**Key terms and concepts:**

**Category 1: Computer science / machine learning**

**Artificial intelligence**

Artificial intelligence (AI) makes it possible for machines to learn from experience, adjust to new inputs and perform human-like tasks. Most AI examples that you hear about today – from chess-playing computers to self-driving cars – rely heavily on deep learning and natural language processing.

**Machine learning**

Machine learning (ML) is a field of study in artificial intelligence concerned with the development and study of statistical algorithms that can learn from data and generalize to unseen data, and thus perform tasks without explicit instructions. Various regression models are essentially machine learning. Recently, artificial neural networks have been able to surpass many previous approaches in performance.

**Deep learning**

Deep learning is the subset of machine learning methods based on artificial neural networks (ANNs) with representation learning. The adjective "deep" refers to the use of multiple layers in the network.

While basic machine learning models do become progressively better at performing their specific functions as they take in new data, they still need some human intervention.

With a deep learning model, an algorithm can determine whether or not a prediction is accurate through its own neural network—minimal to no human help is required. A deep learning model is able to learn through its own method of computing—a technique that makes it seem like it has its own brain.

**Neural network**

In machine learning, a neural network (also artificial neural network or neural net, abbreviated ANN or NN) is a model inspired by the neuronal organization found in the biological neural networks in animal brains

**CNN**

A convolutional neural network (CNN or ConvNet) is a network architecture for deep learning that learns directly from data.

CNNs are particularly useful for finding patterns in images to recognize objects, classes, and categories. They can also be quite effective for classifying audio, time-series, and signal data.

A convolutional neural network can have tens or hundreds of layers that each learn to detect different features of an image. Filters are applied to each training image at different resolutions, and the output of each convolved image is used as the input to the next layer. The filters can start as very simple features, such as brightness and edges, and increase in complexity to features that uniquely define the object.

A CNN is composed of an input layer, an output layer, and many hidden layers in between.

These layers perform operations that alter the data with the intent of learning features specific to the data. Three of the most common layers are convolution, activation or ReLU, and pooling.

Convolution puts the input images through a set of convolutional filters, each of which activates certain features from the images.

Rectified linear unit (ReLU) allows for faster and more effective training by mapping negative values to zero and maintaining positive values. This is sometimes referred to as activation, because only the activated features are carried forward into the next layer. In our study, we use LeakyReLU activation, which allows a small but non-zero for the negative values.

Pooling simplifies the output by performing nonlinear downsampling, reducing the number of parameters that the network needs to learn.

These operations are repeated over tens or hundreds of layers, with each layer learning to identify different features.

**Hyperparameters of neural networks:**

Hyperparameters are the variables which determines the network structure(Eg: number of layers, number of neurons of each layers, etc.) and the variables which determine how the network is trained(Eg: epoches, batch size, learning rate, etc.).

Hyperparameters are set before training(before optimizing the weights and bias).

**Other neural network examples:**

There are of course many neural networks other than CNN, for example, RNN (Recurrent neural network), which employs recurring network that feeds the results back into the network. (As comparison, CNN is a so-called feed-forward neural network which inputs flows to output in one direction). CNN is good at image recognition; RNN is good at temporal/sequential data such as text or video.

ChatGPT uses so-called LLM (large language model), which is based on a type of so-called Transformer model, an architecture developed by [Google](https://en.wikipedia.org/wiki/Google) using a “multi-head [attention](https://en.wikipedia.org/wiki/Attention_(machine_learning))” mechanism, as proposed in a most famous 2017 paper "[Attention Is All You Need](https://en.wikipedia.org/wiki/Attention_Is_All_You_Need)".

There are newer NN models for image recognition based on transformer too, e.g. “Vision Transformers”. As far as I know, in terms of image recognition, the performance of much older CNN is still at least as good as the much newer Vision Transformers.

There are also many other generative neural networks. To name a few: GAN (generative adversarial network), VAE (variational autoencoder), NF (normalizing flows). They are good at generating new content such as new images. If there is enough time and the judges are interested, you can mention very briefly on the project of learning quantum distributions using VAE and NF.

