**Diffusion models for graph generation-summary**

**Diffusion models for graph generation**:

Diffusion models for graph generation are a type of generative model that aim to learn the underlying distribution of a given set of graphs, and then generate new graphs that follow the same distribution.

These models are based on the concept of a diffusion process, which is a stochastic process that describes how particles move through a medium from areas of high concentration to areas of low concentration.

In the context of graph generation, a diffusion process can be thought of as a series of transformations that gradually change one graph into another.

The idea is to start with a simple graph, such as an empty graph or a graph with a single node, and then gradually add nodes and edges in a way that mimics the diffusion process.

**Diffusion Process:**

A diffusion process is a type of stochastic process that describes how particles move through a medium from areas of high concentration to areas of low concentration. It’s used to model many real-life stochastic systems.

For example, in the context of graph generation, a diffusion process can be thought of as a series of transformations that gradually change one graph into another.

**Noise in Diffusion Process:**   
In the context of diffusion models, noise is added at each step of the diffusion process.

This noise is often Gaussian, but other types of noise distributions can also be used.

The noise introduces an element of randomness into the process, which is crucial for the performance of the model.

The scheduling of this noise, i.e., how much noise is added at each step, is an important aspect of the model.

**Reverse Diffusion Process:**  
The reverse diffusion process, also known as the sampling process of a generative model, is the process of recovering the original data by reversing the noising process. By being able to model the reverse process, we can generate new data.

In other words, it’s the process of moving from a noisy state back to the original state.

**Stochastic Differential Equations (SDEs):**

Stochastic Differential Equations (SDEs) are a type of differential equation where one or more of the terms is a stochastic process, resulting in a solution which is also a stochastic process.

In more detail, a stochastic process is a mathematical object usually defined as a collection of random variables.

In the context of SDEs, this randomness introduces an element of uncertainty that can change the behavior of the solution in unexpected ways.

This makes SDEs particularly useful for modeling systems that are influenced by random effects.

The most common form of SDEs is an ordinary differential equation perturbed by a term dependent on a white noise variable.

The white noise variable represents a random process with a mean of zero and a constant variance, and it introduces the stochastic element into the differential equation.

There are three variants of graph diffusion models that we might consider to generate multi-scale graphs using the Graph U-Net model for our project:

**Score Matching with Langevin Dynamics (SMLD)**

Score Matching with Langevin Dynamics (SMLD) is a technique used for learning generative models in unsupervised machine learning.

It combines two important components: score matching and Langevin dynamics.

**Score Matching:**

Score matching is a method for estimating the parameters of a probabilistic model by comparing the model's score (which is the gradient of the log-likelihood) with the score computed from the actual data.

The model is trained to minimize the difference between the score it calculates and the score computed from the real data.

This helps the model to learn the underlying patterns and structure in the data without explicitly computing complex probabilities.

**Langevin Dynamics:**

Langevin dynamics is a mathematical approach inspired by physical systems, used for sampling from complex distributions.

In the context of generative modeling, Langevin dynamics is employed to generate samples from the model distribution.

It involves simulating the movement of a particle, combining a deterministic force (related to the negative gradient of the log-likelihood) and a stochastic force (related to a noise term).

This creates a process that explores the model distribution.

**Combining Score Matching with Langevin Dynamics (SMLD):**

SMLD uses the combination of score matching and Langevin dynamics to both train the generative model and generate new samples from it.

Model parameters are adjusted to minimize the score matching objective, aligning the model's score with the empirical score from the data.

Langevin dynamics are used to efficiently generate new samples from the learned distribution.

The stochastic nature of Langevin dynamics helps explore the model distribution, capturing the underlying data patterns.

**Key Advantages:**

SMLD efficiently generates new samples from the learned generative model, addressing a common challenge in generative modeling.

Score matching provides a computationally efficient way to estimate model parameters without needing to compute the often-intractable partition function, particularly for complex models.

**Considerations:**

The effectiveness of SMLD depends on factors such as model architecture, hyperparameters, and the characteristics of the data being modeled.

Ongoing research aims to enhance techniques within the SMLD framework for better generative modeling performance.

In simpler terms, Score Matching with Langevin Dynamics is a smart way to teach a computer to understand and generate data patterns.

It's like training the computer to create new examples of data by comparing its guesses to real examples and adjusting its understanding over time.

