curve. In fact, it has been shown that using five harmonics (fundamental plus a few) can approximate typical four-bar coupler curves with good accuracy [86]. Similarly, *wavelet descriptors* provide a multiresolution representation of curves: by performing a wavelet decomposition (e.g., Daubechies wavelets) on the $x(t)$ and $y(t)$ coordinate signals, one obtains wavelet coefficients that encode both low-frequency shape and high-frequency details. For example, using a certain number of Daubechies wavelet coefficients (e.g., 38, 76, and 136 coefficients at increasing detail levels) can represent the coupler path with varying fidelity. Prior to the deep learning era, researchers explored using such descriptors in neural network models—Vasiliu and Yannou [87] trained a neural network to map desired path descriptors to linkage dimensions, while Galan-Marin et al. [88] optimized four-bar path generation by combining wavelet-based shape descriptors with neural nets. These studies demonstrated the feasibility of learning a mapping from *shape space* (paths) to *design space* (mechanism parameters) when using compact descriptors of the shape.

However, hand-crafted representations have limitations. They require the curve to be parameterized (e.g., equal-length sampling for Fourier) and often assume a fixed sequence length or ordering of points. To overcome this, recent work has introduced image-based representations of coupler curves that are invariant to the path parameterization. In an image-based model, the 2D workspace is discretized into pixels, and a coupler path is rasterized into an image—for instance, by drawing a thin line or a probability density where pixel intensity reflects the likelihood of the coupler passing through that pixel. This approach offers several advantages: (1) the input size is fixed (e.g., a 64–64 pixel grid) regardless of the number of points on the curve, (2) it captures spatial correlations (nearby points on the curve appear as continuous strokes in the image) that one-dimensional sequences might miss, and (3) it is agnostic to the ordering or starting point of the curve. Effectively, the curve is treated as an un-ordered set of points embedded in a grid. Normalization is applied so that the image focuses on shape rather than extrinsics: typically the coupler path is translated to the origin, scaled to fit a unit bounding box, and rotated to a canonical orientation (e.g., principal axes aligned to image axes). This removes variations in position, size, and rotation ensuring that functionally similar coupler curves map to similar images.

Deshpande and Purwar [61] demonstrated that an image-based representation enables a unified treatment of different coupler curves irrespective of their algebraic complexity. Since any mechanism's coupler curve (from four-/six-bar, or other linkage) can be rendered as an image, a single learning framework can, in principle, handle synthesis for multiple mechanism topologies without specialized formulations for each. This is a significant step toward *type synthesis*: the network is not inherently biased to one linkage type and could potentially generate or identify the appropriate type given the path requirements. In practice, one may still need to label or encode the mechanism type in the output, but the input representation is no longer a limiting factor. Another aspect of representation is how to encode the mechanism design (output) itself for learning. Some approaches directly predict continuous parameters like link lengths or joint coordinates or a graph-based representation. While current deep models for linkage synthesis often output just the dimensional parameters for a fixed topology, future work is considering graph-based neural networks to handle variable topologies as well. Indeed, representing a mechanism as a graph (links = nodes, joints = edges) and using GNNs could allow learning across mechanism types and even suggest new topologies.

10 Deep Learning Techniques for Kinematic Synthesis

Next, we review several deep neural network architectures that have been used in the recent past to map design specifications to the mechanism parameters.

10.1 Variational Autoencoders for Generative Design. A variational autoencoder (VAE) [89] is a deep generative model that consists of an encoder network $q_\phi(z|x)$ that maps an input x (e.g., the image of a coupler curve) to a distribution over latent variables z, and a decoder network $p_\theta(x|z)$ that maps a latent sample back to the data space, attempting to reconstruct the original input. Training maximizes the evidence lower bound (ELBO), which balances reconstruction accuracy and latent space regularity (via Kullback–Leibler divergence to a prior). In mechanism design, VAEs have been applied to learn the distribution of coupler curves from a large dataset of mechanisms. For example, Nurizada et al. [62] trained a convolutional VAE on approximately 3 million coupler curve images (obtained from 4-, 6-, and 8-bar linkages) to obtain a 10-dimensional latent representation of curve shapes.[3] Crucially, in the learned latent space *similar curves are clustered together* as the VAE's encoder abstracts high-level shape features and maps analogous coupler trajectories to nearby points in $\mathbb{R}^{10}$. By working in this latent space, one can perform efficient design retrieval and interpolation. Given a desired path (which might not exactly exist in the database), one can encode it to a latent point z^* and then find its k-nearest neighbors in latent space among the known designs. The decoder alone would merely reconstruct the input curve, which is not useful from the design perspective. Nurizada et al. [62] integrated this with a secondary neural network, which is a fully connected neural network (FCNN), that maps the coupler curve latent vector to the mechanism parameters. In their VAE-FCNN pipeline, the VAE handles the shape abstraction and provides a continuous design space for interpolation, while the FCNN learns to predict mechanism dimensions for a given coupler shape. By not directly decoding the VAE's output, but rather using it to query known mechanism solutions, the approach sidesteps the risk of generating physically invalid mechanisms and instead leverages the repository of defect-free mechanisms in the dataset.

Another powerful capability of generative models like VAEs is *latent space exploration*. Because the latent space is smooth and structured, one can take two known mechanism motions and interpolate between their latent encodings to generate a continuum of intermediate motions, potentially synthesizing new mechanisms that were never explicitly designed before. Deshpande and Purwar [60] demonstrated this by linearly interpolating between latent vectors of two four-bar coupler curves: the decoder generates a series of coupler curves morphing from one shape to the other. Interestingly, many of these intermediate curves are themselves valid coupler curves of some four-bar mechanism, showing that the VAE latent space has captured a meaningful manifold of feasible mechanism motions. This opens up avenues for *computational creativity*, where designers can explore novel motions by traversing latent space and then use a decoding or search mechanism to find a linkage that realizes those motions. In fact, "variational synthesis" of mechanisms has been proposed as a paradigm wherein the user is given high-level control to navigate the latent design space, while the system ensures that any point in this space corresponds to a plausible mechanism. For example, a designer could specify an approximate or partial motion, and the system would generate a family of complete motion variants (via VAE decoding) along with candidate mechanisms for each variant. Such a framework effectively *bridges data-driven generative models with traditional kinematic solvers*: the generative model supplies intelligently perturbed or completed inputs that are likely to produce good results, which are then fed to a solver or evaluator, thus closing the loop with human guidance.

10.2 Conditional Generative Models for End-to-End Synthesis. While a two-step approach (generate curve to search mechanism) is quite effective and fast, ultimately one desires an end-to-end mapping from the task specification to mechanism design. This is challenging because the mapping is one-to-many: a given motion may be produced by many different mechanisms. One solution is to predict a *distribution* of mechanisms rather than a single point. This is precisely what conditional VAEs (cVAEs) are designed to do. In a cVAE, one conditions the generative model on the input (e.g., the coupler path or its image) so that the latent space captures variability in the outputs (mechanism parameters) corresponding to that input. Deshpande and Purwar [60] implemented a cVAE that approximates the conditional distribution of linkage parameters given a coupler trajectory specification. In practice, this was an encoder that takes as input a coupler curve image (the "design specification") and outputs a latent code, and a decoder that outputs a set of mechanism parameters (link lengths, etc.) conditioned on that code. By sampling multiple latent vectors, the cVAE can generate multiple candidate mechanisms for the same input path, effectively providing a distribution of solutions rather than a single point estimate. This approach was shown to produce an array of plausible four-bar linkages for a given target path or motion.

[3] See the original article for a link to publicly available dataset.

