Physics-Informed Machine Learning for Predictive Turbulence Modeling: Status, Perspectives, and Case Studies

Heng Xiao
Assistant Professor
Department of Aerospace & Ocean Engineering
Virginia Tech

Machine Learning Technologies and Their Applications to Scientific and Engineering Domains Workshop, August 17, 2016
Acknowledgment of Collaborators

Jianxun Wang, VT  Jinlong Wu, VT  Dr. Julia Ling, SNL
Publications Related to This Talk

The presentation slides will be made available publicly via ResearchGate, or sent me an email: hengxiao@vt.edu

https://sites.google.com/a/vt.edu/hengxiao/home
Or google: Heng Xiao, VT

Turbulence Modeling with Machine Learning (offline data)


Turbulence Modeling with Data Assimilation (*online data*)


Predictive Modeling & Model Discrepancy

❖ Reynolds Averaged Navier-Stoke (RANS) simulations are widely used in design, optimization, and reliability assessment of aero and space vehicles and gas turbines relevant to NASA missions.

❖ However, it remains challenging to predict system performance with confidence.

❖ Model discrepancy is a major obstacle in predictive modeling with RANS models.
Origin of Model Discrepancy in Low Fidelity Models

1. We do not understand the physics well enough to describe/model them.

2. We cannot afford the computational cost to adequately resolve the physics.

❖ In many cases, model discrepancy originates from the combination of two.

❖ The second reason is dominant for RANS based turbulence modeling, but it also depends on the interpretation.

Using data to complement low fidelity models!
**Simulation in Support System Monitoring**

**Online data:**
Real-time streamed sensor data are available but are sparse

**Data assimilation**
### Simulation in Support of Design and Optimization

<table>
<thead>
<tr>
<th>Calibration Cases (offline data)</th>
<th>Prediction Cases (no data)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A few configuration with data (DNS or measurements)</td>
<td>Similar configuration with different:</td>
</tr>
<tr>
<td></td>
<td>• Twist</td>
</tr>
<tr>
<td></td>
<td>• Sweep angles</td>
</tr>
<tr>
<td></td>
<td>• Airfoil shape</td>
</tr>
</tbody>
</table>

- Machine learning
Scope of This Presentation

❖ Proposition: *Machine learning in conjunction with (offline) data can be used to reduce model discrepancy of low-fidelity models, which are often used in engineering design.*

❖ Here, I share my perspectives and experiences of using physics-informed machine learning to assist modeling of complex physical systems.

❖ RANS turbulence modeling, a typical low fidelity CFD model for turbulent flows, is used as example.

❖ However, the approach is general enough to be relevant for researchers of many other domains, e.g., structures, materials, combustion (flow and chemistry).
Unique Challenges in ML for Computational Physics

❖ Why can’t we take the usual approach and simply use ML to learn what we want to know? Pressure, drag, lift, velocity, failure probability etc. May violate physics laws!

❖ There are many “hard constraints” originating from physics laws, e.g., velocity field is divergence free for incompressible fluid; pressure and velocity fields must be consistent (related via PDE); elasticity tensors must be positive definite, etc.

❖ Popular applications of ML has mostly “soft constraints”. Sentiment analysis in reviews: great, pleasant=5; terrible=1. Targeted advertisement: diapers go with infant toys. Scientific document analysis: abstract, introduction, methodology, results, conclusion.
Physics-Informed Machine Learning: Clarification

Algorithm development has drawn inspirations from physics/biological systems, e.g., simulated annealing, particle swarm, genetic algorithm: Not what I mean.

Our interpretation of PIML: using ML to solve physical problems (mechanics of fluids, solids, materials, combustion).

✓ Incorporate physical constraints (e.g., conservation laws, realizability) in every aspect of ML:
  ✓ formulation of the learning problem
  ✓ choice/normalization of features and responses
  ✓ choice/development of learning algorithm.