The Langevin dynamics part helps efficiently create new examples, and the score matching part makes the learning process more manageable.

**Denoising Diffusion Probabilistic Model (DDPM):**

The Denoising Diffusion Probabilistic Model (DDPM) is a generative model used in machine learning for the task of generative modeling, particularly in image synthesis. It combines ideas from *denoising autoencoders*, *diffusion processes*, and *probabilistic modeling* to generate realistic and high-quality samples from a given data distribution.

**Denoising Autoencoder:**

At the core of DDPM is the concept of denoising.

An autoencoder is a neural network architecture composed of an encoder and a decoder.

In DDPM, the model is trained to reconstruct a clean version of an input sample from a noisy version of the same sample.

**Diffusion Process:**

The term "diffusion" in DDPM refers to a simulated process of adding noise to an image and iteratively denoising it.

This diffusion process helps the model capture the underlying data distribution.

**Probabilistic Modeling:**

DDPM is a probabilistic generative model.

It models the likelihood of the data distribution by learning the parameters of the denoising process.

The model aims to minimize the difference between the data distribution and the distribution of the denoised samples.

**Diffusion Process in Training:**

During the training phase, the model is exposed to a series of noisy samples.

The diffusion process involves applying a series of transformations (adding noise and denoising) to gradually transform the input data into a more accurate representation of the underlying data distribution.

**Likelihood Estimation:**

DDPM learns to estimate the likelihood of data samples by comparing the denoised versions of the samples with the original clean data.

The training process involves adjusting the model parameters to maximize the likelihood of the observed data.

**Generative Process:**

Once the model is trained, it can be used to generate new samples.

During the generative process, the model starts with a noisy input and iteratively denoises it to generate samples that resemble the learned data distribution.

**Sampling Techniques:**

Sampling from a probabilistic model often involves techniques like **Langevin dynamics** or **Markov Chain Monte Carlo (MCMC).**

These methods help explore the distribution and generate diverse samples.

**Applications:**

DDPM has found applications in generating realistic images, particularly in scenarios where capturing fine details and high-quality textures is crucial.

It has been used in image synthesis, super-resolution, and other tasks where high-fidelity generation is required.

**Parameter Tuning:**

As with many machine learning models, DDPM requires careful parameter tuning to achieve optimal performance.

This includes tuning the learning rate, the strength of the noise, and other hyperparameters.

**Score-based Generative Model (SGM):**

Score-based generative models are a class of probabilistic models used for generating data.

These models focus on estimating the gradient of the log-likelihood function with respect to the model parameters.

The gradient, also known as the score or the score function, provides information about the direction in which the model parameters should be adjusted to improve the likelihood of the observed data.

**Score Function:**

The score function, represents the gradient of the log-likelihood with respect to the model parameters.

It provides information about how the log-likelihood changes as the model parameters are adjusted.

**Parameter Updates:**

During training, the model parameters are updated in the direction indicated by the negative score.

This aims to increase the likelihood of the observed data under the model.

**Backpropagation:**

Score-based generative models often use backpropagation, a popular technique in deep learning, to compute the gradients efficiently.

The backpropagation algorithm is applied through the computational graph of the model to compute the gradients of the log-likelihood.

**Difficulties in Computing Likelihood:**

For many complex generative models, computing the likelihood directly can be computationally challenging or even intractable.

Score-based models avoid this issue by focusing on estimating the gradient rather than the full likelihood.

**Flexible Model Architectures:**

Score-based generative models can be implemented using various neural network architectures.

This flexibility allows for the modeling of complex data distributions.

**Generative Process:**

Once the model is trained, it can be used to generate new samples by sampling from the learned distribution.

This is typically achieved by employing sampling methods such as **Langevin dynamics** or **Markov Chain Monte Carlo** (MCMC) techniques.

**Applications:**

Score-based generative models find applications in a variety of domains, including image generation, speech synthesis, and more.

They have been used to generate realistic and diverse samples from complex data distributions.

**Score Matching with Langevin Dynamics (SMLD):**

Score Matching with Langevin Dynamics (SMLD) serves as a sophisticated technique in unsupervised machine learning, specifically tailored for learning generative models.

This approach seamlessly integrates two critical components: score matching and Langevin dynamics.

**Score Matching:**

In the realm of SMLD, score matching emerges as a method for effectively estimating the parameters of a probabilistic model.

The core idea involves comparing the model's score, representing the gradient of the log-likelihood, with the score computed from actual data.

