Prescribed Fire Modeling using Knowledge-Guided Machine Learning for Land Management

Somya Sharma Chatterjee* Kelly Lindsay* Neel Chatterjee* Rohan Patil†
Ilkay Altintas De Callafon† Michael Steinbach * Daniel Giron‡ Mai H. Nguyen†
Vipin Kumar*

Abstract

In recent years, the increasing threat of devastating wildfires has underscored the need for effective prescribed fire management. Process-based computer simulations have traditionally been employed to plan prescribed fires for wildfire prevention. However, even simplified process models are too compute-intensive to be used for real-time decision-making. Traditional ML methods used for fire modeling offer computational speedup but struggle with physically inconsistent predictions, biased predictions due to class imbalance, biased estimates for fire spread metrics (e.g., burned area, rate of spread), and limited generalizability in out-of-distribution wind conditions. This paper introduces a novel machine learning (ML) framework that enables rapid emulation of prescribed fires while addressing these concerns. To overcome these challenges, the framework incorporates domain knowledge in the form of physical constraints, a hierarchical modeling structure to capture the interdependence among variables of interest, and also leverages pre-existing source domain data to augment training data and learn the spread of fire more effectively. Notably, improvement in fire metric (e.g., burned area) estimates offered by our framework makes it useful for fire managers, who often rely on these estimates to make decisions about prescribed burn management. Furthermore, our framework exhibits better generalization capabilities than the other ML-based fire modeling methods across diverse wind conditions and ignition patterns.

Keywords— prescribed fire modeling, knowledge guided machine learning, probabilistic graphical model, surrogate model

1 Introduction

Thousands of wildfires engulf millions of acres in the United States each year alone [1]. These fires threaten wildlife and human lives and destroy personal property. Fire managers (also known as burn bosses) use prescribed fires, the intentional and controlled lighting of fire, to reduce the fuel that feeds extreme fires and help improve forest health by recycling soil nutrients [38; 19]. Prescribed fires are generally ignited only when specific conditions are met, such as lower wind speed conditions. Otherwise, lighting of prescribed burns is likely to jeopardize the fire crew's safety and cause collateral damage, similar to the prescribed fire that led to wildfire in Santa Fe National Forest in 2022 [11]. To minimize the risk of wildfires, fire managers use process-based models to simulate underlying physical processes in a fire system to identify areas that may burn under given weather conditions (such as wind speed and direction), fuel density, and pre-decided ignition pattern. Fire crew use these models to decide if a prescribed fire can be started under expected wind conditions. Among the process-based fire models, QUIC-Fire (QF) is the only process model specifically designed for prescribed fire simulation [28]. QF has been used collaboratively with prescribed fire managers to plan prescribed burns and to better understand the impact of differing weather and ignition conditions on the outcome of a burn. While QF provides significant speedup over other models [16], its computation time still limits its usability for real-time decision-making.

A prescribed fire modeling framework that can emulate the fuel density behavior as exhibited by QF, with a shorter computation time, will be useful to fire managers to ensure the safe lighting of prescribed fires. Such a framework will be useful since the weather conditions can change quickly (from the forecasted scenario), and the crew has to decide if the risk of a wildland fire starting is big enough to call off the burn. Under such rapidly changing situations, ML models can be used as faster alternatives.

ML models have emerged as fast and efficient alternatives to process-based models [39]. While there has been some work on ML-based wildfire modeling, there is little work that can be used in the context of prescribed fire modeling. To model wildfires, CNN [4; 40; 29; 35], ConvLSTM [10; 21] and UNet [21; 43] models have been used. However, due to several differences in the exogenous variables,

*University of Minnesota-Twin Cities. {sharm636, lind0436, chatt097, stei0062, kumar001} @umn.edu

†University of California San Diego {rpatil,ialtintas,mhnguyen }@ucsd.edu

‡Colorado State University. danielrgiron95@gmail.com

ignition conditions, and fire spread mechanisms, methods developed for wildfires are not suitable for modeling prescribed fires (more details in related work). For prescribed fires modeling, a CNN-based model [12] has been proposed that has limited utility since it requires simulated outputs from QUIC-Fire to make forecasts.

Furthermore, the aforementioned ML methods are unable to accurately model the complex nonlinear interaction among physical processes in a prescribed fire system. These methods face three major challenges. First, the predicted fuel density distribution from such ML models has physical inconsistencies, such as inaccurate fuel transportation and inconsistent contraction of burned area. The second challenge is the class imbalance problem. In prescribed fire modeling, predictions from ML models suffer from class imbalance problems where the majority class of unburned pixels bias the fuel predictions, translating into an overestimation of fuel in burned regions, especially in more extreme wind conditions. The third challenge in using an ML model is the inaccurate estimation of summary statistics used by fire managers in operational settings. One such metric is the burned area or number of burned pixels that enables an improved understanding of the burn severity. However, when the ML model inaccurately predicts the fuel density spatial distribution, the number of burned cells is often underestimated.

To overcome the above challenges, we propose using a physics-guided spatiotemporal ML model that addresses the complexities of emulating prescribed fire evolution. In our framework, we propose integrating known **physical constraints and domain knowledge** in the model to enable the ML model to learn physically consistent values [39]. To overcome the second challenge, we leverage preexisting source domain data to accurately learn the spread of fire beyond the ignited cells. This **domain adaptation** mechanism allows us to transfer knowledge for a given ignition pattern from lower wind speed settings to scenarios with a higher wind speed. To address the third challenge, we integrate a **probabilistic graphical structure** to model the interdependence between fuel density and the number of burned cells. This ensures that we can incorporate prior knowledge about these quantities and that the learned embeddings can capture the hierarchical relationship between the two quantities. The probabilistic graphical modeling and the physics-guided learning ensure that the summary statistics, like the number of burned cells, are estimated accurately. The proposed ML-based emulator provides speedup over QUIC-Fire and can be used for *faster than real-time prescribed fire forecasting*[1]. We have released our code in a GitHub repository link. The summary of contributions of this work is provided next.