It is important to note that such end-to-end learning benefits greatly from a large and comprehensive training dataset, which until recently was lacking. The creation of a 3 million mechanism dataset [62] (covering 4-bar, 6-bar, and 8-bar linkages of many types) has been a pivotal enabler for training deep models that generalize. This dataset provides not only the mechanism parameters but also labeled topology information and normalized coupler curve images for each mechanism. The diversity in this dataset (177 mechanism types in total) encourages conditional models to handle variations in topology. Another useful dataset of 100 million mechanism is by Nobari et al. [90], although it is limited the mechanisms that can be simulated using dyadic decomposition only [67]. Indeed, one of the *open questions* is whether a single deep network can handle the diversity of mechanism design tasks; for example, synthesizing different types of linkages or meeting various additional constraints. Conditional deep generative models are a promising route toward that goal, as they can, in principle, learn a unified mapping and produce a multimodal output distribution.

10.3 Convolutional and Feedforward Networks in the Loop. Aside from generative models, researchers have used standard neural network architectures as components of the design pipeline. *Convolutional neural networks (CNNs)* excel at processing image-based representations of coupler curves. For instance, the encoder of the VAE discussed above is a CNN with multiple convolution + pooling layers that compress the 64–64 coupler image into a latent code. CNNs capture local geometric features of the path (curvatures, loops, etc.) that are relevant for determining a suitable mechanism. On the other end, once a specific mechanism type is chosen, a fully connected neural network can be used to regress the continuous design parameters. In the VAE-FCNN approach, a dedicated FCNN was trained for each representation (Fourier, wavelet, point-cloud, image) to map that representation (or its latent features) to the mechanism's link lengths and ground pivot locations. This regression network essentially learns the inverse kinematics mapping in a statistical sense, generalizing across the variety of shapes in the dataset.

Another use of basic neural networks is seen in the work of Deshpande and Purwar [59] for *defect-free four-bar synthesis*. They addressed the classic problem of motion generation without defects by learning an autoencoder embedding of entire motion sequences. Instead of focusing on a coupler path alone, this work considered a prescribed sequence of poses (positions and orientations of the moving link). They computed a motion signature for known defect-free mechanisms and used an autoencoder to compress these signatures into a low-dimensional representation. By clustering thousands of known good four-bar motions in this latent space, they created a "library" of motion clusters. Given a new motion task, one can quickly locate the nearest cluster (i.e., a set of similar motions that are known to be achievable without defects) and select a candidate solution from that cluster as an initial design. This approach of *database clustering + local optimization* proved highly effective: it finds a pool of viable solutions very quickly even in the highly nonlinear solution space of four-bar motion generation. The key was the autoencoder's ability to capture the essential shape of a motion in an invariant form (invariant to rotations, translations, etc., of the entire motion). The distance in latent space provides a meaningful measure of similarity between motions, something hard to achieve with raw angle or coordinate data. This example underscores how even relatively simple neural network models (autoencoders and MLPs) can contribute to mechanism design by acting as advanced feature extractors and search guides.

10.4 Emerging Machine Learning Paradigms in Mechanism Synthesis. While early work in this area relied heavily on CNNs and VAEs or their variants, the landscape has recently broadened considerably. A new development has been the use of reinforcement learning (RL) combined with GNNs to directly search the discrete space of linkage topologies. In this view, candidate mechanisms are represented as evolving graphs, and policies are trained to expand or prune links and joints toward satisfying a target task. The GCP-HOLO [91] framework exemplifies this approach by coupling RL with a GNN-based encoder, showing that policy-gradient methods such as PPO can outperform DQN baselines while scaling to higher-order linkages and achieving order-of-magnitude speedups over classical optimization. Earlier studies had already suggested that formulating synthesis as a sequential decision process was feasible [92], but the incorporation of graph structure learning marks an important advance. Nobari et al. [93] introduced a framework that integrates contrastive learning with hierarchical optimization for path synthesis of planar linkage mechanisms. They also introduced the LINK-ABC benchmark for synthesizing linkages that trace alphabet-shaped trajectories. Similarly, Pan et al. [94] combined graph-based topology encoding with gradient-free optimization to simultaneously determine the optimal topology and trajectory of planar linkages to achieve a desired end-effector path.

Transformers have also entered mechanism design. The MechaFormer model [95] represents design tasks in a domain-specific language and leverages an encoder–decoder Transformer to jointly predict both the topology and continuous parameters of a mechanism. This sequence-learning perspective enables flexible handling of variable-length pose constraints and supports diverse sampling strategies for candidate exploration. Relatedly, GearFormer [96] applies Transformer models to the configuration design of gear trains, embedding physical constraints through a simulator and demonstrating hybrid search pipelines that combine Monte Carlo tree search with autoregressive sequence prediction. These approaches underscore how attention mechanisms can capture long-range dependencies in design tasks that involve both structural and geometric choices.

Diffusion models are also beginning to shape the field. A recently introduced dataset of nearly nine thousand 2D and 3D mechanism sketches has enabled fine-tuning of stable diffusion for design generation and BLIP-2 for captioning [97]. While the 3D results are promising, the study also highlights systematic failure cases such as hallucinated joints or inconsistent paths, pointing to the need for larger and more balanced datasets. This work demonstrates how generative priors from large models can be adapted to specialized engineering domains, but also how data curation and bias remain central challenges. A conditional GAN has been used to synthesize four-bar linkages that satisfy both kinematic accuracy and dynamic performance metrics, such as torque amplification ratios, demonstrating the feasibility of embedding physics criteria directly into generative models [98]. Taken collectively, these developments highlight that mechanism synthesis is no longer confined to CNNs and VAEs/C-VAEs/GANs: RL, GNNs, diffusion models, and transformers are actively reshaping the state-of-the-art.

11 Results and Applications

11.1 Machine Learning-Driven Type and Dimensional Synthesis for Planar Linkages. Deep learning has been successfully applied to both *path generation* and *motion generation* problems for planar linkages. In path generation, perhaps the most salient result is the ability to synthesize both the type and dimensions of a mechanism purely from the desired curve shape. Conventional approaches usually fix the type (say, a four-bar) and solve for dimensions; by contrast, an image-based generative model trained on mixed data of four-, six-, and eight-bars can implicitly perform type synthesis. For example, given a particular coupler curve, the VAE + FCNN model of Nurizada et al. [62] can retrieve candidate mechanisms of different types (4R, 6-bar Watt or Stephenson, etc.) that produce that curve or a similar one. Figure 4 of their article shows a gallery of 4-bar, 6-bar, and 8-bar