The training process revolves around minimizing the disparity between the model's calculated score and the score derived from real data, facilitating the model's grasp of underlying data patterns without intricate probability computations.

**Langevin Dynamics:**

Langevin dynamics, a mathematical technique inspired by physical systems, finds application in SMLD for sampling from complex distributions.

Within generative modeling, Langevin dynamics takes center stage in generating samples from the model distribution.

It orchestrates the movement of a particle by blending a deterministic force (linked to the negative gradient of the log-likelihood) with a stochastic force (tied to a noise term).

This amalgamation creates a process akin to a Markov Chain Monte Carlo (MCMC), allowing for an exploration of the model distribution.

**Combining Score Matching with Langevin Dynamics (SMLD):**

The synergy of score matching and Langevin dynamics in SMLD extends beyond training; it is integral to both training the generative model and generating novel samples.

During training, model parameters adjust to minimize the score matching objective, aligning the model's score with the empirical score from the data.

In sample generation, Langevin dynamics play a pivotal role, efficiently producing new samples from the learned distribution.

The stochastic nature of Langevin dynamics facilitates a comprehensive exploration of the model distribution, capturing intricate data patterns.

**Key Advantages:**

SMLD excels in efficiently generating new samples from the learned generative model, addressing a common challenge in generative modeling.

The use of score matching provides a computationally efficient means to estimate model parameters without grappling with the often-intractable partition function, especially in the context of complex models.

**Considerations:**

The effectiveness of SMLD hinges on factors such as model architecture, hyperparameters, and the unique characteristics of the data being modeled.

Ongoing research continually refines techniques within the SMLD framework to enhance generative modeling performance.

**Denoising Diffusion Probabilistic Model (DDPM):**

The Denoising Diffusion Probabilistic Model (DDPM) stands out as a robust generative model within the realm of machine learning, specifically designed for generative modeling, particularly in the domain of image synthesis.

It amalgamates concepts from denoising autoencoders, diffusion processes, and probabilistic modeling to facilitate the generation of realistic and high-quality samples from a designated data distribution.

**Denoising Autoencoder:**

At its core, DDPM leverages the denoising concept, employing an autoencoder—a neural network architecture comprising an encoder and a decoder.

In DDPM, the model undergoes training to reconstruct a clean version of an input sample from its noisy counterpart.

**Diffusion Process:**

The term "diffusion" in DDPM encapsulates a simulated process involving the addition of noise to an image and its iterative denoising.

This diffusion process plays a pivotal role in enabling the model to comprehend the underlying data distribution.

**Probabilistic Modeling:**

DDPM operates as a probabilistic generative model, aiming to model the likelihood of the data distribution by learning the parameters of the denoising process.

The model strives to minimize the distinction between the data distribution and the distribution of denoised samples.

**Diffusion Process in Training:**

During the training phase, the model encounters a series of noisy samples.

The diffusion process unfolds through a series of transformations, involving the addition of noise and denoising, gradually transforming the input data into a more accurate representation of the underlying data distribution.

**Likelihood Estimation:**

DDPM is adept at learning to estimate the likelihood of data samples by comparing denoised versions with the original clean data.

The training process encompasses adjusting model parameters to maximize the likelihood of observed data.

**Generative Process:**

Once trained, DDPM excels in generating new samples.

The generative process involves starting with a noisy input and iteratively denoising it, thereby producing samples that mirror the learned data distribution.

**Sampling Techniques:**

Sampling from DDPM often involves sophisticated techniques such as Langevin dynamics or Markov Chain Monte Carlo (MCMC), facilitating the exploration of the distribution and the generation of diverse samples.

**Applications:**

DDPM finds practical applications in generating realistic images, particularly in scenarios where capturing fine details and high-quality textures is essential.

Its utility extends to image synthesis, super-resolution, and other tasks demanding high-fidelity generation.

**Parameter Tuning:**

Similar to many machine learning models, DDPM necessitates meticulous parameter tuning for optimal performance.

This encompasses adjusting the learning rate, the strength of the noise, and other hyperparameters.

**Score-based Generative Model (SGM):**

Score-based generative models constitute a distinctive class of probabilistic models dedicated to data generation.

These models focus on estimating the gradient of the log-likelihood function concerning the model parameters, utilizing the score function to guide parameter adjustments for an enhanced likelihood of observed data.

