

Enhancing the Computational Efficiency of Aircraft Assembly Simulations Using Deep Learning Warm-Start

Thi Thu Hoai Trinh

Peter the Great St. Petersburg Polytechnic University

Roman Krasnikov

krasnikov.ra@gmail.com

Peter the Great St. Petersburg Polytechnic University

Sergey Lupuleac

Peter the Great St. Petersburg Polytechnic University

Stanislav Baklanov

Peter the Great St. Petersburg Polytechnic University

Research Article

Keywords: Aerospace informatics, Aircraft smart manufacturing, Contact problems, Deep learning warm-start, Computational efficiency

Posted Date: October 14th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-11124652/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

Enhancing the Computational Efficiency of Aircraft Assembly Simulations Using Deep Learning Warm-Start

Thi Thu Hoai Trinh¹, Roman Krasnikov^{1*}, Sergey Lupuleac¹,
Stanislav Baklanov¹

¹Institute of Physics and Mechanics, Peter the Great St. Petersburg Polytechnic University, Polytechnicheskaya, 29, St. Petersburg, 195251, Russia.

*Corresponding author(s). E-mail(s): krasnikov.ra@gmail.com;
Contributing authors: thuhoai20021707@gmail.com; lupuleac@mail.ru;
baklanov.stas@gmail.com;

Abstract

In modeling the assembly processes of aircraft structures, multiple solutions of quadratic programming problems of a special type are required. Contact problems of stress-strain analysis of structures are reduced to such problems. These numerical models serve as core components of the aircraft assembly digital twin. Despite the use of specialized accelerated solvers, the solution time can be significant, especially for large-scale problems, since the runtime of any solver grows polynomially with increasing problem dimension. One of the acceleration techniques consists of employing initial guesses close to the problem solution, constructed using a neural network. The network is trained on data obtained by solving a large number of problems with synthetic input data. It is known that a finite multilayer perceptron with ReLU activation is capable of accurately reproducing the dependence of the solution of a quadratic programming problem on its input data. In this connection, multi-label classification machine learning models intended for detecting contact in potential contact node pairs are considered. The architecture of the neural network is selected experimentally, and its influence on prediction quality and solver speedup is discussed. The proposed deep learning warm-start method utilizes neural-network-generated initial guesses to accelerate the quadratic programming solver. Evaluated within the context of aerospace informatics, this approach has demonstrated high efficiency on aircraft assembly models. Recommendations are provided for selecting the model

architecture and its quantitative characteristics to achieve a balance between the initial guess construction speed and its accuracy.

Keywords: Aerospace informatics, Aircraft smart manufacturing, Contact problems, Deep learning warm-start, Computational efficiency.

1 Introduction

Modern aircraft manufacturing imposes high precision requirements on the assembly of large-scale structures: even minor deviations in part geometry caused by manufacturing tolerances lead to increased internal stresses and assembly forces. Therefore, digital simulation within the context of digital twins for assembly processes constitutes an important part of computer-aided design systems, enabling the assessment of structure assemblability and the optimization of the technological process already in the design stage.

The theoretical foundations of contact interaction modeling are presented in the monograph by Wriggers [1]. For aircraft assembly problems, these approaches have been developed in the works of Petukhova et al. [2], Lupuleac et al. [3, 4], and Stefanova et al. [5, 6], where mathematical models and numerical methods are proposed that account for elastic deformations of parts, contact constraints, features of the riveting process, and methods to specify the initial gap between panels proposed by Zaitseva et al. [7]. The developed methods are implemented in the ASRP software package (Assembly Simulation of Riveting Process).

The key computational problem in assembly simulation is the contact problem, which, after an appropriate formulation, reduces to a convex quadratic programming (QP) problem with constraints. Its dimension is determined by the number of potential contact pairs and can reach several thousand variables, which leads to significant computational costs in large-scale numerical experiments.

A large number of methods have been developed for solving convex QP problems: classical active-set algorithms proposed by Goldfarb and Idnani [8], including the solver *qpOASES* developed by Ferreau et al. [9], oriented toward parametric problems, and its parallel modification for MPC (Model Predictive Control) [10]; as well as the method with guaranteed complexity proposed by Arnström and Axehill [11]; interior-point methods adapted to contact assembly problems proposed by Stefanova et al. [5]; operator splitting methods implemented by Stellato et al. [12] in the OSQP solver; modern first-order methods for large-scale problems [13–15]. With regard to aircraft assembly, the efficiency of various approaches has been investigated in [6, 16]. Despite the diversity of solvers, their runtime grows polynomially with increasing problem dimension, which makes additional acceleration of computations a relevant task.

An effective technique for such an acceleration is the use of an initial guess close to the problem solution. In recent years, learning-to-optimize methods have been actively employed to construct such guesses; a survey of these methods is provided by Chen et al. [17]. The theoretical justification for the applicability of neural networks to QP problems is provided by the following result. A finite multilayer perceptron with ReLU

(Rectified Linear Unit) activation is capable of accurately reproducing the dependence of the solution of a parametric QP problem on its input data [18, 19]. This result is also supported by unsupervised learning results for large-scale problems [20]. On this basis, a number of approaches have been proposed for predicting the active set and constructing an initial guess: for embedded MPC by Klau?o et al. [21] and for large-scale MPC by Chen et al. [22]; end-to-end learning methods that account for the influence of the initial guess on solver convergence speed by Sambharya et al. [23, 24]; graph neural networks that exploit the structure of constraint matrices by Schmidtobreck et al. [25].