- To the best of our knowledge, we propose the first prescribed fire ML-based emulator to predict fuel density evolution from initial wind conditions and ignition patterns. The proposed model improves the MSE in fuel density predictions by 55% over data-driven ML models and reduces the inference time by 69% over QF.
- The framework leverages domain knowledge, data augmentation from the use of source domain data, and a hierarchical modeling scheme to capture the interdependence between fuel density and burned area to overcome physical inconsistencies in prediction, improving generalization and estimates for fire spread metrics.
- Extensive experiments demonstrate that our approach performs better than other fire modeling ML methods in terms of both prediction performance and physical consistency of the outputs. Measuring success in many scientific domains, including fire modeling, is hard to capture via standard performance metrics commonly used in ML model design. We propose new loss functions and evaluation metrics to evaluate the physical consistency of the outputs from ML models.

2 Related Work

Traditional fire behavior simulator models [13; 27; 37; 36; 17] have been used for simulating wildfires [23; 33; 18; 24]. Though researchers have attempted to adapt some of these models to simulate prescribed fire behavior, the models often make assumptions that may not be suitable for prescribed fire modeling and overlook the fire suppression behavior specific to prescribed fires. Additionally, their computational cost hinders their practical implementation for real-time operational decision-making [31; 14]. QUIC-Fire [28] overcomes some of these challenges and offers faster runtime compared to other process-based models.

ML models have demonstrated success in various wildfire modeling tasks [22; 42; 41; 4; 40; 29; 35; 10; 21; 43]. While physics-guided ML has also been used in wildfire modeling, the physical assumptions made by current methods are not suitable for prescribed fire modeling and may suffer from numeric instability [5; 9]. Additionally, methods developed for wildfire modeling may not generalize to prescribed fire modeling [19; 15; 20; 12; 28]. Wildfires are unplanned and may be caused by natural or accidental ignitions. Prescribed fires are planned to meet management objectives and are ignited under pre-determined environmental conditions. In terms of ignition conditions, wildfires are often modeled as a single ignition point. In contrast, prescribed fires are ignited at several ignition points, often in terms of several fire lines. This creates a more complex interaction among the fire ignited at different points, leading to fire suppression. Wildfire modeling often also looks at fires that have been burning for days, while prescribed fires projects are short-term. Due to differences in time scales, wildfires

[1] An example of "faster than real-time prediction" would be generating forecasts for the next 10 minutes in less than 10 minutes.

are more heavily impacted by moisture content in the atmosphere and weather patterns over several days, while the effect of these exogenous variables may not be that prominent for prescribed fires. This leads to a difference in fire behavior in a wildfire and a prescribed fire [19; 20; 15].

Moreover, previous ML-based studies in prescribed fire modeling have not extensively explored the specific problem of prescribed fire evolution modeling based on initial environmental conditions [12; 3; 2; 30]. This capability is crucial for effective prescribed fire planning. By addressing this gap, our framework fills an important need in the existing ML literature. Relevant ML methods for prescribed fire and wildfire forecasting include CNN [12; 4; 40; 29; 35], ConvLSTM [10; 21], and UNet [21; 43] models. We also show a comparison with these methods in the paper.

3 Problem Setting

In this work, we study the evolution of fuel density, the mass of fire fuel per unit volume, as prescribed fire spreads. We use N simulation runs from the QUIC-Fire model as training examples to learn a data-driven emulator of QUIC-Fire. Our framework leverages simulations from QUIC-Fire, enabling the learning of the two-way interaction between environmental conditions and fire behavior without making strong assumptions. The input tensor for training example, i, can be represented as $X_i \in \mathbb{R}^{n\times M\times P\times C}$, where n is the number of time steps in the sequence, M and P are the number of rows and columns of cells forming the spatial grid and C is the number of input channels. In our experiments, C includes initial wind speed, initial wind direction, ignition pattern, and source domain fuel density maps. Similarly, the fuel density output for the model can be represented as $Y_i \in \mathbb{R}^{n\times M\times P\times 1}$. The initial wind speed and wind direction are static channels - available only for the first time step and repeated throughout the grid and the sequence to obtain the embeddings that are used as inputs to the emulator, whereas ignition patterns and source domain maps are dynamic inputs. This setup allows us to study the driver-response relation in prescribed fire system. Each simulation run represents different initial wind conditions and ignition patterns. For each simulation run i, $X_i = [x_i^1, x_i^2, ..., x_i^n]$ represent the spatiotemporal drivers, where $x_i^t \in \mathbb{R}^{M\times P\times C}$. Similarly, the response in a simulation run i at time step t can be represented as $y_i^t \in \mathbb{R}^{M\times P\times 1}$.

4 Methods

This section details the proposed framework (Fig. 1) for fuel density prediction. Section 4.1 introduces the backbone spatiotemporal model. Section 4.2 outlines the source domain data used for data augmentation. Section 4.3 introduces the physical constraints used to improve the physical consistency of predictions. In Section 4.4, we outline the probabilistic graphical modeling structure.

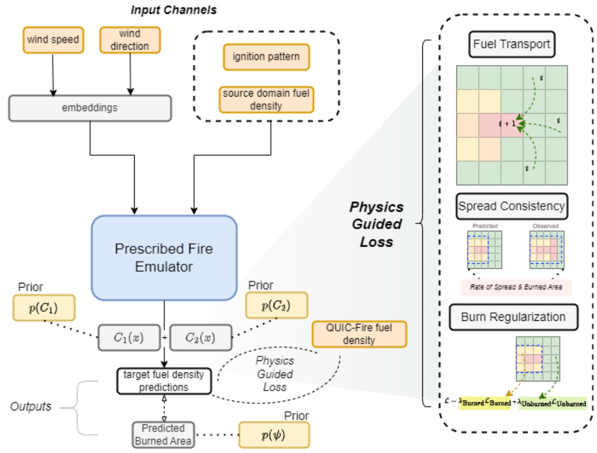

Figure 1: Physics-guided Emulator for fuel density estimation. Inputs include source fuel density, ignition pattern, initial wind speed, and initial wind direction.