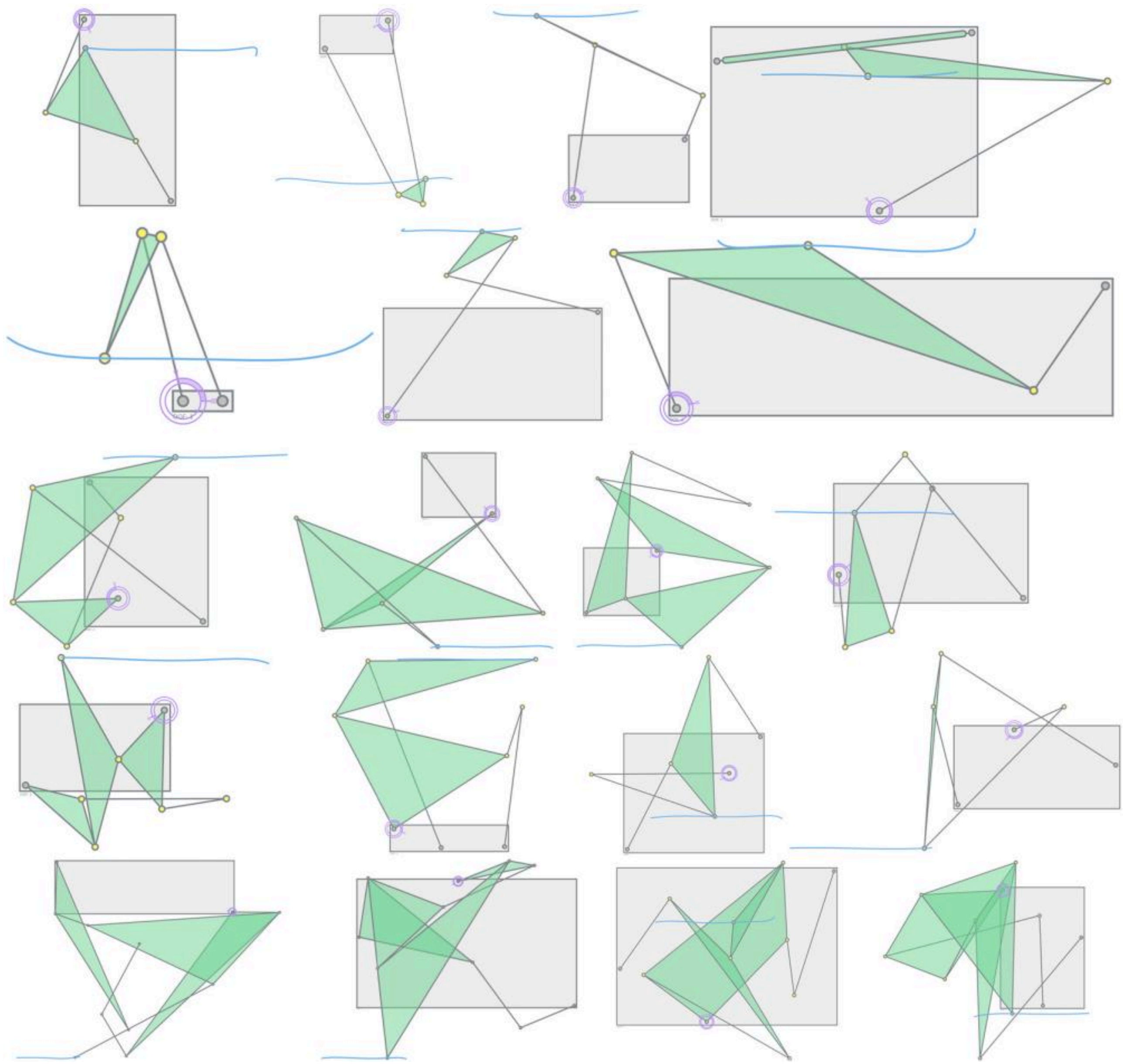

Fig. 3 Various four-, six-, and eight-bar mechanisms generating an approximate straight line

mechanisms from the dataset, each with a distinct coupler curve shape. By searching this rich dataset via learned latent features, the algorithm is essentially exploring type options: for some paths a simple four-bar might suffice, for others the nearest match might be a six-bar or eight-bar mechanism; see Fig. 3 for an example. This addresses the long-standing challenge of mechanism type synthesis in an elegant data-driven way.

For *motion generation (rigid-body guidance)*, data-driven methods have shown dramatic improvements in avoiding defects. The defect-free four-bar synthesis by Deshpande and Purwar [59] is a prime example: by comparing continuous motion histories instead of discrete points, their neural network-based objective function could guide the optimizer toward mechanisms that preserve motion continuity (hence no premature circuit breaks). They reported discovering a *wide pool of defect-free solutions* for benchmark motion problems, far beyond what a precision-point method would return. Moreover, the solutions were found quickly, thanks to the autoencoder reducing the search space and providing good starting candidates. In path generation, the image-based cVAE by Deshpande and Purwar [61] could generate dozens of mechanism designs for a single input curve, many of which were novel or nonintuitive to a human designer. This probabilistic one-to-many mapping means the designer can impose secondary preferences (e.g., prefer simpler mechanisms) after generation, rather than hard-coding such preferences upfront.

11.2 Evaluation Metrics. A notable outcome of the *3M mechanism dataset study* [62] was the introduction of *standardized evaluation metrics for learning-based synthesis*. Six key metrics were proposed: accuracy, novelty, diversity, robustness, computational efficiency, and scalability, to benchmark how well a learning model performs in generating mechanisms for a given task. Accuracy measures how closely the mechanisms output path matches the target curve (e.g., using mean squared error of point distances). Novelty assesses how different the generated designs are from the training examples, promoting truly creative solutions rather than just retrieved ones. Diversity gauges the variety among multiple solutions for the same problem (for instance, does the model output five very similar four-bars or a mix of four- and six-bars?). Robustness reflects performance under imperfect input; e.g., how tolerant is the model to noisy or partially missing path data. Computational efficiency tracks time or resources required, and scalability examines how the approach handles growing complexity (like more points in the path or more links in the mechanism). Using these metrics, the authors compared different VAE architectures on the same dataset and found, for example, that increasing the latent dimension or the capacity of the network improved accuracy up to a point but with diminishing returns on scalability. These metrics provide a framework for quantitatively comparing future data-driven synthesis methods—an important step toward establishing them as reliable design tools. In their results, the VAE-based method achieved high accuracy and diversity, indicating it could closely reproduce target paths and also suggest multiple distinct mechanisms per task. The approach was also robust to input sparsity: it could take as input either a dense image or just a few sketched points (converted to an image) and still infer a plausible full path and mechanism, thanks to the VAE's learned distribution of coupler curves.

11.3 Machine Learning-Driven Dimensional Synthesis of Spatial Mechanisms. Deep learning in spatial mechanism design is nascent but very promising. Sharma and Purwar [70] tackled the *Alt-Burmester problem*—a generalization of Burmester's five-position theory to spatial mechanisms with mixed pose and path constraints. Specifically, they considered a single-DOF 5-SS platform (a spatial mechanism with five spherical–spherical joints) that must guide a moving platform through a set of target poses, where some poses have unspecified orientation (path-points). Analytical synthesis for this problem is extraordinarily difficult, and iterative solutions often get stuck or return mechanisms with branching issues. By introducing a VAE into the pipeline, they reformulated the problem into a pure motion synthesis problem. The VAE was trained on numerous motion trajectories of 5-SS mechanisms, learning the relationship between translational and rotational movement components. When given an Alt-Burmester specification (positions with missing orientations), the encoder could infer a plausible complete motion (i.e., assign likely orientations to those path-points) by sampling from the learned distribution. In essence, the VAE *imputed the missing orientation data* in a statistically informed way, such that the resulting full pose sequence was one that a 5-SS mechanism could execute. This was then passed to a classical spatial motion solver. The net result was a collection of *defect-free spatial mechanism designs* that satisfied the mixed pose/point constraints. The approach demonstrates a general principle: use learning to handle the parts of the problem that are under-constrained or highly nonlinear (here, guessing orientations), and then use traditional synthesis on the well-defined part. It also points to scalability: the authors noted that while they focused on the 5-SS platform, the approach "can be scaled to any single-DOF spatial mechanisms," provided one can generate a dataset of those mechanisms to train the model. Deng et al. [65] recently proposed synthesis of one-degree-of-freedom spatial revolute-spherical-cylindrical-revolute (RSCR) mechanisms using a conditional-β-VAE architecture. In this study, they also built a publicly available database of over 5 million paths and their corresponding RSCR mechanisms. This hints at a future where comprehensive databases and learned models exist for not only planar linkages but also spherical, spatial, and parallel mechanisms, greatly extending the reach of computational design.

