# THE APPLICATION OF ACTIVE LEARNING METHODOLOGIES IN THE DESCRIPTION OF THE SALT EFFECT ON THE SOLUBILITY OF AMINO ACIDS

**Christopher Andrey Piske**

Dissertation presented to

**Escola Superior de Tecnologia e Gestão - Instituto Politécnico de Bragança**

To obtain master's degree in

**Chemical Engineering**

within the scope of the double diploma with

**Universidade Tecnológica Federal do Paraná – Campus Ponta Grossa**

Supervisors:

**João Dinis Oliveira Abranches**

**Simão Pedro de Almeida Pinho**

**Priscilla dos Santos Gaschi Leite**

**Bragança, Portugal**

**January, 2025**

# THE APPLICATION OF ACTIVE LEARNING METHODOLOGIES IN THE DESCRIPTION OF THE SALT EFFECT ON THE SOLUBILITY OF AMINO ACIDS

Dissertation presented to

**Escola Superior de Tecnologia e Gestão - Instituto Politécnico de Bragança**

To obtain master's degree in

**Chemical Engineering**

within the scope of the double diploma with

**Universidade Tecnológica Federal do Paraná – Campus Ponta Grossa**

**Christopher Andrey Piske**

**2025**

**“Technology is a powerful tool, but the human mind remains the source of all true innovation.”**
*(Satya Nadella, 2022).*

This work was developed within the scope of the project CIMO-Centro de Investigação de Montanha, UIDB/00690/2020 (DOI: 10.54499/UIDB/00690/2020), UIDP/00690/2020 (DOI: 10.54499/UIDP/00690/2020); and SusTEC, LA/P/0007/2020 (DOI: 10.54499/LA/P/0007/2020), and CICECO-Aveiro Institute of Materials, UIDB/50011/2020 (DOI 10.54499/UIDB/50011/2020), UIDP/50011/2020 (DOI 10.54499/UIDP/50011/2020) & LA/P/0006/2020 (DOI 10.54499/LA/P/0006/2020), all financed by national funds through the FCT/MCTES (PIDDAC). The financial support from IUPAC Project No. 2022-002-2-500 is highly acknowledged.




# Acknowledgments

First and foremost, I thank God for the health and strength granted to me throughout this journey, allowing me to face challenges and to complete another very important stage of my life.

I am grateful for the opportunity to study in a European country, experiencing a unique academic and cultural environment that has contributed significantly to my personal and professional growth.

To my family, I express my deepest gratitude for their love, support, encouragement, and understanding, especially during difficult times, when their words, phone calls, and gestures of affection were essential in helping me remain focused and committed.

To my friends, I am thankful for their companionship, their words of motivation, and for being present, making this journey lighter and filled with moments that I will never forget.

To my supervisors, I extend my sincere thanks for their patience, dedication, and availability, as well as for their valuable contributions, guidance, and clarifications throughout the entire development of this thesis, in addition to the many opportunities for personal and professional enrichment they have provided.

Finally, I would like to thank everyone who, directly or indirectly, contributed to the completion of this work and to the fulfillment of this dream.

# Abstract

In aqueous solutions containing electrolytes, ions influence both the solubility and the stability of biomolecules. However, inconsistencies across published data highlight the need for a critical review. To address this, a database was constructed on the solubility of glycine in electrolyte solutions spanning from 1996 to 2024, and the experimental data were critically evaluated. Gaussian Process (GP) models were implemented to analyze, predict, and validate solubility behavior. The GP model successfully captures salting-in and salting-out trends, along with specific ion effects reported in the literature. It also provides predictive uncertainty estimates that help identify potentially inconsistent data points or sets. This uncertainty-based analysis enables the reconciliation of conflicting datasets and helps prioritize new experimental measurements in regions where data are sparse or less reliable. By applying a data-filtering method that removes experimental points falling outside the uncertainty range of the model, the influence of inconsistent values is reduced. This results in a more robust model fit and improved prediction accuracy. Therefore, the GP establishes a quantitative foundation for consolidating the current knowledge on the solubility of glycine in saline solutions, identifying methodological inconsistencies in the literature.

**Keywords:** Electrolyte solutions, solubility, Gaussian Process, uncertainty, data reliability.

# Resumo

Em soluções aquosas contendo eletrólitos, os íons influenciam tanto a solubilidade quanto a estabilidade de biomoléculas. No entanto, inconsistências entre dados publicados evidenciam a necessidade de uma revisão crítica. Para abordar essa questão, foi construída uma base de dados sobre a solubilidade da glicina em soluções eletrolíticas abrangendo o período de 1996 a 2024, e os dados experimentais foram avaliados criticamente. Modelos de Processo Gaussiano (GP) foram implementados para analisar, prever e validar o comportamento da solubilidade. O modelo GP captura com sucesso as tendências de *salting-in* e *salting-out*, juntamente com os efeitos específicos dos íons relatados na literatura. Ele também fornece estimativas de incerteza preditiva que auxiliam na identificação de pontos ou conjuntos de dados potencialmente inconsistentes. Essa análise baseada em incerteza permite a reconciliação de conjuntos de dados conflitantes e ajuda a priorizar novas medições experimentais em regiões onde os dados são escassos ou menos confiáveis. Ao aplicar um método de filtragem de dados que remove pontos experimentais que se encontram fora da faixa de incerteza do modelo, a influência de valores inconsistentes é reduzida. Isso resulta em um ajuste de modelo mais robusto e em uma melhoria da precisão das previsões. Portanto, o GP estabelece uma base quantitativa para consolidar o conhecimento atual sobre a solubilidade da glicina em soluções salinas, identificando inconsistências metodológicas na literatura.

**Palavras-chave:** Soluções eletrolíticas, solubilidade, Processo Gaussiano, incerteza, confiabilidade dos dados.

# INDEX

# INDEX OF FIGURES

# INDEX OF TABLES

# INDEX OF ABBREVIATIONS AND ACRONYMS

| | |
|---|---|
| COSMO-RS | Conductor-like Screening Model for Real Solvents |
| DFT | Density Functional Theory |
| GP | Gaussian Processes |
| L-BFGS-B | Limited-memory Broyden-Fletcher-Goldfarb-Shanno |
| SP | Sigma Profile |
| TZVP | Triple Zeta Valence Polarization |

# 1. INTRODUCTION AND MOTIVATION

This introductory chapter provides the scientific framework for the study of the salt effect on the solubility of amino acids and peptides in aqueous systems, with particular emphasis on solubility phenomena and ion-specific effects. It gives fundamental biochemical concepts, practical relevance to biological and industrial processes, and an extensive critical review of existing experimental literature [1].

Aqueous solutions containing electrolytes constitute the natural environment for many biomolecules, such as proteins, nucleic acids, and enzymes [1,2]. These aqueous systems are fundamental for maintaining the three-dimensional structure of these molecules and their functionality [1]. Ions in solution can either stabilise or disrupt intermolecular interactions, thereby influencing molecular behavior [2-4]. As a result, electrolytes play a vital role in regulating the physicochemical properties of biomolecules and are involved in numerous biological processes [1,2].

Current contributions to the knowledge of the solubility of amino acids in the presence of salts are extremely important in several aspects. From the biological point of view, it directly influences the stability of these molecules in aqueous medium [2,5]. Such knowledge is also essential in many industrial applications, especially in the food and pharmaceutical sectors, where product formulation requires precise control of the interactions between biomolecules and ions [2,6].

Among the more than 300 amino acids occurring in nature, only 20 L-α-amino acids are commonly incorporated into proteins [1]. Although these amino acids share fundamental structural components, namely an amino group, a carboxyl group, and a variable side chain, they differ substantially in polarity, charge, and hydrophobicity, as illustrated in Figure 1. Based on their side-chain properties, amino acids can be classified as nonpolar aliphatic, polar uncharged, acidic, basic, or aromatic. These distinctions strongly influence molecular interactions in solution and, consequently, solubility behavior.

The amino acid selected for this dissertation is glycine, because it has the most studied in terms of salt effect on its solubility, with the intention of extending this analysis to alanine, isoleucine, and valine.

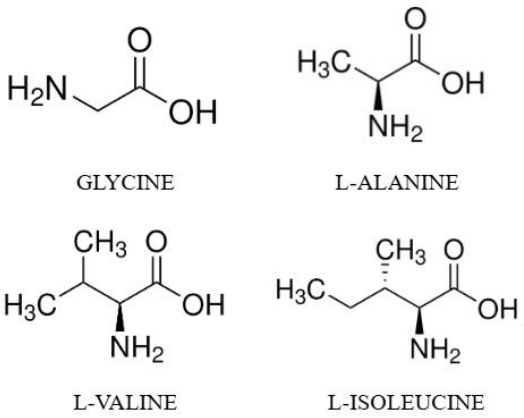


**Fig. 1** Chemical structures of the amino acids studied in this work.

Biomolecules naturally exist in aqueous electrolyte environments, making the study of salt effects on solubility particularly relevant. Salts can induce either salting-in or salting-out effects, depending on ion type, concentration, and molecular structure of the solute [1,2,6]. These effects are closely linked to changes in activity coefficients and are often interpreted using the Hofmeister series (Figure 2), which ranks ions according to their ability to stabilize or destabilize biomolecules in solutions.

The ions selected in this work are those associated with the salts studied, and are monovalent and divalent cations such as sodium, potassium, ammonium, magnesium and anions including chloride, nitrate, thiocyanate and sulfate.

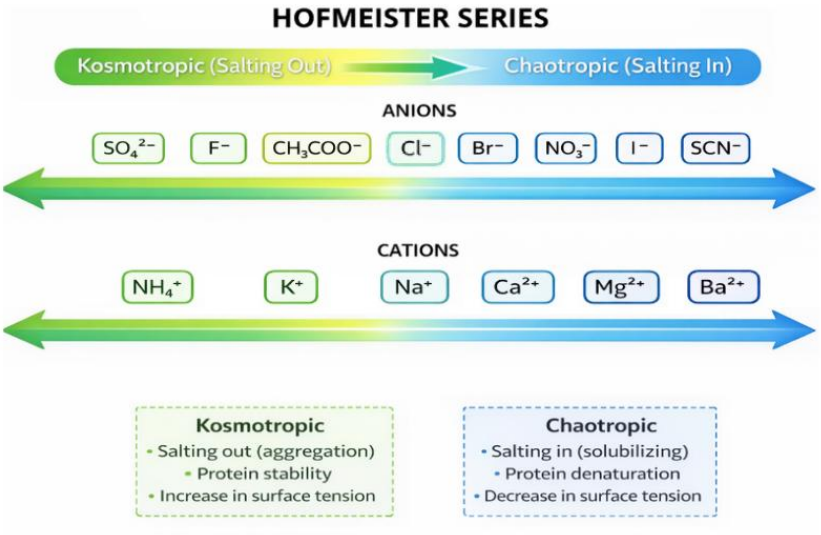


**Fig. 2.** Cations and anions used in this work ordered according to the Hofmeister series.

An extensive literature review highlights a wide variety of solubility data available for amino acids in aqueous electrolyte solutions, with glycine being the most studied system. However, despite the abundance of data, significant gaps persist, and the quality of the data needs a lot of assessment [5,7,8]. That is, across different authors, the effect of the same salt on glycine solubility can be quite distinct. The review reveals inconsistencies and contradictions among published results, underscoring the need for a systematic validation procedure [5,7].

A representative example of a model system exhibiting significant inconsistencies is shown in Figure A2.D, in appendix A, which represents the solubility behavior of glycine in the presence of $K_2SO_4$. In this case, the discrepancies among the reported data cannot be neglected. Considering this variability, the main objective is to employ active learning methodologies to develop a systematic framework capable of identifying the most plausible trends and effects, while simultaneously accounting for all available information in an integrated manner [9]. The full set of experimental data points considered in this analysis is compiled in Table B1, in appendix B.

In this context, machine learning models emerge as a promising alternative for addressing the evaluation of the experimental data [9]. These models are capable of identifying patterns within dispersed and incomplete data sets, enhancing the interpretation of trends and enabling the analysis of inconsistencies [9]. Among the available approaches, Gaussian Processes (GP) stand out, due to their ability to model a large volume of observed data, but also due to their stochastic nature, i.e., their ability to provide a quantification of the uncertainty associated with their predictions [9]. This feature is especially valuable for detecting inconsistencies between different experimental datasets, reinforcing the importance of critically assessing both the methodologies employed and the reliability of reported results[9]. Therefore, GP is a strategic tool for consolidating existing data and guiding future experimental validation efforts in a more rational way.

This study proposes a modeling framework based on GPs to analyze, predict, and validate the solubility of glycine in aqueous solutions containing electrolytes, using an experimental database compiled from multiple literature sources [1-8,10-20]. The GP model enables the identification of inconsistencies among datasets, capturing salting-in/salting-out phenomena and ion-specific effects. It provides an assessment of the uncertainty, helping in the detection of data points deviating from expected trends. In this way, the GP offers a robust quantitative basis for validating existing data, resolving experimental inconsistencies, and guiding the design of future experiments more assertively.

## 2. METHODOLOGY

The structuring of a database related to the solubility of amino acids in aqueous systems containing salts represented the initial and fundamental stage of this research, requiring careful analysis of each author and the respective data reported, especially in the methodologies used, the results achieved, and their relative errors.

Solubility data collection was carried out for glycine in various electrolyte aqueous solutions to map and compare the different salt effects. To ensure the standardization of the experimental conditions, only studies carried out at a temperature of 298.2 K were selected. The database, organized in a spreadsheet, gathers salt molality, solubility, relative solubility, uncertainty and temperature, in addition to the identification of the salt studied and the respective bibliographic reference. At the end of this stage, it was evident that many of the data collected presented inconsistencies in relation to the others, indicating the presence of possibly flawed or methodologically questionable experimental results, and it was necessary to perform a structural chemical analysis of each salt. The analysis is based on relative solubility, defined as the ratio of glycine solubility at a given salt concentration to its solubility in pure water reported by the same authors. This approach also minimizes the impact of variability in the solid crystal structure (α, β, or γ), which remains unidentified in most studies.

### 2.1 Sigma Profiles

Sigma profiles are molecular descriptors derived from quantum chemistry (DFT) calculations in which a molecule is placed in a continuum-solvent model and its screened surface charge density is computed. By partitioning the molecular surface into segments and constructing an unnormalized histogram of these local charge values, the sigma profile encodes the distribution of polarity, electron density and solvation-relevant features in a size-independent vector. This representation captures chemically meaningful information that is often inaccessible to descriptors based solely on atom types, connectivity or graph topology.

Abranches and co-workers [21] have shown that sigma profiles can function as universal molecular descriptors for machine learning workflows. Their studies demonstrate that models trained exclusively on sigma profiles are capable of predicting diverse physicochemical properties such as boiling point, vapor pressure, density and solubility, with excellent accuracy. Because sigma profiles compactly summarize rich electronic information while maintaining fixed dimensionality, they reduce the need for large datasets, simplify model architectures and avoid the scaling issues typical of size-dependent descriptors.

To obtain sigma profiles in this work, each target ion was subjected to a DFT calculation in the software package TURBOMOLE [22], using the def-TZVP basis set and the BP-86 functional, under the continuum-solvent model COSMO. This DFT calculation involves the geometry optimization of the ion in the COSMO solvation environment (with infinite permittivity), yielding the so-called sigma surface, a tessellated molecular surface with associated screened charge (surface charge density) on each surface patch. These calculations were complemented with basic structures from crystallographic data, ensuring accuracy through the comparison of experimental measurements. The basis set and functional chosen are the default used in COSMO-related calculations.

Sigma surfaces are processed into sigma profiles by binning the surface patches according to their local charge densities. The result is an unnormalized histogram (or distribution) of surface area vs. charge density across the molecular surface. In other words, sigma profiles indicate how much surface area of the molecule has a given range of screened charge. In this work, the conversion of sigma surfaces to sigma profiles was carried out using the software package COSMOtherm [23].

2.2 Gaussian Process (GP)

A GP is a set of multivariate normal distributions capable of predicting values of an unknown function, being used as a nonparametric probabilistic model. A GP assumes that the values of the function follow a joint normal distribution and that the relationship between the points is described by a covariance function $k(x, x')$ (kernel) and a mean function M(x).

$$f(x) \sim GP(M(x), k(x, x')) \tag{1}$$

That is, the values of the $f(x)$ function are treated as random variables with joint normal distribution. For a set of N entry points, the joint distribution of the function values can be written as:

$$\begin{bmatrix} f_1 \\ \vdots \\ f_N \end{bmatrix} \sim N(\mu, \Sigma) = N\left(\begin{bmatrix} \mu_1 \\ \vdots \\ \mu_N \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \cdots & \Sigma_{1N} \\ \vdots & \ddots & \vdots \\ \Sigma_{N1} & \cdots & \Sigma_{NN} \end{bmatrix}\right) \tag{2}$$

$$\mu_i = M(x_i) \tag{3}$$

$$\Sigma_{ij} = k(x_i, x_j) \tag{4}$$

Therefore, $\mu$ represents the vector of means and $\Sigma$ is the covariance matrix constructed from the kernel between all pairs of points. When the GP is conditioned to the known data of the function $y(x)$, whether these are represented by $y$ (training set), and the unknown values to

be estimated represented by $f_*$ (test set at the points), it is possible to establish a joint normal distribution according to the following equation:

$$\begin{bmatrix} y \\ f_* \end{bmatrix} \sim N\left(\begin{bmatrix} \mu_y \\ \mu_* \end{bmatrix}, \begin{bmatrix} \Sigma_y & \Sigma_{y*} \\ \Sigma_{y*}^T & \Sigma_* \end{bmatrix}\right) \quad (5)$$

Therefore, for a given test point, the GP allows the predictive distribution of $f_*$, conditioned to the observed data $y$. This distribution is also Gaussian, with mean and variance updated according to the data.

$$f_*|y \sim N(\mu', \Sigma') \quad (6)$$

This equation represents the prediction made by the GP, where $\mu'$ is the predicted mean and $\Sigma'$ is the associated uncertainty. This characteristic makes Gaussian Processes especially valuable in problems where the estimation of uncertainty is crucial for a detailed analysis.

The Python GPflow package (version 2.5.2) was the fundamental tool used to perform all GP-related calculations. The input (features) of the GP model trained in this work are the sigma profiles of each salt ($SP_{salt}$), obtained by summing the sigma profiles of each individual ion of the salt ($SP_{cation}$ and $SP_{anion}$), followed by multiplying by its mole fraction in water ($x_{salt}$):

$$SP_{salt} = x_{salt}(SP_{cation} + SP_{anion}) \quad (7)$$

Because the sigma profiles obtained in this work range from -0.062 e/Å² to 0.062 e/Å², in intervals of 0.001 e/Å², $SP_{salt}$ represents a vector of size 125. The output (labels) of the GP model are the relative solubility of glycine in the specific water/salt solution. Here, relative solubility means the ratio between the solubility of glycine in the water/salt solution and the solubility of glycine in pure water.

The variables were normalized using methods based on logarithmic transformations followed by standardization (normalization). For the independent variables, the Log+bStand transformation was applied, while for the dependent variable, the LogStand transformation was applied. In both cases, the resulting variables were rescaled to a range close to the standardized range with zero mean and unit standard deviation. In the GP model, the zero-mean function was implemented, while the variance of the Gaussian probability function (likelihood) was initially set at a reduced value ($10^{-3}$). The kernel parameters were adjusted by maximizing the marginal log-likelihood using the L-BFGS-B algorithm. In addition, a White kernel was used, which adds an overall estimate of the uncertainty associated with the data.

The Python code developed in this work is freely available in appendix C. Its GP-based structure can be readily reused in new studies by adapting the input data and parameters, enabling further analyses and applications in related research.