4.1 Spatiotemporal Model In this work, convLSTM [32] models are used to leverage both spatial correlations among the cells in a spatial grid and temporal relation in time-series measurements for each cell. Our spatiotemporal model uses ConvLSTM layers to encode information in the three-dimensional domain - time, height (rows of cells), and width (columns of cells). The ConvLSTM uses the following set of equations to generate embeddings for a sequence,

$$
\begin{aligned}
\boldsymbol{i^t} &= \sigma(\boldsymbol{W_i}\left[[\boldsymbol{x^t}]; \boldsymbol{h^{t-1}}; \boldsymbol{c^{t-1}}\right] + \boldsymbol{b_i}),\\
\boldsymbol{f^t} &= \sigma(\boldsymbol{W_f}\left[[\boldsymbol{x^t}]; \boldsymbol{h^{t-1}}; \boldsymbol{c^{t-1}}\right] + \boldsymbol{b_f}),\\
\boldsymbol{g^t} &= tanh(\boldsymbol{W_g}\left[[\boldsymbol{x^t}]; \boldsymbol{h^{t-1}}\right] + \boldsymbol{b_g}),\\
\boldsymbol{c^t} &= \boldsymbol{f^t} \odot \boldsymbol{c^{t-1}} + \boldsymbol{i^t} \odot \boldsymbol{g^t},\\
\boldsymbol{o^t} &= \sigma(\boldsymbol{W_o}\left[[\boldsymbol{x^t}]; \boldsymbol{h^{t-1}}; \boldsymbol{c^t}\right] + \boldsymbol{b_o}),\\
\boldsymbol{h^t} &= \boldsymbol{o^t} \odot \tanh(\boldsymbol{c^t}).
\end{aligned}
\tag{4.1}
$$

Here, inputs x^t, cell states $\mathbf{c}^t$, hidden states h^t and gates i^t, $\mathbf{o}^t$, $\mathbf{f}^t$ are 3D tensors. Similar to standard LSTM cell, ConvLSTM cells contain a cell state $\mathbf{c}^t$ that preserves the memory from the past. The forget gate $\mathbf{f}^t$ filters the information obtained from $\mathbf{c}^{t-1}$, and the input gate filters information from the cell state. The new cell state and hidden state are computed as $\mathbf{c}^t$ and $\mathbf{h}^t$. Predicted fuel density is estimated from the hidden units as $\hat{y} = \mathbf{W}_y\mathbf{h}^t$. Therefore, each hidden state $\mathbf{h}^t$ is obtained from hidden and cell states from the prior time step as $\mathbf{h}^{t-1}$ and $\mathbf{c}^{t-1}$. These ConvLSTM layers are used as building blocks to model complex, non-linear interactions in the data. In our model, we stacked several sets of ConvLSTMs and batch normalization layers with ReLU activation. The output layer is a convolution layer that outputs a 3D tensor with one channel for fuel density.

4.2 Transferring Knowledge from Source Domain In a prescribed fire system, the complex interaction of fire lines is challenging to model, especially under extreme wind conditions, which make the dynamics of the fire spread more chaotic. Under this challenging setup, conventional ML model predictions are biased towards the unburned class, leading to an overestimation of fuel in burned regions. In addition to the asymmetric loss formulation (discussed in section 4.3), we propose data augmentation using pre-existing source domain data to address the class imbalance bias. In our framework, the source and target domains are fuel density data where similar ignition patterns are used for lighting the prescribed fires. However, the environmental conditions, including wind speed and direction, may differ. To maintain a realistic operational setting, we use the same source maps during inference as well. In our experiments, to study the impact of source data on training, we evaluate the model performance with different source data scenarios: (a) no source data, and (b) source fuel density generated with an initial wind speed of 1 m/s and wind direction 230° (*standard setting*, Appendix, Fig. 5b). While source maps with higher wind speeds and disparate wind conditions were also explored, the standard source maps resulted in the most skillful prediction model (Appx. Fig. 10). Since higher wind speeds accelerate the spread of fire and cause complex interactions between fire and atmosphere, other source settings increased uncertainty and added bias toward the over-burning of fuel. Notably, since we use one source setting for training, predicting all the target fuel densities with higher wind speeds and disparate wind directions involves overcoming distributional shifts. Next, to overcome the problem of physically inconsistent predictions, we integrate known physical constraints in the modeling framework.

4.3 Integrating Knowledge

4.3.1 Transferring knowledge from the physics-based model: The problem of fuel transport

DEFINITION 1. (FUEL TRANSPORT) *This problem relates to the transportation of fuel by wind from one location to another. For fuel density $y_{t,s}$ at location s and time step t, fuel transportation results in $y_{t,s} = f(y_{t-1,s}) + \rho y_{t,s'}$, where f represents the non-linear effect of other physical processes on fuel density, ρ refers to the fraction of fuel that is transported from location s' to s. Thus, the probability of fuel increasing in an ignited cell s is non-zero, $P(y_{t,s} > y_{t-1,s}) > \delta$ where $\delta \in [0, 1]$.*

Fuel transport is more common in the case of wildfires that generally spread under higher wind speeds. Since prescribed fires are lit under very specific wind conditions (low wind speed), fuel transportation is unlikely. This assumption is encoded in the QUIC-Fire model. In simulations generated by QUIC-Fire with a homogenous initial fuel profile, this helps avoid any uncharacteristic expansion and contraction of burned fuel density arising from fuel transportation from observed or unobserved regions. To emulate this behavior, we incorporate the assumption that a cell's fuel density cannot increase over time by fuel transportation. Therefore, we penalize those predictions that have an uncharacteristic increase in fuel density over time, given as,

$$\mathcal{L}_{\mathcal{FT}} = \frac{\sum_t ||Y_t - \hat{Y}_t|| \odot \mathbb{1}((\hat{Y}_t - \hat{Y}_{t-1}) > \epsilon)}{T}. \tag{4.2}$$

To put a soft constraint, ϵ, a non-negative value, is used to set the tolerance for what will be defined as a significant increase in fuel density over two consecutive time steps. T is the number of time steps

4.3.2 Spread Consistency Fire managers often rely on estimates of fire spread metrics like rate of spread and burned area indices to understand how the fire is going to spread. Since these quantities are estimated from fuel density, we regularize those fuel density predictions that lead to higher errors in estimates for the fire spread metrics. We, therefore, ensure the spread consistency by estimating the average rate of spread and burned area percentage from the predicted fuel density and minimizing the difference with the estimates from the QUIC-Fire simulations.