12 Summarizing Advantages of Machine Learning Methods in Mechanism Synthesis

Machine learning augments classical kinematic synthesis by turning noisy, partial, or under-specified tasks into tractable design problems and by widening the space of viable mechanisms the designer can explore. Learning models like VAEs learn compact, invariant representations of motion and path data, which makes them resilient to sketchy or under-specified inputs while preserving the essential characteristic of the desired behavior. In practice, this means a hand-drawn curve, a sparse set of poses, or a motion fragment can be interpreted as a family of clean, physically plausible specifications rather than a single brittle target. From there, learning-based pipelines naturally support one-to-many mappings: for a given task, they surface diverse candidate mechanisms. These solutions often span different topologies and dimensions, so the designer can balance accuracy, size, manufacturability, or transmission characteristics without restarting the synthesis from scratch.

Crucially, these advantages complement unified algebraic methods rather than replace them. Learned encoders and generative models act upstream to denoise, complete, and parameterize the task, while downstream, geometric fitting and kinematic solvers enforce feasibility and extract concrete link dimensions with guarantees. This division of labor delivers both robustness and speed: once trained, models produce near-instant feedback suitable for interactive workflows, while the kinematics layer filters out branch or circuit defects and honors exact constraints when required. The net result is a more forgiving and creative design loop, one that simultaneously reduces sensitivity to input imperfections, accelerates exploration across linkage types, and maintains the rigor needed to deliver realizable mechanisms.

13 Open Challenges and Opportunities

The convergence of deep learning and mechanism design, while full of potential, is still in its early stages. In this section, we identify several *open research questions*.

13.1 Datasets, Coverage, and Bias. A fundamental question is how can we systematically generate and curate large databases of mechanisms for training? The 3M planar linkage dataset [62] and 100M MIT dataset [90] is a start, but we need similar data for other mechanism classes including spherical and spatial linkages and multi-degree-of-freedom manipulators. Simulation-based data generation must be validated to ensure the data covers only feasible designs (e.g., filtering out mechanisms with unrealistic link length ratios or unreachable positions). Another question is related to distribution of mechanisms and motions used to train/test learning models (e.g., percentages of four-/six-/eight-/ten-bar variants with revolute and prismatic joints) and failure cases (rare coupler shapes, near-singular motions).

Emerging datasets enable pretraining and fine-tuning but remain limited in size and balance. The recent ∼9k-image mechanisms dataset [95] shows that fine-tuned diffusion models can capture 3D sketch regularities but struggle with 2D drawings and exhibit hallucinations, underscoring coverage gaps and potential biases from source distributions and curation. Data efforts should follow *FAIR principles* (findable, accessible, interoperable, reusable) to encourage community-wide use and consistency in evaluations. We also encourage reporting accuracy, novelty, diversity, robustness, computation time, and scalability [62] for generated results.

13.2 Representation Learning for Mechanism Topology. Mechanism structures are graph-like and not easily encoded as fixed-length vectors. Developing graph neural networks or other structure-aware models to represent mechanism topology is an open frontier. This could enable a network to suggest not just dimensional parameters but also novel topologies (imagine a network proposing a six-bar over a four-bar because the learned graph embedding indicates the six-bar can better realize a given motion). Initial steps in this direction might involve training networks on the matrix representations or adjacency lists of mechanisms and exploring latent spaces of mechanism graphs.

13.3 End-to-End Synthesis Architecture. What is the best neural network architecture to directly map from a design specification (path, motion, and other specifications) to mechanism parameters? Thus far, cVAEs and some cGANs have been tried on relatively constrained cases. The question remains if a single deep network can be made to handle multiple types of tasks (path versus motion generation) or constraints (like avoiding certain regions for ground pivots). A promising idea is *multitask learning*, where a network is trained simultaneously on several synthesis tasks, potentially sharing latent representations between them. Additionally, hybrid architectures that combine convolution (for interpreting geometry) with graph neural layers (for encoding candidate mechanism designs) might be needed for an end-to-end system. The role of *transformers* [99] (which have revolutionized sequence-to-sequence mappings in other domains) is also worth exploring for mechanism design sequences. Bolanos et al. [95] proposed MechaFormer, a transformer based architecture for path synthesis of mechanisms.

13.4 Generality and Scalability. Most current ML models for mechanism design are bespoke (trained for a specific DOF or class of mechanism). A grand challenge is to scale these models up to more complex machines. For example, can we extend planar linkage synthesis models to *spherical or spatial linkages*? The latter have more degrees-of-freedom and more complex motion spaces, which might require larger networks or fundamentally new representations. Preliminary work with spatial 5-SS mechanisms [70] is encouraging, but scaling to higher DOF or parallel mechanisms will likely require innovations in network design and training strategy (perhaps incorporating kinematic and quasi-static knowledge into the network via *physics-informed neural networks* or symmetry constraints). Scalability also concerns the size of the problem specification: e.g., if a path has 100 points instead of 5 or vice-versa, can the network handle it? Techniques like sequence models or attention mechanisms might be needed to allow variable-length inputs.

13.5 Integration With Human Design Process. Ultimately, data-driven mechanism design tools should *augment human creativity*, not replace it. Thus, research should focus on human-in-the-loop systems where the algorithms can explain or visualize their suggestions in ways designers can trust. Providing explanations for why a certain mechanism was suggested (perhaps by pointing to similar prior mechanisms in the database) could build confidence. Moreover, interfaces that allow designers to tweak the generated solutions (adjusting a link length and seeing the effect on the coupler path instantly through the surrogate model) will make these tools more practical. This brings up the need for real-time or near-instantaneous evaluation, which is where the speed of neural networks is a major asset. Once trained, these models can produce results in milliseconds, enabling interactive design sessions.

13.6 Benchmarking and Validation. As the field progresses, it will be important to establish benchmark problems and compare learning-based approaches against traditional ones and against each other. The six metrics proposed by Nurizada et al. [62] are a good starting point. Future benchmarks might include: (a) a set of challenging path and motion synthesis tasks (with known optimal or feasible solutions) to test algorithms' accuracy and diversity, (b) stress-test scenarios with very noisy inputs to probe robustness, and (c) cross-validation where a model is trained on one class of tasks and tested on another to evaluate generality. *Physical validation* of synthesized mechanisms is also crucial—ultimately the true test is building a mechanism and seeing it perform the intended motion. Integrating design-for-manufacturing considerations (e.g., limiting link lengths to practical ranges, ensuring joints can be realized) into the learning process is an open direction that intersects with mechanical engineering practice.

13.7 Large Language Model-Driven Mechanism Co-Pilot. Another important opportunity is using large language models as a mechanism co-pilot in design tools like MotionGen. The co-pilot would turn text, sketches, and pose data into clean, machine-checkable tasks. It would then call learned motion encoders and unified geometric solvers, and output editable, parametric feature trees with short, clear rationales. For example, when the designer prompts, "design a compact slider-crank to lift 60 mm, avoid a 40 mm keep-out, transmission angle greater than 50 degrees, Grashof preferred," or sketches a rough path in MotionGen, the co-pilot captures intent and compiles it into a normalized constraint set: target poses/paths, keep-out regions, joint preferences, bounds, and soft versus hard priorities. Key research problems include multimodal specification capture with uncertainty, policies that choose among fitting, retrieval, and generative models under latency and compute limits, tight CAD integration for bidirectional conversational edits, and evaluation that blends user metrics (time-to-first-feasible, acceptance rate) with technical scores (accuracy, robustness, diversity). Solving these issues can turn large language models into dependable co-pilots that make mechanism synthesis conversational, fast, and grounded in kinematics.