# 3. RESULTS AND DISCUSSION

To enable a thorough analysis of glycine solubility in the presence of different electrolytes, the database was developed to cover the widest possible variety of salts. This diversity facilitates a more comprehensive analysis about the influence of both cations and anions on solubility trends. The dataset encompasses a wide array of anions ($F^-$, $Cl^-$, $Br^-$, $I^-$, $NO_3^-$, $SCN^-$, $CH_3COO^-$ and $SO_4^{2-}$) and cations ($Na^+$, $K^+$, $NH_4^+$, $Ba^{2+}$, $Ca^{2+}$ and $Mg^{2+}$). This ensures the analysis is broad, allowing for detailed insights into the chemical interactions governing the solubility of amino acids in saline environments.

## 3.1 Dataset

Our dataset, available in the repository associated with this work, contains the solubility of glycine in aqueous solutions of different salts, as a function of salt concentration. The dataset contains a total of 262 entries. With the database systematically organized, the distribution of collected data for each salt investigated can be clearly visualized in Figure 3 and appendix B (Table B1). It presents the amount of data available for each salt, and an overview of the number of different datasets within a given salt. The dataset also contains information on the reported experimental uncertainty of each data point, which is used to produce experimental error bars throughout this manuscript.

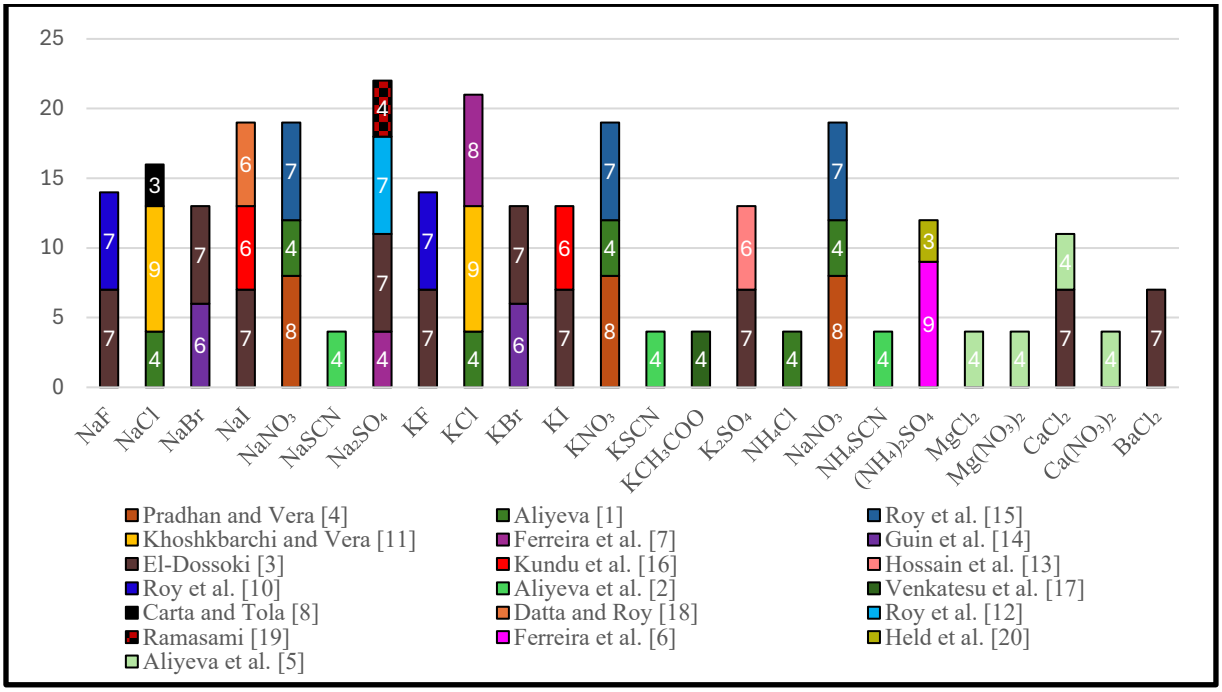


**Fig. 3.** Experimental data points (numbers) and datasets (different colors) collected for the salt effect on the solubility of glycine in aqueous solutions at 298.2 K.

For the construction of the GP model, it was necessary to calculate the sigma profiles of the ions included in the database. These sigma profiles, obtained as explained in Section 2.1, allowed us to define the molecular structures of the salts and to distinguish each system within the model. By considering the sigma profiles of both cations and anions, the GP can account for ion-specific effects, giving a more faithful description of the different physicochemical behaviors associated with each salt. This approach also supports more reliable predictions of the systems under study. Figure 4 presents the calculated sigma profiles for the cations (top) and anions (bottom) that constitute the investigated salts.

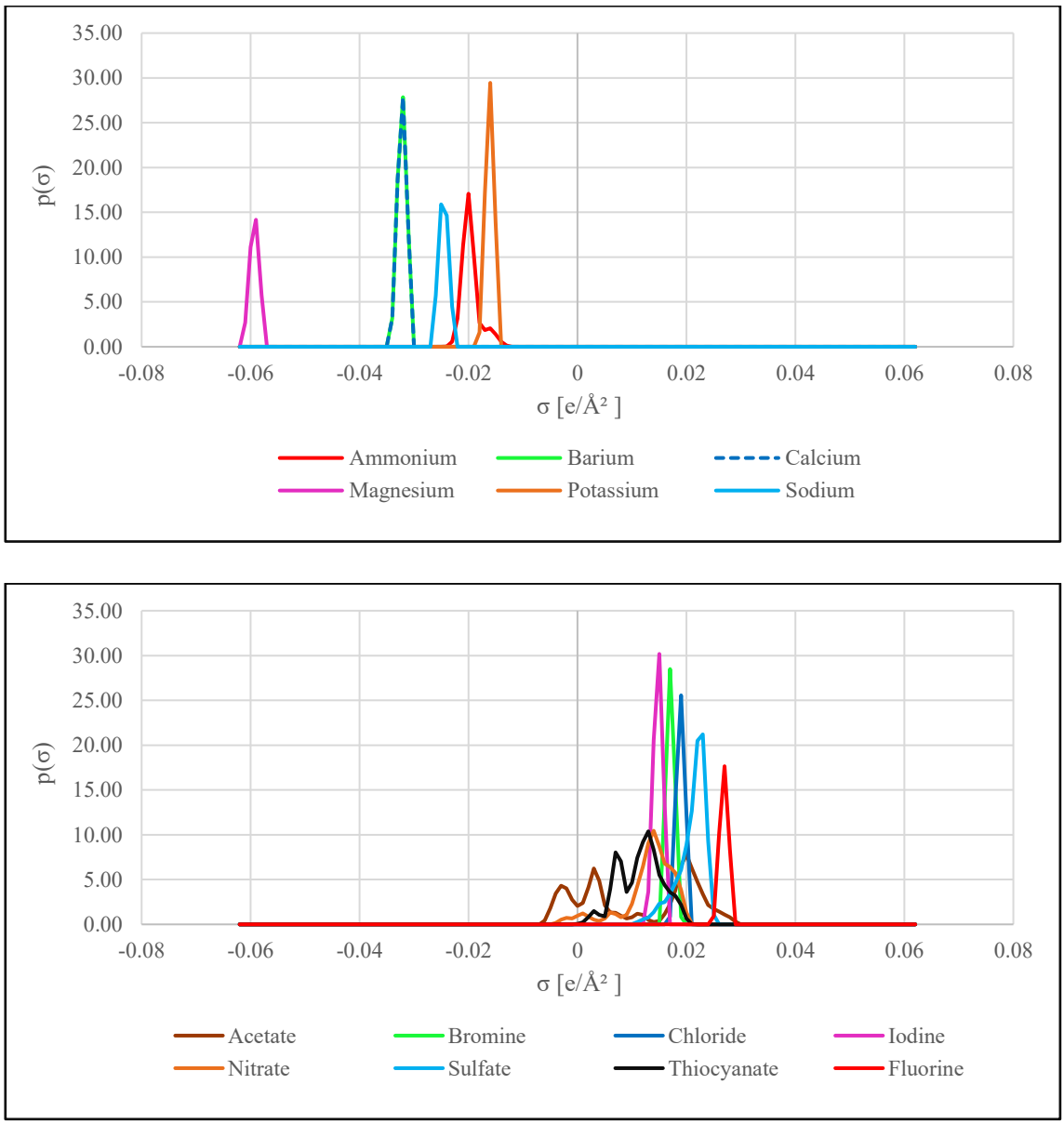


**Fig. 4.** Calculated sigma profiles for cations (top) and anions (bottom) forming the salts of the studied database, obtained from COSMO-RS surface charge density distributions.

To provide a comprehensive overview, a global evaluation of the model's performance was conducted. Specifically, as illustrated in Figure 5, the experimental relative solubility values ($S/S_0$), are compared with the corresponding predictions by the GP model, which indicates a good predictive capacity of the model. In this case, most of the data points cluster around the diagonal line, indicating a satisfactory performance of the GP model. This statement is supported by the coefficient of determination ($R^2 = 0.750$), which confirms that the model captures the experimental data trends with reasonable accuracy, even considering the diversity of salts included in the database.

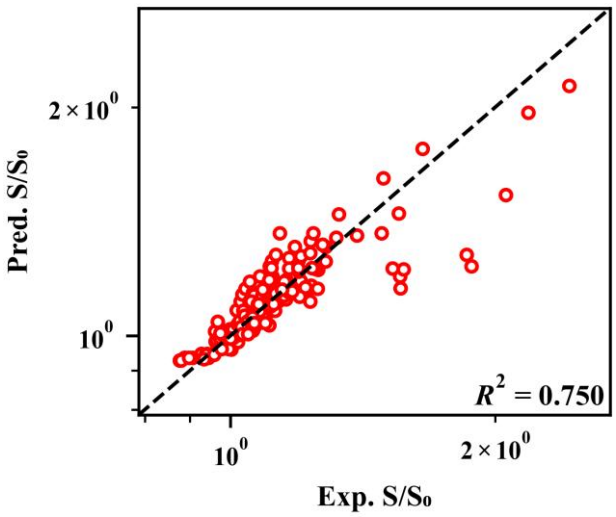


**Fig. 5.** Comparison between experimental (Exp. $S/S_0$) and predicted (Pred. $S/S_0$) relative solubility values for glycine in salt aqueous solutions at 298.2 K, using the GP model.

As can be seen in Figure 6, a detailed analysis of the data reveals that, for salt systems with data reported by multiple authors, there is a lack of consensus regarding their effect on glycine solubility. This variability underscores the importance of applying the GP model to evaluate and validate the data. Such an approach enables the identification of inconsistencies and contributes to enhancing the overall reliability of the dataset.

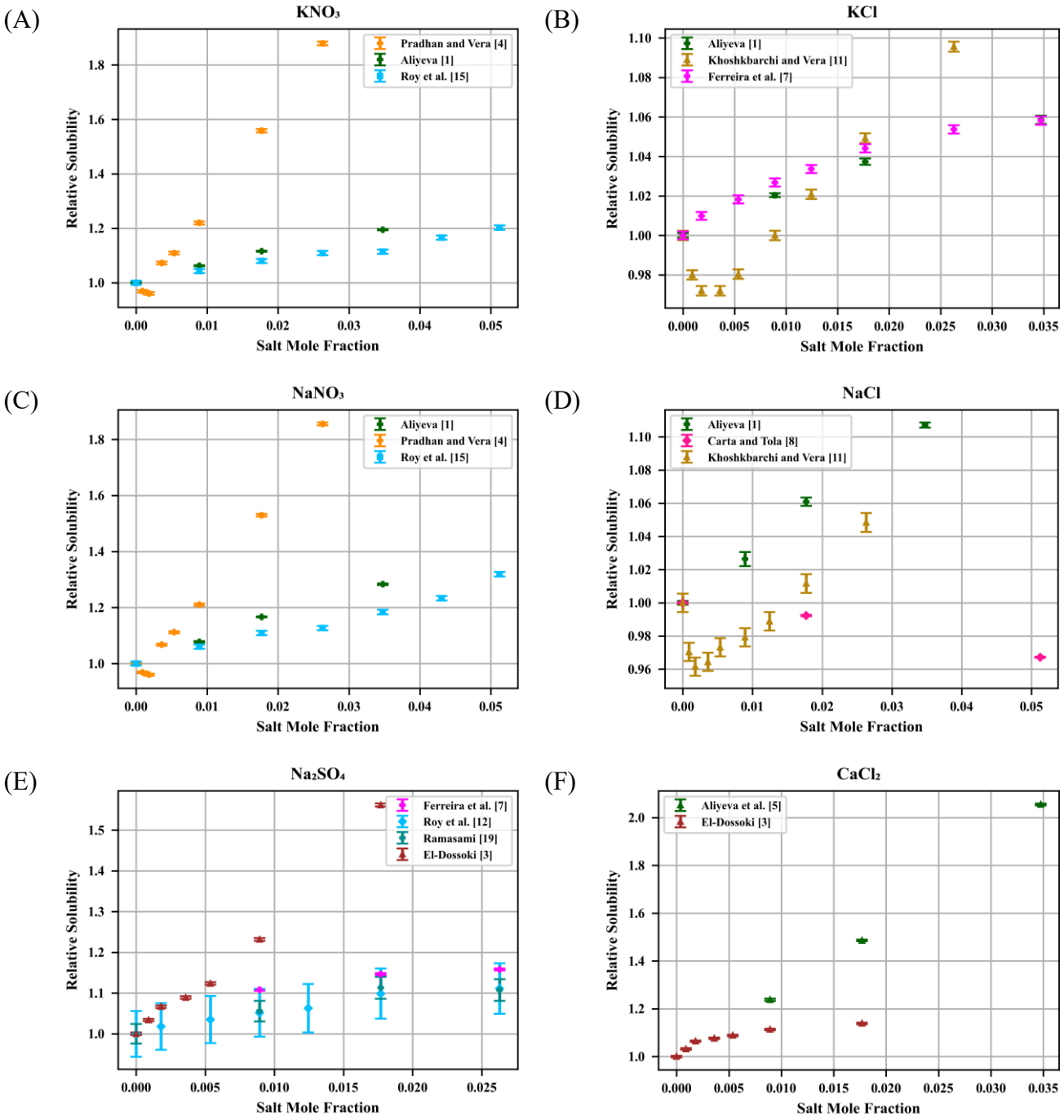


**Fig. 6.** Experimental relative solubility of glycine in $KNO_3$ (A), KCl (B), $NaNO_3$ (C), NaCl (D), $Na_2SO_4$ (E), $CaCl_2$ (F) aqueous solutions at 298.2 K, from different authors.

For the aqueous solution containing $KNO_3$ (Figure 6A) or $NaNO_3$ (Figure 6C), it is observed that the results reported by Pradhan and Vera [4] present relative solubility values significantly higher than those of the other authors. This discrepancy suggests that methodological differences or specific experimental conditions may have influenced the results, causing them to deviate from the general trend observed in the other datasets.

For the KCl and NaCl systems (Figures 6B and 6D), Khoshkbarchi and Vera [11] report a salting-out effect at low salt mole fractions for KCl, whereas Ferreira et al. [7] and Aliyeva [1] the opposite salting-in effect. For NaCl, Carta and Tola [8] present only two values that deviate from the trend shown by the other authors.

In the case of $Na_2SO_4$ (Figure 6E), there is a good general agreement among most authors, except for El-Dossoki's data [3], which present higher values of relative solubility throughout the concentration range. As a last example, for $CaCl_2$ (Figure 6F), Aliyeva et al. [5] report a clear, near-linear increase in relative solubility with salt mole fraction, whereas El-Dossoki [3] observes only a slight increase across the measured range.

The combined global and individual analyses of the experimental data enabled the identification of both consistent patterns and discrepancies among datasets for each salt studied. Understanding these variations is crucial for the proper curation of the database and for the development of reliable predictive models. This approach reinforces the usefulness of the GP as a tool for assessing the reliability of experimental results and revealing potential methodological biases or variations in measurement conditions. It further emphasizes the importance of implementing models capable of identifying which data are closest to the expected behavior.

### 3.2 GP Regression

The development of the GP model implies establishing a nonlinear and probabilistic relationship between the physico-chemical description of the system and the relative solubility of glycine. To this end, the input independent variables were defined as the product between the mole fraction of the salt and its sigma profile, a parameter that describes the distribution of molecular charge density. The relative solubility of glycine was used as the dependent variable.

The output variable of the model corresponds to the relative solubility predicted by the GP model. As the data are sequentially introduced from the database, the model dynamically updates its predictions, estimating the expected values for each combination of mole fraction and sigma profile. In addition to point predictions, the GP model provides a confidence interval for each estimate, allowing not only to assess global and individual trends, but also to assess the uncertainty associated with the results quantitatively. This feature is fundamental for interpreting the results, as it facilitates direct comparison between the GP predictions and the experimental values.

An analysis of the model output for the $KNO_3$ containing system (Figure 7A), reveals that the prediction curve (red line) captures the overall trend of the experimental data,

illustrating an increase in the relative solubility of glycine with increasing mole fraction of the salt. However, the shaded region, which represents the model confidence interval, indicates that a substantial portion of the experimental points lie outside this interval, with only a limited number falling within the predicted uncertainty bounds, and discarding most of the data. The results reported by Pradhan and Vera [4] predominantly fall outside, presenting the larger distances to the uncertainty range predicted by the GP model, suggesting that these measurements are inconsistent with the expected pattern for the relative solubility of glycine in this system. A similar situation occurs in the systems containing $NaNO_3$ (Figure 7C).

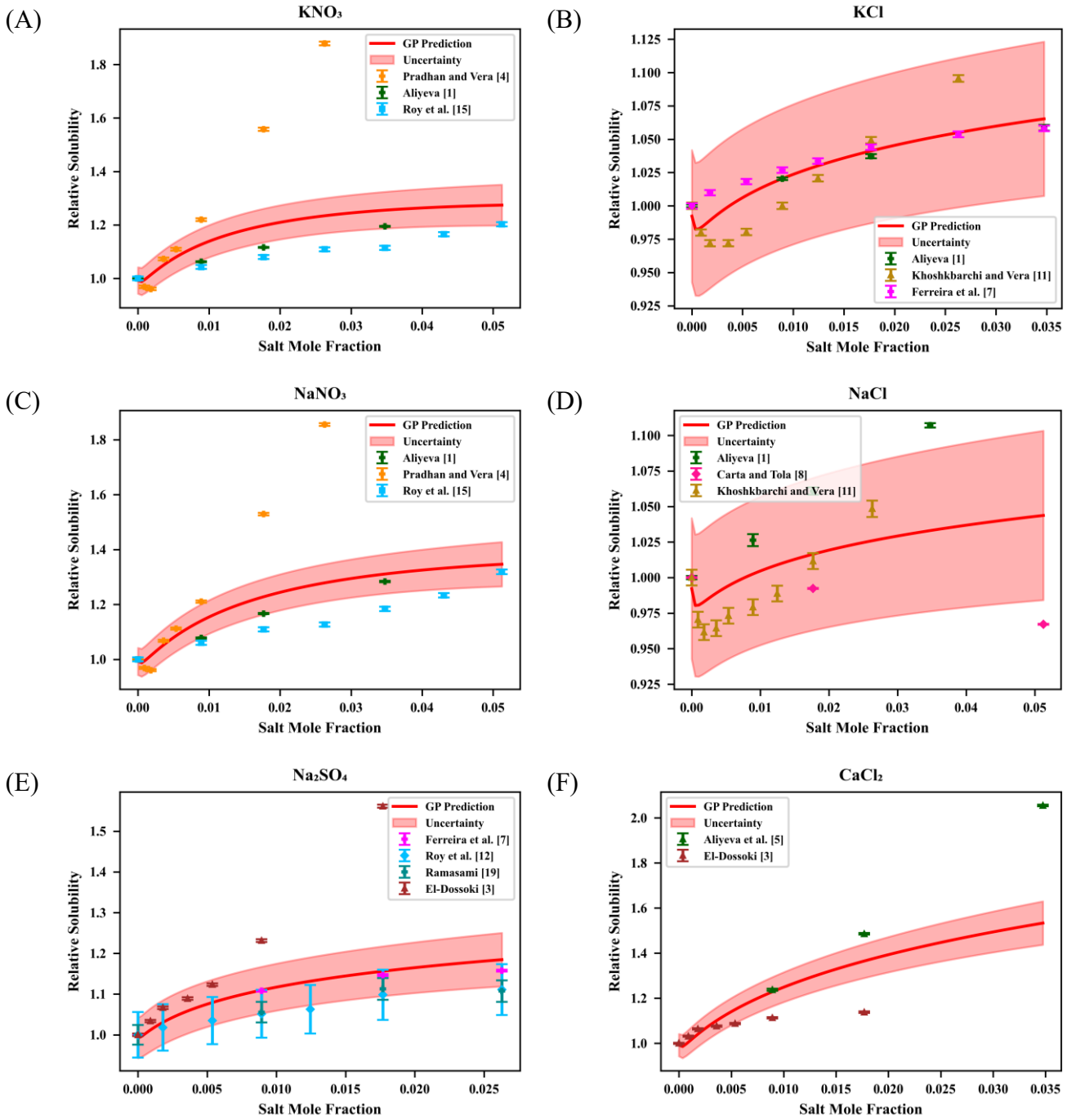


**Fig. 7.** GP model prediction (red line) for the relative solubility of glycine in $KNO_3$ (A), KCl (B), $NaNO_3$ (C), NaCl (D), $Na_2SO_4$ (E), $CaCl_2$ (F) aqueous solutions at 298.2 K, and the associated prediction uncertainty.