Existing studies in the field of learning-to-optimize are mainly focused on control problems and general-form QP problems with specialized solvers. Contact problems in aircraft assembly possess a number of specific features: a large number of potential contact pairs, strong imbalance of active constraints, high conditioning, and dense Hessian matrices. For solving such problems, projection methods of both first and second order are often employed [16, 26]. These features complicate the direct application of existing approaches. To the best of the authors' knowledge, the construction of initial guesses for such problems has remained virtually unexplored.

The present paper proposes a neural network-based method for constructing an initial guess for contact problems in the aircraft assembly. The problem is formulated as multi-label classification: for each potential contact pair, the model predicts a binary indicator of whether contact is present in the solution. Due to the class imbalance characteristic of assembly models, appropriate quality metrics and training strategies are employed [27–29]. The approach is integrated into ASRP and is intended to accelerate the QP solver without modifying the optimization algorithm used.

The paper is organized as follows. Section 2 presents the mathematical formulation of the contact problem and discusses methods for its solution. Section 3 describes the proposed neural network method: data preparation, model architecture, and training procedure. Section 4 reports the results of computational experiments. The conclusion summarizes the main findings and outlines directions for future research.

2 Mathematical formulation of the contact problem

2.1 Contact problem in assembly

In modeling the assembly processes of aircraft structures, it is necessary to determine the stress-strain state of the parts after the installation of the fasteners. Prior to assembly, there exist initial technological gaps between the mating surfaces, caused by deviations in part geometry and errors in their relative positioning. After the fasteners are installed, the parts deform, as a result of which some of the potential contact pairs come into contact, while for the remaining pairs a residual gap is preserved.

After finite element discretization, the assembly contact problem is formulated as a convex quadratic programming problem with a special structure [2, 6]. The solution of this problem determines the stress-strain state of the structure and constitutes the most computationally expensive part of the variational assembly analysis.

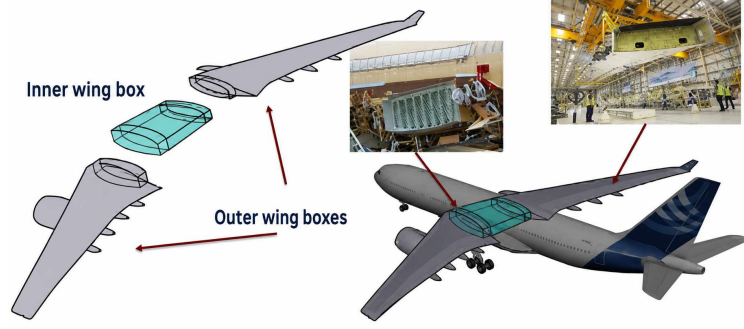


Fig. 1: Wing-fuselage assembly

2.2 Quadratic programming problem

The quadratic programming problem, to which the contact problem is reduced, can be written in the following form:

$$\min_{\mathbf{x}} \frac{1}{2} \mathbf{x}^T \mathbf{K} \mathbf{x} + \mathbf{f}^T \mathbf{x}, \quad \mathbf{A} \mathbf{x} \leq \mathbf{g}, \quad (1)$$

where $\mathbf{x}$ is the vector of unknown displacements, $\mathbf{K}$ is the positive definite stiffness matrix of the structure, $\mathbf{f}$ is the vector of forces, $\mathbf{A}$ is the matrix of contact constraints, and $\mathbf{g}$ is the vector of initial gaps.

Convex quadratic programming problems often arise in stress-strain analysis and are widely studied in the literature on numerical optimization [14, 30].

In this work, three interrelated formulations of the contact problem are considered: the primal, the dual, and the relative formulation [6]. For generating the training data, the relative formulation is used, in which the unknowns are the gaps between the parts.

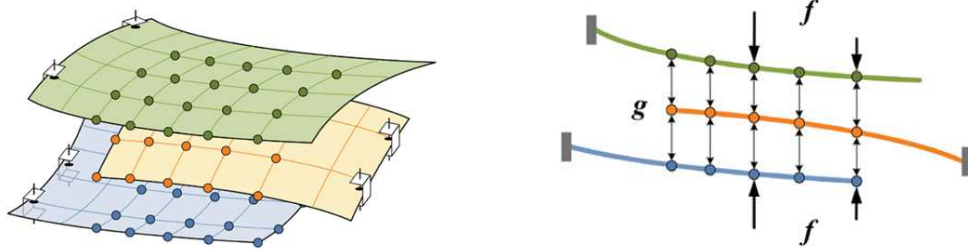


Fig. 2: Schematic view and cross-section of the assembly, adapted from [6]

2.3 Active set of constraints

The structure of the active set of constraints plays a key role in solving the quadratic programming problem:

$$\mathcal{I}_{\text{act}} = \{i \in \{1, \dots, m\} : g_i - u_i^* = 0\}, \quad (2)$$

where u_i^* denotes the optimal value of the relative displacement in the i -th contact pair.