DEFINITION 2. (RATE OF SPREAD) *Average rate of spread is computed as the distance traveled by fire over the time spent from the initial time step to time t [34], $\mathcal{ROS}(Y_t)$ = Distance/Time Taken = $(\eta(Y_t) - \eta(Y_{t_0}))/(t - t_0)$. Here, t_0 is the initial time step, and $\eta(.)$ estimates the number of columns impacted by the fire.*

Here, Y_{t_0} refers to where the fire is ignited. In the dataset, the wind generally starts from the west and blows toward the east. This allows us to estimate a unilateral rate of spread based on how many columns of cells are impacted by the fire, enabling us to reduce the number of FLOPS for ROS estimation in the loss computation. **Rate of Spread (ROS) loss** is the mean squared error in the rate of the spread between predicted and observed fuel density values is minimized as part of the loss, as

$$\mathcal{L}_{\mathcal{ROS}} = \frac{\sum_t (\mathcal{ROS}(Y_t) - \mathcal{ROS}(\hat{Y}_t))^2}{T}. \tag{4.3}$$

DEFINITION 3. (BURNED AREA PERCENTAGE) *Burned area percentage refers to the percentage of cells burned in the whole grid at time t, $\mathcal{BA}(Y_t) = (\sum_i^M \sum_j^P \mathbb{1}(Y_{i,j,t} < \epsilon_b) \times 100)/(M \times P)$.*

Burned Area (BA) Percentage loss is the mean squared error between the predicted and observed burned area metric is minimized, as

$$\mathcal{L}_{\mathcal{BA}} = \frac{\sum_t (\mathcal{BA}(Y_t) - \mathcal{BA}(\hat{Y}_t))^2}{T}. \tag{4.4}$$

Incorporating $\mathcal{L}_{\mathcal{FM}} = \lambda_{FM}(\mathcal{L}_{\mathcal{ROS}} + \mathcal{L}_{\mathcal{BA}})$ ensures that the statistical properties of the predicted fuel density match the ones of the QUIC-Fire simulations.

4.3.3 Burn Regularization To overcome the class imbalance problem, we can penalize the over-estimation of fuel in burned regions in the loss term by including a weighted loss on the burned cells. The burned loss is given as follows,

$$\mathcal{L}_{\text{Burned}} = \frac{\sum_t ||Y_t - \hat{Y}_t|| \odot \mathbb{1}((Y_t) < \epsilon_b)}{T}, \tag{4.5}$$

Similarly, to address the model bias towards lower fuel density values under high wind speeds, we can add weighted loss to regularize the over-burning behavior in unburned area. The unburned loss is given as follows,

$$\mathcal{L}_{\text{Unburned}} = \frac{\sum_t ||Y_t - \hat{Y}_t|| \odot \mathbb{1}((Y_t) > \epsilon_u)}{T}. \tag{4.6}$$

To enforce these physical constraints, we formulate the physics-guided loss function as,

$$\begin{aligned} \mathcal{L} =& ||Y - \hat{Y}|| + \lambda_{FT}\mathcal{L}_{\mathcal{FT}} + \mathcal{L}_{\mathcal{FM}} \\ &+ \lambda_{Burned}\mathcal{L}_{Burned} + \lambda_{Unburned}\mathcal{L}_{Unburned} \end{aligned} \tag{4.7}$$

More details on the choice of loss penalty coefficient terms λ are provided in the results section.

4.4 Probabilistic Graphical Modeling (PGM) To integrate prior knowledge about the physical processes, we incorporate a probabilistic graphical modeling structure in our framework. This enables us to represent different variables of interest at multiple levels of abstraction, allowing us to leverage the interdependence among these variables for better fuel density estimation. Scientific problems like prescribed fire modeling have a high degree of complexity arising from the interaction of different physical processes. In our case, we model two quantities - fuel density and number of ignited cells. Modeling of the number of ignited cells captures more global similarities in the observed and predicted values, whereas fuel density modeling focuses more on local similarities in the observed and predicted values.

We incorporate mixture density modeling [7] to factorize the effect of these multiple physical modalities in the latent space and improve response posterior estimation of fuel density. We model fuel density as a mixture of Gaussian components in our framework. Additionally, incorporating prior knowledge about burned area estimation can also improve fuel density modeling. Drawing inspiration from epidemic modeling [26] and wildfire modeling [25], where Poisson distribution is used to model count datasets with extreme values, we use Poisson priors to estimate the number of ignited cells in the grid at each time t.

4.4.1 Gaussian Mixture Density Networks To model the response as a mixture of Gaussian components, we estimate the parametric distributions C_j and membership of each component represented by the mixing parameter, π_j, based on the parameter vector learned by the neural network $\phi(x)$. The fuel density response can be represented as, $\hat{y} = \pi_1(\phi(x))C_1(y|\phi(x)) + \pi_2(\phi(x))C_2(y|\phi(x))$ where component C_j can be represented as realizations from a Normal distribution as, $C_j \sim N(\theta(x))$. For each Gaussian component j, parameters $\theta = \{\mu, \sigma\}$ and π are learned as outputs of the neural network $\phi(x)$. The parameters can be estimated by minimizing the negative logarithm of likelihood [7],

$$\mathcal{L}_{MDN} = -\sum_i log(\sum_j \pi_j(\phi(x_i))C_j(Y_i|\phi(x_i))) \tag{4.8}$$

4.4.2 Burned Area Estimation from Poisson Processes We also consider the point pattern of fire ignitions as a realization of a Poisson process. For a random count measure ψ that represents the number of cells that are ignited in a grid at time t, we model the rate function as $\lambda(y|x,t) = \mathbb{E}(\psi) = \mathbb{E}\sum_i^\psi \mathbb{1}(y_i|x_i \in I)$, where I represents the Borel set representing the ignited cells in the observed region at time t. Poisson processes characterize the number of ignited cells as estimated from fuel density predictions, $\psi \sim Pois(\lambda(y|x,t))$. We employ the variational free energy function to estimate ψ as,

$$\mathcal{L}_{PP} = KL(q(\psi)||p(\psi)) - \mathbb{E}_{q(\psi)}[logP(\mathcal{D}|\Phi_\psi)] \tag{4.9}$$

The KL term ensures that the posterior q learned for ψ is parsimonious to maintain similarity with the prior $p(\psi)$. The negative log-likelihood cost helps ensure that appropriate λ is learned to fit framework Φ_ψ to our dataset $\mathcal{D}$. To incorporate the probabilistic graphical structure in the framework, the loss function can be modified from Eq. 4.7 to Eq. 4.10. A perceptual model for the PGM is given in Figure 6 in Appendix.

$$\begin{aligned} \mathcal{L} =& \mathcal{L}_{MDN} + \lambda_{FT}\mathcal{L}_{\mathcal{FT}} + \mathcal{L}_{\mathcal{FM}} \\ &+ \lambda_{Burned}\mathcal{L}_{Burned} + \lambda_{Unburned}\mathcal{L}_{Unburned} \\ &+ \lambda_{PP}\mathcal{L}_{PP} \end{aligned} \tag{4.10}$$

The framework is summarized in Algorithm 1 in Appendix. In each training step, we first estimate the latent representations learned from the spatiotemporal model $\phi(x)$ for each batch. These representations are further used to estimate the Gaussian mixture model components to compute the fuel density $\hat{y}_i$. ψ is sampled from $Pois(\lambda)$ where λ depends on the learned conditional response distribution $y|x$ at each time t. To update all the parameters in the model Φ, compute gradient with respect to the loss function given in Eq. 4.10 and update all the parameters. The model without the graphical model component is called PGCL (loss Eq. 4.7), while the one with the graphical modeling component is called PGCL+ (loss Eq. 4.10).