In stark contrast, the systems containing KCl and NaCl (Figures 7B and 7D) exhibit markedly different behavior from the nitrates. One of the key findings in these cases is the presence of distinct salting-in and salting-out effects in the dilute solutions, which introduces significant variability into the data. Despite the relative proximity of the experimental values, this variability leads to broader uncertainty intervals in the GP predictions, wide enough that

most data points fall within the model's confidence limits. Specifically, for the NaCl system, the data reported by Aliyeva [1] and Carta and Tola [8] include several values, particularly at higher mole fractions, that fall outside the model's predicted uncertainty range. This suggests that portions of these datasets may be inconsistent with the expected solubility behavior. Moreover, none of the datasets examined align well with the trend predicted by the GP model, reinforcing the need for a more critical assessment of the experimental methodologies and conditions used in this system.

For the $Na_2SO_4$ containing system (Figure 7E), the model predicts a moderate and continuous increase in the relative solubility of glycine with increasing salt mole fraction. The uncertainty region encompasses many data points, as three independent studies report values within a similar range. However, the dataset from El-Dossoki [12] reports significantly higher values than the others, particularly at higher salt concentrations. This discrepancy contributes to an expanded uncertainty interval and results in a more noticeable deviation between the model prediction and the remaining experimental data at higher mole fractions. Among the sets evaluated, only the data from Ferreira et al. [7] are fully within the model's uncertainty region, being most consistent with the predicted trend.

As a final remark, for the $CaCl_2$ system (Figure 7F) the GP model predicts a smooth increase in the relative solubility of glycine with rising salt mole fraction. The uncertainty band covers most measurements at low to intermediate mole fractions, particularly those from El-Dossoki [3], which cluster around the prediction. In contrast, the values reported by Aliyeva et al. [5] tend to lie above the trend, and the highest concentration point clearly falls outside the uncertainty region, yielding the largest deviation. The model captures the global tendency, but the high-concentration behavior remains weakly constrained due to sparse and inconsistent data within the sampled range. For this system, El-Dossoki's measurements are the most consistent with the predicted trend.

Overall, the predictions of the GP model effectively captured the trends in the relative solubility of glycine across the various saline systems, even in cases where discrepancies existed among experimental datasets. The model showed sensitivity to the individual contributions of the constituent ions of each salt, reflecting their influence on solubility behavior. This allowed for the identification of consistent patterns among systems containing common ions, further reinforcing the model's capacity to reveal underlying physicochemical relationships. Considering the other systems in the database, a similar analysis can be carried out by observing appendices C.

Analysis of the results indicates that the model adapts its predictive behavior, to align with the observed trends. However, a further refinement is needed to improve its ability to distinguish which data points are most consistent with the predicted values. In addition, the assessment of predictive uncertainties helps identify regions of greater reliability, guiding the selection of new experimental points to determine and improve the model, thereby increasing the accuracy of the predictions.

### 3.3 GP Meta Analysis

The implementation of the data filtering strategy, removing the experimental points that are outside the uncertainty bounds of the predictions, emerges as a promising approach to improve the GP model performance. This method reduces the impact of potentially inconsistent data, enabling a more accurate and robust fit. By delimiting the data that present better agreement with the predicted behavior, the model can refine its parameters more effectively, thereby improving both its predictive accuracy and the overall reliability of the results. Figure 8 illustrates the workflow proposed for this modeling approach.

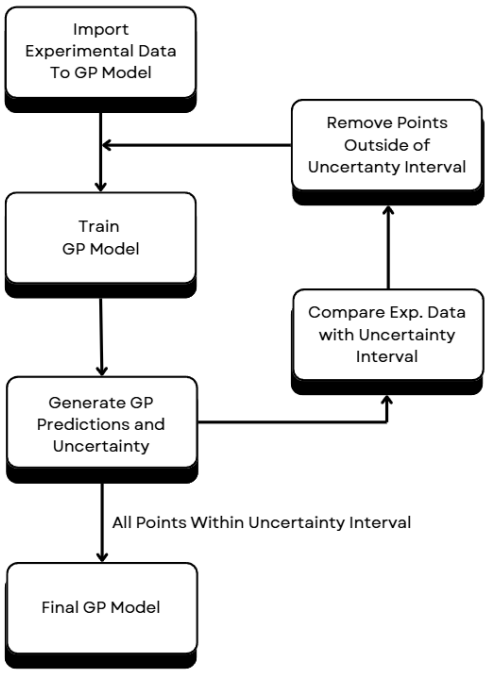


**Fig. 8.** A data-filtering workflow for training GPs to describe the relative solubility of glycine in different salt solutions.

The workflow follows a simple, iterative cycle: first, the model is trained with the initial dataset; next, the predictions of relative solubility and the respective uncertainty are computed for each salt; then, these predictions are compared with the corresponding experimental measurements at each salt mole fraction; Finally, any points falling clearly outside the

uncertainty range of the model are excluded from subsequent training iterations. The procedure is applied to all systems in a synchronized manner: at each iteration, all inconsistent points are removed from each system, ensuring balance and preventing favoritism. In this way, outliers from one system do not affect the others.

It is important to note that the filtering procedure was applied using a 99.5% confidence interval. No experimental data points are permanently removed from the database. Instead, the exclusions are made only in a copy of the dataset used for model predictions. In the Figures, points excluded from new GP predictions and calculations are marked with a red circle, clearly indicating that they were filtered out. This iterative filtering process contributes to greater model stability, improved predictive accuracy, and increased confidence in the results, while ensuring that the data selection remains traceable and scientifically robust.

By applying the iterative filtering method, the GP model was able to identify inconsistent points within the dataset. A detailed account of the number of excluded data points for each system is provided in appendix A (Table A1). Furthermore, Figure 9 illustrates the systems after the application of the filtering procedure, highlighting the improvement in model consistency once the inconsistent data points are removed from the GP predictions.

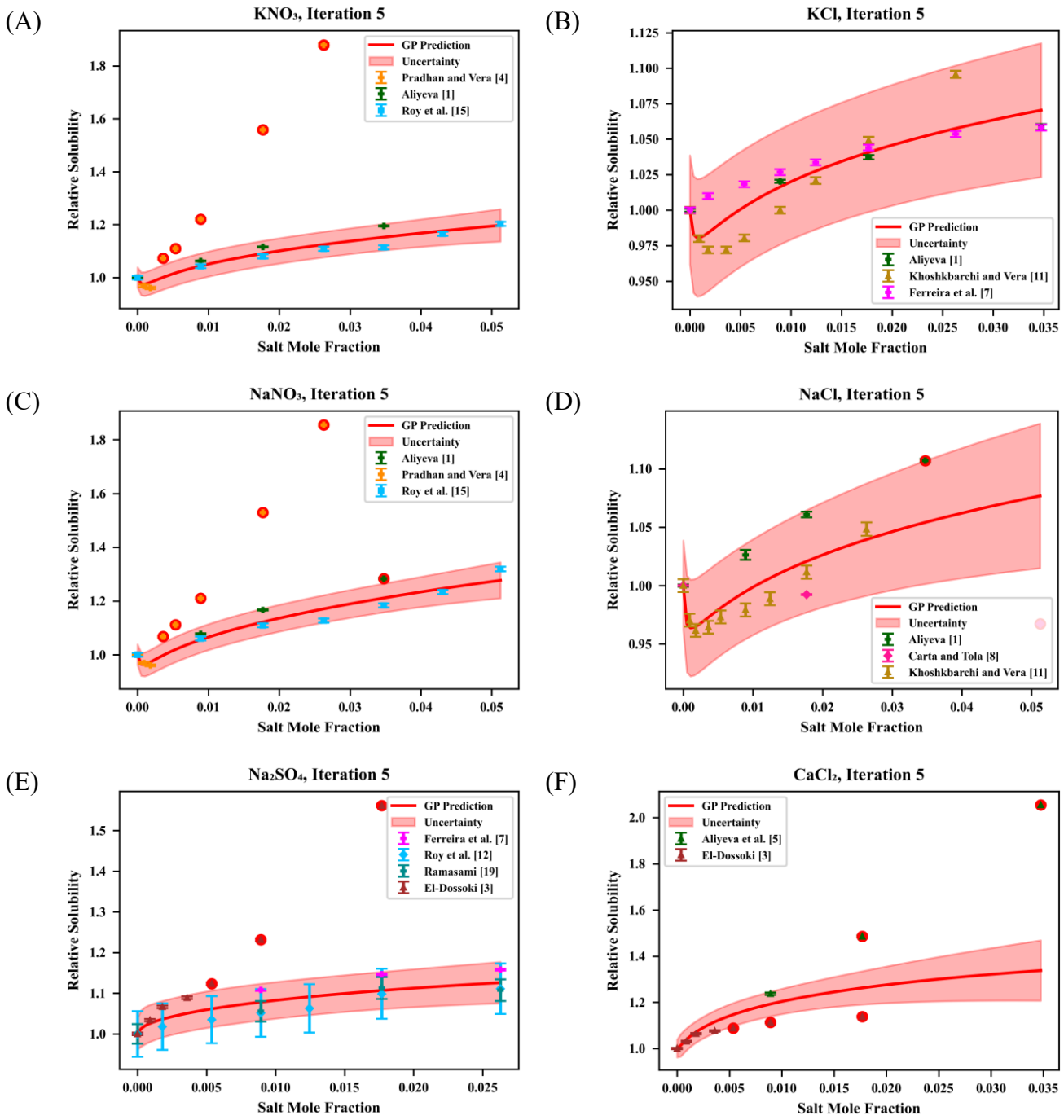


**Fig. 9.** GP prediction of glycine relative solubility in $KNO_3$ (A), KCl (B), $NaNO_3$ (C), NaCl (D), $Na_2SO_4$ (E), $CaCl_2$ (F) aqueous solutions at 298.2 K, with uncertainty bands after iterative removal of points outside the uncertainty interval.

When analyzing the behavior of the model for the $KNO_3$ (Figure 9A) and $NaNO_3$ (Figure 9C) systems, it is observed that, after applying the iterative filtering procedure, the results of Roy et al. [15] consistently showed the highest agreement with the GP predictions. In both systems, the narrowing of the uncertainty interval reinforces the quality of the fit and the high predictive confidence of the model. For Pradhan and Vera [4], the results at lower mole

fractions were considered consistent with the model, while the higher values were excluded during the filtering process. The data of Aliyeva [1] were also found to be consistent overall, with only the final value for the $NaNO_3$ system being disregarded after the iterative filtering.

In the evaluation of the GP model applied to the KCl (Figure 9B) and NaCl (Figure 9D) systems, it is observed that most of the experimental data reported by different authors are consistent with the predictions. For KCl, the model tends to follow more closely the results of Khoshkbarchi and Vera [11] at lower mole fractions, while the data of Ferreira et al. [7] become more consistent as the mole fraction increases. In contrast, for NaCl, the predictions are predominantly aligned with Khoshkbarchi and Vera [11], and only the final points of Carta and Tola [8] and Aliyeva [1] were excluded by the uncertainty criterion, leading to greater dispersion of the model and broader uncertainty at higher concentrations.

For the $Na_2SO_4$ system (Figure 9E), the results of El-Dossoki [3] showed some inconsistencies at higher mole fractions, whereas the data reported by Ferreira et al. [7], Ramasami [19], and Roy et al. [12] remained consistent with the predictions of the GP model across the evaluated range. However, among these authors, it is not evident which dataset aligns most closely with the model predictions.

Finally, for the $CaCl_2$ system (Figure 9F), the GP predictions indicate that the expected effect is a slight increase in relative solubility with increasing mole fraction. This outcome contrasts with the experimental results of both El-Dossoki [3] and Aliyeva et al. [5], whose reported trends at higher concentrations are inconsistent with the model, leading to the remotion of these points during filtering.

After applying the iterative filtering of inconsistent data, the uncertainty regions are significantly reduced, and the datasets reported by different authors become more coherent with each other. This process allows the model to emphasize which points are truly consistent, resulting in greater stability and improved predictive accuracy. Figure 10 illustrates this step-by-step filtering procedure for the $KNO_3$ system, showing how the progressive exclusion of inconsistent values strengthens the reliability of the model outcomes.

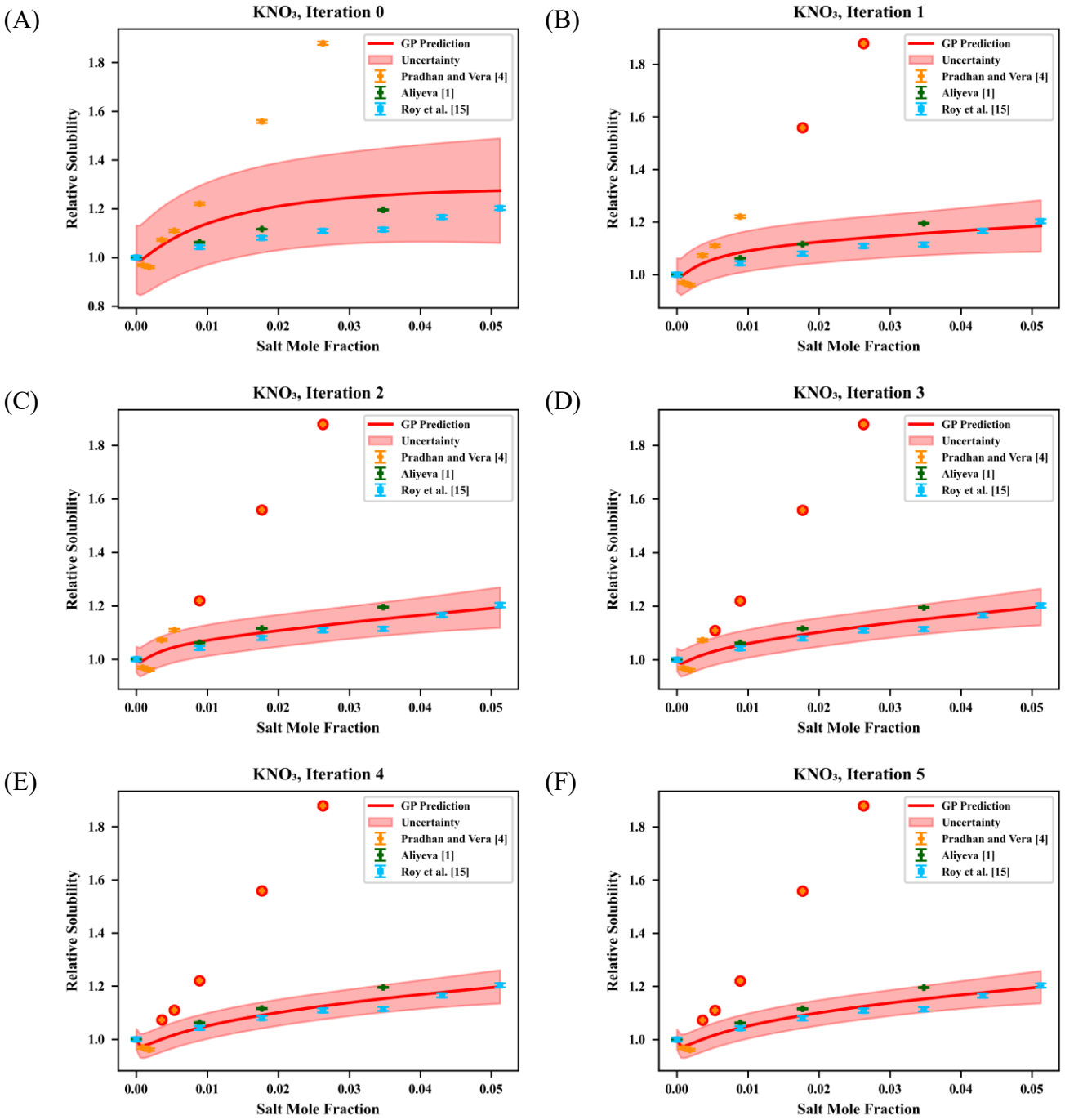


**Fig. 10.** Stepwise iterative filtering of inconsistent experimental data in the $KNO_3$ aqueous system at 298.2 K, using the GP model (confidence interval of 99.5%).

From Figure 10A to 10B, corresponding to the first iteration, the GP model already shows a noticeable reduction in the uncertainty band. Two points from Pradhan and Vera [4], located outside the confidence interval, are removed, which increases the precision of the prediction. At this stage, the datasets of Aliyeva [1] and Roy et al. [15] become clearly aligned with the model trend, reinforcing their consistency. In the subsequent steps (Figures 10C and 10D), corresponding to iterations 2 and 3, additional points from Pradhan and Vera [4] are progressively removed. This indicates that, for the most part, their reported results diverge from

the predicted behavior. Meanwhile, the uncertainty interval gradually adjusts around the datasets of Aliyeva [1] and Roy et al. [15], as well as the low mole fraction values of Pradhan and Vera [4], which remain consistent with the GP predictions. During the final stages (Figures 10E and 10F), only minor refinements occur, with a slight narrowing of the uncertainty band. This confirms the stability of the model and the consistency of the results reported by Aliyeva [1] and Roy et al. [15]. By the fifth iteration, all inconsistent points have been filtered out based on the 99.5% confidence criterion, leaving only the data that best represent the expected solubility trend for glycine in the $KNO_3$ system.

Therefore, we can conclude that the application of the iterative method of filtering by the uncertainty area proved to be effective in making the GP model more reliable and accurate. By removing the datapoints lying outside the bands, it reduces the influence of inconsistent results between authors and stabilizes the model adjustment process, allowing the GP to clearly identify the data set that is most consistent with their predictions, resulting in a model with minimal uncertainty areas and improved prediction. In addition, data filtering highlights the need to verify potentially inconsistent data and guides new studies in regions of greater uncertainty.