In conclusion, deep learning and data-driven approaches are reshaping the landscape of mechanism design. They offer fresh ways to address age-old challenges like type synthesis, approximate motion generation, and creative conceptual design. By learning from data—whether generated via simulation or collected from the troves of mechanisms devised over centuries—these methods can surmount the limitations of purely analytical techniques, handling imprecise inputs and vast design spaces with aplomb. The research so far has demonstrated *feasibility and advantages*: models that capture the essence of coupler curve families, algorithms that output myriad defect-free linkages in seconds, and systems that guide designers rather than requiring designers to guide them. As we move forward, the synergy of deep learning with classical kinematics and human expertise will likely yield powerful hybrid workflows. The ultimate vision is a computational design assistant that can instantly suggest mechanism ideas for any arbitrary motion problem, backed by the knowledge of millions of past designs and the creative interpolation between them. The work reviewed in this section brings us several steps closer to that vision, laying the groundwork for a new era of mechanism synthesis that is data-rich, intelligence-driven, and innovative.

14 Five-Year Research Roadmap (2025–2030)

Over the next five years, we envision that our community will make a coordinated push that deepens the union of unified geometry and learning while making the resulting tools dependable and easy to use in real design settings. In the foregoing, we are presenting a multithreaded roadmap, where our use of the pronoun *we* represents not just the authors, but our research fraternity as a whole.

The first thread is data and representation. We will expand today's planar mechanism datasets into balanced, FAIR datasets for spherical and spatial linkages, with rigorous filters for feasibility and diversity, and with consistent annotations of topology, constraints, and defect states. In parallel, we will move beyond fixed-form parameter vectors toward structure-aware encodings: graph representations of mechanisms paired with motion embeddings that are invariant to pose, scale, and parameterization so that models can reason jointly about what a mechanism is (its topology) and how it moves (its kinematics).

Building on that foundation, the second thread is end-to-end, physics-grounded learning. Rather than choosing between algebraic fitting and deep models, we will develop hybrid pipelines in which learned components impute missing task information and denoise sketches, while geometric solvers enforce feasibility, exact constraints, and defect avoidance. The goal is a single, modular architecture that accepts variable-length, mixed specifications (poses, paths, velocities, and fabrication limits) and returns families of realizable designs across linkage types. Such architectures will provide certificates of feasibility, sensitivity, and branch/circuit safety. For spatial mechanisms, we will pursue dual-quaternion and Study-quadric–aware networks that respect group structure, enabling the same guarantees long available in planar mechanism synthesis.

A third thread is generality, scalability, and speed. We will scale models to higher mobility systems and richer tasks without sacrificing runtime by combining attention mechanisms for long sequences with graph neural layers for topology, plus fast surrogate evaluators that approximate expensive kinematic checks. The practical target is interactive performance: immediate feedback for editing constraints, exploring alternatives, and steering solutions, so designers can scrub through a spectrum of mechanisms while the system maintains feasibility in the background.

The fourth thread centers on human-in-the-loop design and interpretability. Interfaces will surface why a design was proposed (nearest exemplars, sensitivity maps, constraint attributions) and how adjustments will propagate (predictive previews and counterfactuals). Preference modeling, learned from designer choices, will guide search toward mechanisms that balance accuracy with manufacturability, size, and robustness. Throughout, we will treat defect detection not as a post hoc filter but as a learned prior and a hard constraint integrated into training and solving.

Finally, we will establish benchmarks, metrics, and validation that the community can trust: standardized task suites spanning planar, spherical, and spatial cases; metrics that jointly score accuracy, diversity, novelty, robustness to noise, computational cost, and scalability; and open reference implementations that reproduce results. Wherever possible, we will close the loop with hardware, thus rapidly prototyping representative designs to test tracking accuracy, transmission quality, and tolerance sensitivity. Taken together, these directions aim to turn mechanism synthesis into an interactive, data-rich discipline where theory, learning, and practice reinforce one another.

15 Conclusions

Kinematic synthesis of mechanisms has historically been a dichotomous process: type synthesis followed by dimensional synthesis, each typically addressed through separate analytical frameworks. However, recent developments in algebraic fitting techniques have unified these traditionally distinct tasks into a single, coherent computational pipeline. By leveraging tools from projective geometry, linear algebra, and kinematic mapping, these methods enable simultaneous optimization of linkage topology and dimensions, even under overconstrained or approximate task specifications. The use of singular value decomposition (SVD) and constraint manifold formulations has revealed the continuous nature of the design space, in which various mechanism types (e.g., four-bar, six-bar, revolute-prismatic) coexist and can be algorithmically explored.

Yet, as powerful and elegant as these geometric methods are, they still face limitations when dealing with uncertain, noisy, or high-dimensional design inputs, or when tasked with generating diverse and novel mechanism solutions. Here, machine learning has emerged as a transformative complement. It enables mechanism synthesis to go beyond deterministic equation solving into the realm of data-driven design exploration, where the process is informed not only by task constraints but also by patterns learned from large databases of successful mechanisms.

In particular, generative models such as variational autoencoders and conditional VAEs have demonstrated the ability to synthesize mechanisms directly from high-level design intent, such as an image of a desired coupler path or a hand-drawn sketch. These models learn compact, invariant representations of motion data, enabling them to interpolate across a latent space of known mechanisms and propose multiple plausible solutions, even for under-defined tasks. This probabilistic, one-to-many mapping transforms synthesis from a point-solution problem into a spectrum-generation framework, enabling computational creativity.

Importantly, the integration of ML and algebraic synthesis is not just theoretical. Systems like MotionGen implement these ideas in user-friendly interfaces, allowing engineers to input motion requirements and interactively explore the resulting mechanisms. Likewise, ML models have been trained on millions of planar and spatial mechanisms (e.g., 4R, 6R, 5-SS, RSCR) to not only predict linkage dimensions but also assist in type synthesis, a task previously considered intuitive or intractable. This has allowed synthesis for complex spatial linkages and hybrid joint configurations that would be arduous to explore analytically.

Looking ahead, the field faces several open challenges: the need for more systematic and FAIR-compliant mechanism datasets across different linkage types; development of graph-based representations to generalize across topologies; and end-to-end synthesis architectures that merge perception (e.g., of motion sketches) with structural reasoning. The integration of human-in-the-loop design tools, where ML assists but does not replace human creativity, will be essential in making these approaches practical and trustworthy. Additionally, real-time evaluation, manufacturability constraints, and hybrid workflows that combine optimization with learning are areas of active exploration.

In summary, mechanism design is undergoing a paradigm shift. Where once it was largely analytical and experience-driven, it is now becoming computational, data-enriched, and intelligence-augmented. The fusion of algebraic synthesis with machine learning has not only unified type and dimensional synthesis but also expanded the solution space to include novel, robust, and designer-guided alternatives. As research progresses, we can expect these methods to become even more generalized, more robust, and more tightly integrated into CAD and simulation tools, thereby accelerating the invention and deployment of novel mechanisms in engineering systems. The journey that began with Burmester's five positions has led us to a future where a few lines of algebra and code can design a variety of mechanisms instantaneously.

Acknowledgment

The review presented here draws upon several key articles in the field. The authors of those works (A. Purwar, Q. J. Ge, P. Zhao, S. Deshpande, Z. Lyu, A. Nurizada, A. Halwah, X. Deng, X. Li, S. Sharma, R. Dhaipule, N. Chakraborty, and W. Liao) are gratefully acknowledged for their foundational contributions. Their publications in ASME journals (*Journal of Mechanical Design*, *Journal of Mechanisms and Robotics*, *Journal of Computing and Information Science in Engineering*), Advances in Mechanisms, Robotics, and Design Education and Research, Mechanisms and Machine Science, Frontiers of Mechanical Engineering, IFToMM World Congress, USCToMM Symposium on Mechanical Systems and Robotics, and International Design Engineering and Technical Conferences (IDETC/CIE) form the core of the content synthesized in this article.

Conflict of Interest

There are no conflicts of interest.

Data Availability Statement

No data, models, or code were generated or used for this article.