The active set corresponds to the contact pairs for which the initial gap is completely eliminated after assembly. Conversely, if $g_i - u_i^* > 0$, the corresponding constraint remains inactive, and no contact occurs between the given pair of nodes.

In many algorithms for solving quadratic programming problems, the active set structure significantly affects the convergence rate and the total number of iterations [8, 10, 11]. If information about the expected active set is known in advance with a certain degree of confidence, a substantially better initial guess can be constructed compared to the standard solver initialization [26].

Information about the expected active set has been used to construct an initial guess in a number of previous works [21–23]. Before proceeding to the description of the neural network method proposed in the present work (Section 3), the solver for which the initial guess is constructed is briefly described.

2.4 Method for solving the quadratic programming problem

In the present paper, the first-order gradient projection solver [16] is accelerated using a neural network approximation for the initial guess. This solver is based on catalyst acceleration of coordinate descent, which is a variant of the A-HPE framework by [31] that uses a fully implicit composite optimization problem statement and a base first-order solver for handling inexact proximal operator calculation:

$$\mathbf{x}_{k+1} \approx_{\delta} \text{prox}_{h_k H}(\mathbf{y}_k) = \arg \min_{\mathbf{x} \in \mathbb{R}^n} \left[H(\mathbf{x}) + \frac{1}{2h_k} \|\mathbf{x} - \mathbf{y}_k\|^2 \right], \quad (3)$$

where $H : \mathbb{R}^n \rightarrow \mathbb{R}$ is the proper closed μ -strongly convex composite objective and h_k denotes the step size. It can be shown that such a calculation, namely, a proximal subproblem, has the same form as the original QP problem, but with a lower condition number of the Hessian. Although the A-HPE framework is too general for direct implementation, the envelope by d’Aspremont et al. [32, Algorithm 24] is used.

The coordinate descent is implemented to solve the proximal subproblems. In general, the use of a non-accelerated first-order solver for handling proximal inexactness results in an accelerated solver within the catalyst acceleration. This has been shown by d’Aspremont et al. [32] for deterministic global optimization algorithms and by Lin et al. [33] for both deterministic and stochastic algorithms. Although the analysis of coordinate descent provided by Nesterov [34] focuses on stochastic coordinate descent, the deterministic strategy of index selection based on the most violated optimality condition (namely, the largest absolute value of the objective subgradient) is used.

The performance of such a solver is highly dependent on a good initial guess. This is not only an intuitive proposition, but also a result of an a priori analysis of estimates for the functional gap by Nesterov [34], which uses the value $R(\mathbf{x}_0)$ as a measure of the initial guess sublevel set. Although the set of $\mathbf{x}$ such that $F(\mathbf{x}) \leq F(\mathbf{x}_0)$, where F is the quadratic objective of the QP problem, is considered nontrivial in the original Nesterov’s analysis, the strong convexity of the QP problem objective implies that the optimal set contains only the problem solution $\mathbf{x}_*$. Therefore, the value $R(\mathbf{x}_0)$ can be considered a measure of the quality of the initial guess. Nesterov’s a priori estimate for coordinate descent, together with the functional gap estimates by Lin et al. [33] for catalyst acceleration, forms a tight a priori functional gap estimate that tends to zero whenever $\mathbf{x}_0$ is close to $\mathbf{x}_*$. This observation provides not only intuitive but also theoretical motivation for generating an initial guess $\mathbf{x}_0$ as precise as possible for the described accelerated first-order solver.

This method, along with others [5, 26], is implemented in the ASRP software package, which is designed to model the assembly process of aircraft structures. The same package also incorporates the neural-network-based procedure for generating an initial guess, which is the subject of the present work.

3 Neural network-based method for constructing the initial guess

The proposed method belongs to the class of learning-to-optimize approaches. Instead of directly recovering the solution of the quadratic programming problem, the neural network is used to predict the structure of the active set of constraints, which determines the initial guess for the numerical solver. This separation of functions preserves the mathematical correctness of the solution, since the final result is always computed by the solver rather than by the model. At the same time, it simplifies the learning task: instead of recovering a high-dimensional continuous solution vector, the network solves a multi-label binary classification problem.

The general scheme of the approach consists of two stages (Fig. 3). In the training stage, a training dataset is generated from the results of solving a series of contact problems in ASRP, and a multilayer perceptron is trained on this dataset. The output of this stage is the weights of the trained network. In the inference stage, the trained network predicts an initial guess for a new contact problem based on its input data. This initial guess is then passed to the QP solver to obtain the final solution.