# 4. CONCLUSIONS

This research demonstrated that, even when starting from a database with inconsistencies and contradictory results between different saline solutions, the GP model allowed us to quantify and interpret the specific influence of cations and anions on the relative solubility of glycine in aqueous solutions. Despite these adversities, the GP adequately reproduced the main experimental behaviors and identified which datasets are most consistent.

The proposed meta-analysis, based on iterative filtering of points outside the uncertainty area of the model itself, proved to be crucial to improve the predictive capacity and identify inconsistencies. This procedure narrowed the area of uncertainty and highlighted the data most compatible with the model, without definitively removing the results from the database. As a result, the GP has become more stable, predictive, and reliable to support conclusions and guide new studies.

It is therefore concluded that the GP, when properly modeled and continuously improved by methods sensitive to uncertainty and data quality, helps the validation of different databases, even in the presence of inconsistencies. The model quantifies measurement confidence, anticipates the effects on the system, and accurately identifies inconsistent data and poorly sampled regions. GP-based methodology also offers practical guidance for future experimental work. By quantifying predictive uncertainty across salts and compositions, the model highlights regions where existing solubility data are sparse, inconsistent, or otherwise unreliable. These high-uncertainty areas represent the most valuable targets for new measurements, enabling experimental efforts to focus on conditions where additional data would substantially improve confidence in the solubility surface. Conversely, regions with low uncertainty correspond to well-supported measurements that are unlikely to benefit from further experimentation. Thus, the approach provides a systematic way to prioritize remeasurement and to allocate experimental resources more efficiently.

Regarding future work, it is suggested to start by developing a similar approach to aqueous electrolyte systems containing alanine, isoleucine and valine, for which a large set of experimental data is already compiled the SP for the amino acids available as shown in Fig. D1 of Appendix D. It is immediately evident the differences in the SPs what can be an important test to the robustness of the GP model. Finally, it would be of very high relevance to establish a program of new experimental measurements to check the real behavior of glycine solubility

at low salt concentration and extend the studies for other salts not included in the database to serve as a test of the consistency of the calculated solubility values.

# REFERENCES


[1] M. Aliyeva, “Ion Effects on Protein Model Compounds in Aqueous Systems: Experimental and Computational Studies,” Ph.D. Thesis, University of Aveiro, Aveiro, 2022.

[2] M. Aliyeva, P. Brandão, J. A. P. Coutinho, O. Ferreira, and S. P. Pinho, “Solubilities of Amino Acids in the Presence of Chaotropic Anions,” *J Solution Chem*, vol. 53, no. 4, pp. 527–537, Apr. 2024, doi: 10.1007/s10953-023-01282-3.

[3] F. I. El-Dossoki, “Effect of the Charge and the Nature of Both Cations and Anions on the Solubility of Zwitterionic Amino Acids, Measurements and Modeling,” *J Solution Chem*, vol. 39, no. 9, pp. 1311–1326, Oct. 2010, doi: 10.1007/s10953-010-9580-3.

[4] A. A. Pradhan and J. H. Vera, “Effect of Anions on the Solubility of Zwitterionic Amino Acids,” *J Chem Eng Data*, vol. 45, no. 1, pp. 140–143, Jan. 2000, doi: 10.1021/je9902342.

[5] M. Aliyeva, P. Brandão, J. R. B. Gomes, J. A. P. Coutinho, O. Ferreira, and S. P. Pinho, “Solubilities of Amino Acids in Aqueous Solutions of Chloride or Nitrate Salts of Divalent (Mg2+or Ca2+) Cations,” *J Chem Eng Data*, vol. 67, no. 6, pp. 1565–1572, Jun. 2022, doi: 10.1021/acs.jced.2c00148.

[6] L. A. Ferreira, E. A. Macedo, and S. P. Pinho, “The Effect of Ammonium Sulfate on the Solubility of Amino Acids in Water at (298.15 and 323.15) K,” *Journal of Chemical Thermodynamics*, vol. 41, no. 2, pp. 193–196, Feb. 2009, doi:10.1016/j.jct.2008.09.019.

[7] L. A. Ferreira, E. A. Macedo, and S. P. Pinho, “Effect of KCl and Na2so4 on the Solubility of Glycine and DL-Alanine in Water at 298.15 K,” *Ind Eng Chem Res*, vol. 44, no. 23, pp. 8892–8898, Nov. 2005, doi: 10.1021/ie050613q.

[8] R. Carta and G. Tola, “Solubilities of L-Cystine, L-Tyrosine, L-Leucine, and Glycine in Aqueous Solutions at Various pHs and NaCl Concentrations,” *Ind Eng Chem Res*, vol. 41, pp. 414–417, 1996, doi: 10.1021/je9501853.

[9] D. O. Abranches, E. J. Maginn, and Y. J. Colón, “Activity Coefficient Acquisition with Thermodynamics-Informed Active Learning for Phase Diagram Construction,” *AIChE Journal*, vol. 69, no. 8, Aug. 2023, doi: 10.1002/aic.18141.

[10] S. Roy, P. Guin, K. Mahali, and B. Dolui, “Solubility and Transfer Gibbs Free Energetics of Glycine, DL-Alanine, DL-Nor-Valine and DL-Serine in Aqueous Sodium Fluoride and Potassium Fluoride Solutions at 298.15 K,” *Indian Jounal of Chemistry*, vol. 56, pp. 399–406, Apr. 2017.

[11] M. K. Khoshkbarchi and J. H. Vera, “Effect of NaCl and KCl on the Solubility of Amino Acids in Aqueous Solutions at 298.2 K: Measurements and Modeling,” *Ind Eng Chem Res*, vol. 36, pp. 2445–2451, 1997, doi: 10.1021/ie9606395#.

[12] S. Roy, P. S. Guin, K. Mahali, A. Hossain, and B. K. Dolui, “Evaluation and Correlation of Solubility and Solvation Thermodynamics of Glycine, DL-Alanine and DL-Valine in Aqueous Sodium Sulphate Solutions at Two Different Temperatures,” *J Mol Liq*, vol. 234, pp. 124–128, May 2017, doi: 10.1016/j.molliq.2017.03.068.

[13] A. Hossain, K. Mahali, B. K. Dolui, P. S. Guin, and S. Roy, “Solubility Analysis of Homologous Series of Amino Acids and Solvation Energetics in Aqueous Potassium Sulfate Solution,” *Heliyon*, vol. 5, no. 8, Aug. 2019, doi: 10.1016/j.heliyon.2019.e02304.

[14] P. S. Guin, K. Mahali, B. K. Dolui, and S. Roy, “Solubility and Thermodynamics of Solute-Solvent Interactions of Some Amino Acids in Aqueous Sodium Bromide and Potassium Bromide Solutions,” *J Chem Eng Data*, vol. 63, no. 3, pp. 534–541, Mar. 2018, doi: 10.1021/acs.jced.7b00647.

[15] S. Roy, P. S. Guin, S. Mondal, S. Ghosh, and B. K. Dolui, “Solubility of Glycine and DL-Nor-Valine in Aqueous Solutions of NaNO3 and KNO3 and Measurements of Transfer Thermodynamics,” *J Mol Liq*, vol. 222, pp. 313–319, Oct. 2016, doi: 10.1016/j.molliq.2016.07.050.

[16] S. Kundu, K. Mahali, and S. Roy, “Solvation Thermodynamics of Four Amino Acids in Electrolytic Solutions of Sodium and Potassium Iodide Salts at 298.15 K,” *Can J Chem*, vol. 101, no. 4, pp. 224–234, Apr. 2023, doi: 10.1139/cjc-2022-0251.

[17] P. Venkatesu, M. J. Lee, and H. M. Lin, “Transfer Free Energies of Peptide Backbone Unit from Water to Aqueous Electrolyte Solutions at 298.15 K,” *Biochem Eng J*, vol. 32, no. 3, pp. 157–170, Dec. 2006, doi: 10.1016/j.bej.2006.09.015.

[18] A. Datta and S. Roy, “Thermodynamics of Solute–Solvent Interactions and Solubility of Some Amino Acids in Aqueous Sodium Iodide Solutions at T = 298.15 K,” *Russian Journal of Physical Chemistry A*, vol. 95, pp. S62–S70, Apr. 2021, doi: 10.1134/S0036024421140041.

[19] P. Ramasami, “Solubilities of Amino Acids in Water and Aqueous Sodium Sulfate and Related Apparent Transfer Properties,” *J Chem Eng Data*, vol. 47, no. 5, pp. 1164–1166, Sep. 2002, doi: 10.1021/je025503u.

[20] C. Held, T. Reschke, R. Müller, W. Kunz, and G. Sadowski, “Measuring and Modeling Aqueous Electrolyte/Amino-Acid Solutions with ePC-SAFT,” *Journal of Chemical Thermodynamics*, vol. 68, pp. 1–12, 2014, doi: 10.1016/j.jct.2013.08.018.

[21] D. O. Abranches, E. J. Maginn, and Y. J. Colón, “Stochastic Machine Learning Via Sigma Profiles to Build a Digital Chemical Space,” *Proceedings of the National Academy of Sciences*, vol. 121, no. 31, Jul. 2024, doi: 10.1073/pnas.2404676121.

[22] TURBOMOLE V7.1 2016, a development of University of Karlsruhe and Forschungszentrum Karlsruhe GmbH, 1989-2007, TURBOMOLE GmbH, since 2007; available from http://www.turbomole.com. (accessed December 11, 2025).

[23] BIOVIA COSMOtherm, Release 2021," *Dassault Systèmes. http://www.3ds.com*. (accessed December 11, 2025).

# APPENDICES

## A. Experimental and GP model results for glycine solubility data

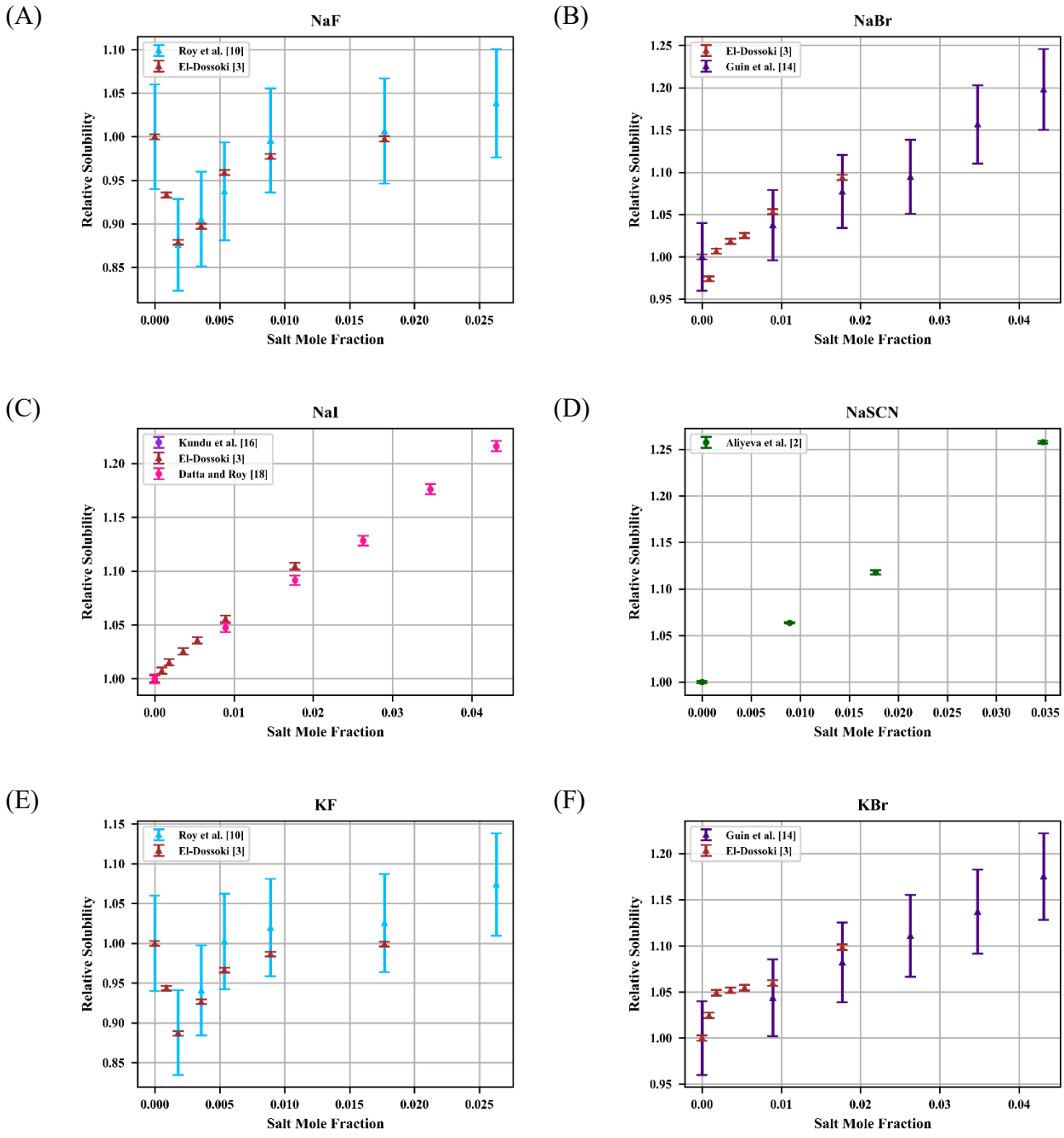


**Fig. A1.** Experimental relative solubility of glycine in NaF (A), NaBr (B), NaI (C), NaSCN (D), KF (E), KBr (F) aqueous solutions from different authors.

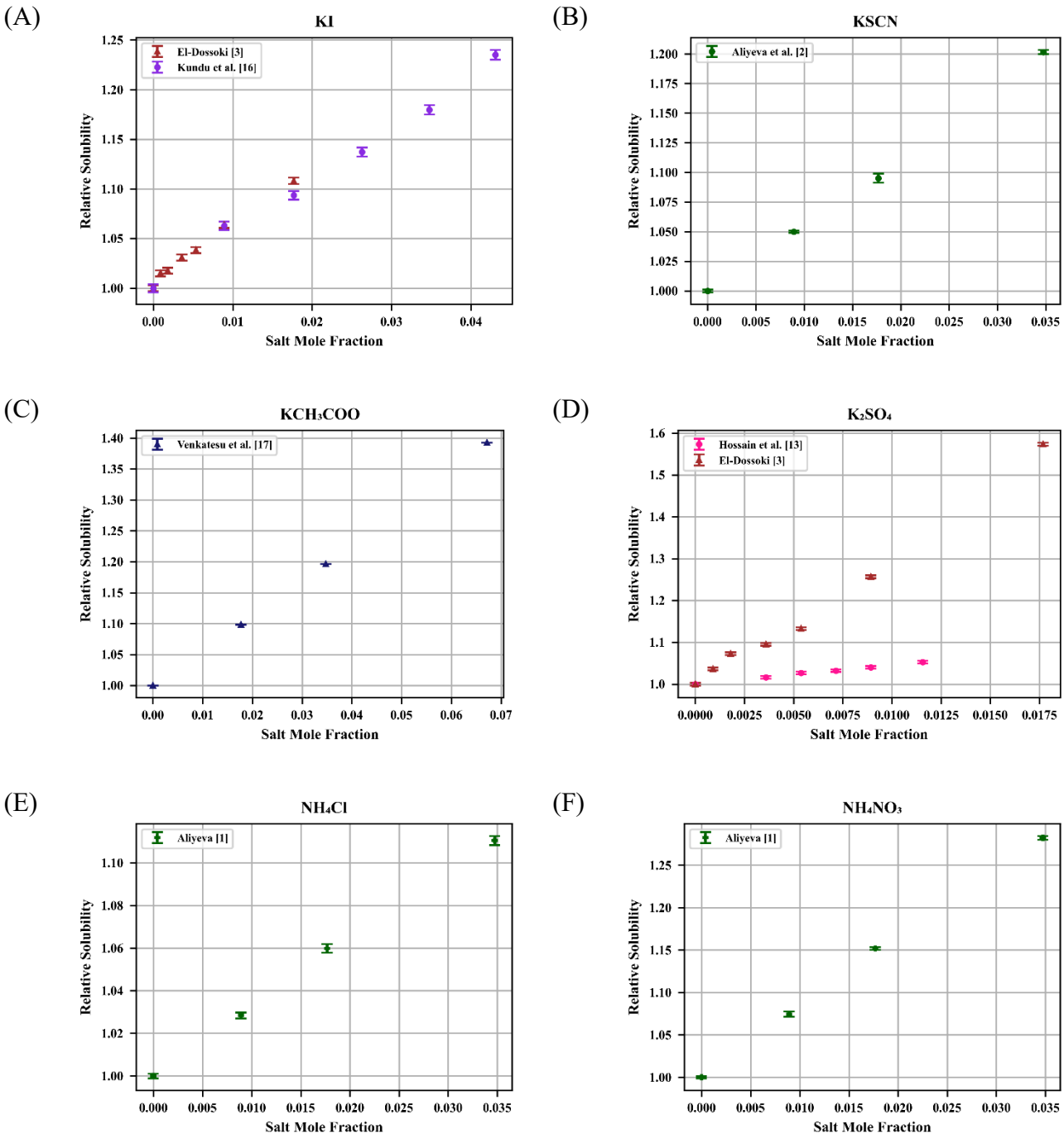


**Fig. A2.** Experimental relative solubility of glycine in KI (A), KSCN (B), $KCH_3COO$ (C), $K_2SO_4$ (D), $NH_4Cl$ (E), $NH_4NO_3$ (F) aqueous solutions from different authors.