☐ Co-design in (1) formulation of physical problem for learning; (2) ML algorithm development; (3) hardware.
Physics-Informed Machine Learning: Perspectives

Assist but respect models: Machine learning should be used to correct/improve existing models, not to replace them. Thus, we learn the model discrepancy, not the model output directly. (consensus)

1. Choose quantities that have physical bounds/constraints/interpretation to learn (allow for anchoring to physics).

2. Learned quantities should be universal to some extent: same functional form in training and prediction flows! Note the limitation of universality though…

3. Obey physical constraints in the learning as much as possible (e.g., hard constraints such as positive semi-definiteness of Reynolds stress; conservations of mass).
Case Study: RANS-Based Turbulence Modeling

❖ Description of the challenge
❖ Problem formulation
❖ Procedure:
  
  \[ y = f(q) \text{ or } f : q \rightarrow y \]

❖ Choice of features and responses
❖ Choice of machine learning algorithm
❖ Results
❖ Possible extension to other systems
RANS as Work-Horse Tool in CFD

- RANS (Reynolds Averaged Navier-Stokes) solvers with turbulence closures are still the low-fidelity work-horse tool in industrial CFD simulations.

- High-fidelity methods such as LES and DNS are still too expensive for practical flows.

- The drawback of RANS: poor performance in flows with separation, mean pressure gradient, curvature, or swirling …

- Need to quantify and reduce model discrepancy in RANS simulations.
Source of Uncertainty in RANS Models

- **Reynolds stress closure** is the source of model form uncertainty in RANS simulations.

\[
\frac{\partial U_i}{\partial t} + \frac{\partial (U_i U_j)}{\partial x_j} + \frac{1}{\rho} \frac{\partial p}{\partial x_i} - \nu \frac{\partial^2 U_i}{\partial x_j \partial x_j} = \text{div. Reynolds Stresses} \quad \nabla \cdot \tau
\]

\( \tau \): hub of turbulence models

Physics-based prior knowledge
Source of Uncertainty in RANS Models

- Reynolds stress closure is the source of model form uncertainty in RANS simulations.

\[
\frac{\partial U_i}{\partial t} + \frac{\partial (U_i U_j)}{\partial x_j} + \frac{1}{\rho} \frac{\partial p}{\partial x_i} - \nu \frac{\partial^2 U_i}{\partial x_j \partial x_j} = \nabla \cdot \mathbf{T}
\]

\( \mathbf{T} \): hub of turbulence models

\[
\frac{D\omega}{Dt} = P(\omega, \mathbf{U}) - D(\omega, \mathbf{U}) + T(\omega, \mathbf{U})
\]

Physics-based prior knowledge
Source of Uncertainty in RANS Models

- Reynolds stress closure is the source of model form uncertainty in RANS simulations.

\[
\frac{\partial U_i}{\partial t} + \frac{\partial (U_i U_j)}{\partial x_j} + \frac{1}{\rho} \frac{\partial p}{\partial x_i} - \nu \frac{\partial^2 U_i}{\partial x_j \partial x_j} = \nabla \cdot \tau
\]

\[
\frac{D \omega}{Dt} = P(\omega, U) - D(\omega, U) + T(\omega, U)
\]

\[
\frac{D \tilde{\nu}_t}{Dt} = P(\tilde{\nu}_t, U) - D(\tilde{\nu}_t, U) + T(\tilde{\nu}_t, U)
\]

\[\tau: \text{hub of turbulence models}\]

Physics-based prior knowledge
Source of Uncertainty in RANS Models

- Reynolds stress closure is the source of model form uncertainty in RANS simulations.

\[
\frac{\partial U_i}{\partial t} + \frac{\partial (U_i U_j)}{\partial x_j} + \frac{1}{\rho} \frac{\partial p}{\partial x_i} - \nu \frac{\partial^2 U_i}{\partial x_j \partial x_j} = \nabla \cdot \tau
\]

Div. Reynolds Stresses \( \nabla \cdot \tau \)

\( \tau \): hub of turbulence models

- k-\( \omega \) model

\[
\frac{D\omega}{Dt} = P(\omega, \mathbf{U}) - D(\omega, \mathbf{U}) + T(\omega, \mathbf{U})
\]

- SA model

\[
\frac{D\tilde{\nu}_t}{Dt} = P(\tilde{\nu}_t, \mathbf{U}) - D(\tilde{\nu}_t, \mathbf{U}) + T(\tilde{\nu}_t, \mathbf{U})
\]

Reynolds Stress Transport Model (RSTM)

\[
\frac{D\tau}{Dt} = P(\tau, \mathbf{U}) - D(\tau, \mathbf{U}) + T(\tau, \mathbf{U})
\]

Physics-based prior knowledge
Injecting Uncertainty in RANS Modeling

Injecting uncertainties directly to the Reynolds stresses: output of the turbulence closure [Xiao et al.] Our approach.