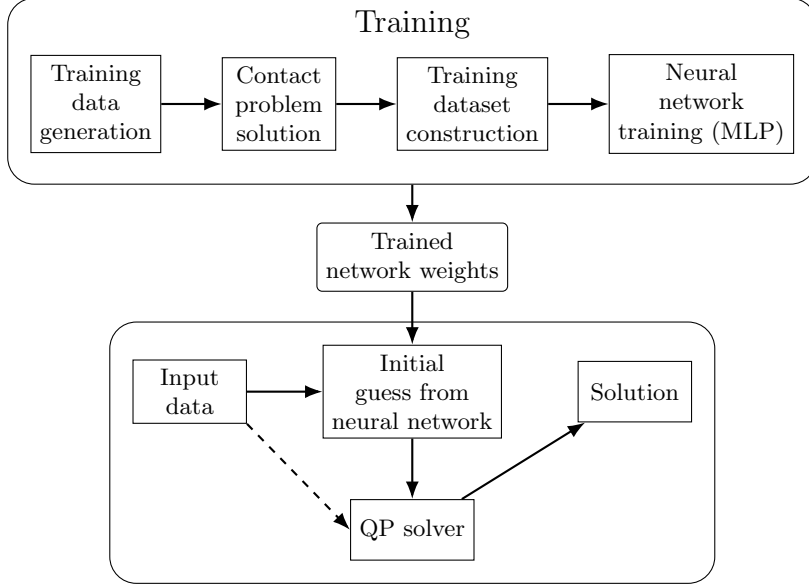


Fig. 3: General scheme of the proposed approach

3.1 Training data

The training dataset is generated fully automatically in the ASRP software package by repeatedly solving the contact problem for randomly generated fields of initial gaps. For generating such fields, the methodology of Zaitseva et al. [7] is used. In this methodology, the initial gap is described as a realization of a Gaussian random process using a parametric model. This approach makes it possible to obtain statistically representative gap realizations even with a limited amount of experimental measurements from actual assembly. The resulting gap fields reflect possible geometric deviations of the mating parts and are used as input data for solving the contact problem.

For each initial gap field, a random fastener configuration is automatically generated. The contact problem is then solved, and the results are used to construct a training example. The number of possible fastener configurations is combinatorially large even for medium-size models; therefore, the dataset is constructed with random statistical sampling rather than exhaustive enumeration. In the present work, 10 000 examples are generated for the model with $m = 121$, and 30 000 examples for the model with $m = 621$.

For each potential contact pair, two features are used: the contact force f_i and the initial gap g_i . Thus, the input vector has the form

$$\mathbf{x} = (f_1, \dots, f_m, g_1, \dots, g_m)^T, \quad (4)$$

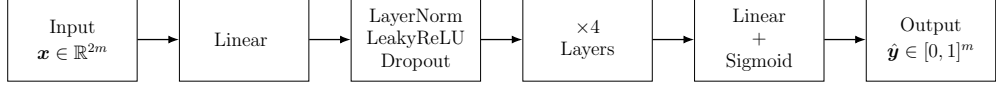


Fig. 5: Architecture of the multilayer perceptron for predicting the active set of contact constraints

where m is the number of potential contact pairs. The target variable is a binary label vector $\mathbf{y} = (y_1, \dots, y_m)^T$ determined by whether the constraint is active or not (2):

$$y_i = \begin{cases} 1, & g_i - u_i = 0, \\ 0, & g_i - u_i > 0. \end{cases} \quad (5)$$

The features are normalized by the mean and standard deviation computed on the training set. The same normalization parameters are applied at the inference stage. The dataset is divided into training, validation, and test parts in a 80:10:10 ratio, with the original class ratio preserved (stratified split).

The label distribution is substantially imbalanced and depends on the model characteristics. On average, the proportion of contact pairs is 16.4% for the model with $m = 121$ and 33.3% for the model with $m = 621$ (Fig. 4). To compensate for this imbalance, the parameter w_{pos} is used, which increases the contribution of the positive class to the loss function [27, 28].

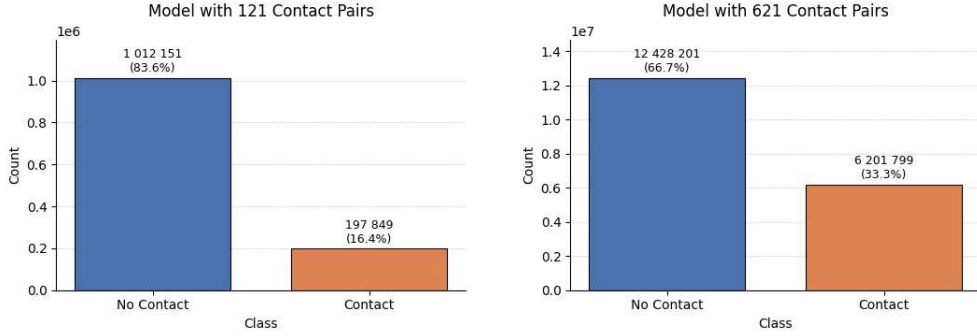


Fig. 4: Distribution of contact labels for models with $m = 121$ and $m = 621$

3.2 Model architecture and training

A multilayer perceptron is used. The input and output layers have fixed dimensions of $2m$ and m , respectively. There are four fully connected hidden layers with LayerNorm, LeakyReLU, and Dropout (Fig. 5). The width of the hidden layers was selected experimentally as 512 for $m = 121$ and 1024 for $m = 621$.