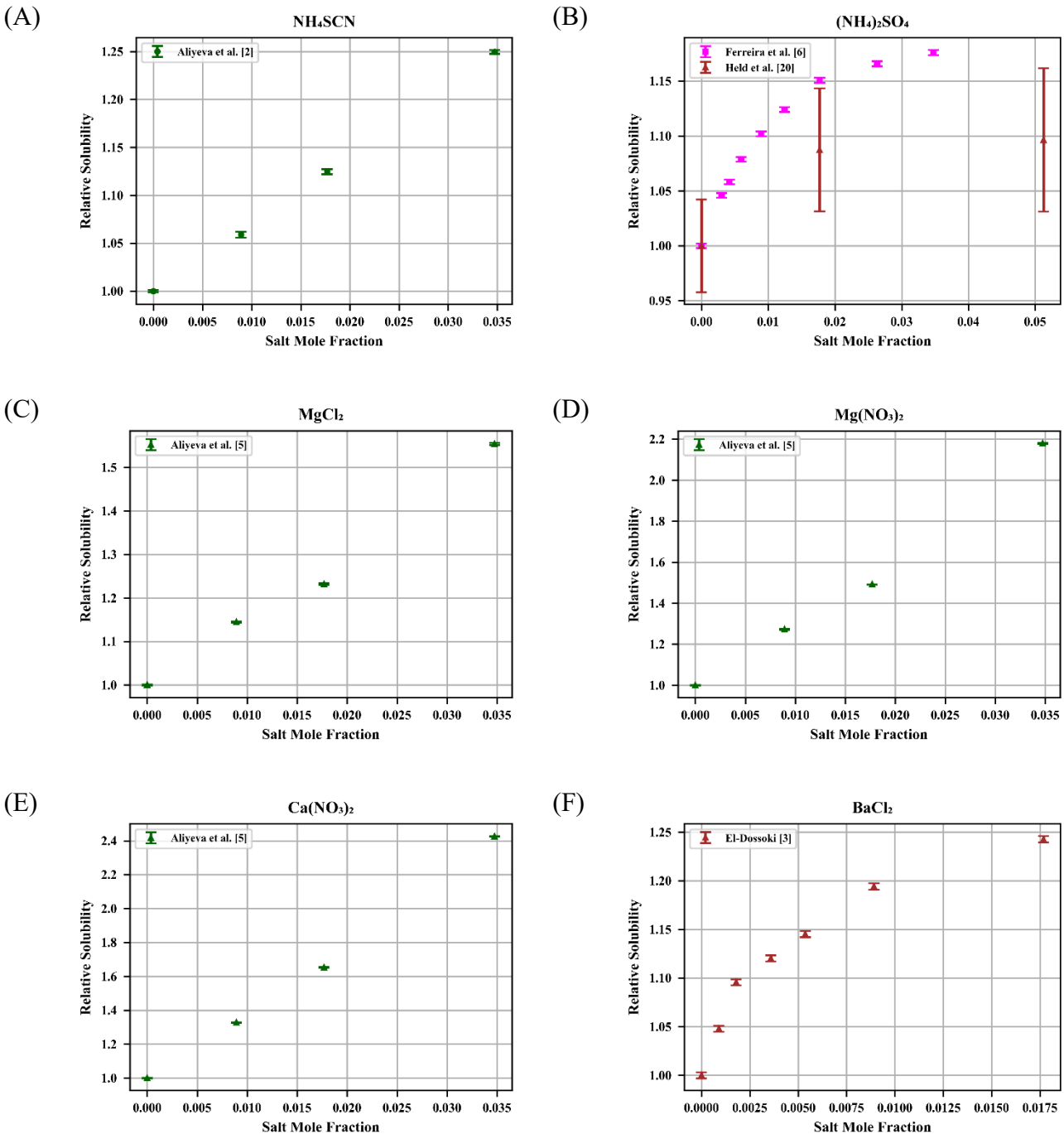


**Fig. A3.** Experimental relative solubility of glycine in $NH_4SCN$ (A), $(NH_4)_2SO_4$ (B), $MgCl_2$ (C), $Mg(NO_3)_2$ (D), $Ca(NO_3)_2$ (E), $BaCl_2$ (F) aqueous solutions from different authors.

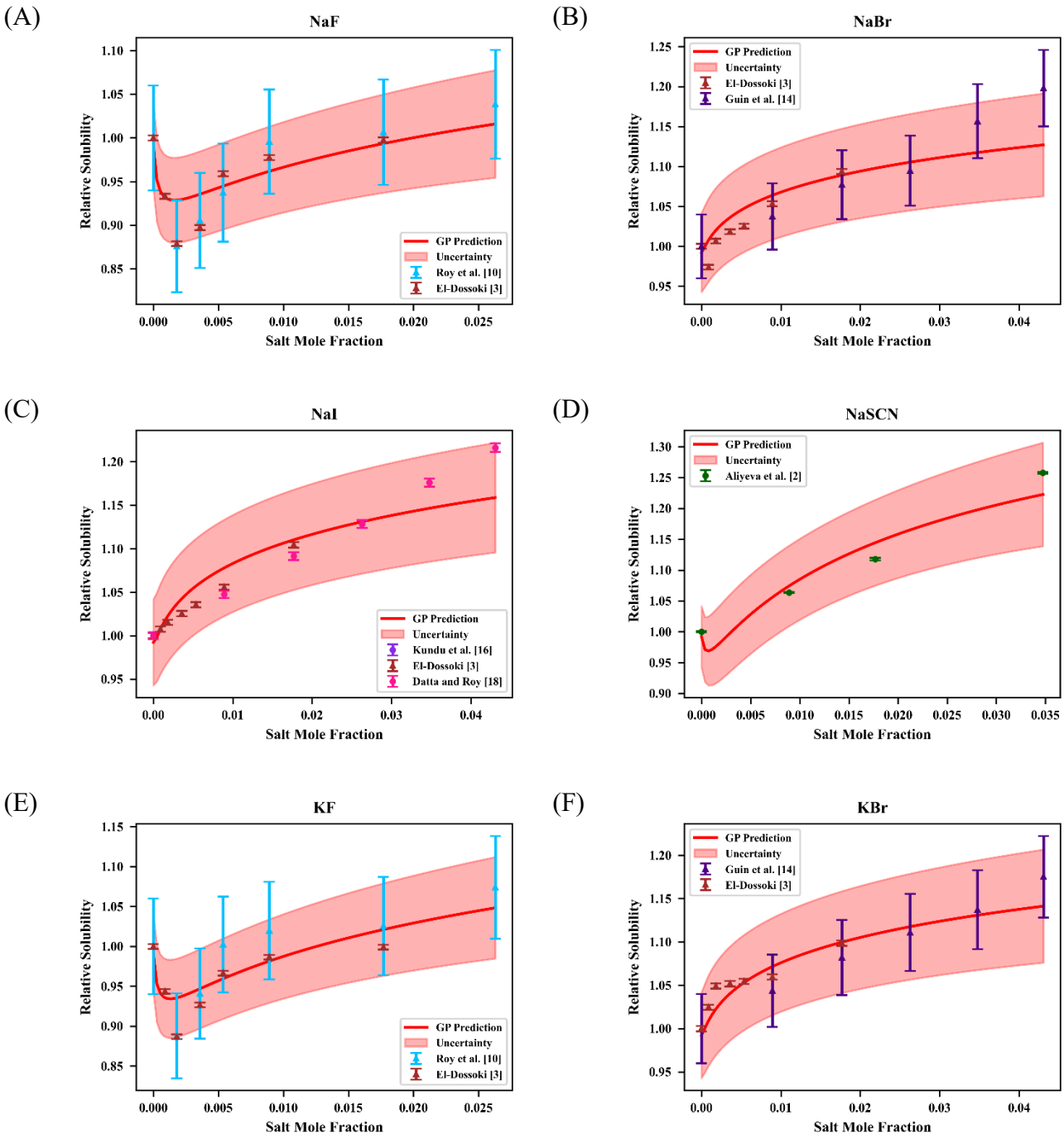


**Fig. A4.** Gaussian Process (GP) model prediction (red line) for the relative solubility of glycine in NaF (A), NaBr (B), NaI (C), NaSCN (D), KF (E), KBr (F) aqueous solutions at 298.2 K, and the associated prediction uncertainty.

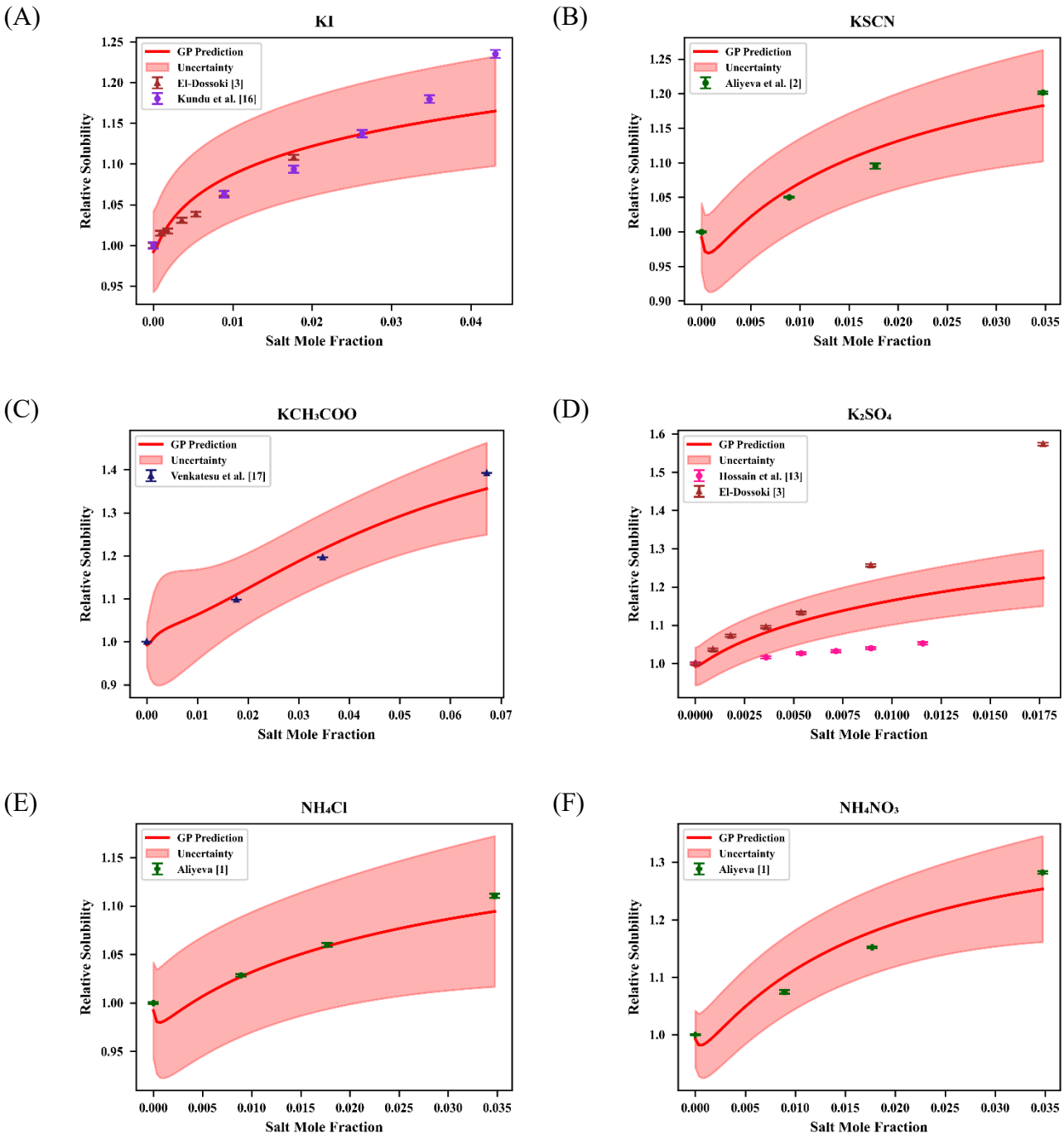


**Fig. A5.** GP model prediction (red line) for the relative solubility of glycine in KI (A), KSCN (B), $KCH_3COO$ (C), $K_2SO_4$ (D), $NH_4Cl$ (E), $NH_4NO_3$ (F) aqueous solutions at 298.2 K, and the associated prediction uncertainty.

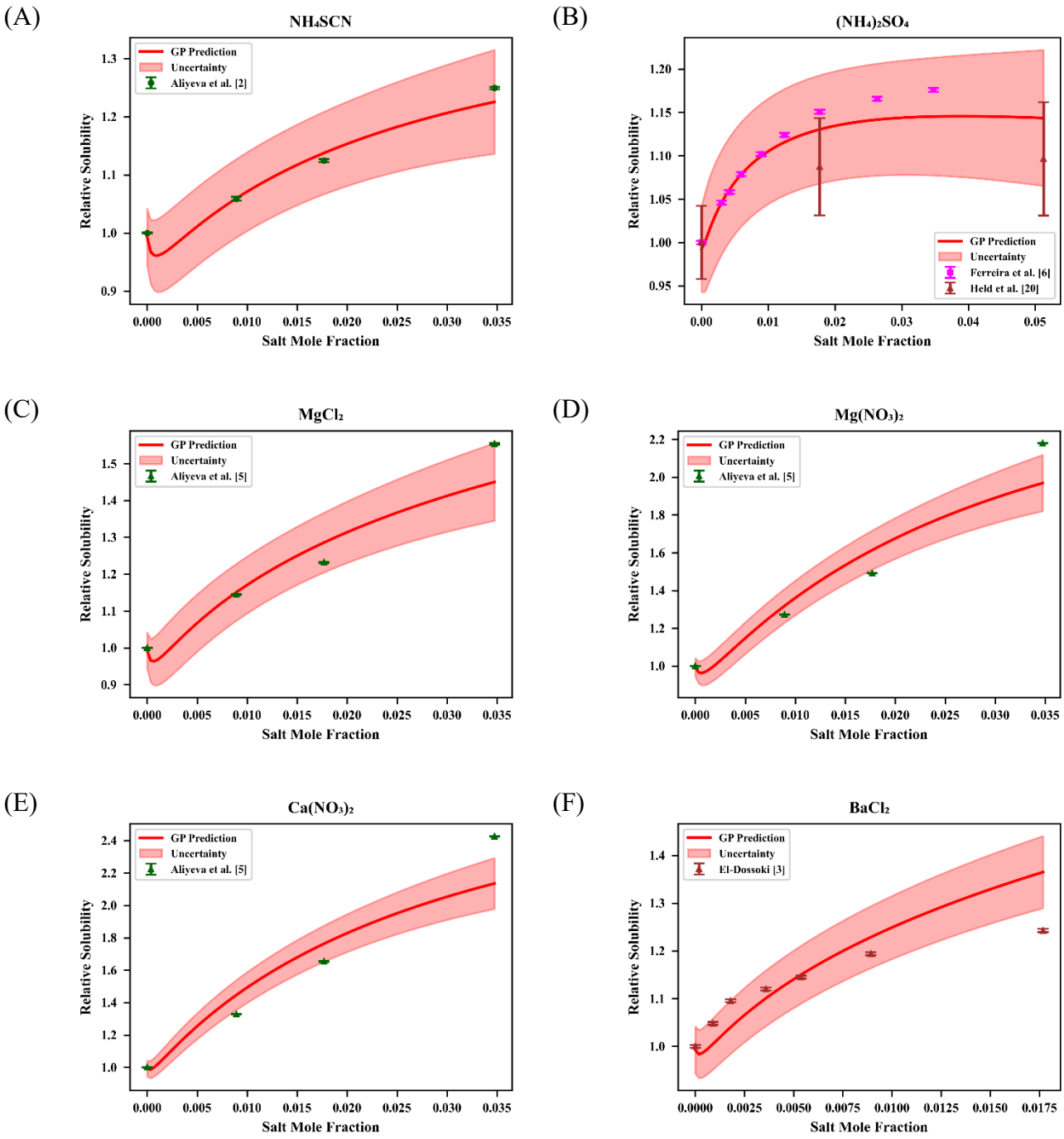


**Fig. A6**. GP model prediction (red line) for the relative solubility of glycine in $NH_4SCN$ (A), $(NH_4)_2SO_4$ (B), $MgCl_2$ (C), $Mg(NO_3)_2$ (D), $Ca(NO_3)_2$ (E), $BaCl_2$ (F) aqueous solutions at 298.2 K, and the associated prediction uncertainty.

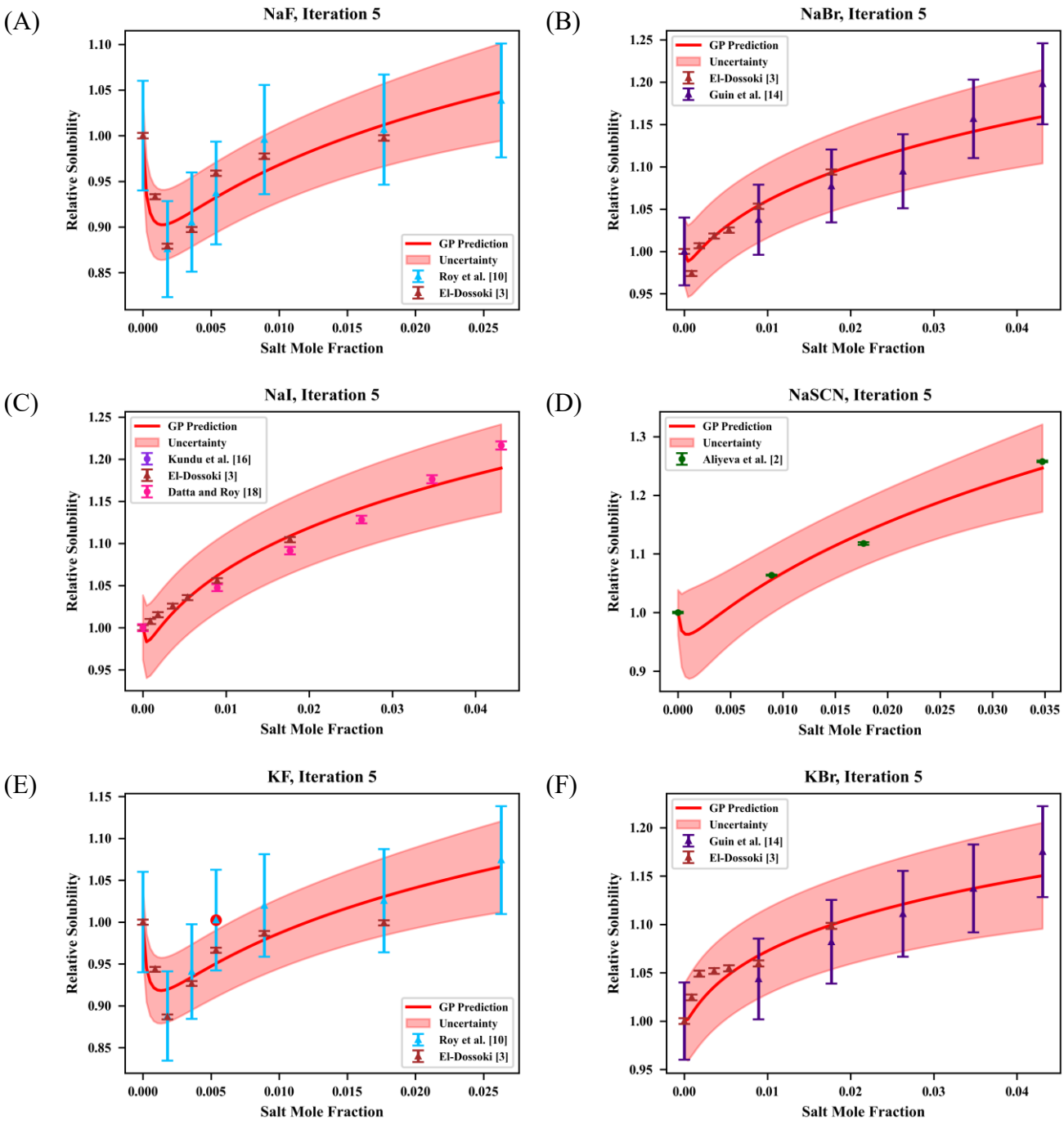


**Fig. A7.** GP prediction of glycine relative solubility in NaF (A), NaBr (B), NaI (C), NaSCN (D), KF (E), KBr (F) aqueous solutions with uncertainty bands after iterative removal of points outside the uncertainty interval.