5 Results

5.1 Experimental Setup

5.1.1 Baselines We compare the proposed models in the methods section with other baselines in wildfire and prescribed fire modeling. We compare our results with CNN [12; 4; 40; 29; 35], ConvLSTM (CL) [10; 21] and UNet [21; 43]. We also include Firefront UNet model [8] that learns representation for fire state, spatial forcing and weather in separate encoders. We include a CL-GL model that further modifies the CL model with Gram loss-based regularization to match the statistical similarities between the predicted and observed fuel density. Additionally, we include the *Match baselines* that uses historical fuel maps as prediction for how fuel may change in the future. To estimate fuel in the test set, we consider two baselines - Match ignition baseline and Match wind baseline. Match ignition baseline looks at the first fuel map in the historical data with the closest ignition pattern - while the wind conditions may vary. Match wind looks at the first fuel map in the historical data with the closest wind conditions - while the ignition pattern may vary.

5.1.2 Dataset and Experimental Details We use simulation runs from the QUIC-Fire model [1] to learn the prescribed fire emulator. To test generalization under different environmental factors, the simulation runs include 5 different ignition patterns, 7 wind speeds, and 11 wind directions. The simulation runs are for grasslands with two-dimensional evolution of fires captured in 300 x 300 cells grid over n time steps at 1 second time intervals. Each cell is at a 2m x 2m resolution. In the experimental setup, we randomly split the 100 runs and put 50% of the data into training and the rest into test dataset, with each comprising 50 samples. Therefore, each of the datasets has input data with dimensionality 50 simulation runs $\times$ 50 time steps $\times$ 300 rows $\times$ 300 columns $\times$ 4 features . Features are standardized using min-max scaling. With batch size 1 and using the Adam optimization method for gradient estimation, we train each model for 250 epochs [1]. The code is implemented using Tensorflow 2.0 and NVIDIA A40 GPU. Hyperparameters, including penalty coefficients in the loss terms, are fine-tuned using random grid search in the models. Learning rate is 0.001, $\lambda_{FT}, \lambda_{Burned} = 0.001$, $\lambda_{Unburned}, \lambda_{FM} = 0.0001$. In the experiments, we use $\epsilon = 0.001$ for the physical constraint loss masking. More details on the selection of hyperparameters and an ablation study are given in the appendix. We also use ϵ_b =0.1 and ϵ_u =0.65 for the burned and unburned loss masking, respectively. In the generalization results, we sample test runs into different datasets with different physical properties. Description on test datasets is given as follows: $\mathcal{D}_{\text{Low Wind}}$: initial wind speed $< 10m/s$ and $\mathcal{D}_{\text{High Wind}}$: initial wind speed $\geq 10m/s$. $\mathcal{D}_{\text{NW Wind}}$: initial wind direction : from northwest, and $\mathcal{D}_{\text{SW Wind}}$: initial wind direction: from southwest. $\mathcal{D}_{\text{Aerial}}$, $\mathcal{D}_{\text{Outward}}$, $\mathcal{D}_{\text{Strip South}}$, $\mathcal{D}_{\text{Inward}}$ and $\mathcal{D}_{\text{Strip North}}$ datasets include samples with different ignition patterns for igniting the fire.

5.1.3 Metrics Measuring success in many scientific domains, including fire modeling, is hard to capture via standard performance metrics commonly used in ML model design. We propose new loss functions and evaluation metrics to evaluate the physical consistency of the outputs from ML models. We evaluate MSE, burned area MSE, unburned area MSE and fire metric MSE (ROS MSE + BA MSE) on test set. We formulate a metric, **Dynamic MSE (DMSE)**, that evaluates model performance based on rate of change in fuel density. For time steps with bigger change in the observed fuel density, we ensure that the error in predictions are penalized more, $DMSE = (\sum_N (Y_t - Y_{t-1}) \cdot (||\hat{Y}_t - Y_t||))/(\sum_N (Y_t - Y_{t-1}))$. We further validate the physical consistency of the predictions. We consider cells to either be unburned (U), burning (B) or completely burned (C). $\mathbf{M}_{\text{FT}}$ is the percentage of cells that do not follow the fuel transport constraint in the predicted values. $\mathbf{M}_{\text{U-C}}$ is the percentage of unburned cells that are predicted to be burned. $\mathbf{M}_{\text{U-B}}$ is the percentage of unburned cells predicted as burning. $\mathbf{M}_{\text{B-C}}$ is the percentage of cells that reached burned stage too early. $\mathbf{M}_{\text{B-U}}$ is the percentage of burning cells that are predicted to be unburned.

5.2 Results and Discussion Table 1 reports results on the impact of source domain data. Furthermore, we show the impact of including physics (PGCL) and including both physics and probabilistic graphical modeling (PGCL+) in Table 2. We further present results on physical consistency of predictions, inference time for models, and generalization under varying environmental conditions. Results on ablation study, data sparse scenarios and temporal evolution of MSE are presented in the Appendix.

5.2.1 Spatiotemporal Modeling Table 1 compares CNN, UNet, CL, CL-GL, and Firefront models in learning fire behavior. Among the models, the CL model captures the spatiotemporal changes in fuel density and has the highest predictive skill, suggesting that including temporal information is important for skillful prediction of fuel density. In CL-GL, enforcing the statistical information between the observed and predicted fuel density to be similar is not helpful in capturing the evolution of fire. In contrast, enforcing a more physically meaningful similarity measure, such as in the case of the fire spread metric loss, improves model performance (see ablation study in Appendix).