**Supervised vs. Unsupervised Learning:**

In general, all machine learning algorithms are categorized as either supervised learning or unsupervized learning. The former means like in the training process, we provide the answer of the training samples to the network. Our paper uses supervised learning, because we tell the 5 IM parameter answer to the network during the training process. The most basic example of recognizing the MNIST hand-written digits, is also supervized learning, because the training process requires people to label the training dataset before it is fed to the network training.

Unsupervised learning does need such answers during the training process; it only needs the data itself. Most generative networks, such as GAN, VAE, NFs, are unsupervised learning. I.e. we only provides the images (or distributions) to the network for training, but no need to label these images. The network learning these images and how to generate new images.

**Category 2: Simulation algorithm**

**Markov Chain**

A Markov chain or Markov process is a stochastic model describing a sequence of possible events in which the probability of each event depends only on the state attained in the previous event. Informally, this may be thought of as, "What happens next depends only on the state of affairs now; it does not depend on the path of reaching this state."

**Monte Carlo Simulation**

Monte Carlo methods, or Monte Carlo experiments, are a broad class of computational algorithms that rely on repeated random sampling to obtain numerical results.

In statistics, Markov chain Monte Carlo (MCMC) methods comprise a class of algorithms for sampling from a probability distribution. By constructing a Markov chain that has the desired distribution as its equilibrium distribution, one can obtain a sample of the desired distribution by recording states from the chain. The more steps that are included, the more closely the distribution of the sample matches the actual desired distribution. Various algorithms exist for constructing chains, including the Metropolis–Hastings algorithm.

**Metropolis Hastings**

The Metropolis–Hastings algorithm generates a sequence of sample values in such a way that, as more and more sample values are produced, the distribution of values more closely approximates the desired distribution. These sample values are produced iteratively, with the distribution of the next sample being dependent only on the current sample value, thus making the sequence of samples into a Markov chain. Specifically, at each iteration, the algorithm picks a candidate for the next sample value based on the current sample value. Then, with some probability, the candidate is either accepted, in which case the candidate value is used in the next iteration, or it is rejected in which case the candidate value is discarded, and current value is reused in the next iteration.

In terms of implementation, from the starting point x, it generates a candidate random nearby point x’, then compare the probability ration r = P(x’)/P(x). If r>=1, it always accepts x’; if r<1 it accepts x’ with probability r. It has been mathematically proven that such a sequence will eventually match the target distribution after sequence that’s long enough. Our simulation follows this algorithm, i.e. we accept spin flip at probability equal to the ratio.

**Category 3: Physics and Ising model**

**Latent heat:**

The in the poster and paper, we say that, the newly added , intuitively speaking, this term represents the natural resistance to any state change, and can also be thought of as an analog to the latent heat needed for the ice/water phase transition in classical thermodynamics. What is latent heat?

**Latent heat** (also known as **latent energy** or **heat of transformation**) is energy released or absorbed, by a body, during a constant-temperature process—usually a [phase transition](https://en.wikipedia.org/wiki/First-order_phase_transition), like melting or condensation. The most famous one is phase transition of water and ice at 0 Celsius. Both water and ice can exist at 0-celcius. However, to melt a 0-celsius ice to 0-celsius water, it needs to absorb a lot of heat. This amount of heat (or call it energy) is called latent heat. So even the temperature of the system stays unchanged (in this case 0 degree), it absorbs a lot of heat. And vice versa, to freeze 0-degree water to ice, it needs to release the amount of latent heat.

**What’s ferromagnetic? What’s paramagnetic?**

Ferromagnetism is a property of certain materials (such as iron, nickel, cobalt) that results in a significant, observable [magnetic permeability](https://en.wikipedia.org/wiki/Magnetic_permeability), allowing the material to form a [permanent magnet](https://en.wikipedia.org/wiki/Permanent_magnet). Ferromagnetic materials are familiar metals that are noticeably attracted to a magnet, a consequence of their substantial magnetic permeability.

Paramagnetism is a form of [magnetism](https://en.wikipedia.org/wiki/Magnetism) whereby some materials are only weakly attracted by an externally applied [magnetic field](https://en.wikipedia.org/wiki/Magnetic_field).