Injecting uncertainties to turbulence model transport equations: form of the turbulence closure [Duraisamy et al.]
Critical Questions in Physics-Informed Machine Learning

Objective: Reduce RANS model discrepancy by learning from data.

❖ Where does the training data come from?
❖ What are the quantities to learn (responses, targets, dependent variables)? Are they universal, at least to some extent?
❖ What are the features (predictors, independent variables)?
❖ What learning algorithm should be used?
Objective: Reduce RANS model discrepancy by learning from data.

❖ Where does the training data come from?

❖ What are the quantities to learn (responses, targets, dependent variables)? Are they universal, at least to some extent?

❖ What are the features (predictors, independent variables)?

❖ What learning algorithm should be used?
Formulation: Inference (optional) + Machine Learning

Data: Pressure, Skin friction, Velocity

Inverse Modeling

Information: Spatial discrepancy

Machine Learning

Knowledge: Functional discrepancy

Embedding

Prediction: Injection into solver

[Xiao et al., 2016. JSMC]

[Duraisamy et al., 2015. AIAA]
Formulation: Inference (optional) + Machine Learning

Data: Pressure, Skin friction, Velocity

Inverse Modeling

Information: Spatial discrepancy

Machine Learning

Knowledge: Functional discrepancy

Embedding

Prediction: Injection into solver

[Duraisamy et al., 2015. AIAA]
Formulation: Inference (optional) + Machine Learning

Data: Pressure, Skin friction, Velocity

Inverse Modeling

Information: Spatial discrepancy

Machine Learning

Knowledge: Functional discrepancy

Embedding

\[ \mathcal{N}(U) = \nabla \cdot (\tau_{\text{rans}} + \delta \tau) \]

or

\[ \frac{D\omega}{Dt} = P(\omega, U) - D(\omega, U) + T(\omega, U) + \delta \]

Xiao et al.

[Duraisamy et al., 2015.AIAA]
Formulation: Inference (optional) + Machine Learning

Data: Pressure, Skin friction, Velocity

Inverse Modeling

Information: Spatial discrepancy

Machine Learning

Knowledge: Functional discrepancy

Embedding

Prediction: Injection into solver

[Xiao et al., 2014]

\[
\mathcal{N}(U) = \nabla \cdot (\mathbf{\tau}_{\text{rans}} + \delta \mathbf{\tau})
\]

or

\[
\frac{D\omega}{Dt} = P(\omega, U) - D(\omega, U) + T(\omega, U) + \delta
\]

[Duraisamy et al., 2015. AIAA]

\[
\delta \mathbf{\tau}(x) \quad \text{or} \quad \delta(x)
\]

\[
\hat{\delta} \mathbf{\tau}(q) \quad \text{or} \quad \hat{\delta}(q)
\]

\[
\mathcal{N}(U) = \nabla \cdot (\mathbf{\tau}_{\text{rans}} + \hat{\delta} \mathbf{\tau}(q))
\]

or

\[
\frac{D\omega}{Dt} = P - D + T + \hat{\delta}(q)
\]
Formulation: Inference (optional) + Machine Learning

Data
Pressure, Skin friction, Velocity

DNS Data of
Reynolds Stress for
Elementary Flows

Knowledge
functional discrepancy

Machine Learning

Embedding

\[ \mathcal{N}(U) = \nabla \cdot (\mathbf{\tau}_{\text{rans}} + \delta \mathbf{\tau}) \]

\[ \frac{D\omega}{Dt} = P(\omega, \mathbf{U}) - D(\omega, \mathbf{U}) + T(\omega, \mathbf{U}) + \delta \]

Duraisamy et al.

\[ \delta \mathbf{\tau}(x) \quad \text{or} \quad \delta(x) \]

Xiao et al.

\[ \hat{\delta} \mathbf{\tau}(q) \quad \text{or} \quad \hat{\delta}(q) \]

\[ \mathcal{N}(U) = \nabla \cdot (\mathbf{\tau}_{\text{rans}} + \hat{\delta} \mathbf{\tau}(q)) \]

\[ \text{or} \quad \frac{D\omega}{Dt} = P - D + T + \hat{\delta}(q) \]

Prediction: Injection into solver

[Duraisamy et al., 2015. AIAA]
Machine learning encompasses a variety of data-driven methods that include classifiers, regression algorithms, and clustering algorithms. Supervised machine learning algorithms use a set of labeled training examples, which are used to train a model that can predict the output of new, unseen data. An important aspect of training is to ensure that the model generalizes well to new data, which means that the model does not simply memorize the training data but instead learns the underlying patterns.