References

[1] Hunt, K., 1978, *Kinematic Geometry of Mechanisms*, Clarendon Press, Oxford.
[2] Bottema, O., and Roth, B., 1979, *Theoretical Kinematics*, Dover Publication Inc., New York.

[3] Erdman, A. G., and Sandor, G. N., 1991, *Mechanism Design: Analysis and Synthesis*, Vol. 1, 2nd ed., Prentice-Hall, Englewood Cliffs, NJ.
[4] McCarthy, J. M., 1990, *Introduction to Theoretical Kinematics*, The MIT Press, Cambridge, MA.
[5] McCarthy, J. M., and Soh, G. S., 2010, *Geometric Design of Linkages*, Springer, New York, NY.
[6] Tsai, L., 2001, *Mechanism Design: Enumeration of Kinematic Structures According to Function*, CRC Press LLC, Boca Raton, FL.
[7] Moon, F., 2009, "History of the Dynamics of Machines and Mechanisms From Leonardo to Timoshenko," International Symposium on History of Machines and Mechanisms, H. S. Yan and M. Ceccarelli, eds.
[8] Libraries, C. U., 2002, "KMODDL—Kinematic Models for Digital Design," http://kmoddl.library.cornell.edu/index.php
[9] Norton, R., 2011, *Design of Machinery: An Introduction to the Synthesis and Analysis of Mechanisms and Machines*, 5th ed. ed., SUNY Stony Brook, New York.
[10] Denavit, J., and Hartenberg, R. S., 1955, "A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices," ASME J. Appl. Mech., **22**(2), pp. 215–221.
[11] Mruthyunjaya, T., 2003, "Kinematic Structure of Mechanisms Revisited," Mech. Mach. Theory, **38**(4), pp. 279–320.
[12] Kong, X., and Gosselin, C. M., 2007, *Type Synthesis of Parallel Mechanisms*, 7th ed., Springer Publishing Company, Incorporated, Heidelberg, Germany.
[13] Rico, J. M., and Ravani, B., 2007, "On Calculating the Degrees of Freedom or Mobility of Overconstrained Linkages: Single-Loop Exceptional Linkages," ASME J. Mech. Des., **129**(3), pp. 301–311.
[14] Wampler, C., Larson, B. T., and Erdman, A. G., 2007, "A New Mobility Formula for Spatial Mechanisms," ASME Conf. Proc., **2007**, pp. 561–570.
[15] Jin, Q., and Yang, T.-L., 2002, "Structure Synthesis of Parallel Manipulators With 3-Dimension Translation and 1-Dimension Rotation," ASME Conf. Proc., **2002**, pp. 907–915.
[16] Jin, Q., and Yang, T.-L., 2004, "Theory for Topology Synthesis of Parallel Manipulators and Its Application to Three-Dimension-Translation Parallel Manipulators," ASME J. Mech. Des., **126**(4), pp. 625–639.
[17] Yang, J., Gao, F., Ge, Q. J., Zhao, X., Guo, W., and Jin, Z., 2011, "Type Synthesis of Parallel Mechanisms Having the First Class G_F Sets and One-Dimensional Rotation," Robotica, **29**(6), pp. 895–902.
[18] Gao, F., Yang, J., and Ge, Q. J., 2011, "Type Synthesis of Parallel Mechanisms Having the Second Class G_F Sets and Two Dimensional Rotations," ASME J. Mech. Rob., **3**(1), p. 011003.
[19] Angeles, J., and Bai, S., 2022, *Kinematics of Mechanical Systems: Fundamentals, Analysis and Synthesis*, Springer International Publishing, Cham, Switzerland.
[20] McCarthy, J. M., 2011, "Kinematics, Polynomials, and Computers—A Brief History," ASME J. Mech. Rob., **3**(1), p. 010201.
[21] Erdman, A., and Lehman, M., 2024, "A Review of Kinematic Theories and Practices Compiled for Biomechanics Students and Researchers," ASME J. Biomech. Eng., **146**(5), p. 050801.
[22] Erdman, A. G., 1993, *Modern Kinematics: Developments in the Last Forty Years*, John Wiley and Sons, New York.
[23] Lee, W.-T., and Russell, K., 2018, "Developments in Quantitative Dimensional Synthesis (1970-Present): Four-Bar Path and Function Generation," Inverse Problems Sci. Eng., **26**(9), pp. 1280–1304.
[24] Lee, W.-T., and Russell, K., 2018, "Developments in Quantitative Dimensional Synthesis (1970-Present): Four-Bar Motion Generation," Inverse Problems Sci. Eng., **26**(1), pp. 133–148.
[25] Yang, W., Ding, H., and Kecskemethy, A., 2022, "Structural Synthesis Towards Intelligent Design of Plane Mechanisms: Current Status and Future Research Trend," Mech. Mach. Theory, **171**, p. 104715.
[26] Söylemez, E., 2023, *Kinematic Synthesis of Mechanisms Using Excel® and Geogebra*, 1 ed., Springer Cham.
[27] Han, X., Zhao, P., Zhao, X., and Zi, B., 2025, "Review on Machine Learning-Based Approaches for the Kinematic Analysis and Synthesis of Mechanisms," Front. Mech. Eng., **20**, p. 11.
[28] Perez, A., and McCarthy, J., 2004, "Dual Quaternion Synthesis of Constrained Robotic Systems," ASME J. Mech. Des., **126**(3), pp. 425–435.
[29] Lee, E., and Mavroidis, C., 2006, "An Elimination Procedure for Solving the Geometric Design Problem of Spatial 3R Manipulators," ASME J. Mech. Des., **128**(1), pp. 142–145.
[30] Su, H. J., McCarthy, J. M., and Watson, L. T., 2004, "Generalized Linear Product Homotopy Algorithms and the Computation of Reachable Surfaces," ASME J. Comput. Inf. Sci. Eng., **4**(3), pp. 226–234.
[31] Su, H.-J., 2004, "Computer-Aided Constrained Robot Design Using Mechanism Synthesis Theory," Ph.D. dissertation, University of California, Irvine, CA.
[32] Raghavan, M., and Roth, B., 1995, "Solving Polynomial Systems for the Kinematic Analysis and Synthesis of Mechanisms and Robot Manipulators," ASME J. Mech. Des., **117**(1), pp. 71–79.
[33] Wampler, C. W., Morgan, A. P., and Sommese, A. J., 1990, "Numerical Continuation Methods for Solving Polynomial Systems Arising in Kinematics," ASME J. Mech. Des., **112**(1), pp. 59–68.
[34] Krovi, V., Ananthasuresh, G., and Kumar, V., 2001, "Kinematic Synthesis of Spatial R-R Dyads for Path Following With Applications to Coupled Serial Chain Mechanisms," ASME J. Mech. Des., **123**(3), pp. 359–366.
[35] Su, H. J., Wampler, C. W., and McCarthy, J., 2004, "Geometric Design of Cylindric PRS Serial Chains," ASME J. Mech. Des., **126**(1), pp. 269–277.