**Score Function:**

The score function, representing the gradient of the log-likelihood concerning the model parameters, provides valuable insights into how adjustments impact the log-likelihood.

**Parameter Updates:**

Throughout training, model parameters evolve in the direction indicated by the negative score, a strategy aimed at amplifying the likelihood of observed data under the model.

**Backpropagation:**

Score-based generative models leverage backpropagation, a widely employed deep learning technique, for efficient gradient computation.

This algorithm traverses the model's computational graph, computing gradients of the log-likelihood.

**Difficulties in Computing Likelihood:**

For many complex generative models, directly computing likelihood can be computationally challenging or intractable.

Score-based models circumvent this challenge by focusing on estimating the gradient rather than the full likelihood.

**Flexible Model Architectures:**

Score-based generative models boast adaptability with various neural network architectures, enabling the modeling of intricate data distributions.

**Generative Process:**

Once trained, score-based models excel in generating new samples.

This involves sampling from the learned distribution, typically achieved through methods like Langevin dynamics or Markov Chain Monte Carlo (MCMC).

**Applications:**

Score-based generative models find diverse applications, spanning image generation, speech synthesis, and more.

Their capacity to generate realistic and varied samples from complex data distributions has contributed to their widespread use.

**Score Matching with Langevin Dynamics (SMLD):**

SMLD is a smart technique in machine learning that helps computers learn without supervision.

It's specialized in creating generative models, combining two main ideas: score matching and Langevin dynamics.

**Score Matching:**

Estimates model parameters by comparing the model's guesses with real data.

Trains the model to minimize the difference between its guesses and the real data's details.

This simplifies learning without complex probability calculations.

**Langevin Dynamics:**

Generates new examples by simulating a particle's movement in a complex environment.

Utilizes a mix of a directed force (related to learning from mistakes) and random force (related to uncertainty) to explore and understand the data's structure.

**Combining Score Matching with Langevin Dynamics (SMLD):**

During training, adjusts the model to make its guesses more accurate (minimizing the score matching objective).

In sample generation, Langevin dynamics helps create new examples efficiently.

**Key Advantages:**

Efficiently creates new examples, solving a common problem in generative modeling.

Efficiency: Smartly estimates model parameters without dealing with complex calculations.

**Considerations:**

Success depends on factors like model design, settings, and the unique features of the data.

**Denoising Diffusion Probabilistic Model (DDPM):**

DDPM is a powerful tool in machine learning for creating new and realistic images.

It blends ideas from denoising autoencoders, diffusion processes, and probabilistic modeling.

**Denoising Autoencoder:**

Cleans up noisy data using a neural network structure called an autoencoder.

**Diffusion Process:**

Simulates a process of cleaning up noisy images step by step.

Helps the model understand the underlying patterns in the data.

**Probabilistic Modeling:**

Models the likelihood of data distribution by learning from the cleaning-up process.

Makes the model's understanding of data distribution closer to the real distribution.

**Training Process:**

Exposes the model to noisy samples, refining its ability to clean up and understand the data distribution.

**Generative Process:**

Creates new examples once trained, starting with noisy input and making it more realistic.

**Sampling Techniques:**

Uses smart methods like Langevin dynamics or Markov Chain Monte Carlo to explore and generate different samples.

**Applications:**

Perfect for generating lifelike images, especially when capturing fine details and textures matters.

**Parameter Tuning:**

Needs careful adjustment of settings like learning rate and noise strength for the best performance.

**Score-based Generative Model (SGM):**

SGM is a unique type of model for generating data.

It focuses on understanding the direction in which model parameters should change to improve the likelihood of observed data.

**Score Function:**

Represents how the likelihood changes as the model parameters adjust.

Guides adjustments to improve the likelihood of observed data.

**Parameter Updates:**

Parameters change in the direction indicated by the negative score to enhance data likelihood.

**Backpropagation:**

Uses backpropagation, a handy deep learning technique, for efficient gradient computation.

**Difficulties in Computing Likelihood:**

Skips the challenge of directly computing likelihood for complex models.

**Flexible Model Architectures:**

Can use various neural network structures, making it versatile for complex data distributions.

**Generative Process:**

Excellent at generating new samples, usually by smart methods like Langevin dynamics or Markov Chain Monte Carlo.

**Applications:**

Widely used for creating realistic and varied samples in areas like image generation and speech synthesis.