In microscope or statistical physics, Ferromagnetism means that the spin (magnetic orientation) of the atoms are strongly aligned. Paramagnetism means that the spin of the atoms are mostly randomly oriented; alignment only occurs infrequently in small local areas

**What is partition function**

A partition function describes the [statistical](https://en.wikipedia.org/wiki/Statistics) properties of a system in [thermodynamic equilibrium](https://en.wikipedia.org/wiki/Thermodynamic_equilibrium). Partition functions are [functions](https://en.wikipedia.org/wiki/Function_(mathematics)) of the thermodynamic [state variables](https://en.wikipedia.org/wiki/State_function), such as the [temperature](https://en.wikipedia.org/wiki/Temperature) and [volume](https://en.wikipedia.org/wiki/Volume). Most of the aggregate [thermodynamic](https://en.wikipedia.org/wiki/Thermodynamics) variables of the system, such as the [total energy](https://en.wikipedia.org/wiki/Energy), [free energy](https://en.wikipedia.org/wiki/Thermodynamic_free_energy), [entropy](https://en.wikipedia.org/wiki/Entropy), and [pressure](https://en.wikipedia.org/wiki/Pressure), can be expressed in terms of the partition function or its [derivatives](https://en.wikipedia.org/wiki/Derivative). The partition function is dimensionless.

It usually takes quite some mathematical efforts to derive/simplify the partition function of a system. But once partition function is derived, most of physical properties of the system can be calculated fairly easily. It took years for Onsager to calculate the exact solution for 2D Ising lattice partition function. Because “he had a lot of time during World War II” (according to himself), as a physics, chemistry professor at Yale Univ.

**Boltzmann Distribution**

In [statistical mechanics](https://en.wikipedia.org/wiki/Statistical_mechanics) and [mathematics](https://en.wikipedia.org/wiki/Mathematics), a Boltzmann distribution (also called Gibbs distribution) is a [probability distribution](https://en.wikipedia.org/wiki/Probability_distribution) that gives the probability that a system will be in a certain [state](https://en.wikipedia.org/wiki/Microstate_(statistical_mechanics)) as a function of that state's energy and the temperature of the system.

where pi is the probability of the system being in state i,  Ei is the energy of that state, and a constant kT of the distribution is the product of the [Boltzmann constant](https://en.wikipedia.org/wiki/Boltzmann_constant) k and [thermodynamic temperature](https://en.wikipedia.org/wiki/Thermodynamic_temperature) T. The symbol ∝ denotes [proportionality](https://en.wikipedia.org/wiki/Proportionality_(mathematics)). As you can see that, lower energy state is always more favored than higher energy state. And the higher temperature T, the more random the system will be, i.e. the probability difference across lower Ei or higher Ei becomes smaller at higher temperature T. Whereas if T=0 Kelvin (absolute zero temperature), only the lowest energy state (so called ground state) can exist, the system always stays at the ground state if T=0 Kelvin. If T goes to infinite, then every state is equally possible (the most random state)

**Potential questions:**

1. **NSIDC data format?**

They are stored in netCDF (.nc) format. There is convenient python package (package name netCDF4) to retrieve data from these files.

NSIDC provides python code to download files over a large period of time in a single batch.

1. **Intuitively, why 1-dimensional Ising model does not show phase transition between ferromagnetism and paramagnetism? Why 2-dimensional (or 3-dimensional or higher Ising model) shows phase transition?**

Mathematically 1D and 2D partition function can be solved with matrix transformations, which reveals a critical temperature Tc =2.27 J / KB for 2D Ising lattice. Below this critical temperature, the lattice can be easily magnetized, also called very high magnetic susceptibility or permeability. Above this critical temperature, the magnetic susceptibility drops drastically; the lattice becomes paramagnetic.

While in 1D Ising model, it remains paramagnetic no matter how low the temperature is. In other words, there is no long-term order of the lattice (which a line for 1D case). Spins of atoms far away are only very weakly correlated with each other. Ironically, Ising solved this 1D model mathematically, but incorrectly generalized that this model does not exhibit phase behavior in any higher dimension.

Intuitively, for a 1D lattice (a line), each spin only has 2 direct neighbors; the interaction between atoms decays fast with distance. For 2D lattice, each spin has 4 direct neighbors even excluding the diagonal ones, the interactions between spins can propagate farther which eventually bring long-term order if the temperature is low enough (when temperature is low, there is less randomness from thermos fluctuations). Consequently, 3D or higher Ising lattices also shows phase transition at certain critical temperatures.