Overfitting is a common issue in supervised learning, where the model learns the training data too well, including the noise and outliers, which results in poor generalization to new data. This is evident in the plot where the training error remains below the validation error, even when six cases are used for training, indicating overfitting. As the number of data sets used for training increases, the training error increases slightly and the validation error decreases significantly, which is a typical sign of overfitting.

The accuracy of the model can be evaluated by comparing the predictions with the actual outcomes. In the case of CFD simulations, the difference between the training and validation error is an indicator of the degree of overfitting. Small differences suggest that the model is likely to generalize well, while large differences indicate overfitting.

The velocity magnitude is normalized by the bulk velocity in cases 1, 4, and 5, by the free stream velocity in cases 2 and 7, and by the average jet velocity in cases 3 and 6. The figures show contours of the normalized velocity magnitude as predicted by RANS. Some figures are adopted from Ling et al. POF 2015; www.turbostream-cfd.com; youtube.com.
Objective: Reduce RANS model discrepancy by learning from data.

❖ Where does the training data come from?
❖ What are the quantities to learn (responses, targets, dependent variables)? Are they universal?
❖ What are the features (predictors, independent variables)?
❖ What learning algorithm should be used?
Injecting Uncertainty into Reynolds Stresses

- Iaccarino et al. perturbed towards three limiting states in Barycentric triangle (realizability map) for uncertainty estimation

\[ \tau = 2k \left( \frac{1}{3} I + a \right) = 2k \left( \frac{1}{3} I + V \Lambda V^T \right) \]

\[ \tau \rightarrow \left( k, \xi, \eta, \varphi_1, \varphi_2, \varphi_3 \right) \]

magnitude, aspect ratio, orientation

Physics-based “normalization”: potential to be universal quantities; Physical constraints respected.
Is The Discrepancy of Anisotropy Universal?

![Graph showing the comparison of $\delta \xi$ and $\delta \eta$ for different Reynolds numbers. The graphs illustrate the discrepancy in anisotropy between two sets of data, one at $Re=2800$ and another at $Re=10595$. The graphs show a significant difference, indicating a potential universality issue.]

Probably!

From Physical Space to Feature Space: Learning

- Construct discrepancy function based on “mean flow features” \( q \! \)

\[
\delta \tau (x) \quad \text{Inferred or DNS, not universal (specific to the geometry)}
\]

\[
\Rightarrow \delta \tau (q) \quad \text{Machine learning}
\]

- Responses are discrepancies in TKE (log), eigenvalues and eigenvectors.

\[
\tau = 2k \left( \frac{1}{3} \mathbf{I} + a \right) = 2k \left( \frac{1}{3} \mathbf{I} + \mathbf{V} \Lambda \mathbf{V}^T \right)
\]

\[
\delta \log(k)(q), \delta \xi(q), \delta \eta(q), \delta \varphi_1(q)
\]
Critical Questions in PIML

Objective: Reduce RANS model discrepancy by learning from data.

❖ Where does the training data come from?
❖ What are the quantities to learn (responses, targets, dependent variables)? Are they universal, at least to some extent?
❖ What are the features (predictors, independent variables)?
❖ What learning algorithm should be used?
Construction of Feature Space

\{S, \Omega, \nabla p, \nabla k, Re_d, \mathcal{P}/\varepsilon, k/\varepsilon, \kappa\}

4 tensors/vectors; 47 invariants (integrity bases)

- Invariants of 4 tensors/vectors: strain rate (S), rotation rate (\Omega), pressure (p) gradient, TKE (k) gradient;
- 4 scalars: streamline curvature (\kappa), wall-distance based Reynolds number (Re_d), turbulent time scale
- (Normalized) feature vector \(q\) has a length of \(\sim 50\).
- Very high dimension feature space: beyond human comprehension: interpretation in progress.