Training is performed by minimizing the weighted binary cross-entropy

$$\mathcal{L} = -\frac{1}{m} \sum_{i=1}^m [w_{\text{pos}} y_i \log p_i + (1 - y_i) \log(1 - p_i)], \quad (6)$$

where $p_i = \sigma(o_i)$, and $w_{\text{pos}} = n_{\text{neg}}/n_{\text{pos}}$ compensates for the class imbalance described in Section 3.1. In the inference stage, the probability is binarized with a threshold τ selected on the validation set using the Macro-F1 metric [28]:

$$\hat{y}_i = \begin{cases} 1, & p_i > \tau, \\ 0, & p_i \leq \tau. \end{cases} \quad (7)$$

Optimization was performed using the Adam algorithm [35] with early stopping based on the Macro-F1 metric on the validation set. The specific hyperparameter values for both models are presented in Section 4.1 (Table 1) together with the results of computational experiments.

3.3 Construction of the initial guess

The predicted binary vector $\hat{\mathbf{y}}$ is not used as the state variable values, but rather as the initial active set $\mathcal{I}_{\text{act}}^{(0)}$. An index i is included in $\mathcal{I}_{\text{act}}^{(0)}$ if the label $\hat{y}_i = 1$ for primal and relative problems and $\hat{y}_i = 0$ for dual problem. The initial values of the residual and Lagrange multipliers are set to zero. Such an initialization provides the solver with a starting point that is substantially closer to the desired active set compared to the standard initialization, which reduces the number of required iterations. The subsequent computational process is performed entirely by the first-order solver described in Section 2.4, without modifying its computational scheme.

The next section provides an experimental evaluation of the quality of the active set prediction and the efficiency of the proposed initial guess construction method.

4 Computational experiment

4.1 Experimental setup

The goal of the experiment is to evaluate the quality of the active set prediction and the resulting speedup in solving contact problems using the proposed initial guess. All computations were performed in the ASRP software package. The machine learning model was implemented in PyTorch, and interaction with the C++ solver was performed via the built-in Python interpreter.

The study was conducted on two aircraft assembly models with $m = 121$ and $m = 621$ potential contact pairs. For each model, a training dataset was generated according to the methodology described in Section 3 (10,000 and 30,000 examples, respectively). The quality of the model was evaluated on an independent test set, and the solution time was compared for 200 independent test problems using the standard solver initialization and the proposed initialization. In all cases, identical input data

was used; the only difference was in the method of constructing the initial guess. The main experimental parameters are presented in Table 1.

Table 1: Main parameters of the computational experiment and model training for $m = 121$ and $m = 621$

Parameter	$m = 121$	$m = 621$
Input dimension n	242	1242
Output dimension m	121	621
Training set size	10 000	30 000
Hidden layer width	512	1024
Number of hidden layers	4	4
Dropout	0.1	0.15
Optimizer	Adam	
Learning rate	$5 \cdot 10^{-4}$	$2 \cdot 10^{-4}$
Batch size	128	
Maximum number of epochs	300	
Early stopping (patience)	20	
Loss function	BCEWithLogitsLoss	
Model selection criterion	Macro-F1	
Binarization threshold τ	0.7	0.6

4.2 Prediction quality and solution speedup

The quality of the active set prediction on the test set is presented in Table 2. For the model with $m = 121$, a Macro-F1 score of 0.91 was achieved, and for the more complex model with $m = 621$, a score of 0.80 was obtained. The decrease in the metric with increasing dimension is expected and is related to the growth in the number of potential contact pairs, the more complex structure of the active set, and the more pronounced class imbalance. As the results show, perfectly accurate recovery of the active set is not required to construct an effective initial guess. This observation is consistent with similar findings reported in [21, 23].

The solution time of the contact problem was compared with the standard solver initialization and with the initial guess generated by the neural network for two formulations: dual and relative. The summarized results are presented in Table 3 and Table 4. The use of the neural network initial guess reduces the average solution time by 24–41%, depending on the problem dimension and the formulation. The highest speedup is achieved for the model with $m = 121$, where the active set prediction quality is the highest. For the model with $m = 621$, the speedup is somewhat lower, which is consistent with the decrease in classification accuracy; nevertheless, it remains substantial.

The distributions of the solving time (Fig. 6, Fig. 7) show that the use of the neural network systematically shifts the solution time to lower values and noticeably reduces the right tail of the distribution. This means that the probability of unfavorable

Table 2: Quality of active set prediction on the test set

Metric	$m = 121$	$m = 621$
Threshold τ	0.70	0.60
Macro-F1	0.9103	0.8001
Micro-F1	0.9505	0.8196
Balanced Accuracy	0.9111	0.8057
Precision (contact)	0.8483	0.7127
Recall (contact)	0.8522	0.7644
F1 (contact)	0.8503	0.7376

Table 3: Summary of solution time characteristics for the model with 121 contact pairs

Mode	Total, s	Mean, s	Median, s	Range, s	Reduction, %
Dual, no NN	19.997	0.100	0.104	0.002–0.368	37%
Dual, with NN	12.560	0.063	0.064	0.007–0.158	
Relative, no NN	140.255	0.701	0.654	0.271–1.548	41%
Relative, with NN	82.700	0.414	0.400	0.004–0.914	

Table 4: Summary of solution time characteristics for the model with 621 contact pairs

Mode	Total, s	Mean, s	Median, s	Range, s	Reduction, %
Dual, no NN	938.980	4.695	4.065	0.596–14.350	28%
Dual, with NN	675.661	3.378	3.303	0.606–11.250	
Relative, no NN	6619.937	33.100	30.997	11.377–107.392	24%
Relative, with NN	5023.725	25.119	23.339	7.155–68.921	

scenarios with a large number of solver iterations is reduced, and the predictability of the numerical solver performance is improved.