1. **How can spin value be continuous?**

We know that spins have discrete values in quantum mechanics. In this research we use spin value in an Ising lattice to represent the water/ice concentration therefore we introduce continuous spin value. This is a practical choice, without strict compliance with quantum mechanics per se. I am happy to discuss more details with you offline on how to improve this model either theoretically or practically.

1. **Why do we need this additional term: Inertial factor?**

Our innovation of inertia factor in our research is kind-of inspired by the latent heat of phase transition as explained before. To change the spin value of any cell, we add this term accounts for the energy needed to overcome the inertia of the spin change.

Another thought/inspiration of the inertia factor is that: the standard Ising model with spin +1 and -1 means that each flip will simply change the sign of the cell. However, when spin value is continuous, we want to differentiate the probability of flipping to different spin values. So it’s also natural to introduce this inertia term to differentiate the probability to differentiate to states with different spin values.

At end of the day, this is a practical choice of my research. My results with this inertia term are much better than without it.

1. **Why choose simulation period to be half month?**

See paper section 4.2 Simulation Periods

1. **Why can you set to 1?**

See paper. What we care is only the relative strength of J, I and B, not their absolute values. Because the whole model only depends on the products like , , , but not individual J, I or B, so setting to a number different from 1 will still just scale J,I and B inversely with keeping the same , , . So we can set to 1 without loss of generality.

1. **Why are J and I almost constant across the full lattice? But B is set to be dependent on the location of each spin?**

See paper. Simply put, J represents binding between water/water or ice/ice, naturally it should not depend on the timing and the location of water or ice. Similarly, I represents the inherent inertia factor of the transition, it should not depend on the location of water and ice and timing either. B is the external field i.e. can be different ambient temperature, sunlight etc, which can be function of time and location.

1. **Why do we choose B to be linear function of location x and y?**

See paper and question #27. Our focus area, the north and west side are farther from north pole, close to continent; whereas east/south is very near north pole. Intuitively speaking, we think ambient temperature is colder where closer to pole (south/east), warmer when farther (north/west). Therefore we choose a simple linear function of (x,y) for B and it works pretty well. This choice, of course, can be further explored and enhanced.

1. **What is albedo effect?**

See paper section 1.3.

1. **What is jet streams which is mentioned in the poster and paper?**

Jet streams are fast flowing, narrow, [meandering](https://en.wikipedia.org/wiki/Meander) [air currents](https://en.wikipedia.org/wiki/Thermal_wind) in the [atmospheres](https://en.wikipedia.org/wiki/Atmosphere_of_Earth) of the [Earth](https://en.wikipedia.org/wiki/Earth). The strongest jet streams are the polar jets around the [polar vortices](https://en.wikipedia.org/wiki/Polar_vortex), at 9–12 km above sea level. Jet stream is similar to polar vortex except that polar vortex is wind located 16-50 kilometers above Earth's surface, higher than jet stream. Certain research says that that jet streams and arctic sea-Ice loss affect each other.

1. **What’s dual annealing optimization? And why does the JAP paper uses dual annealing but this research uses CNN?**

Dual Annealing is an optimization method and combines local gradient (fast local optimization) and simulated annealing (probabilistic global optimization). The first part of this research uses Dual annealing which works well for this optimization, and the software package is readily available in python Scipy package.

Later last year, I learned quite some machine learning last summer from Prof. Streltruk in Cambridge Univ and the course at WCSU. So I also incorporated CNN machine learning to this research to solve the inverse ising problem. Even though the current performance of CNN vs. dual-annealing is comparable for this study, the advantage of CNN vs. dual-annealing is that, each optimization iteration of DA requires significant amount of time (several hours); CNN training requires several hours, but once CNN is trained once the model is saved, when we feed the initial and final images to the CNN, it can predict the Ising parameters in subsecond. We only need to train the CNN once.

Additionally, using CNN to solve the inverse Ising problem gives us much more control and future enhancements. Because DA is a software in scipy that’s a blackbox to me; while CNN we can control and improve the performance by fine-tuning the network. This also gives CNN better scalability, i.e. if we further enrich the Ising parameters to more complex functional forms, we will be able to adapt the CNN to work for that too; whereas DA might not be able to achieve that because the software package is fixed.