Objective: train discrepancy functions \(\delta \mathcal{T}_i(q)\)
(Ling et al. 2016)
Should Feature Variables Be Invariant?

❖ Invariants is not only desirable, but essential!
❖ Very different from other applications of ML, e.g., handwritten digits recognition.

Features should not be fully invariant here!
Should Feature Variables Be Invariant?

❖ Invariants is not only desirable, but essential!

❖ Very different from other applications of ML, e.g., handwritten digits recognition.

Features should **not** be fully invariant here!

Fully invariant is essential

III. MACHINE LEARNING ALGORITHMS

Machine learning encompasses a variety of data-driven methods that include classifiers, regressors, and clustering algorithms. Supervised machine learning algorithms use a set of labeled training...
Critical Questions in PIML

Objective: Reduce RANS model discrepancy by learning from data.

❖ Where does the training data come from?
❖ What are the quantities to learn (responses, targets, dependent variables)? Are they universal, at least to some extent?
❖ What are the features (predictors, independent variables)?
❖ What algorithm should be used?
Machine Learning Algorithm: Random Forests

- Machine learning is an umbrella term for many algorithms.
- We used Random Forests regression: (1) suitable for high-dimension feature space and (2) robust in tolerating unimportant features; no linear regression = more robust
- Key lesson: choice of algorithm is dictated by the physical problem.

Physics-informed machine learning
Stratifying baseball player salary data (color coded from low (blue, green) to high (yellow, red)
Decision tree for these data

(Figure credit: Hastie and Tibshirani, Introduction to Statistical Learning)
Results

- Overall, the tree stratifies or segments the players into three regions of predictor space: $R_1 = \{X \mid \text{Years} < 4.5\}$, $R_2 = \{X \mid \text{Years} \geq 4.5, \text{Hits} < 117.5\}$, and $R_3 = \{X \mid \text{Years} \geq 4.5, \text{Hits} \geq 117.5\}$.

(Slide credit: Hastie and Tibshirani, Introduction to Statistical Learning)
From Decision Tree to Random Forests

❖ Individual decision trees are usually bad decision makers: greedy algorithm may miss globally optimal stratification.

❖ Random forests: an ensemble of trees built from bootstrap samples.

❖ Use only a subset of features to de-correlate the trees.

❖ Physical intuition!
Other Machine Learning Algorithms

❖ With a feature space dimension of 50, many ML algorithms susceptible to “curse of dimensionality” are ruled out: e.g., linear regression and its variants; Gaussian Process.

❖ Neural network seems to be viable choice with several potential benefits (yet to be explored):
  ✓ More natural for coupled regression.
  ✓ More flexible and possibly better predictive skills.
  ✓ Co-design?
Case Study: Flow in a Square Duct

The flow features in-plane secondary flow vortexes, which cannot be predicted by standard RANS models.
DNS Data for Duct Flows

Re$_b$ = 2200

Training
Re2200

Training
Re2600

Re$_b$ = 2900

Training
Re2900

Prediction
Re3500

Re$_b$ = 3200

Re$_b$ = 3500
$Re_b = 2200$

**Arrows:** In-plane velocity

**Color:** Streamwise velocity

**Lines:** Contours $U/U_{\text{max}} = 0.5$ and $0.8$
$Re_b = 2600$

Arrows: In-plane velocity

Color: Streamwise velocity

Lines: Contours $U/U_{max} = 0.5$ and $0.8$
Arrows: In-plane velocity
Color: streamwise velocity
Lines: contours $U/U_{\text{max}} = 0.5$ and $0.8$
Re$_b$ = 3500

Arrows: In-plane velocity
Color: streamwise velocity
Lines: contours $U/U_{max} = 0.5$ and 0.8
Velocity *Prediction* with Machine Learning *Corrected* Reynolds Stresses
Secondary velocity pattern near corner

Significance success in using ML towards predictive turbulence modeling.
Case Study: Separated Flows in Different Geometries

Training flows
- curved backward facing step
- wavy channel
- converging-diverging channel
- backward facing step

Prediction
- Periodic hill
  Re=10595

Source: http://turbmodels.larc.nasa.gov/other_dns.html
Predicted Anisotropy in Separated Region