In addition, the inference time through the neural network was measured on the same 200 test problems. For the model with $m = 121$, the average time to construct the initial guess was 2.83 ms, which constitutes approximately 2.8% of the average solution time in the dual formulation. For the model with $m = 621$, the corresponding value was 5.81 ms, or approximately 0.12% of the dual solution time. Thus, the computational cost of the neural network prediction is negligible. At the same time, the use of the neural network initial guess leads to a reduction in the number of solver iterations, thereby reducing the total time of the computational procedure.

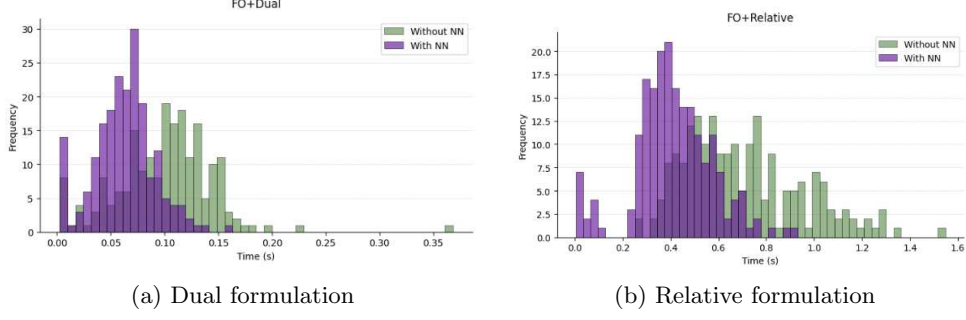


Fig. 6: Solution time distribution for the model with 121 contact pairs

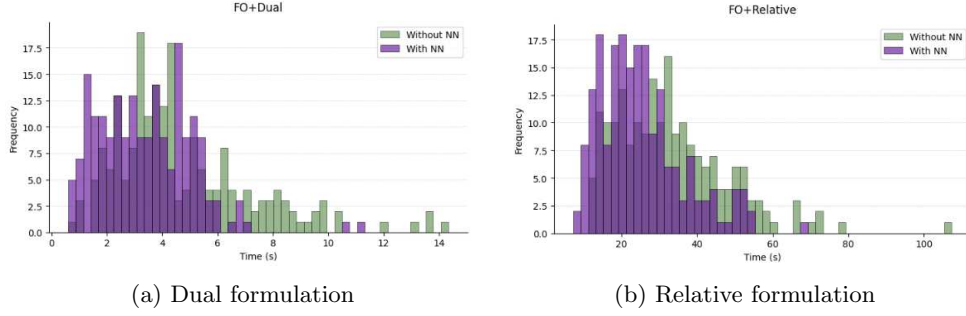


Fig. 7: Solution time distribution for the model with 621 contact pairs

4.3 Discussion of results

The obtained results confirm that the proposed method for constructing the initial guess provides a consistent reduction in the solution time of contact problems without modifying the mathematical formulation of the problem or the numerical algorithm used. This finding is consistent with the general conclusions of the works [21–23] on the application of machine learning to accelerate numerical optimization methods.

A key observation is the relationship between classification quality and the resulting speedup. The highest Macro-F1 score (0.91 for $m = 121$) corresponds to the highest speedup, while a lower score (0.80 for $m = 621$) still provides a substantial gain. This confirms the initial idea of the method: for an effective initial guess, an approximate recovery of the active set structure is sufficient, and exact recovery is not required.

5 Conclusion

A deep learning warm-start method based on active set prediction has been proposed to solve contact quadratic programming problems that arise in the simulation of aircraft assembly processes. In contrast to the approach of Sambharya et al. [23, 24], which is based on end-to-end learning of the initial guess without explicit active set

prediction, and to existing active set prediction approaches for model predictive control problems [21, 22], the proposed method is adapted to the specific features of contact assembly problems in aircraft structures: a large number of potential contact pairs and the characteristic class imbalance.

A fully automated procedure to generate training data has been developed and integrated into the ASRP software package to enhance its computational efficiency. On this basis, training datasets have been constructed, and neural networks have been trained for two models of aircraft assembly, containing 121 and 621 potential contact pairs, respectively.

The computational experiments performed have demonstrated that the proposed method provides high quality active set structure prediction (Macro-F1 above 0.91 for the model with 121 contact pairs and approximately 0.80 for the model with 621 contact pairs) and a consistent reduction in contact problem solving time by 24–41%, depending on the problem dimension and formulation, without loss of accuracy in the final solution. This confirms that exact recovery of the full solution of the quadratic programming problem is not required for effective acceleration of the numerical method. It is sufficient to determine the structure of the active set of constraints with acceptable accuracy, after which the numerical solver rapidly refines the remaining components of the solution.

Based on these outcomes, a practical recommendation can be formulated for the choice of model architecture. For problems with a relatively small number of potential contact pairs, a network with narrower hidden layers is sufficient, providing high classification quality with minimal training and inference costs. For problems of larger dimension, increasing the width of the hidden layers partially compensates for the growing complexity of the active set structure but does not eliminate the natural decrease in prediction accuracy. The choice of architecture should therefore be determined by the trade-off between the initial guess construction time and the required classification accuracy.