5.2.2 Source Domain Data The MSE values in Table 1 reflect that by including the source fuel density sequence data as one of the input channels, all the models overcome

[1] Simulation runs can be generated using the code provided in this GitHub repository

[1] Code link provided here

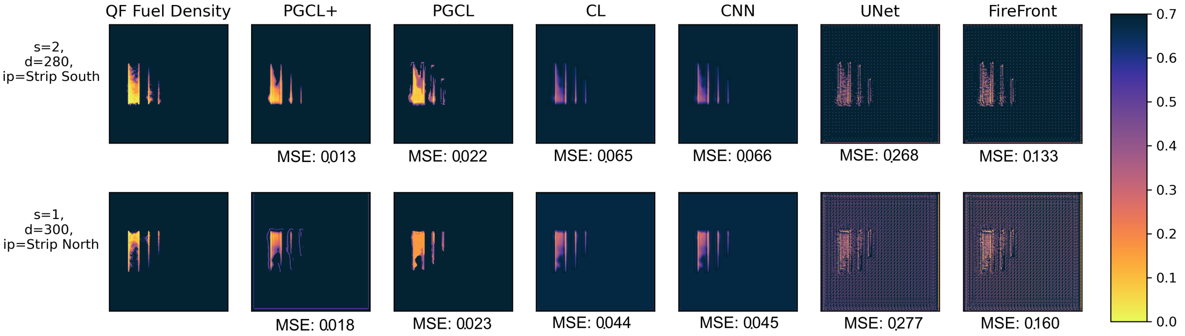

Figure 2: Predicted fuel density by different models. s: initial wind speed, d: initial wind direction, ip: ignition pattern. QUIC-Fire (QF) Fuel density is the ground truth we are emulating.

the bias towards over-estimation of fuel. We see the biggest improvement in CNN model performance. Since CNN and U-Net are unable to capture the temporal information, including source domain data is able to significantly reduce the overestimation bias. CL, FireFront, UNet, and CL-GL leverage both spatial and temporal information, resulting in lower MSE before the addition of source domain data compared to CNN while adding the source domain data helps all the frameworks. For further experiments, we explore CL model (with source domain data) for physics-guided learning since it achieves the best prediction skill.

Table 1: Test set model performance. CL with source domain input data outperforms other methods.

Method	Source Domain	Overall MSE (↓)	Unburned MSE(↓)	Burned MSE(↓)	Fire Metrics MSE(↓)	DMSE(↓)
CNN	w/o	4.9809	4.9808	9.9618	4.9809	0.0003
U-Net	w/o	2.9754	2.9754	5.9508	3.1712	0.0002
CL	w/o	1.1745	1.1745	2.3490	1.1745	0.0003
FireFront	w/o	1.6238	1.6238	3.2476	2.6008	0.0013
CL-GL	w/o	1.7971	1.7971	3.5943	1.7982	0.0004
CNN	w	0.0294	0.0294	0.0594	0.0748	**0.0001**
U-Net	w	0.0352	0.0352	0.0704	0.0354	0.0002
CL	w	**0.0282**	**0.0282**	**0.0416**	**0.0209**	**0.0001**
FireFront	w	0.0293	0.0293	0.0586	0.0745	**0.0001**
CL-GL	w	0.2336	0.2336	0.4672	0.2342	0.0002

5.2.3 Integrating Knowledge

We investigate the effect of incorporating physical constraints (PGCL) and graphical structure (PGCL+) on prescribed fire emulation in terms of the test set evaluation metrics in Table 2. The physics-guided models outperform other baseline models. PGCL and PGCL+ also show a reduction in fire metric MSE values, suggesting that the overall rate of spread and burned area percentage in predictions is similar to the observed values. PGCL+ also improves the fire metric estimation over PGCL and other methods since it captures the interdependence between fuel density values and burned area estimates.

Table 2: Test set model performance. Physics-guided ConvLSTM model (PGCL) and Physics-guided ConvLSTM model + PGM (PGCL+) outperform other methods.

Method	Overall MSE (↓)	Unburned MSE (↓)	Burned MSE (↓)	Fire Metrics MSE (↓)	DMSE (↓)
CL	0.0282	0.0282	0.0416	0.0209	0.0001
PGCL	0.0157	0.0157	0.0312	0.0307	0.0002
PGCL+	**0.0126**	**0.0126**	**0.0252**	**0.0127**	**0.0001**
Match Ignition	0.0275	0.0275	0.0265	0.0279	0.0051
Match Wind	0.0490	0.0490	0.0403	0.0352	0.0048

5.2.4 Evaluating Beyond MSE

Apart from MSE, evaluating burned and fire metric MSE helps improve our understanding of how well the models are handling the class imbalance problem and the estimation of fire metrics (rate of spread and burned area). Due to the imbalance in the number of burned and unburned cells, unburned cell MSEs have a bigger influence on the overall MSE score. This is reflected in similar values of MSE and unburned MSE (difference seen in 6^{th} decimal place). Moreover, the impact of class imbalance problem is seen in the higher MSE in the burned cells. The addition of source domain data, physical constraints, and graphical structure lead to improvement in estimates for fuel density in burned region and estimate for fire metrics as well, as evident in Table 1 and 2.

5.2.5 Qualitative Evaluation

While the MSE values are useful for evaluating the average model performance, investigating individual prediction examples can be useful to ensure generalization under different distributional settings. Figure 2 shows the individual predictions at the 50th second and also outlines the test sample attributes for comparison. We compare the PGCL+, PGCL, CL, CNN, UNet, and FireFront models. We notice that the physics-guided model has the most stable predictions, while other methods are underburning burned areas (class imbalance problem) and overburning unburned cells. In models that are not informed by physical constraints in the loss term, we also notice physi-

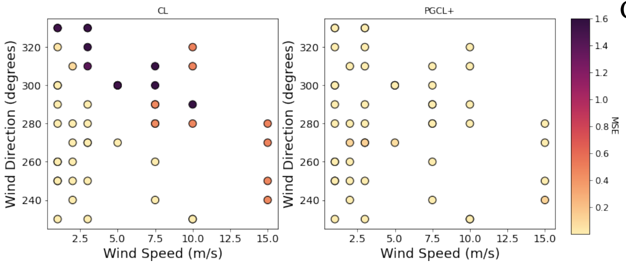

Figure 3: Average MSE by wind conditions for CL and PGCL+. Color represents the MSE values.

cal inconsistencies, such as higher fuel density predictions in cells surrounding the fire perimeter. This behavior arises due to complex wind dynamics in that region that is not being captured by these models. Without physics guidance, the standard ML models inaccurately predict that region's fuel density. Furthermore, in the cells predicted to be burned, the physics-guided model accurately predicts the lower fuel density values. The physics-guided model achieves the most accurate predictions for fuel density.