[36] Su, H.-J., and McCarthy, J., 2005, "The Synthesis of an RPS Serial Chain to Reach a Given Set of Task Positions," Mech. Mach. Theory, **40**(7), pp. 757–775.
[37] Lee, E., and Mavroidis, D., 2002, "Solving the Geometric Design Problem of Spatial 3R Robot Manipulators Using Polynomial Homotopy Continuation," ASME J. Mech. Des., **124**(4), pp. 652–661.
[38] Eberhard, P., Gaugele, T., and Sedlaczek, K., 2005, "Topology Optimized Synthesis of Planar Kinematic Rigid Body Mechanisms," *Advanced Design of Mechanical Systems: From Analysis to Optimization*. J. A. C.Ambrosio and P. Eberhard, eds., CISM International Centre for Mechanical Sciences, vol. 511, Springer, Vienna.
[39] Frecker, M. I., Ananthasuresh, G. K., Nishiwaki, S., Kikuchi, N., and Kota, S., 1997, "Topological Synthesis of Compliant Mechanisms Using Multi-criteria Optimization," ASME J. Mech. Des., **119**(2), pp. 238–245.
[40] Hayes, M., and Zsombor-Murrary, P., 2004, "Towards Integrated Type and Dimensional Synthesis of Mechanisms for Rigid Body Guidance," Proceedings of the 2004 CSME Forum, London, ON, Canada, June 1–4.
[41] Faltings, B., 1990, "Qualitative Kinematics in Mechanisms," Artif. Intell., **44**(1–2), pp. 89–119.
[42] Ge, Q. J., Zhao, P., and Purwar, A., 2013, *Volume 14 of Advances in Mechanisms, Robotics, and Design Education and Research, Mechanisms and Machine Science*, Springer, Heidelberg, pp. 145–157.
[43] Ge, Q. J., Zhao, P., Purwar, A., and Li, X., 2012, "A Novel Approach to Algebraic Fitting of a Pencil of Quadrics for Planar 4R Motion Synthesis," ASME J. Comput. Inf. Sci. Eng., **12**(4), p. 041003.
[44] Zhao, P., Li, X., Purwar, A., Thakkar, K., and Ge, Q. J., 2013, "A Task Driven Approach to the Synthesis of Spherical 4R Linkages Using Algebraic Fitting," ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, DETC2013-13179.
[45] Li, X., Zhao, P., Purwar, A., and Ge, Q. J., 2013, "A Task Driven Approach to Simultaneous Type Synthesis and Dimensional Optimization of Planar Parallel Manipulator Using Algebraic Fitting of a Family of Quadrics," ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, DETC2013-13197.
[46] Zhao, P., Purwar, A., and Ge, Q. J., 2013, "Constraint Decomposition Via Kinematic Mapping: A New and Efficient Approach to the Solution of the Classical Burmester Problem," ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, DETC2013-12977.
[47] Ge, Q., Purwar, A., Zhao, P., and Deshpande, S., 2017, "A Task-Driven Approach to Unified Synthesis of Planar Four-Bar Linkages Using Algebraic Fitting of a Pencil of G-Manifolds," ASME J. Comput. Inf. Sci. Eng., **17**(3), p. 031011.
[48] Purwar, A., Deshpande, S., and Ge, Q. J., 2017, "MotionGen: Interactive Design and Editing of Planar Four-Bar Motions Via a Unified Framework for Generating Pose- and Geometric-Constraints," ASME J. Mech. Rob., **9**(2), p. 024504.
[49] Ge, Q. J., Ge, X., Purwar, A., and Li, X., 2015, "A Null-Space Analysis Method for Solving Bilinear Equations in Kinematic Synthesis of Planar and Spherical Dyads," Proceedings of the 14th IFToMM World Congress, Taipei, Taiwan, Oct. 25–30.
[50] Sharma, S., Purwar, A., and Ge, Q. J., 2019, "An Optimal Parametrization Scheme for Path Generation Using Fourier Descriptors for Four-Bar Mechanism Synthesis," ASME J. Comput. Inf. Sci. Eng., **19**(1), p. 014501.
[51] Sharma, S., Purwar, A., and Ge, Q. J., 2019, "A Motion Synthesis Approach to Solving Alt-Burmester Problem by Exploiting Fourier Descriptor Relationship Between Path and Orientation Data," ASME J. Mech. Rob., **11**(1), p. 011016.
[52] Li, X., Zhao, P., Purwar, A., and Ge, Q., 2018, "A Unified Approach to Exact and Approximate Motion Synthesis of Spherical Four-Bar Linkages Via Kinematic Mapping," ASME J. Mech. Rob., **10**(1), p. 011003.
[53] Ge, X., Purwar, A., and Ge, Q. J., 2016, "From 5-SS Platform Linkage to Four-Revolute Jointed Planar, Spherical and Bennett Mechanisms," ASME 2016 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Charlotte, NC, Aug. 21–24.
[54] Ge, X., Purwar, A., and Ge, Q. J., 2017, "Finite Position Synthesis of 5-SS Platform Linkages Including Partially Specified Joint Locations," ASME 2017 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Cleveland, OH, Aug. 6–9.
[55] Zhao, P., Li, X., Purwar, A., and Ge, Q. J., 2016, "A Task-Driven Unified Synthesis of Planar Four-Bar and Six-Bar Linkages With R- and P-Joints for Five-Position Realization," ASME J. Mech. Rob., **8**(6), p. 061003.
[56] Li, X., Ge, X., Purwar, A., and Ge, Q. J., 2015, "A Unified Algorithm for Analysis and Simulation of Planar Four-Bar Motions Defined With R- and P-Joints," ASME J. Mech. Rob., **7**(1), p. 011014.
[57] Deshpande, S., and Purwar, A., 2022, "A Unified Approach to Dyad and Triad Synthesis for Planar Mechanisms for Motion Generation," *Proceedings of the 2022 USCToMM Symposium on Mechanical Systems and Robotics*, P. Larochelle and J. M.. McCarthy, eds., Springer International Publishing, Cham, pp. 243–261.
[58] Sharma, S., and Purwar, A., 2020, "Using a Point-Line-Plane Representation for Unified Simulation of Planar and Spherical Mechanisms," ASME J. Comput. Inf. Sci. Eng., **20**(6), pp. 1–13.
[59] Deshpande, S., and Purwar, A., 2019, "A Machine Learning Approach to Kinematic Synthesis of Defect-Free Planar Four-Bar Linkages," ASME J. Comput. Inf. Sci. Eng., **19**(2), p. 021004.
[60] Deshpande, S., and Purwar, A., 2019, "Computational Creativity Via Assisted Variational Synthesis of Mechanisms Using Deep Generative Models," ASME J. Mech. Des., **141**(12), p. 121402.
[61] Deshpande, S., and Purwar, A., 2020, "An Image-Based Approach to Variational Path Synthesis of Linkages," ASME J. Comput. Inf. Sci. Eng., **21**(2), p. 021005.