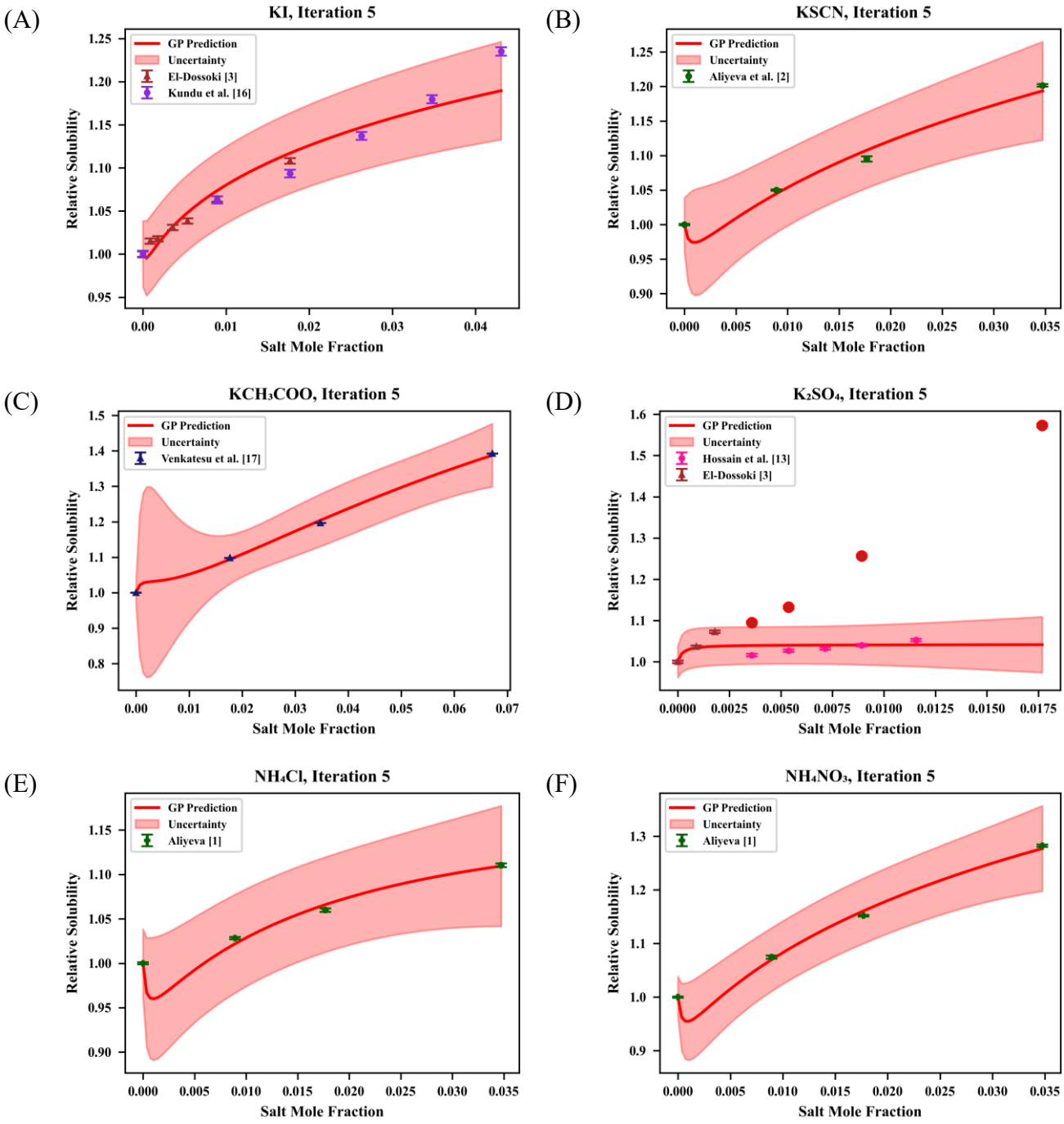


**Fig. A8.** GP prediction of glycine relative solubility in KI (A), KSCN (B), $KCH_3COO$ (C), $K_2SO_4$ (D), $NH_4Cl$ (E), $NH_4NO_3$ (F) aqueous solutions with uncertainty bands after iterative removal of points outside the uncertainty interval.

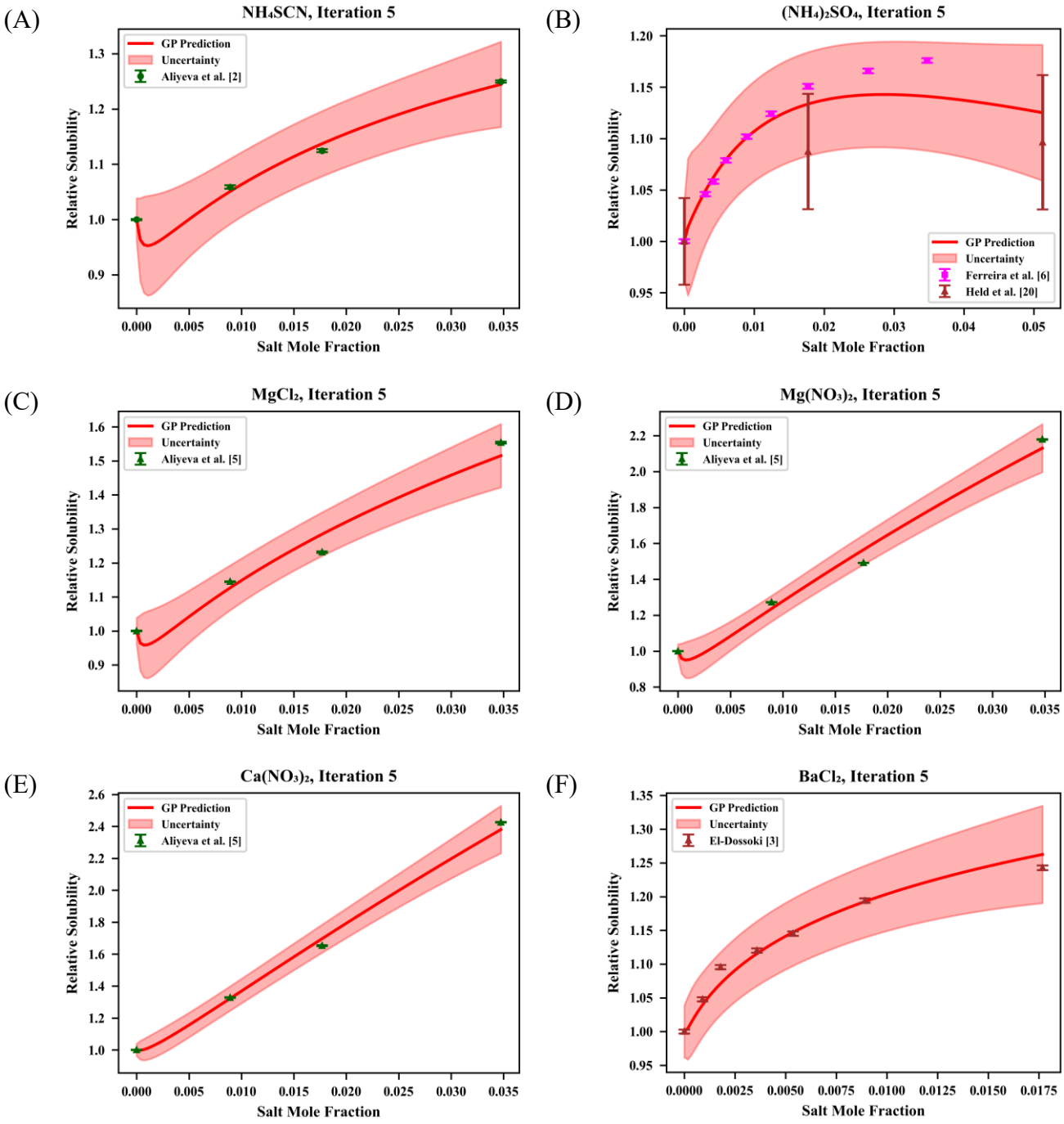


**Fig. A9.** GP prediction of glycine relative solubility in $NH_4SCN$ (A), $(NH_4)_2SO_4$ (B), $MgCl_2$ (C), $Mg(NO_3)_2$ (D), $Ca(NO_3)_2$ (E), $BaCl_2$ (F) aqueous solutions with uncertainty bands after iterative removal of points outside the uncertainty interval.

**Tab. A1.** Number of experimental data points removed from GP model training for each electrolyte system after applying the iterative filtering method with a 99.5% confidence interval.

| Salt | Number of Data | Number of Removed Data |
|---|---|---|
| NaF | 14 | 0 |
| NaCl | 16 | 2 |
| NaBr | 13 | 0 |
| NaI | 19 | 0 |
| $NaNO_3$ | 19 | 6 |
| NaSCN | 4 | 0 |
| $Na_2SO_4$ | 22 | 3 |
| KF | 14 | 1 |
| KCl | 21 | 0 |
| KBr | 13 | 0 |
| KI | 13 | 0 |
| $KNO_3$ | 19 | 5 |
| KSCN | 4 | 0 |
| $KCH_3COO$ | 4 | 0 |
| $K_2SO_4$ | 13 | 4 |
| $NH_4Cl$ | 4 | 0 |
| $NH_4NO_3$ | 4 | 0 |
| $NH_4SCN$ | 4 | 0 |
| $(NH_4)_2SO_4$ | 12 | 0 |
| $MgCl_2$ | 4 | 0 |

| | | |
|---|---|---|
| $Mg(NO_3)_2$ | 4 | 0 |
| $CaCl_2$ | 11 | 5 |
| $Ca(NO_3)_2$ | 4 | 0 |
| $BaCl_2$ | 7 | 0 |

## B. Database of glycine solubility in aqueous electrolyte systems

**Tab. B1.** Solubility data of glycine in different aqueous salt solutions at 298.2 K found in the open literature.

| Salt | Molality | Molar Fraction | Solubility (g/1000g $H_20$) | $S/S_0$ | Relative Uncertainty | Uncertainty S/S0 |
|---|---|---|---|---|---|---|
| $KNO_3$ | 0.0000 | 0.0000 | 250.21 | 1.0000 | 0.6006 | 0.0048 |
| $KNO_3$ | 0.0500 | 0.0009 | 242.40 | 0.9688 | 0.6006 | 0.0047 |
| $KNO_3$ | 0.1000 | 0.0018 | 240.37 | 0.9607 | 0.6006 | 0.0047 |
| $KNO_3$ | 0.2000 | 0.0036 | 268.38 | 1.0726 | 0.6006 | 0.0050 |
| $KNO_3$ | 0.3000 | 0.0054 | 277.53 | 1.1092 | 0.6006 | 0.0051 |
| $KNO_3$ | 0.5000 | 0.0089 | 305.23 | 1.2199 | 0.6006 | 0.0053 |
| $KNO_3$ | 1.0000 | 0.0177 | 389.91 | 1.5584 | 0.6006 | 0.0061 |
| $KNO_3$ | 1.5000 | 0.0263 | 470.16 | 1.8791 | 0.6006 | 0.0069 |
| $KNO_3$ | 0.0000 | 0.0000 | 238.33 | 1.0000 | 0.1270 | 0.0011 |
| $KNO_3$ | 0.5000 | 0.0089 | 253.18 | 1.0623 | 0.2810 | 0.0017 |
| $KNO_3$ | 1.0000 | 0.0177 | 265.95 | 1.1159 | 0.2700 | 0.0017 |
| $KNO_3$ | 2.0000 | 0.0347 | 284.80 | 1.1950 | 0.3100 | 0.0019 |
| $KNO_3$ | 0.0000 | 0.0000 | 250.66 | 1.0000 | 0.9008 | 0.0072 |
| $KNO_3$ | 0.5000 | 0.0089 | 261.39 | 1.0428 | 0.9008 | 0.0073 |
| $KNO_3$ | 1.0000 | 0.0177 | 270.70 | 1.0800 | 0.9008 | 0.0075 |
| $KNO_3$ | 1.5000 | 0.0263 | 277.91 | 1.1087 | 0.9008 | 0.0076 |
| $KNO_3$ | 2.0000 | 0.0347 | 279.26 | 1.1141 | 0.9008 | 0.0076 |
| $KNO_3$ | 2.5000 | 0.0431 | 292.17 | 1.1656 | 0.9008 | 0.0078 |
| $KNO_3$ | 3.0000 | 0.0512 | 301.48 | 1.2028 | 0.9008 | 0.0079 |

| | | | | | | |
|---|---|---|---|---|---|---|
| KCl | 0.0000 | 0.0000 | 238.33 | 1.0000 | 0.1270 | 0.0011 |
| KCl | 0.5000 | 0.0089 | 243.17 | 1.0203 | 0.1080 | 0.0010 |
| KCl | 1.0000 | 0.0177 | 247.23 | 1.0373 | 0.2400 | 0.0016 |
| KCl | 2.0000 | 0.0347 | 252.28 | 1.0585 | 0.3700 | 0.0021 |
| KCl | 0.0000 | 0.0000 | 249.90 | 1.0000 | 0.3000 | 0.0024 |
| KCl | 0.0500 | 0.0009 | 244.90 | 0.9800 | 0.3000 | 0.0024 |
| KCl | 0.1000 | 0.0018 | 242.90 | 0.9720 | 0.3000 | 0.0024 |
| KCl | 0.2000 | 0.0036 | 242.90 | 0.9720 | 0.3000 | 0.0024 |
| KCl | 0.3000 | 0.0054 | 245.00 | 0.9804 | 0.3000 | 0.0024 |
| KCl | 0.5000 | 0.0089 | 249.90 | 1.0000 | 0.3000 | 0.0024 |
| KCl | 0.7000 | 0.0124 | 255.10 | 1.0208 | 0.3000 | 0.0024 |
| KCl | 1.0000 | 0.0177 | 262.20 | 1.0492 | 0.3000 | 0.0025 |
| KCl | 1.5000 | 0.0263 | 273.80 | 1.0956 | 0.3000 | 0.0025 |
| KCl | 0.0000 | 0.0000 | 235.70 | 1.0000 | 0.2357 | 0.0020 |
| KCl | 0.1000 | 0.0018 | 238.03 | 1.0099 | 0.2380 | 0.0020 |
| KCl | 0.3000 | 0.0054 | 240.00 | 1.0182 | 0.2400 | 0.0020 |
| KCl | 0.5000 | 0.0089 | 242.01 | 1.0268 | 0.2420 | 0.0021 |
| KCl | 0.7000 | 0.0124 | 243.63 | 1.0336 | 0.2436 | 0.0021 |
| KCl | 1.0000 | 0.0177 | 246.09 | 1.0441 | 0.2461 | 0.0021 |
| KCl | 1.5000 | 0.0263 | 248.35 | 1.0537 | 0.2484 | 0.0021 |
| KCl | 2.0000 | 0.0347 | 249.40 | 1.0581 | 0.2494 | 0.0021 |
| KBr | 0.0000 | 0.0000 | 254.41 | 1.0000 | 5.0882 | 0.0400 |
| KBr | 0.5000 | 0.0089 | 265.52 | 1.0437 | 5.3105 | 0.0417 |
| KBr | 1.0000 | 0.0177 | 275.28 | 1.0820 | 5.5056 | 0.0433 |
| KBr | 1.5000 | 0.0263 | 282.64 | 1.1109 | 5.6528 | 0.0444 |

| | | | | | | |
|---|---|---|---|---|---|---|
| KBr | 2.0000 | 0.0347 | 289.32 | 1.1372 | 5.7864 | 0.0455 |
| KBr | 2.5000 | 0.0431 | 299.00 | 1.1753 | 5.9801 | 0.0470 |
| KBr | 0.0000 | 0.0000 | 250.58 | 1.0000 | 0.3754 | 0.0030 |
| KBr | 0.0500 | 0.0009 | 256.74 | 1.0246 | 0.3754 | 0.0030 |
| KBr | 0.1000 | 0.0018 | 262.90 | 1.0491 | 0.3754 | 0.0031 |
| KBr | 0.2000 | 0.0036 | 263.57 | 1.0518 | 0.3754 | 0.0031 |
| KBr | 0.3000 | 0.0054 | 264.25 | 1.0545 | 0.3754 | 0.0031 |
| KBr | 0.5000 | 0.0089 | 265.52 | 1.0596 | 0.3754 | 0.0031 |
| KBr | 1.0000 | 0.0177 | 275.28 | 1.0986 | 0.3754 | 0.0031 |
| KI | 0.0000 | 0.0000 | 250.58 | 1.0000 | 0.3754 | 0.0030 |
| KI | 0.0500 | 0.0009 | 254.34 | 1.0150 | 0.3754 | 0.0030 |
| KI | 0.1000 | 0.0018 | 255.01 | 1.0177 | 0.3754 | 0.0030 |
| KI | 0.2000 | 0.0036 | 258.32 | 1.0309 | 0.3754 | 0.0030 |
| KI | 0.3000 | 0.0054 | 260.19 | 1.0383 | 0.3754 | 0.0031 |
| KI | 0.5000 | 0.0089 | 266.57 | 1.0638 | 0.3754 | 0.0031 |
| KI | 1.0000 | 0.0177 | 277.68 | 1.1081 | 0.3754 | 0.0032 |
| KI | 0.0000 | 0.0000 | 254.41 | 1.0000 | 0.5088 | 0.0040 |
| KI | 0.5000 | 0.0089 | 270.40 | 1.0629 | 0.5408 | 0.0043 |
| KI | 1.0000 | 0.0177 | 278.21 | 1.0935 | 0.5564 | 0.0044 |
| KI | 1.5000 | 0.0263 | 289.32 | 1.1372 | 0.5786 | 0.0045 |
| KI | 2.0000 | 0.0347 | 300.13 | 1.1797 | 0.6003 | 0.0047 |
| KI | 2.5000 | 0.0431 | 314.24 | 1.2352 | 0.6285 | 0.0049 |
| $K_2SO_4$ | 0.0000 | 0.0000 | 250.13 | 1.0000 | 0.3754 | 0.0030 |
| $K_2SO_4$ | 0.2000 | 0.0036 | 254.19 | 1.0162 | 0.3754 | 0.0030 |
| $K_2SO_4$ | 0.3000 | 0.0054 | 256.89 | 1.0270 | 0.3754 | 0.0030 |

| | | | | | | |
|---|---|---|---|---|---|---|
| $K_2SO_4$ | 0.4000 | 0.0071 | 258.24 | 1.0324 | 0.3754 | 0.0030 |
| $K_2SO_4$ | 0.5000 | 0.0089 | 260.19 | 1.0402 | 0.3754 | 0.0031 |
| $K_2SO_4$ | 0.6500 | 0.0116 | 263.35 | 1.0528 | 0.3754 | 0.0031 |
| $K_2SO_4$ | 0.0000 | 0.0000 | 250.58 | 1.0000 | 0.3754 | 0.0030 |
| $K_2SO_4$ | 0.0500 | 0.0009 | 259.74 | 1.0365 | 0.3754 | 0.0031 |
| $K_2SO_4$ | 0.1000 | 0.0018 | 268.83 | 1.0728 | 0.3754 | 0.0031 |
| $K_2SO_4$ | 0.2000 | 0.0036 | 274.46 | 1.0953 | 0.3754 | 0.0031 |
| $K_2SO_4$ | 0.3000 | 0.0054 | 283.84 | 1.1327 | 0.3754 | 0.0032 |
| $K_2SO_4$ | 0.5000 | 0.0089 | 314.92 | 1.2567 | 0.3754 | 0.0034 |
| $K_2SO_4$ | 1.0000 | 0.0177 | 394.27 | 1.5734 | 0.3754 | 0.0039 |
| KF | 0.0000 | 0.0000 | 249.23 | 1.0000 | 7.4770 | 0.0600 |
| KF | 0.1000 | 0.0018 | 221.31 | 0.8880 | 6.6392 | 0.0533 |
| KF | 0.2000 | 0.0036 | 234.52 | 0.9410 | 7.0356 | 0.0565 |
| KF | 0.3000 | 0.0054 | 249.83 | 1.0024 | 7.4950 | 0.0601 |
| KF | 0.5000 | 0.0089 | 254.19 | 1.0199 | 7.6256 | 0.0612 |
| KF | 1.0000 | 0.0177 | 255.61 | 1.0256 | 7.6684 | 0.0615 |
| KF | 1.5000 | 0.0263 | 267.70 | 1.0741 | 8.0310 | 0.0644 |
| KF | 0.0000 | 0.0000 | 250.58 | 1.0000 | 0.3754 | 0.0030 |
| KF | 0.0500 | 0.0009 | 236.47 | 0.9437 | 0.3754 | 0.0029 |
| KF | 0.1000 | 0.0018 | 222.21 | 0.8868 | 0.3754 | 0.0028 |
| KF | 0.2000 | 0.0036 | 232.19 | 0.9266 | 0.3754 | 0.0029 |
| KF | 0.3000 | 0.0054 | 242.18 | 0.9664 | 0.3754 | 0.0029 |
| KF | 0.5000 | 0.0089 | 247.21 | 0.9865 | 0.3754 | 0.0030 |
| KF | 1.0000 | 0.0177 | 250.36 | 0.9991 | 0.3754 | 0.0030 |
| KSCN | 0.0000 | 0.0000 | 238.33 | 1.0000 | 0.1300 | 0.0011 |