Separation region

Trained on cases with different geometry

Black: DNS
Red: ML
Blue: Baseline

Figure 9: Barycentric map of the predicted Reynolds stress anisotropy for the test flow (PH10595) learned from training flows with different geometries and at different Reynold numbers (WC360 and CS13200). The prediction results on four streamwise locations at $x/H=1$, $2$, $3$, $4$ are compared with the corresponding baseline and benchmark results in panels (a)–(d), respectively. These four locations correspond to the beginning, center, downstream, and end of the separation bubble, which are indicated in the insets of each panel. The arrow denotes the order of samples plotted in the triangle.
Beyond Turbulence Modeling

- Constitutive modeling of complex materials
- Dynamics of atmospheric, ocean, and climate system
- Combustion
- … …

Similar challenges to turbulent flows:

1. We do not **understand** the physics well enough to describe/model them (e.g., chemical reactions)

2. We cannot **afford** the computational cost to adequately resolve the physics (e.g., micro-fibers, grains; cloud, ABL, terrain)
Analogy between turbulent flows and dynamics of complex materials
Analogy Between Turbulence & Solid Mechanics

❖ Turbulence can be considered “a fluid with complex constitutive behavior”.

\[ \tau = 2\nu_t S \]
\[ S = \frac{1}{2}(\nabla U + \nabla^t U) \]

U = velocity

\nu_t = turbulent viscosity

❖ Complex materials can be considered “a solid with complex constitutive behavior (stress/strain relation)”.

\[ \tau = E\varepsilon \]
\[ \varepsilon = \frac{1}{2}(\nabla U + \nabla^t U) \]

U = displacement

E = effective modulus
Experimental samples of heterogeneous material

Subjected to representative loading (computational or laboratory test)

Finite element mesh

$V_{\text{test}} \geq V_{\text{fe}} \geq V$

apparently stress-dependent material property at the macro-scale

[Das & Ghanem, 2009]
Where does the Complex Behavior come from?

- The “complex behaviors” in both problems do not really exist if they were fully resolved, i.e.,
  - directly resolve all scales in turbulent flows (Direct Numerical Simulation).
  - directly resolve all meso-scale constituents (fully resolved FEM)
- The “apparently” complex constitutive behavior is due to the modeling of unresolved scales.
- As a result the constitutive coefficients ($\nu_t$ or $E$) are properties of the flow dynamics or structural dynamics, and not the property of the materials (fluid or solid).
Summary

❖ Using ML in computational physics has unique challenges.

❖ In physics-informed machine learning, we utilize physical constraints in all aspects of machine learning to address these challenges.

❖ Choose universal quantities based on physical prior knowledge.

❖ Preliminary success in RANS based turbulence modeling. The objective is co-design of ML algorithm and problem formulation.

❖ Has potential well beyond turbulence modeling.
Now: separate functions for each flow class; Eventually: ML algorithm choose data automatically.

Need benchmark database: elementary flows (free shear, plane channel), flows of medium complexity (separation, airfoil), to realistic flows (wing-body junction) and industrial flows.

Collaborations Ideas

III. MACHINE LEARNING ALGORITHMS

The velocity magnitude is normalized by the bulk velocity in cases 1, 4, and 5, by the free stream velocity in cases 2 and 7, and by the average jet velocity in cases 3 and 6.

FIG. 1. Schematics of each case in the database showing contours of normalized velocity magnitude as predicted by RANS.
Collaboration Ideas

- Evaluation of the PIML method in turbulence models relevant to NASA (SA, k-ω SST, maybe an EARSM/RSTM)

- Dissemination by implementing/distributing in NASA codes (e.g., CFL3D, FUN3D, OVERFLOW). Current implementations are in open-source code OpenFOAM with Python scripts.

- Extensions beyond RANS-based turbulence modeling.
Thank you!

Heng Xiao, Virginia Tech
hengxiao@vt.edu

https://sites.google.com/a/vt.edu/hengxiao/home
Physics-Informed Machine Learning: Perspectives

Assist but respect models: Machine learning should be used to correct/improve existing models, not to replace them. Thus, we learn the model discrepancy, not the model output directly. (consensus)

1. Choose quantities that have physical bounds/constraints/interpretation to learn (allow for anchoring to physics).

2. Learned quantities should be universal to some extent: same functional form in training and prediction flows! Note the limitation of universality though…

3. Obey physical constraints in the learning as much as possible.