Effectively, CNN allows me to use deep learning technology to implement my own optimization package customized for my purpose, and vast potential for future enhancements. The CNN also gives flexibility and fun of coding too.

1. **Why do we choose 50,000 steps for each period?**

See paper

1. **What does these CNN predicted Ising parameters mean?**

See paper. But we focus less on the absolute values of these parameters. As we know, we have chosen Beta=1. If Beta set to different number, all these J, B and I will be scaled inversely accordingly. So absolute value of these parameters do not mean much. See question 6 too.

Worth noting that J and I and relatively stable across different time periods which makes sense: why should binding between ice/ice or water/water change much w.r.t time?

But B apparently varies a lot across different time: positive in the melting cycle (June to Sep) and negative in the freezing cycle (Sep to Dec). All these makes perfect sense intuitively

As always, after we find mathematical solution either by solving an equation manually or via computer, we always try to understand them intuitively.

(they might ask whey By shows spikes across period if they look really carefully, then tell them these could be due to irregularities of the ambient environment)

1. **How do produce daily results?**

Read the paper. We simply divide 50,000 steps by 15 to get daily results. Understand it yourself.

the Ising parameter values used in these daily results are from the corresponding period as on table 1.

1. **What is quantum ising model?**

The **transverse field Ising model** is a quantum version of the classical [Ising model](https://en.wikipedia.org/wiki/Ising_model). It features a lattice with nearest neighbour interactions determined by the alignment of spins along the z axis, and an external field perpendicular to z axis, e.g. in the x axis. An important feature of this setup is that, in a quantum sense, the spin projection along the z axis and along x axis are not commuting observable quantities. That is, they cannot both be observed simultaneously. This means classical statistical mechanics cannot describe this model, and a quantum treatment is needed. This adds a rich amount of properties to the model. I am just trying to start learning this, but definitely happy to discuss further offline.

More general questions:

1. **How does this paper fit into the wider literature of Ising Model application, especially those on environmental study? list 2-3 academic papers that are closely related to or serve as an inspiration for this study?**

Ising model has found wide application across various fields as stated in the introduction. My research was inspired by this 2019 paper "Ising model for melt ponds on Arctic sea ice," by Prof Golden of Utah University. Pdf attached. And her is a news link on it: <https://unews.utah.edu/ising-melt-pond/>

Their paper focuses on the equilibrium configuration of melting pond shape and sizes via Ising model but my research studies the dynamic water/ice transition via the kinetic Ising model. There is another paper that simulate dynamic Ising for land pattern (deforestation) transition, as well as a few that have introduced continuous spins to Ising.

1. **How does this paper contribute to the literature?**

This is the first (I believe it is) Ising model study of the dynamics of sea ice, which not only fits the data very well but also shed light on the importance of environmental factors that can be further investigated in environmental studies and modeling. It can also definitely be further enhanced for deeper and/or wider studies of Ising model application on similar fields

1. **What is the limitation of this paper? How can this study be improved?**

The model is very simplified. disregarding many real world features e.g. thickness of ice, idiosyncratic variation of external force B across different locations etc. So apparently the results are far from perfect. However it is still able to demonstrate the power of the simple Ising model.

1. **Can you think of any other environmental investigations that the Ising Model can be applied to? (relating to question #16)**

There are plenty of others, for example, this paper “The kinetic Ising model encapsulates essential dynamics of land pattern change” by Stepinski and Nowosad studies forest pattern change using classical Ising model.

1. **What would be the biggest challenge MODELING WISE if the 2-D model is extended to Quantum Ising?**

We only thought about quantum computing, qubits conceptually. Haven’t explored quantum computing in practice. Quantum qubits will be able to represent the continuous spin values, but apparently quantum computer of 3600 qubits is far from practical use yet.

1. **What are the environmental factors you can think of that have or have not been incorporated into this study?**

Same as question #18

1. **How did you find this research topic? How much did your mentor help?**

My mentor Joanne Wang, a physics Prof at Xiamen University at Malaysia, had been a visiting scholar at Yale University in the past year. During one of our conversations she described this fantastic Ising Model (the legendary Ernst Ising and Lars Onsager, who is a Yale Professor, Nobel Prize winner); last year, I came across this Ising ice melting pond paper by Prof Golden and was inspired to use it to model sea ice dynamics. The research turned out to be very interesting and rewarding. I learned a lot of physics in this process. An earlier version of this research has been accept by Journal of Applied Physics to be published in next few weeks.