| | | | | | | |
|---|---|---|---|---|---|---|
| KSCN | 0.5000 | 0.0089 | 250.24 | 1.0500 | 0.0900 | 0.0010 |
| KSCN | 1.0000 | 0.0177 | 261.00 | 1.0951 | 0.7700 | 0.0038 |
| KSCN | 2.0000 | 0.0347 | 286.38 | 1.2016 | 0.2800 | 0.0018 |
| $KCH_3COO$ | 0.0000 | 0.0000 | 250.90 | 1.0000 | 0.0000 | 0.0000 |
| $KCH_3COO$ | 1.0000 | 0.0177 | 275.53 | 1.0982 | 0.0000 | 0.0000 |
| $KCH_3COO$ | 2.0000 | 0.0347 | 300.15 | 1.1963 | 0.0000 | 0.0000 |
| $KCH_3COO$ | 4.0000 | 0.0672 | 349.40 | 1.3926 | 0.0000 | 0.0000 |
| $NaNO_3$ | 0.0000 | 0.0000 | 238.33 | 1.0000 | 0.1270 | 0.0011 |
| $NaNO_3$ | 0.5000 | 0.0089 | 256.83 | 1.0776 | 0.4290 | 0.0024 |
| $NaNO_3$ | 1.0000 | 0.0177 | 278.03 | 1.1666 | 0.3290 | 0.0020 |
| $NaNO_3$ | 2.0000 | 0.0347 | 305.87 | 1.2834 | 0.4930 | 0.0028 |
| $NaNO_3$ | 0.0000 | 0.0000 | 250.21 | 1.0000 | 0.4504 | 0.0036 |
| $NaNO_3$ | 0.0500 | 0.0009 | 242.40 | 0.9688 | 0.4504 | 0.0035 |
| $NaNO_3$ | 0.1000 | 0.0018 | 240.30 | 0.9604 | 0.4504 | 0.0035 |
| $NaNO_3$ | 0.2000 | 0.0036 | 267.10 | 1.0675 | 0.4504 | 0.0037 |
| $NaNO_3$ | 0.3000 | 0.0054 | 278.21 | 1.1119 | 0.4504 | 0.0038 |
| $NaNO_3$ | 0.5000 | 0.0089 | 302.83 | 1.2103 | 0.4504 | 0.0040 |
| $NaNO_3$ | 1.0000 | 0.0177 | 382.63 | 1.5293 | 0.4504 | 0.0046 |
| $NaNO_3$ | 1.5000 | 0.0263 | 464.23 | 1.8554 | 0.4504 | 0.0051 |
| $NaNO_3$ | 0.0000 | 0.0000 | 250.66 | 1.0000 | 0.9008 | 0.0072 |
| $NaNO_3$ | 0.5000 | 0.0089 | 265.75 | 1.0602 | 0.9008 | 0.0074 |
| $NaNO_3$ | 1.0000 | 0.0177 | 278.06 | 1.1093 | 0.9008 | 0.0076 |
| $NaNO_3$ | 1.5000 | 0.0263 | 282.41 | 1.1267 | 0.9008 | 0.0076 |
| $NaNO_3$ | 2.0000 | 0.0347 | 296.75 | 1.1839 | 0.9008 | 0.0078 |
| $NaNO_3$ | 2.5000 | 0.0431 | 309.14 | 1.2333 | 0.9008 | 0.0080 |

| | | | | | | |
|---|---|---|---|---|---|---|
| $NaNO_3$ | 3.0000 | 0.0512 | 330.61 | 1.3190 | 0.9008 | 0.0083 |
| NaCl | 0.0000 | 0.0000 | 238.33 | 1.0000 | 0.1270 | 0.0011 |
| NaCl | 0.5000 | 0.0089 | 244.62 | 1.0264 | 0.8740 | 0.0042 |
| NaCl | 1.0000 | 0.0177 | 252.85 | 1.0609 | 0.4520 | 0.0025 |
| NaCl | 2.0000 | 0.0347 | 263.86 | 1.1071 | 0.2290 | 0.0016 |
| NaCl | 0.0000 | 0.0000 | 205.77 | 1.0000 | 0.0375 | 0.0004 |
| NaCl | 1.0000 | 0.0177 | 204.19 | 0.9923 | 0.0375 | 0.0004 |
| NaCl | 3.0000 | 0.0512 | 199.01 | 0.9672 | 0.0375 | 0.0004 |
| NaCl | 0.0000 | 0.0000 | 249.90 | 1.0000 | 0.7000 | 0.0056 |
| NaCl | 0.0500 | 0.0009 | 242.50 | 0.9704 | 0.7000 | 0.0055 |
| NaCl | 0.1000 | 0.0018 | 240.30 | 0.9616 | 0.7000 | 0.0055 |
| NaCl | 0.2000 | 0.0036 | 241.00 | 0.9644 | 0.7000 | 0.0055 |
| NaCl | 0.3000 | 0.0054 | 243.20 | 0.9732 | 0.7000 | 0.0055 |
| NaCl | 0.5000 | 0.0089 | 244.70 | 0.9792 | 0.7000 | 0.0055 |
| NaCl | 0.7000 | 0.0124 | 247.10 | 0.9888 | 0.7000 | 0.0056 |
| NaCl | 1.0000 | 0.0177 | 252.80 | 1.0116 | 0.7000 | 0.0056 |
| NaCl | 1.5000 | 0.0263 | 262.00 | 1.0484 | 0.7000 | 0.0057 |
| NaI | 0.0000 | 0.0000 | 254.41 | 1.0000 | 0.5088 | 0.0040 |
| NaI | 0.5000 | 0.0089 | 266.50 | 1.0475 | 0.5330 | 0.0042 |
| NaI | 1.0000 | 0.0177 | 277.68 | 1.0915 | 0.5554 | 0.0044 |
| NaI | 1.5000 | 0.0263 | 287.07 | 1.1284 | 0.5741 | 0.0045 |
| NaI | 2.0000 | 0.0347 | 299.23 | 1.1762 | 0.5985 | 0.0047 |
| NaI | 2.5000 | 0.0431 | 309.44 | 1.2163 | 0.6189 | 0.0049 |
| NaI | 0.0000 | 0.0000 | 250.58 | 1.0000 | 0.3754 | 0.0030 |
| NaI | 0.0500 | 0.0009 | 252.46 | 1.0075 | 0.3754 | 0.0030 |

| | | | | | | |
|---|---|---|---|---|---|---|
| NaI | 0.1000 | 0.0018 | 254.41 | 1.0153 | 0.3754 | 0.0030 |
| NaI | 0.2000 | 0.0036 | 256.96 | 1.0255 | 0.3754 | 0.0030 |
| NaI | 0.3000 | 0.0054 | 259.52 | 1.0357 | 0.3754 | 0.0030 |
| NaI | 0.5000 | 0.0089 | 264.55 | 1.0557 | 0.3754 | 0.0031 |
| NaI | 1.0000 | 0.0177 | 276.78 | 1.1046 | 0.3754 | 0.0032 |
| NaI | 0.0000 | 0.0000 | 254.41 | 1.0000 | 0.5088 | 0.0040 |
| NaI | 0.5000 | 0.0089 | 266.50 | 1.0475 | 0.5330 | 0.0042 |
| NaI | 1.0000 | 0.0177 | 277.68 | 1.0915 | 0.5554 | 0.0044 |
| NaI | 1.5000 | 0.0263 | 287.07 | 1.1284 | 0.5741 | 0.0045 |
| NaI | 2.0000 | 0.0347 | 299.23 | 1.1762 | 0.5985 | 0.0047 |
| NaI | 2.5000 | 0.0431 | 309.44 | 1.2163 | 0.6189 | 0.0049 |
| $Na_2SO_4$ | 0.0000 | 0.0000 | 235.70 | 1.0000 | 0.2357 | 0.0020 |
| $Na_2SO_4$ | 0.5000 | 0.0089 | 261.16 | 1.1080 | 0.2612 | 0.0022 |
| $Na_2SO_4$ | 1.0000 | 0.0177 | 270.24 | 1.1465 | 0.2702 | 0.0023 |
| $Na_2SO_4$ | 1.5000 | 0.0263 | 272.87 | 1.1577 | 0.2729 | 0.0023 |
| $Na_2SO_4$ | 0.0000 | 0.0000 | 250.13 | 1.0000 | 7.0037 | 0.0560 |
| $Na_2SO_4$ | 0.1000 | 0.0018 | 254.64 | 1.0180 | 7.1298 | 0.0570 |
| $Na_2SO_4$ | 0.3000 | 0.0054 | 258.84 | 1.0348 | 7.2476 | 0.0579 |
| $Na_2SO_4$ | 0.5000 | 0.0089 | 263.05 | 1.0516 | 7.3653 | 0.0589 |
| $Na_2SO_4$ | 0.7000 | 0.0124 | 265.82 | 1.0627 | 7.4430 | 0.0595 |
| $Na_2SO_4$ | 1.0000 | 0.0177 | 274.76 | 1.0984 | 7.6932 | 0.0615 |
| $Na_2SO_4$ | 1.5000 | 0.0263 | 277.91 | 1.1110 | 7.7815 | 0.0622 |
| $Na_2SO_4$ | 0.0000 | 0.0000 | 250.20 | 1.0000 | 3.0024 | 0.0240 |
| $Na_2SO_4$ | 0.5000 | 0.0089 | 264.11 | 1.0556 | 3.1694 | 0.0253 |
| $Na_2SO_4$ | 1.0000 | 0.0177 | 278.43 | 1.1128 | 3.3412 | 0.0267 |

| | | | | | | |
|---|---|---|---|---|---|---|
| $Na_2SO_4$ | 1.5000 | 0.0263 | 277.06 | 1.1074 | 3.3248 | 0.0266 |
| $Na_2SO_4$ | 0.0000 | 0.0000 | 250.58 | 1.0000 | 0.3754 | 0.0030 |
| $Na_2SO_4$ | 0.0500 | 0.0009 | 259.07 | 1.0339 | 0.3754 | 0.0030 |
| $Na_2SO_4$ | 0.1000 | 0.0018 | 267.25 | 1.0665 | 0.3754 | 0.0031 |
| $Na_2SO_4$ | 0.2000 | 0.0036 | 272.88 | 1.0890 | 0.3754 | 0.0031 |
| $Na_2SO_4$ | 0.3000 | 0.0054 | 281.51 | 1.1234 | 0.3754 | 0.0032 |
| $Na_2SO_4$ | 0.5000 | 0.0089 | 308.61 | 1.2316 | 0.3754 | 0.0033 |
| $Na_2SO_4$ | 1.0000 | 0.0177 | 391.26 | 1.5614 | 0.3754 | 0.0038 |
| NaF | 0.0000 | 0.0000 | 249.23 | 1.0000 | 7.4770 | 0.0600 |
| NaF | 0.1000 | 0.0018 | 218.30 | 0.8759 | 6.5491 | 0.0526 |
| NaF | 0.2000 | 0.0036 | 225.66 | 0.9054 | 6.7698 | 0.0543 |
| NaF | 0.3000 | 0.0054 | 233.62 | 0.9373 | 7.0085 | 0.0562 |
| NaF | 0.5000 | 0.0089 | 248.18 | 0.9958 | 7.4454 | 0.0597 |
| NaF | 1.0000 | 0.0177 | 250.88 | 1.0066 | 7.5265 | 0.0604 |
| NaF | 1.5000 | 0.0263 | 258.84 | 1.0386 | 7.7652 | 0.0623 |
| NaF | 0.0000 | 0.0000 | 250.58 | 1.0000 | 0.3754 | 0.0030 |
| NaF | 0.0500 | 0.0009 | 233.84 | 0.9332 | 0.3754 | 0.0029 |
| NaF | 0.1000 | 0.0018 | 220.26 | 0.8790 | 0.3754 | 0.0028 |
| NaF | 0.2000 | 0.0036 | 224.83 | 0.8972 | 0.3754 | 0.0028 |
| NaF | 0.3000 | 0.0054 | 240.30 | 0.9590 | 0.3754 | 0.0029 |
| NaF | 0.5000 | 0.0089 | 244.95 | 0.9775 | 0.3754 | 0.0030 |
| NaF | 1.0000 | 0.0177 | 249.98 | 0.9976 | 0.3754 | 0.0030 |
| NaBr | 0.0000 | 0.0000 | 250.58 | 1.0000 | 0.3754 | 0.0030 |
| NaBr | 0.0500 | 0.0009 | 244.05 | 0.9739 | 0.3754 | 0.0030 |
| NaBr | 0.1000 | 0.0018 | 252.24 | 1.0066 | 0.3754 | 0.0030 |

| | | | | | | |
|---|---|---|---|---|---|---|
| NaBr | 0.2000 | 0.0036 | 255.16 | 1.0183 | 0.3754 | 0.0030 |
| NaBr | 0.3000 | 0.0054 | 256.89 | 1.0252 | 0.3754 | 0.0030 |
| NaBr | 0.5000 | 0.0089 | 263.95 | 1.0533 | 0.3754 | 0.0031 |
| NaBr | 1.0000 | 0.0177 | 274.08 | 1.0938 | 0.3754 | 0.0031 |
| NaBr | 0.0000 | 0.0000 | 254.41 | 1.0000 | 5.0882 | 0.0400 |
| NaBr | 0.5000 | 0.0089 | 263.95 | 1.0375 | 5.2789 | 0.0415 |
| NaBr | 1.0000 | 0.0177 | 274.08 | 1.0773 | 5.4816 | 0.0431 |
| NaBr | 1.5000 | 0.0263 | 278.51 | 1.0947 | 5.5702 | 0.0438 |
| NaBr | 2.0000 | 0.0347 | 294.27 | 1.1567 | 5.8855 | 0.0463 |
| NaBr | 2.5000 | 0.0431 | 304.78 | 1.1980 | 6.0957 | 0.0479 |
| NaSCN | 0.0000 | 0.0000 | 238.33 | 1.0000 | 0.1300 | 0.0011 |
| NaSCN | 0.5000 | 0.0089 | 253.51 | 1.0637 | 0.0400 | 0.0007 |
| NaSCN | 1.0000 | 0.0177 | 266.40 | 1.1178 | 0.3800 | 0.0022 |
| NaSCN | 2.0000 | 0.0347 | 299.76 | 1.2578 | 0.1800 | 0.0014 |
| $(NH_4)_2SO_4$ | 0.0000 | 0.0000 | 235.70 | 1.0000 | 0.2357 | 0.0020 |
| $(NH_4)_2SO_4$ | 0.1670 | 0.0030 | 246.52 | 1.0459 | 0.2465 | 0.0021 |
| $(NH_4)_2SO_4$ | 0.2330 | 0.0042 | 249.40 | 1.0581 | 0.2494 | 0.0021 |
| $(NH_4)_2SO_4$ | 0.3330 | 0.0060 | 254.26 | 1.0787 | 0.2543 | 0.0022 |
| $(NH_4)_2SO_4$ | 0.5000 | 0.0089 | 259.71 | 1.1019 | 0.2597 | 0.0022 |
| $(NH_4)_2SO_4$ | 0.7000 | 0.0124 | 264.95 | 1.1241 | 0.2650 | 0.0022 |
| $(NH_4)_2SO_4$ | 1.0000 | 0.0177 | 271.24 | 1.1508 | 0.2712 | 0.0023 |
| $(NH_4)_2SO_4$ | 1.5000 | 0.0263 | 274.77 | 1.1658 | 0.2748 | 0.0023 |
| $(NH_4)_2SO_4$ | 2.0000 | 0.0347 | 277.17 | 1.1759 | 0.2772 | 0.0024 |
| $(NH_4)_2SO_4$ | 0.0000 | 0.0000 | 249.23 | 1.0000 | 5.2549 | 0.0422 |
| $(NH_4)_2SO_4$ | 1.0000 | 0.0177 | 271.00 | 1.0873 | 8.2577 | 0.0561 |

| | | | | | | |
|---|---|---|---|---|---|---|
| $(NH_4)_2SO_4$ | 3.0000 | 0.0512 | 273.25 | 1.0964 | 10.5098 | 0.0653 |
| $NH_4SCN$ | 0.0000 | 0.0000 | 238.33 | 1.0000 | 0.1300 | 0.0011 |
| $NH_4SCN$ | 0.5000 | 0.0089 | 252.36 | 1.0589 | 0.6200 | 0.0032 |
| $NH_4SCN$ | 1.0000 | 0.0177 | 268.02 | 1.1246 | 0.5000 | 0.0027 |
| $NH_4SCN$ | 2.0000 | 0.0347 | 297.83 | 1.2497 | 0.3500 | 0.0022 |
| $NH_4NO_3$ | 0.0000 | 0.0000 | 238.33 | 1.0000 | 0.1270 | 0.0011 |
| $NH_4NO_3$ | 0.5000 | 0.0089 | 256.11 | 1.0746 | 0.6060 | 0.0031 |
| $NH_4NO_3$ | 1.0000 | 0.0177 | 274.53 | 1.1519 | 0.2090 | 0.0015 |
| $NH_4NO_3$ | 2.0000 | 0.0347 | 305.61 | 1.2823 | 0.3550 | 0.0022 |
| $NH_4Cl$ | 0.0000 | 0.0000 | 238.33 | 1.0000 | 0.1270 | 0.0011 |
| $NH_4Cl$ | 0.5000 | 0.0089 | 245.10 | 1.0284 | 0.1990 | 0.0014 |
| $NH_4Cl$ | 1.0000 | 0.0177 | 252.60 | 1.0599 | 0.3420 | 0.0020 |
| $NH_4Cl$ | 2.0000 | 0.0347 | 264.66 | 1.1105 | 0.3640 | 0.0021 |
| $MgCl_2$ | 0.0000 | 0.0000 | 238.33 | 1.0000 | 0.1270 | 0.0011 |
| $MgCl_2$ | 0.5000 | 0.0089 | 272.84 | 1.1448 | 0.1430 | 0.0012 |
| $MgCl_2$ | 1.0000 | 0.0177 | 293.52 | 1.2315 | 0.3400 | 0.0021 |
| $MgCl_2$ | 2.0000 | 0.0347 | 370.32 | 1.5538 | 0.3940 | 0.0025 |
| $Mg(NO_3)_2$ | 0.0000 | 0.0000 | 238.33 | 1.0000 | 0.1270 | 0.0011 |
| $Mg(NO_3)_2$ | 0.5000 | 0.0089 | 303.19 | 1.2721 | 0.1680 | 0.0014 |
| $Mg(NO_3)_2$ | 1.0000 | 0.0177 | 355.42 | 1.4913 | 0.0520 | 0.0010 |
| $Mg(NO_3)_2$ | 2.0000 | 0.0347 | 519.40 | 2.1793 | 0.1530 | 0.0018 |
| $CaCl_2$ | 0.0000 | 0.0000 | 238.33 | 1.0000 | 0.1270 | 0.0011 |
| $CaCl_2$ | 0.5000 | 0.0089 | 294.98 | 1.2377 | 0.9850 | 0.0048 |
| $CaCl_2$ | 1.0000 | 0.0177 | 354.15 | 1.4859 | 0.7000 | 0.0037 |
| $CaCl_2$ | 2.0000 | 0.0347 | 489.82 | 2.0552 | 0.6500 | 0.0038 |

| | | | | | | |
|---|---|---|---|---|---|---|
| $CaCl_2$ | 0.0000 | 0.0000 | 250.58 | 1.0000 | 0.3754 | 0.0030 |
| $CaCl_2$ | 0.0500 | 0.0009 | 258.54 | 1.0318 | 0.3754 | 0.0030 |
| $CaCl_2$ | 0.1000 | 0.0018 | 266.72 | 1.0644 | 0.3754 | 0.0031 |
| $CaCl_2$ | 0.2000 | 0.0036 | 269.73 | 1.0764 | 0.3754 | 0.0031 |
| $CaCl_2$ | 0.3000 | 0.0054 | 272.65 | 1.0881 | 0.3754 | 0.0031 |
| $CaCl_2$ | 0.5000 | 0.0089 | 278.96 | 1.1132 | 0.3754 | 0.0032 |
| $CaCl_2$ | 1.0000 | 0.0177 | 285.27 | 1.1384 | 0.3754 | 0.0032 |
| $Ca(NO_3)_2$ | 0.0000 | 0.0000 | 238.33 | 1.0000 | 0.1270 | 0.0011 |
| $Ca(NO_3)_2$ | 0.5000 | 0.0089 | 316.46 | 1.3278 | 0.3120 | 0.0020 |
| $Ca(NO_3)_2$ | 1.0000 | 0.0177 | 393.99 | 1.6531 | 0.3780 | 0.0025 |
| $Ca(NO_3)_2$ | 2.0000 | 0.0347 | 578.20 | 2.4260 | 0.0940 | 0.0017 |
| $BaCl_2$ | 0.0000 | 0.0000 | 250.58 | 1.0000 | 0.3754 | 0.0030 |
| $BaCl_2$ | 0.0500 | 0.0009 | 262.59 | 1.0479 | 0.3754 | 0.0031 |
| $BaCl_2$ | 0.1000 | 0.0018 | 274.53 | 1.0956 | 0.3754 | 0.0031 |
| $BaCl_2$ | 0.2000 | 0.0036 | 280.69 | 1.1201 | 0.3754 | 0.0032 |
| $BaCl_2$ | 0.3000 | 0.0054 | 286.95 | 1.1451 | 0.3754 | 0.0032 |
| $BaCl_2$ | 0.5000 | 0.0089 | 299.23 | 1.1941 | 0.3754 | 0.0033 |
| $BaCl_2$ | 1.0000 | 0.0177 | 311.39 | 1.2427 | 0.3754 | 0.0034 |