The proposed approach does not depend on a specific algorithm for solving the quadratic programming problem, which makes it promising for a broad class of problems in computational mechanics and engineering optimization. Future work includes investigating the application of graph neural networks and transformer architectures for active set prediction, increasing the volume of automatically generated training data, and extending the approach to contact problems of larger dimension and to other formulations of quadratic programming problems.

Acknowledgements. The research was supported by Russian Science Foundation (project No. 22-19-00062-II, <https://rscf.ru/en/project/22-19-00062/>).

Statements and Declarations

Funding

The research was supported by Russian Science Foundation (project No. 22-19-00062-II, <https://rscf.ru/en/project/22-19-00062/>).

Competing interests

The authors have no competing interests to declare that are relevant to the content of this article.

Ethics approval and Consent to participate

Not applicable.

Data availability

Replication of the results of the numerical experiments is supported by the detailed description of the algorithm. The specific model data and numerical experiment results are available from the corresponding author upon reasonable request.

Author contribution

Thi Thu Hoai Trinh developed the deep learning warm-start method and tested it on specific assembly models. Roman Krasnikov provided implementation and detailed description on the Catalyst-accelerated coordinate descent – the base solver in the paper. Sergey Lupuleac described the problem statement, reviewed references on engineering challenges in aircraft assembly simulation, and designed the physical formulation for numerical experiments. Stanislav Baklanov reviewed references on warm-starting methods and validated numerical experiments results. All authors contributed to the writing and review of the manuscript.

References

- [1] Wriggers, P.: Computational Contact Mechanics, 2nd edn., p. 518. Springer, Berlin, Heidelberg (2006). <https://doi.org/10.1007/978-3-540-32609-0>
- [2] Petukhova, M.V., Lupuleac, S.V., Shinder, Y.K., Smirnov, A.B., Yakunin, S.A., Bretagnol, B.: Numerical approach for airframe assembly simulation. *Journal of Mathematics in Industry* **4**(1), 8 (2014) <https://doi.org/10.1186/2190-5983-4-8>
- [3] Lupuleac, S., Petukhova, M., Shinder, Y., Bretagnol, B.: Methodology for solving contact problem during riveting process. *SAE International Journal of Aerospace* **4**(2), 952–957 (2011) <https://doi.org/10.4271/2011-01-2582>
- [4] Lupuleac, S., Zaitseva, N., Stefanova, M., Berezin, S., Shinder, J., Petukhova, M., Bonhomme, E.: Simulation of the wing-to-fuselage assembly process. *ASME Journal of Manufacturing Science and Engineering* **141**(6), 061009 (2019) <https://doi.org/10.1115/1.4043365>
- [5] Stefanova, M., Yakunin, S., Petukhova, M., Lupuleac, S., Kokkolaras, M.: An interior-point method-based solver for simulation of aircraft parts riveting. *Engineering Optimization* **50**(5), 781–796 (2018) <https://doi.org/10.1080/0305215X.2017.1350262>

- [6] Stefanova, M., Minevich, O., Baklanov, S., Petukhova, M., Lupuleac, S., Grigor'ev, B., Kokkolaras, M.: Convex optimization techniques in compliant assembly simulation. *Optimization and Engineering* **21**(4), 1665–1690 (2020) <https://doi.org/10.1007/s11081-020-09493-z>
- [7] Zaitseva, N., Lupuleac, S., Khashba, V., Shinder, J., Bonhomme, E.: Approaches to initial gap modeling in final aircraft assembly simulation. In: *ASME International Mechanical Engineering Congress and Exposition*, vol. 84492, pp. 02–02059 (2020). <https://doi.org/10.1115/IMECE2020-24229> . American Society of Mechanical Engineers
- [8] Goldfarb, D., Idnani, A.: A numerically stable dual method for solving strictly convex quadratic programs. *Mathematical Programming* **27**(1), 1–33 (1983) <https://doi.org/10.1007/BF02591962>
- [9] Ferreau, H.J., Kirches, C., Potschka, A., Bock, H.G., Diehl, M.: qpOASES: A parametric active-set algorithm for quadratic programming. *Mathematical Programming Computation* **6**(4), 327–363 (2014) <https://doi.org/10.1007/s12532-014-0071-1>
- [10] Ferreau, H.J., Kozma, A., Diehl, M.: A parallel active-set strategy to solve sparse parametric quadratic programs arising in MPC. *IFAC Proceedings Volumes* **45**(17), 74–79 (2012) <https://doi.org/10.3182/20120823-5-NL-3013.00015>
- [11] Arnström, D., Axehill, D.: Semi-explicit linear MPC using a warm-started active-set QP algorithm with exact complexity guarantees. In: *2021 60th IEEE Conference on Decision and Control (CDC)*, pp. 2557–2562 (2021). <https://doi.org/10.1109/CDC45484.2021.9683133> . IEEE
- [12] Stellato, B., Banjac, G., Goulart, P., Bemporad, A., Boyd, S.: Osqp: An operator splitting solver for quadratic programs. *Mathematical Programming Computation* **12**(4), 637–672 (2020) <https://doi.org/10.1007/s12532-020-00179-2>
- [13] Nesterov, Y.E.: A method of solving a convex programming problem with convergence rate $O(1/k^2)$. *Soviet Mathematics Doklady* **27**(2), 372–376 (1983)
- [14] Beck, A.: *First-Order Methods in Optimization*. MOS-SIAM Series on Optimization, vol. 25. SIAM, ??? (2017). <https://doi.org/10.1137/1.9781611974997>
- [15] Lu, H., Yang, J.: A practical and optimal first-order method for large-scale convex quadratic programming. *Mathematical Programming* **215**(1), 771–808 (2026) <https://doi.org/10.1007/s10107-024-02108-6>
- [16] Krasnikov, R., Titova, M., Baklanov, S., Lupuleac, S.: Comparative analysis of projection-based methods in aircraft assembly modelling. In: *Accepted for OPTIMA 2025* (2025)