5.2.6 Physical Consistency This is further evident in Table 3 that summarizes the physical consistency metrics for test samples. The results suggest that the physics-guided models achieve better physical consistency in the predictions. The physics-guided models have a lower proportion of cells where the fuel transport constraint is not met ($\mathbf{M}_{\text{FT}}$), have a lower proportion of falsely predicted burned cells ($\mathbf{M}_{\text{U-C}}$) and have a lower proportion of unburned cells that are predicted as burning ($\mathbf{M}_{\text{U-B}}$). Lower values of $\mathbf{M}_{\text{B-C}}$ and $\mathbf{M}_{\text{B-U}}$ suggest more accurate estimation of burn severity and rate of change in fuel density.

Table 3: Physical consistency evaluations metric for test samples.

Sample	FT (↓)	U-C (↓)	U-B (↓)	B-C (↓)	B-U (↓)
CL	0.07	20.30	57.51	20.21	50.03
PGCL	0.03	2.59	1.79	2.63	38.55
PGCL+	**0.01**	**0.12**	**0.44**	**0.18**	**33.02**

Table 4: Model inference time for predicting test sequence with 50 time steps (in seconds) averaged over 10 repetitions

	QUIC-Fire	PGCL+	PGCL	CL	CNN	UNET
Minimum	16 s	8 s	**6 s**	24 s	12 s	16 s
Mean	42 s	13 s	**12 s**	16 s	13 s	17 s
Maximum	116 s	20 s	17 s	27 s	**16 s**	19 s

5.2.7 Inference Time In Table 4, we also compare the proposed framework with QUIC-Fire in terms of inference time for predicting 50 seconds of fuel density. We see that PGCL+ achieves 3 to 5.8 times speedup compared to QUIC-Fire, and PGCL achieves 4 to 6.8 times speedup when using CPU.

Table 5: Generalization under different physical conditions.

Category	Data	PGCL+	PGCL	CL
Wind Speed	$\mathcal{D}_{\text{Low Speed}}$	**0.0073**	0.0115	0.0103
	$\mathcal{D}_{\text{High Speed}}$	**0.0177**	0.0317	0.1142
Wind Direction	$\mathcal{D}_{\text{NW Wind}}$	**0.0290**	0.0293	0.0302
	$\mathcal{D}_{\text{SW Wind}}$	0.0042	**0.0010**	0.0172
Ignition Pattern	$\mathcal{D}_{\text{Aerial}}$	**0.0054**	0.0194	0.0411
	$\mathcal{D}_{\text{Outward}}$	**0.0028**	0.0065	0.0419
	$\mathcal{D}_{\text{Strip South}}$	**0.0014**	0.0608	0.0180
	$\mathcal{D}_{\text{Inward}}$	0.0301	**0.0091**	0.0122
	$\mathcal{D}_{\text{Strip North}}$	**0.0022**	0.0637	0.0077

5.2.8 Generalization under Different Environmental Conditions Figure 3 summarizes test set MSE values by the initial wind direction and wind speed conditions. The physics-guided model achieves better generalization compared to data-driven spatiotemporal model, especially under higher wind speed conditions and under wind direction greater than 280°. This is because higher wind speeds introduce increased uncertainty and complexity into fire behavior prediction. This is further evident in Table 5, where we report the test MSE values by different wind speeds, wind directions, and ignition patterns to evaluate the generalization under different environmental conditions. Model performance for test samples with wind originating from NW is lower due to the distributional shift between the target and the source maps since standard source maps include only winds originating from SW.

6 Conclusion

This work proposes a novel knowledge-guided ML model that integrates physical knowledge, models interdependence of fuel density and burned area, and leverages source domain fuel maps to improve emulation of the fuel density changes. Our approach reduces different physical inconsistencies in predictions, including fuel transport, over-burning in unburned areas, and class imbalance problem. Since the framework achieves faster-than-real-time inference, it provides avenues for fire managers to adapt their plans to achieve a burn's intended objective with changing environmental conditions. Our framework can be extended to other spatiotemporal models for learning physically consistent fuel densities. Several future directions can be explored. The proposed method can be modified to include other physical constraints. We will extend the study to longer time series where a bigger improvement in inference time can be highly useful to fire managers in operational decision-making.

7 Acknowledgements

This research was funded by NSF grants 2134904 and 1934721.

References

[1] John T Abatzoglou et al. 2016. Impact of anthropogenic climate change on wildfire across western US forests. *Proceedings of the National Academy of Sciences* 113, 42 (2016), 11770–11775.

[2] Sadia Afrin and Fernando Garcia-Menendez. 2021. A machine-learning model for prescribed fire smoke in the Southeastern US. In *AGU Fall Meeting Abstracts*, Vol. 2021. A33G–07.

[3] Anxhelo Agastra et al. 2020. Evaluating a Probability-Based Model for Prescribed Fire Forecasting with Machine Learning. In *100th American Meteorological Society Annual Meeting*. AMS.

[4] Amila Akagic and Emir Buza. 2022. Lw-fire: A lightweight wildfire image classification with a deep convolutional neural network. *Applied Sciences* 12, 5 (2022), 2646.

[5] Frederic Allaire, Vivien Mallet, and Jean-Baptiste Filippi. 2021. Emulation of wildland fire spread simulation using deep learning. *Neural networks* 141 (2021), 184–198.

[6] Reza Azad et al. 2019. Bi-directional ConvLSTM U-Net with densley connected convolutions. In *Proceedings of the IEEE/CVF international conference on computer vision workshops*. 0–0.

[7] Christopher M Bishop. 1994. Mixture density networks. (1994).

[8] Andrew Bolt et al. 2022. A spatio-temporal neural network forecasting approach for emulation of firefront models. In *2022 Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA)*. IEEE, 110–115.

[9] Luca Bottero et al. 2020. Physics-informed machine learning simulator for wildfire propagation. *arXiv preprint arXiv:2012.06825* (2020).