## C. Python code for Gaussian Process modeling of glycine solubility data

```
import os
import warnings
import time

# Specific
import numpy
import pandas
from sklearn import preprocessing, metrics
import gpflow
from matplotlib import pyplot as plt
import itertools

# ================================================================
# Configuration
# ================================================================

# Path to database folder
dbPath=r'C:/Users/win11/Desktop/DUPLA DIPLOMAÇÃO/Tese de mestrado/'
# Define normalization methods
featureNorm="Log+bStand" # None,Standardization,MinMax,LogStand,Log+bStand
labelNorm='LogStand' # None,Standardization,MinMax,LogStand,Log+bStand
# GP Configuration
gpConfig={'kernel':'RBF',
        'useWhiteKernel':True,
        'trainLikelihood':True}

# ================================================================
# Auxiliary Functions
# ================================================================

def normalize(inputArray,skScaler=None,method='Standardization',reverse=False):
```

```
"""
normalize() normalizes (or unnormalizes) inputArray using the method
specified and the skScaler provided.

Parameters
----------
inputArray : numpy array
    Array to be normalized. If dim>1, array is normalized column-wise.
skScaler : scikit-learn preprocessing object or None
    Scikit-learn preprocessing object previosly fitted to data. If None,
    the object is fitted to inputArray.
    Default: None
method : string, optional
    Normalization method to be used.
    Methods available:
        . Standardization - classic standardization, (x-mean(x))/std(x)
        . MinMax - scale to range (0,1)
        . LogStand - standardization on the log of the variable,
                    (log(x)-mean(log(x)))/std(log(x))
        . Log+bStand - standardization on the log of variables that can be
                      zero; uses a small buffer,
                      (log(x+b)-mean(log(x+b)))/std(log(x+b))
    Defalt: 'Standardization'
reverse : bool
    Whether  to normalize (False) or unnormalize (True) inputArray.
    Defalt: False

Returns
-------
inputArray : numpy array
    Normalized (or unnormalized) version of inputArray.
skScaler : scikit-learn preprocessing object
    Scikit-learn preprocessing object fitted to inputArray. It is the same
```

```
    as the inputted skScaler, if it was provided.

"""
# If inputArray is a labels vector of size (N,), reshape to (N,1)
if inputArray.ndim==1:
    inputArray=inputArray.reshape((-1,1))
    warnings.warn('Input to normalize() was of shape (N,). It was assumed'\
                +' to be a column array and converted to a (N,1) shape.')
# If skScaler is None, train for the first time
if skScaler is None:
    # Check method
    if method=='Standardization' or method=='MinMax': aux=inputArray
    elif method=='LogStand': aux=numpy.log(inputArray)
    elif method=='Log+bStand': aux=numpy.log(inputArray+10**-3)
    else: raise ValueError('Could not recognize method in normalize().')
    if method!='MinMax':
        skScaler=preprocessing.StandardScaler().fit(aux)
    else:
        skScaler=preprocessing.MinMaxScaler().fit(aux)
# Do main operation (normalize or unnormalize)
if reverse:
    # Rescale the data back to its original distribution
    inputArray=skScaler.inverse_transform(inputArray)
    # Check method
    if method=='LogStand': inputArray=numpy.exp(inputArray)
    elif method=='Log+bStand': inputArray=numpy.exp(inputArray)-10**-3
elif not reverse:
    # Check method
    if method=='Standardization' or method=='MinMax': aux=inputArray
    elif method=='LogStand': aux=numpy.log(inputArray)
    elif method=='Log+bStand': aux=numpy.log(inputArray+10**-3)
    else: raise ValueError('Could not recognize method in normalize().')
    inputArray=skScaler.transform(aux)
```

```
    # Return
    return inputArray,skScaler


def buildGP(X_Train,Y_Train, gpConfig={}):
    """
    buildGP() builds and fits a GP model using the training data provided.

    Parameters
    ----------
    X_Train : numpy array (N,K)
        Training features, where N is the number of data points and K is the
        number of independent features (e.g., sigma profile bins).
    Y_Train : numpy array (N,1)
        Training labels (e.g., property of a given molecule).
    gpConfig : dictionary, optional
        Dictionary containing the configuration of the GP. If a key is not
        present in the dictionary, its default value is used.
        Keys:
            . kernel : string
                Kernel to be used. One of:
                    . 'RBF' - gpflow.kernels.RBF()
                    . 'RQ' - gpflow.kernels.RationalQuadratic()
                    . 'Matern32' - gpflow.kernels.Matern32()
                    . 'Matern52' - gpflow.kernels.Matern52()
                The default is 'RQ'.
            . useWhiteKernel : boolean
                Whether to use a White kernel (gpflow.kernels.White).
                The default is True.
            . trainLikelihood : boolean
                Whether to treat the variance of the likelihood of the modeal
                as a trainable (or fitting) parameter. If False, this value is
                fixed at 10^-5.
                The default is True.
```

```
    The default is {}.
Raises
------
UserWarning
    Warning raised if the optimization (fitting) fails to converge.

Returns
-------
model : gpflow.models.gpr.GPR object
    GP model.

"""
# Unpack config
kernel = gpConfig.get('kernel', 'RBF')
useWhiteKernel = gpConfig.get('useWhiteKernel', True)

# Kernel
if kernel == 'RBF':
    gpkernel = gpflow.kernels.SquaredExponential()
elif kernel == 'RQ':
    gpkernel = gpflow.kernels.RationalQuadratic()
elif kernel == 'Matern32':
    gpkernel = gpflow.kernels.Matern32()
elif kernel == 'Matern52':
    gpkernel = gpflow.kernels.Matern52()
else:
    raise ValueError(f"Kernel {kernel} not recognized.")

if useWhiteKernel:
    gpkernel += gpflow.kernels.White()

likelihood = gpflow.likelihoods.Gaussian()
```

```
    model = gpflow.models.GPR(data=(X_Train, Y_Train), kernel=gpkernel, likelihood=likelihood)

    # Otimização
    opt = gpflow.optimizers.Scipy()
    opt.minimize(model.training_loss, model.trainable_variables, method="L-BFGS-B")

    return model

def gpPredict(model,X):
    """
    gpPredict() returns the prediction and standard deviation of the GP model
    on the X data provided.

    Parameters
    ----------
    model : gpflow.models.gpr.GPR object
        GP model.
    X : numpy array (N,K)
        Training features, where N is the number of data points and K is the
        number of independent features (e.g., sigma profile bins).

    Returns
    -------
    Y : numpy array (N,1)
        GP predictions.
    STD : numpy array (N,1)
        GP standard deviations.

    """
    # Do GP prediction, obtaining mean and variance
    GP_Mean,GP_Var=model.predict_f(X)
    # Convert to numpy
    GP_Mean=GP_Mean.numpy()
```

```
    GP_Var=GP_Var.numpy()
    # Prepare outputs
    Y=GP_Mean
    STD=numpy.sqrt(GP_Var)
    # Output
    return Y,STD

# ========================================================================
# Main Script
# ========================================================================

# Iniate timer
ti=time.time()
# Path to CSV files
trainDB_Path=os.path.join(dbPath,'GP_dataset_glycine - ARTICLE.csv')
# Load CSV file
trainDB=pandas.read_csv(trainDB_Path,delimiter=";")
# Create copy
trainDB_iteration=trainDB.copy()
# Get target variable denominator
varName=trainDB.columns[4]

x_mol=trainDB.iloc[:,2].to_numpy()
X_Train=x_mol.reshape(-1,1)*trainDB.iloc[:,7:-2].to_numpy()
Y_Train=trainDB.iloc[:,4].to_numpy().reshape(-1,1)
Y_err = trainDB.iloc[:, 6].to_numpy().reshape(-1,1) # incerteza na solubilidade relativa
# Get target variable denominator
varName=trainDB.columns[4]
# Normalize
if featureNorm is not None:
    X_Train_N,skScaler_X_Train=normalize(X_Train,method=featureNorm)

else:
```

```
    X_Train_N=X_Train

if labelNorm is not None:
    Y_Train_N,skScaler_Y_Train=normalize(Y_Train,method=labelNorm)
    if labelNorm=='Standardization':
        Y_err_N=Y_err/skScaler_Y_Train.scale_
    elif labelNorm=='LogStand':
        Y_err_N= Y_err/(Y_Train_N*skScaler_Y_Train.scale_)
else:
    Y_Train_N= Y_Train
    Y_err_N= Y_err

# Train GP
model=buildGP(X_Train_N,Y_Train_N,gpConfig=gpConfig)
# Get GP predictions
Y_Train_Pred_N,STD_Train_N=gpPredict(model,X_Train_N)

# Unnormalize
if labelNorm is not None:
    Y_Train_Pred,__=normalize(Y_Train_Pred_N,skScaler=skScaler_Y_Train,
                        method=labelNorm,reverse=True)

else:
    Y_Train_Pred=Y_Train_Pred_N

if labelNorm=='Standardization':
    STD_Train=STD_Train_N*skScaler_Y_Train.scale_
elif labelNorm=='LogStand':
    STD_Train=Y_Train_Pred*STD_Train_N*skScaler_Y_Train.scale_
else:
    STD_Train=STD_Train_N

# ============================================================
```

```
# Plots
# ================================================================

# Pyplot Configuration
plt.rcParams['figure.dpi']=600
plt.rcParams['savefig.dpi']=600
plt.rcParams['text.usetex']=False
plt.rcParams['font.family']='serif'
plt.rcParams['font.serif']='Times New Roman'
plt.rcParams['font.weight']='bold'
plt.rcParams['mathtext.rm']='serif'
plt.rcParams['mathtext.it']='serif:italic'
plt.rcParams['mathtext.bf']='serif:bold'
plt.rcParams['mathtext.fontset']='custom'
plt.rcParams['axes.titlesize']=8
plt.rcParams['axes.labelsize']=8
plt.rcParams['xtick.labelsize']=7
plt.rcParams['ytick.labelsize']=7
plt.rcParams['font.size']=8
plt.rcParams["savefig.pad_inches"]=0.02

# Histogram of the target data
print('Training set size: '+str(Y_Train.shape[0]))
print('Training set min: ' +str(Y_Train.min()))
print('Training set max: ' +str(Y_Train.max()))

plt.figure(figsize=(2.3,2))

R2_Train=metrics.r2_score(numpy.log(Y_Train),numpy.log(Y_Train_Pred))

MAE_Train=metrics.mean_absolute_error(Y_Train,Y_Train_Pred)

# Plot
```

```
r2 = metrics.r2_score(Y_Train, Y_Train_Pred)
plt.figure(figsize=(2.3,2))
plt.loglog(Y_Train,Y_Train_Pred,'ow',markersize=3,mec='red')

lims=[numpy.min([plt.gca().get_xlim(),plt.gca().get_ylim()]),
      numpy.max([plt.gca().get_xlim(),plt.gca().get_ylim()])]
plt.axline((lims[0],lims[0]),(lims[1],lims[1]),color='k',linestyle='--',linewidth=1)
plt.xlabel('Exp. '+varName,weight='bold')
plt.ylabel('Pred. '+varName,weight='bold')

xlim = plt.gca().get_xlim()
ylim = plt.gca().get_ylim()

x_text = xlim[1] * 0.7
y_text = ylim[1] * 0.3
plt.text(x_text, y_text, f'$R^2$ = {r2:.3f}', fontsize=8, fontweight='bold')
# Prevent wrong units from showing for RI

# Print elapsed time
tf=time.time()
print('Time elapsed: '+'{:.2f}'.format(tf-ti)+' s')

# ====================================================================
# Individual GP Predictions
# ====================================================================

to_delete_global=[]
# Initialize iterations
for iteration in range(100):
    # Extract sigma profiles and target variable as numpy arrays
    x_mol=trainDB_iteration.iloc[:,2].to_numpy()
    X_Train=x_mol.reshape(-1,1)*trainDB_iteration.iloc[:,7:-2].to_numpy()
    Y_Train=trainDB_iteration.iloc[:,4].to_numpy().reshape(-1,1)
```

```
    # Normalize
    if featureNorm is not None:
        X_Train_N,skScaler_X_Train=normalize(X_Train,method=featureNorm)
    else:
        X_Train_N=X_Train
    if labelNorm is not None:
        Y_Train_N,skScaler_Y_Train=normalize(Y_Train,method=labelNorm)
    else:
        Y_Train_N=Y_Train
    # Train GP
    model=buildGP(X_Train_N,Y_Train_N,gpConfig=gpConfig)


    # Plot
    # mol_list=['KNO₃','NaNO₃','KCl',"NaCl",'Na₂SO₄',"CaCl₂"]
    mol_list=['KNO₃']
    for mol in trainDB.iloc[:, 0].unique(): #trainDB.iloc[:, 0].unique(): # if you want individual results change
to mol_list
        mol_data = trainDB[trainDB.iloc[:, 0] == mol]
        # Entrada para malha (gráfico)
        x_mol=numpy.linspace(0,mol_data.iloc[:,2].to_numpy().max(),
                             100).reshape(-1,1)
        X_Plot=x_mol*mol_data.iloc[0,7:-2].to_numpy('float').reshape(1,-1)
        # Dados reais
        X_Real=mol_data.iloc[:,2].to_numpy().reshape(-1,1)\
            *mol_data.iloc[0,7:-2].to_numpy('float').reshape(1,-1)
        Y_Real=mol_data.iloc[:,4].to_numpy().reshape(-1,1)
        # Normalização
        if featureNorm is not None:
            X_Plot_N,__=normalize(X_Plot,skScaler=skScaler_X_Train,
                                  method=featureNorm)
            X_Real_N,__=normalize(X_Real,skScaler=skScaler_X_Train,
                                  method=featureNorm)
        else:
            X_Plot_N = X_Plot
```

```
    X_Real_N = X_Real_N
# Predições
Y_Plot_Pred_N, Y_Plot_STD_N = gpPredict(model, X_Plot_N)
Y_Real_Pred_N, Y_Real_STD_N = gpPredict(model, X_Real_N)
# Desnormalização
if labelNorm is not None:
    Y_Plot_Pred, _ = normalize(Y_Plot_Pred_N,
            skScaler=skScaler_Y_Train, method=labelNorm,
            reverse=True)
    Y_Real_Pred, _ = normalize(Y_Real_Pred_N,
            skScaler=skScaler_Y_Train, method=labelNorm,
            reverse=True)
else:
    Y_Plot_Pred = Y_Plot_Pred_N
    Y_Real_Pred = Y_Real_Pred_N
```

## D. Sigma profiles of different amino acids (glycine, alanine, isoleucine, and valine)

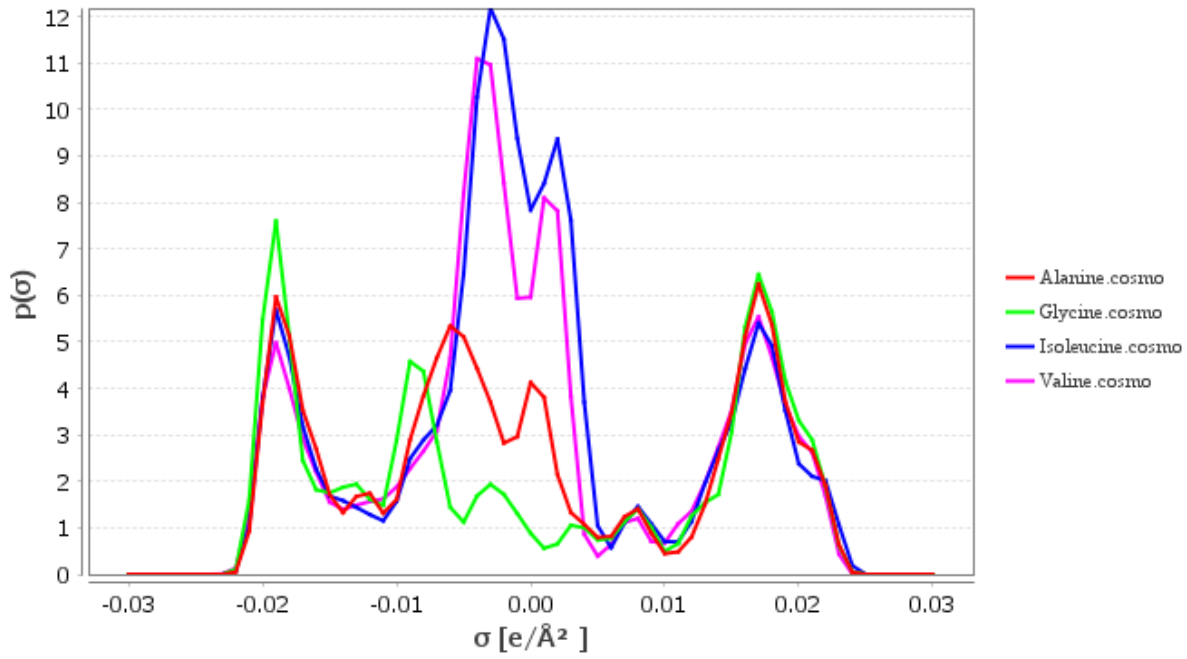


**Fig. D1.** Comparison of COSMO sigma profiles of glycine, alanine, isoleucine, and valine.