I learned quite some machine learning since last summer from Prof. Streltruk in Cambridge Univ and the course at WCSU. So at the later stage, I also incorporated CNN machine learning to this research.

More technical questions:

1. **What machine (configuration) did you use for simulation and CNN training? How fast was it ?**

All simulations are done on my personal PC. Intel i7 CPU with 12 cores and 64 GB memory; we used parallel processing on CPU; but this research does not even need to use the GeForce GPU on my computer. The computation complexity for Ising model and the CNN training process is very reasonable. So this research itself is “green”, very environmental friendly. This research only consumed very small amount of electricity.

* each 50K iteration took less than a second; The time is proportional to iterations, so, for example, doubling it to 100k will take twice as much time.
* CNN training is the heavy part, which takes about 10 hours. But it only needs to be trained once; then the model is saved and can be used to predict the Ising parameters for any pair of initial and final ice/water images. The Ising parameter prediction again is super fast, taking sub-second.
* Another heavy part is to generate and training samples. We generate 10,000 samples with different (J,I,B0,Bx,By) for each initial state. There are 13 initial states for each year (from June 16th, July 1st, to December 16th). So the whole sample generation process takes tens of hours. But again, this only needs to be run once and the training samples are saved in the hard disk for future usage. So this is not a concern.
* If being asked, each dual annealing fitting process tries over 10000 set of parameters, so a few hours to complete the each process of fitting for each simulation period

1. **Why did you choose Metropolis for simulation purpose? Any other paper that also use Metropolis for simulating Ising?**

read the paper. there is another Glauber method for Ising simulation which is similar but slightly different from Metropolis. Metropolis is probably the most widely used MCMC method.

1. **Is the B function your innovation? Or has it been used in other paper?**

Read the paper. This is just a practical choice so it works better for the location of our focus area. of course I started with a constant B for the full lattice. but the best fit result was not as great. To accommodate for different external environment for the focus area, a function for B linearly dependent on x and y is a practical choice.

1. **What is the biggest challenge of the coding part for this study?**

Ising model coding is very straightforward. the time consuming part is the effort to enhance the model to improve results. For instance, starting without Inertia factor, constant B, then gradually explore these to get better results. These apparently can be further explored and enhanced.

I also started without the inertia factor, but the results were not satisfactory, therefore I further introduced this inertia factor term and results became much better.

Another very time consuming part is to fine tune the CNN architecture and hyper-parameters to achieve better results gradually. The architecture of this CNN is apparently not obtained on the first day. I tested many different numbers of layers, different forms of convolutional kernel and sizes, different strides and padding, different forms of activation functions, etc. Eventually the architecture illustrated here is the one giving best performance so far. Of course, my debugging and enhancing efforts are still limited, it can be further enhanced for sure.

1. **What is Bx and what is By? What do they capture? What are the (x, y ) coordinates used?**

read the paper page 16 the paragraph above table 1, pasted below:

The values of *Bx* are mostly negative due to the geographic distribution of ice coverage. For our Ising lattice representing the focus area, *x* coordinates corresponding to the rows in the lattice increase from top to bottom; *y* coordinates for the columns increase from left to right. Interestingly, ice coverage near the bottom of our area, the Canadian Arctic Archipelago marked by the red oval in Figure 1, is much thicker than elsewhere including the north pole (the gray circular mask). In fact, many scientists believe this region will have the last piece of ice standing in the Arctic if the Blue Ocean Event happens. As the lower part of the focus area tends to have greater ice coverage, *Bx* is mostly negative, except for very few periods when the ice coverage remains relatively unchanged. *By* is less negative, implying that the impact of the geographic location along the *y* direction is less pronounced than that of *x.* This is because the ice at the north pole is thinner than in Archipelago, which mitigates the impact of the *y* coordinate of a cell. In addition, the values of *Bx* and *By* exhibit greater fluctuations than other parameters, indicating that our simplified linear functional form of *Bi =* is far from perfectly modeling the full effect of external fields; it can be further enriched by linking to actual geographical and environmental factors to enhance the power of the Ising model, which is left for our future research.