[10] John Burge, Matthew Bonanni, Matthias Ihme, and Lily Hu. 2020. Convolutional LSTM neural networks for modeling wildland fire dynamics. *arXiv preprint arXiv:2012.06679* (2020).

[11] Calf Canyon Fire Information. 2022. *Calf Canyon Incident Overview*. https://inciweb.wildfire.gov/incident-information/nmsnf-calf-canyon

[12] Zachary Cope. 2021. Using Convolutional Neural Networks to Predict QUIC-Fire Outputs. In *AGU Fall Meeting Abstracts*, Vol. 2021. NH15A–0448.

[13] Mark A Finney. 1998. *FARSITE, Fire Area Simulator–model development and evaluation*. Number 4. US Department of Agriculture, Forest Service, Rocky Mountain Research Station.

[14] Nicolas Frangieh et al. 2018. Numerical simulation of grassland fires behavior using an implicit physical multiphase model. *Fire safety journal* 102 (2018), 37–47.

[15] James H Furman and Rodman Linn. 2018. What is FIRETEC (and why should I care)? *Fire Management Today* 76, 3 (2018), 33–36.

[16] Michael R Gallagher et al. 2021. Reconstruction of the Spring Hill Wildfire and Exploration of Alternate Management Scenarios Using QUIC-Fire. *Fire* 4, 4 (2021), 72.

[17] Jonathan Gaudreau et al. 2016. BorealFireSim: A GIS-based cellular automata model of wildfires for the boreal forest of Quebec in a climate change paradigm. *Ecological Informatics* 32 (2016), 12–27.

[18] Michael Gollner et al. 2015. Towards data-driven operational wildfire spread modeling: A report of the NSF-funded WIFIRE workshop. (2015).

[19] J Kevin Hiers et al. 2020. Prescribed fire science: The case for a refined research agenda. *Fire Ecology* 16 (2020), 1–15.

[20] Chad M Hoffman et al. 2018. Advancing the science of wildland fire dynamics using process-based models. *Fire* 1, 2 (2018), 32.

[21] Fantine Huot, R Lily Hu, Matthias Ihme, Qing Wang, John Burge, Tianjian Lu, Jason Hickey, Yi-Fan Chen, and John Anderson. 2020. Deep learning models for predicting wildfires from historical remote-sensing data. *arXiv preprint arXiv:2010.07445* (2020).

[22] Piyush Jain et al. 2020. A review of machine learning applications in wildfire science and management. *Environmental Reviews* 28, 4 (2020), 478–505.

[23] Paul Johnston et al. 2008. Efficient simulation of wildfire spread on an irregular grid. *International Journal of Wildland Fire* 17, 5 (2008), 614–627.

[24] Jeffrey Katan and Liliana Perez. 2021. ABWiSE v1. 0: toward an agent-based approach to simulating wildfire spread. *Natural Hazards and Earth System Sciences* 21, 10 (2021), 3141–3160.

[25] Jonathan Koh et al. 2023. Spatiotemporal wildfire modeling through point processes with moderate and extreme marks. *The annals of applied statistics* 17, 1 (2023), 560–582.

[26] Cécile Kremer et al. 2021. Quantifying superspreading for COVID-19 using Poisson mixture distributions. *Scientific reports* 11, 1 (2021), 1–11.

[27] Rodman Linn et al. 2002. Studying wildfire behavior using FIRETEC. *International journal of wildland fire* 11, 4 (2002), 233–246.

[28] Rodman Ray Linn et al. 2020. QUIC-Fire: A fast-running simulation tool for prescribed fire planning. *Environmental Modelling & Software* 125 (2020), 104616.

[29] M Marjani and MS Mesgari. 2023. The Large-Scale Wildfire Spread Prediction Using a Multi-Kernel Convolutional Neural Network. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 10 (2023), 483–488.

[30] Luis A Pérez-Rodríguez et al. 2020. Evaluation of prescribed fires from unmanned aerial vehicles (UAVs) imagery and machine learning algorithms. *Remote Sensing* 12, 8 (2020), 1295.

[31] Nathan H Schumaker et al. 2022. HexFire: a flexible and accessible wildfire simulator. *Land* 11, 8 (2022), 1288.

[32] Xingjian Shi et al. 2015. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems* 28 (2015).

[33] Andrew L Sullivan. 2009. Wildland surface fire spread modelling, 1990–2007. 3: Simulation and mathematical analogue models. *International Journal of Wildland Fire* 18, 4 (2009), 387–403.

[34] Andrew L Sullivan and James S Gould. 2020. Wildland Fire Rate of Spread. In *Encyclopedia of Wildfires and Wildland-Urban Interface (WUI) Fires*. Springer, 1095–1098.

[35] Samuel Sung, Yuping Li, and Leonard Ortolano. 2020. A 3D Convolutional Neural Network for Predicting Wildfire Profiles. (2020).

[36] Kevin Tolhurst et al. 2008. Phoenix: development and application of a bushfire risk management tool. *Australian journal of emergency management* 23, 4 (2008), 47–54.

[37] Cordy Tymstra et al. 2010. Development and structure of Prometheus: the Canadian wildland fire growth simulation model. *Natural Resources Canada, Canadian Forest Service, Northern Forestry Centre, Information Report NOR-X-417.(Edmonton, AB)* (2010).

[38] Christine Wiedinmyer and Matthew D Hurteau. 2010. Prescribed fire as a means of reducing forest carbon emissions in the western United States. *Environmental science & technology* 44, 6 (2010), 1926–1932.

[39] Jared Willard et al. 2022. Integrating scientific knowledge with machine learning for engineering and environmental systems. *Comput. Surveys* 55, 4 (2022), 1–37.

[40] Suwei Yang, Massimo Lupascu, and Kuldeep S Meel. 2021. Predicting forest fire using remote sensing data and machine learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 14983–14990.

[41] Yanzhu Zan et al. 2022. Emulation of Forest Fire Spread Using ResNet and Cellular Automata. In *2022 7th International Conference on Computer and Communication Systems (ICCCS)*. IEEE, 109–114.

[42] Guoli Zhang et al. 2022. Dynamic prediction of global monthly burned area with hybrid deep neural networks. *Ecological Applications* (2022), e2610.

[43] Puzhao Zhang, Yifang Ban, and Andrea Nascetti. 2021. Learning U-Net without forgetting for near real-time wildfire monitoring by the fusion of SAR and optical time series. *Remote Sensing of Environment* 261 (2021), 112467.