[62] Nurizada, A., Dhaipule, R., Lyu, Z., and Purwar, A., 2024, "A Dataset of 3M Single-DOF Planar 4-, 6-, and 8-Bar Linkage Mechanisms With Open and Closed Coupler Curves for Machine Learning-Driven Path Synthesis," ASME J. Mech. Des., **147**(4), p. 4067014.
[63] Lyu, Z., and Purwar, A., 2024, "Point-Based Models: A Unified Approach for Geometric Constraint Analysis of Planar N-Bar Mechanisms With Rotary, Sliding, and Rolling Joints," ASME J. Mech. Rob., **17**(4), p. 044514.
[64] Lyu, Z., and Purwar, A., 2025, "Construction, Reduction, and Decomposition of Planar Mechanisms Via an Extended Pebble Game Framework," ASME J. Mech. Rob., **17**(12), pp. 1–31.
[65] Deng, X., Nurizada, A., and Purwar, A., 2025, "Path Synthesis of Spatial Revolute-Spherical-Cylindrical-Revolute Mechanisms Using Deep Learning," Front. Mech. Eng., **20**(2), p. 9.
[66] Purwar, A., and Chakraborty, N., 2023, "Deep Learning-Driven Design of Robot Mechanisms," ASME J. Comput. Inf. Sci. Eng., **23**(6), p. 060811.
[67] Lyu, Z., Purwar, A., and Liao, W., 2023, "A Unified Real-Time Motion Generation Algorithm for Approximate Position Analysis of Planar N-Bar Mechanisms," ASME J. Mech. Des., **146**(6), p. 063302.
[68] Nurizada, A., and Purwar, A., 2023, "Transforming Hand-Drawn Sketches of Linkage Mechanisms Into Their Digital Representation," ASME J. Comput. Inf. Sci. Eng., **24**(1), p. 011010.
[69] Nurizada, A., and Purwar, A., 2023, "An Invariant Representation of Coupler Curves Using a Variational AutoEncoder: Application to Path Synthesis of Four-Bar Mechanisms," ASME J. Comput. Inf. Sci. Eng., **24**(1), p. 011008.
[70] Sharma, S., and Purwar, A., 2022, "A Machine Learning Approach to Solve the Alt-Burmester Problem for Synthesis of Defect-Free Spatial Mechanisms.," ASME J. Comput. Inf. Sci. Eng., **22**(2), p. 021003.
[71] Nurizada, A., Lyu, Z., and Purwar, A., 2024, "Path Generative Model Based on Conditional–Variational Auto Encoder for Four-Bar Mechanism Design," ASME J. Mech. Rob., **17**(6), p. 061004.
[72] Ravani, B., and Roth, B., 1983, "Motion Synthesis Using Kinematic Mappings," J. Mech. Trans. Autom. Des. Trans. ASME, **105**(3), pp. 460–467.
[73] Deng, X., Purwar, A., and Ge, Q. J., 2023, "A Unified Design Equation to Represent Geometric Constraints of Spatial SS, ES and SE Dyads," ASME 2023 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Volume 8: 47th Mechanisms and Robotics Conference (MR), Boston, MA, Aug. 20–23, ASME, p. V008T08A025.
[74] Sharma, S., and Purwar, A., 2020, "Unified Motion Synthesis of Spatial Seven-Bar Platform Mechanisms and Planar-Four Bar Mechanisms," ASME 2020 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Volume 10: 44th Mechanisms and Robotics Conference (MR), Virtual, Online, Aug. 17–19, ASME, p. V010T10A033.
[75] Li, X., Ge, Q. J., and Gao, F., 2018, "A Computational Geometric Approach for Motion Generation of Spatial Linkages With Sphere and Plane Constraints," ASME J. Mech. Rob., **11**(1), p. 014504.
[76] Deng, X., and Purwar, A., 2024, "A Matrix-Based Approach to Unified Synthesis of Planar Four-Bar Mechanisms for Motion Generation With Position, Velocity, and Acceleration Constraints," ASME J. Comput. Inf. Sci. Eng., **24**(12), p. 121003.
[77] Deshpande, S., and Purwar, A., 2017, "A Task-Driven Approach to Optimal Synthesis of Planar Four-Bar Linkages for Extended Burmester Problem," ASME J. Mech. Rob., **9**(6), p. 061005.
[78] Yao, J., and Angeles, J., 2000, "Computation of All Optimum Dyads in the Approximate Synthesis of Planar Linkages for Rigid-Body Guidance," Mech. Mach. Theory, **35**(8), pp. 1065–1078.
[79] Chen, C., Bai, S., and Angeles, J., 2008, "A Comprehensive Solution of the Classical Burmester Problem," Trans. CSME, **32**(2), pp. 137–154.
[80] Chen, C., Bai, S., and Angeles, J., 2008, "The Synthesis of Dyads With One Prismatic Joint," ASME J. Mech. Des., **130**(3), p. 034501.
[81] Chen, C., and Angeles, J., 2008, "A Novel Family of Linkages for Advanced Motion Synthesis," Mech. Mach. Theory, **43**(7), pp. 882–890.
[82] Kaipa, K., Purwar, A., Ayala, O., and Bawab, S., 2025, "Transforming Engineering Education: Project-Based Learning and Technology Integration in a Senior-Level Mechanisms Course," ASEE 2025 Annual Conference & Exposition ASEE, Paper No. 49376.
[83] Purwar, A., 2023, "Machine-Learning Driven Robot-Motion Design: Introducing a Web-Based Mechanism Design Software," https://peer.asee.org/43510
[84] Schröcker, H.-P., Husty, M. L., and McCarthy, J. M., 2007, "Kinematic Mapping Based Assembly Mode Evaluation of Planar Four-Bar Mechanisms," ASME J. Mech. Des., **129**(9), pp. 924–929.
[85] Robson, N. P., McCarthy, J. M., and Tumer, I. Y., 2008, "The Algebraic Synthesis of a Spatial TS Chain for a Prescribed Acceleration Task," Mech. Mach. Theory, **43**(10), pp. 1268–1280.
[86] Li, X., Wu, J., and Ge, Q. J., 2016, "A Fourier Descriptor-Based Approach to Design Space Decomposition for Planar Motion Approximation," ASME J. Mech. Rob., **8**(6), p. 064501.
[87] Vasiliu, A., and Yannou, B., 2001, "Dimensional Synthesis of Planar Mechanisms Using Neural Networks: Application to Path Generator Linkages," Mech. Mach. Theory, **36**(2), pp. 299–310.
[88] Galan-Marin, G., Alonso, F. J., and Del Castillo, J. M., 2009, "Shape Optimization for Path Synthesis of Crank-Rocker Mechanisms Using a Wavelet-Based Neural Network," Mech. Mach. Theory, **44**(6), pp. 1132–1143.
[89] Kingma, D. P., and Welling, M., 2019, "An Introduction to Variational Autoencoders," Found. Trends® Mach. Learn., **12**(4), pp. 307–392.
[90] Nobari, A. H., Srivastava, A., Gutfreund, D., and Ahmed, F., 2022, "LINKS: A Dataset of a Hundred Million Planar Linkage Mechanisms for Data-Driven Kinematic Design," Proceedings of the ASME 2022 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. Volume 3A:48th Design Automation Conference (DAC, St. Louis, MO, Aug. 14–17, ASME, p V03AT03A013.
[91] Fogelson, M. B., Tucker, C., and Cagan, J., 2023, "GCP-HOLO: Generating High-Order Linkage Graphs for Path Synthesis," ASME J. Mech. Des., **145**(7), p. 073303.
[92] Vermeer, K., Kuppens, R., and Herder, J., 2018, "Kinematic Synthesis Using Reinforcement Learning," Proceedings of the ASME 2018 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. Volume 2A: 44th Design Automation Conference. Quebec City, Quebec, Canada, Aug. 26–29, ASME, p. V02AT03A009.
[93] Nobari, A. H., Chen, W., and Ahmed, F., 2022, "Range-Constrained Generative Adversarial Network: Design Synthesis Under Constraints Using Conditional Generative Adversarial Networks," ASME J. Mech. Des., **144**(2), p. 021708.
[94] Pan, Z., Min, L., Gao, X., and Manocha, D., 2025, "Joint Search of Optimal Topology and Trajectory for Planar Linkages," Int. J. Rob. Res., **42**, pp. 176–195.
[95] Bolanos, D., Ataei, M., and Jayaraman, P., 2025, "MechaFormer: Sequence Learning for Kinematic Mechanism Design Automation," Arxiv.
[96] Etesam, Y., Cheong, H., Ataei, M., and Jayaraman, P., 2025, "Deep Generative Model for Mechanical System Configuration Design," Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence and Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence and Fifteenth Symposium on Educational Advances in Artificial Intelligence, Philadelphia, PA, February.
[97] Ghezelbash, F., Eskandari, A., and Bidhendi, A., 2024, "A Dataset for Mechanical Mechanisms," Arxiv.
[98] Lee, S., Kim, J., and Kang, N., 2024, "Deep Generative Model-Based Synthesis Framework of Four-Bar Linkage Mechanisms With Target Conditions," J. Comput. Des. Eng., **11**(5), pp. 318–332.
[99] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., et al., 2019, "Transformers: State-of-the-Art Natural Language Processing," Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Online, Association for Computational Linguistics, pp. 38–45.