- [17] Chen, T., Chen, X., Chen, W., Heaton, H., Liu, J., Wang, Z., Yin, W.: Learning to optimize: A primer and a benchmark. *Journal of Machine Learning Research* **23**(189), 1–59 (2022)
- [18] Amos, B., Kolter, J.Z.: Optnet: Differentiable optimization as a layer in neural networks. In: *Proceedings of the 34th International Conference on Machine Learning (ICML)*. *Proceedings of Machine Learning Research*, vol. 70, pp. 136–145 (2017)
- [19] Magoon, C., Yang, F., Aigerman, N., Kovalsky, S.: Differentiation through black-box quadratic programming solvers. *Advances in Neural Information Processing Systems* **38**, 108486–108517 (2026)
- [20] Yang, L., Li, B., Ding, T., Wu, J., Wang, A., Wang, Y., Tang, J., Sun, R., Luo, X.: An efficient unsupervised framework for convex quadratic programs via deep unrolling. *arXiv preprint* (2024). [arXiv:2412.01051](https://arxiv.org/abs/2412.01051)
- [21] Klaučo, M., Kalúz, M., Kvasnica, M.: Machine learning-based warm starting of active set methods in embedded model predictive control. *Engineering Applications of Artificial Intelligence* **77**, 1–8 (2019) <https://doi.org/10.1016/j.engappai.2018.09.014>
- [22] Chen, S.W., Wang, T., Atanasov, N., Kumar, V., Morari, M.: Large scale model predictive control with neural networks and primal active sets. *Automatica* **135**, 109947 (2022) <https://doi.org/10.1016/j.automatica.2021.109947>
- [23] Sambharya, R., Hall, G., Amos, B., Stellato, B.: End-to-end learning to warm-start for real-time quadratic optimization. In: *Proceedings of the 5th Annual Conference on Learning for Dynamics and Control (L4DC)*. *Proceedings of Machine Learning Research*, vol. 211, pp. 220–234 (2023)
- [24] Sambharya, R., Hall, G., Amos, B., Stellato, B.: Learning to warm-start fixed-point optimization algorithms. *Journal of Machine Learning Research* **25**(166), 1–46 (2024)
- [25] Schmidtreich, E.J., Arnström, D., Häusner, P., Sjölund, J.: Warm-starting active-set solvers using graph neural networks. *arXiv preprint* (2025). [arXiv:2511.13174](https://arxiv.org/abs/2511.13174)
- [26] Baklanov, S., Stefanova, M., Lupuleac, S.: Newton projection method as applied to assembly simulation. *Optimization Methods and Software* **37**(2), 577–604 (2022)
- [27] He, H., Garcia, E.A.: Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering* **21**(9), 1263–1284 (2009) <https://doi.org/10.1109/TKDE.2008.239>

- [28] Saito, T., Rehmsmeier, M.: The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PLoS ONE* **10**(3), 0118432 (2015) <https://doi.org/10.1371/journal.pone.0118432>
- [29] Goodfellow, I., Bengio, Y., Courville, A.: *Deep Learning*. MIT Press, Cambridge, MA (2016)
- [30] Nocedal, J., Wright, S.J.: *Numerical Optimization*, 2nd edn. Springer, New York (2006). <https://doi.org/10.1007/978-0-387-40065-5>
- [31] Monteiro, R.D.C., Svaiter, B.F.: An accelerated hybrid proximal extragradient method for convex optimization and its implications to second-order methods. *SIAM J. Optim.* **23**(2), 1092–1125 (2013) <https://doi.org/10.1137/110833786>
- [32] d’Aspremont, A., Scieur, D., Taylor, A.: Acceleration methods. *Foundations and Trends in Optimization* **5**(1-2), 1–245 (2021) <https://doi.org/10.1561/24000000036>
- [33] Lin, H., Mairal, J., Harchaoui, Z.: Catalyst acceleration for first-order convex optimization: from theory to practice. *Journal of Machine Learning Research* **18** (2017) <https://doi.org/10.48550/arXiv.1712.05654>
- [34] Nesterov, Y.: Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM J. Optim.* **22**(2), 341–362 (2012) <https://doi.org/10.1137/100802001>
- [35] Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: *Proceedings of the 3rd International Conference on Learning Representations (ICLR)* (2015). <https://arxiv.org/abs/1412.6980>